

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
1 May 2003 (01.05.2003)

PCT

(10) International Publication Number
WO 03/036419 A2

- (51) International Patent Classification⁷: **G06F**
- (21) International Application Number: PCT/US02/33275
- (22) International Filing Date: 18 October 2002 (18.10.2002)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
10/086,764 19 October 2001 (19.10.2001) US
- (71) Applicant: **THE DETSKY GROUP, L.P.** [US/US];
13707 Springmint Drive, Cypress, TX 77429 (US).
- (72) Inventors: **ROGERS, Heath, W.**; 7797 S. Old State Rd. 37, Bloomington, IN 47403 (US). **METCALF, Jonathan, H.**; 12419 Logan Mill Drive, Houston, TX 77070 (US). **O'LEARLY, Robert, J.**; 13707 Springmint Drive, Cypress, TX 77429 (US). **BARTH, Merlyn, W.**; 13102 Roundstone Ct., St. Louis, MO 63146 (US). **GUNTHER, Jacob, H.**; 725 East 2160 North, North Logan, UT 84341 (US). **MALIK, Rabia, B.**; 12613 Seattle Slew

Drive #2828, Houston, TX 77065 (US). **PETERSON, Erik, S.**; 12710 Ivy Forest, Cypress, TX 77429 (US). **SASHNIKOV, Mikhail, M.**; 13555 Breton Ridge #338, Houston, TX 77070 (US).

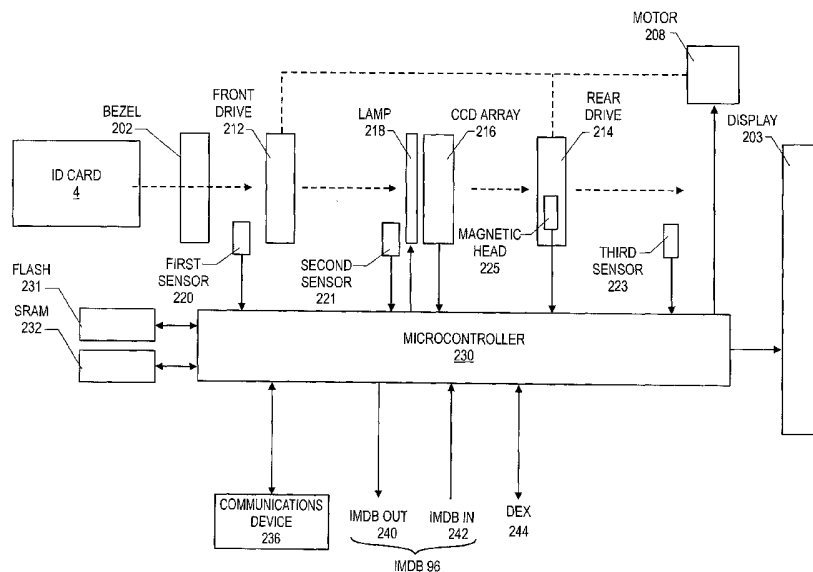
(74) Agent: **LEWIS, Terril, G.**; Howrey Simon Arnold & White, LLP, 750 Bering Drive, Houston, TX 77057-2198 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SYSTEM FOR VENDING PRODUCTS AND SERVICES USING AN IDENTIFICATION CARD AND ASSOCIATED METHODS



(57) Abstract: Disclosed is a highly integrated and flexible system for vending products and services to consumers. The system receives information in advance of the vend by having the consumer insert an identification (ID) card, preferably a driver's license, into a point-of-purchase terminal (referred to as an OSU device). The OSU device preferably contains an Optical Scanning Unit (OSU), capable of scanning the textual information on the ID card. In one embodiment, the scanned information is compared against optical templates present in the system to discern or verify the information on the ID card, and is then used by the system to enable or disable the vending transaction, and/or to allow access to several preregistered system accounts.



WO 03/036419 A2



Published:

— without international search report and to be republished
upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

- 1 -

SYSTEM FOR VENDING PRODUCTS AND SERVICES USING AN IDENTIFICATION CARD AND ASSOCIATED METHODS

CROSS-REFERENCE TO RELATED APPLICATIONS

5 This application is a continuation-in-part of U.S. Patent Application Serial No. 10/086,764, filed October 19, 2001, which is incorporated by reference and to which priority is claimed.

BACKGROUND OF THE INVENTION

10 The present invention relates generally to a system for vending products or services by the use of a standard ID card, such as a driver's license.

It is sometimes desirable to vend products or to provide services only after certain information has been provided by the consumer. For example, in order to vend age-restricted products, such as alcohol or cigarettes, the age of the consumer must be verified in advance of the purchase, typically by having the vendor visually check the consumer's driver's license to 15 verify his date of birth. In another example, it may be desirable to vend gasoline to a consumer only after the validity of his driver's license has been verified.

To make the vending process more efficient, it is desirable to electronically automate the receipt of such pertinent information from the customer. But this is generally only possible if the consumer has some form of identification capable of storing such information in an 20 electronic form. When one reviews the forms of identification typically held and carried by consumers, one finds two primary forms of identification—credit cards and driver's licenses. In this respect, "credit cards" should be understood to refer to other similar types of issued cards, such as debit cards, store-issued credit cards, bank-issued automatic teller machine (ATM) cards, and "smart cards" which contain integrated circuitry. However, both of these 25 forms of identification have drawbacks when applied to automating the process of gathering information about the consumer in advance of the vending of products and services.

Credit cards typically contain magnetic strips or integrated circuitry that contain some amount of consumer information. However, credit cards are of limited utility in facilitating the automated information gathering process discussed above. First, not all consumers carry credit 30 cards, especially many younger consumers. Second, the electronic information contained on credit cards is not always sufficient to allow an assessment of the propriety of vending a particular product to a given consumer. For example, credit cards typically do not contain

- 2 -

information concerning the consumer's age or date of birth, a necessary piece of information for automating the vending of age-restricted products. Third, credit cards, especially store-issued credit cards, typically only allow for the purchase of those products or services sold by that store, and are therefore of limited utility. Fourth, the electronic information contained on credit cards is sometimes encrypted, or stored in formats unknown and undecipherable to the vendors. In short, credit cards, in their various formats, are generally not a suitable mechanism for gathering information about a consumer in advance of the vending of products and services.

Driver's licenses present an attractive means of gathering consumer information because they are widely held. However, driver's licenses, like credit cards, have historically been of limited utility for this purpose. First, driver's licenses come in many different formats, with each state issuing its own unique license. This makes automatic information gathering difficult for a vending system, which is to operate on a nationwide (or international) scale. Second, not all states' driver's licenses contain a means for electronically storing information about the consumer. For example, not all states issue driver's licenses that contain a magnetic strip element. Third, even as to the driver's licenses that do contain electronic means of storing consumer information, the information may be limited, encrypted, or stored in formats unknown and undecipherable to the vendors, and thus suffer from the same problems as credit cards. Fourth, even if driver's licenses were suitable to automate the information gathering process, they lack the means for allowing consumers to pay for the purchase, and therefore have been of limited utility in automating the entire vending process.

A specific problem already mentioned is the vending of age-restricted products. Most, if not all, states impose minimum age requirements for the purchase of certain products such as alcohol, tobacco products, and other age-restricted products. In order to purchase such products, the customer traditionally must present identification to the seller to verify his or her age prior to the transaction. The inability to verify the customer's age prevents age-restricted products from being sold in vending machines in an automated fashion. This verification process is particularly problematic in the vending machine industry since vending machines, by their very nature, involve unattended point-of-purchase transactions. Some examples of prior approaches to this problem or related problems can be found in the following U.S. patents, all of which are incorporated herein by reference in their entirety: 4,884,212;

- 3 -

5,139,384; 5,146,067, 5,273,183; 5,352,876; 5,371,346; 5,450,980; 5,523,551; 5,641,050; 5,641,092; 5,647,505; 5,696,908; 5,722,526; 5,734,150; 5,774,365; 5,819,981; 5,859,779; 5,927,544; 5,988,346; 5,147,021; 4,982,072; 4,915,205; and 4,230,214.

Some prior art vending approaches, such as that of Sharrard, U.S. Pat No. 5,722,526, have contemplated using drivers licenses or other identification cards to verify the customer's age. In the Sharrard system, a customer inputs money into the vending machine and makes his or her selection. Thereafter, the customer is prompted to input an identification card such as a state government issued identification card or a driver's license containing the customer's birth date. The vending machine either optically reads the written birth date on the face of the card, or reads the birth date data from a magnetic strip contained on the back of the card. A processor unit compares this data with the present date that is keyed into the vending machine by its operator, and determines whether the customer is of a sufficient age to purchase the product.

Sharrard's disclosure notwithstanding, it is difficult to implement Sharrard's technique for age verification. As noted previously, not all drivers' licenses contain magnetic strips, and even for those that do, age data may not be present on the strip or may be difficult to extract. Further, despite Sharrard's general disclosure of the idea of optically scanning a driver's license to extract age data, such a process is not disclosed or enabled in Sharrard, but is merely noted as a good idea.

Some prior art approaches such as U.S. Patent No. 5,927,544, issued to Kanoh, suggests that age information can be "recorded on the [credit] card" to verify a vending customer's age for the purpose of vending age-restricted products, see Kanoh, Col. 4, ll. 55-58, but the present inventors submit that such information is in fact rarely present on a standard credit card. Although consumer reporting agencies, such as TRW and Equifax, and other credit card companies such as VISA or MasterCard, store information in databases for a large number of consumers, conventional vending machines are unable to access such information to verify the age of a purchaser. Those prior art vending machines that have connectivity to such databases contemplate using the database to verify credit or password information, but do not disclose or suggest using such databases to verify age. See Kanoh, Col. 4, ll. 37-42 (noting that the microprocessor in his vending machine enables "a credit card company to check credit

- 4 -

card numbers, personal identification code numbers, and other data via a communications link," but not mentioning age data).

What is needed is a highly flexible system for vending products and services that (1) can be implemented on a nationwide (or international) scale, (2) is fully automated, (3) is capable of extracting necessary information from a consumer to assist in the vending process, and (4) is capable of remotely managing and updating an unlimited number of vending machines. Additionally, such a system would be further advantaged by (1) providing means for allowing for the payment of the products and services vended, (2) being implementable by making only minor modifications to otherwise standard vending equipment, and (3) having the capability to vend a wide array of products and services. Such a system is disclosed herein.

SUMMARY OF THE INVENTION

Disclosed is a highly integrated and flexible system for vending products and services to consumers. The system receives information in advance of the vend by having the consumer insert an identification (ID) card, preferably a driver's license, into a point-of-purchase terminal (referred to as an Optical Scanning Unit (OSU) device). The OSU device preferably contains an Optical Scanning Unit (OSU), capable of scanning the textual and graphical information (such as a validation seal or other picture) on the ID card. The scanned information, such as the consumer's age, is then compared against optical templates present in the system (preferably in the OSU) to discern or verify the information on the ID card, and is then used by the system to enable or disable the vending transaction.

The system preferably contains several components that may be distributed on a nationwide basis depending on the desired system functionality and geographic scope of the proposed system. To add flexibility to and to enhance the performance of the system, a protocol that allows for the OSU devices to communicate with the remainder of the system has been developed and is disclosed. Additionally, optical character recognition (OCR) algorithms have been developed and are disclosed to facilitate the analysis of the ID cards, a process that presents special problems not encountered in OCR analysis generally. Furthermore, a design for an OSU, capable of reading and interpreting optical data and magnetic strip data, is disclosed.

In a related embodiment, the disclosed system allows a consumer's ID card to act as a smart card useable for purchasing a wide array of products and services, including food, gas,

- 5 -

money, phone service, rental cars, etc., which are sold through the OSU devices connected to the system. The system may also be used to tap into or establish consumer accounts useable for paying for system products and services. The system may be used more generally to determine information about a person or consumer who accesses the system, for example, by tapping into law enforcement or immigration status databases after OSU analysis of their ID cards. Additionally, methods are disclosed for initializing an OSU device upon its installation in the system and for configuring and/or update its functionality. Because the ID card of different states may be used on the system, the system may be implemented on a nationwide scale.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a block diagram of the system.

Figure 2 shows an elevated perspective view of a left side of the optical scanning unit (OSU), including the faceplate.

Figure 3 shows an elevated perspective view of the left side of the OSU, with the faceplate removed.

Figure 4 shows a plan view of the topside of the OSU, with the faceplate removed.

Figure 5 shows an elevated perspective view of the right side of the OSU, with the faceplate removed.

Figure 6 shows a schematic showing the relationship of the components in the OSU.

Figure 7 shows an illustration of the interaction of the various layers utilized in the Davis Terminal Protocol.

Figure 8 shows an exemplary driver's license capable of being optically analyzed by the system.

Figure 9A shows an exemplary form and cluster information file structure used during optical character recognition (OCR).

Figure 9B shows an exemplary font information file structure used during optical character recognition (OCR).

Figure 10 shows an embodiment of an OSU having a mounting block, two charge-coupled-device (CCD) arrays, and a light source.

Figure 11 shows another embodiment of an OSU having a mounting block, a CCD array, and two light sources.

- 6 -

Figure 12 shows an exemplary scan of a card having ultraviolet lettering.

Figure 13 shows yet another embodiment of an OSU having a mounting block, a CCD array, and offset light sources.

Figure 14 shows an isometric view of the mounting block of Figure 13.

5 Figures 15A-B show exemplary images of a driver's license scanned from different angles.

Figure 16A shows an exemplary image resulting from a subtractive process performed on the images of Figures 15A-B.

Figure 16B shows an inverted image of the image in Figure 16A.

10 Figure 17 shows a filtered image of the image in Figure 16B.

Figure 18 shows an embodiment of an OSU having a CCD array and light source offset by an obtuse angle.

Figure 19 shows an embodiment of an OSU having two mounting blocks for scanning both sides of a document or card. Figures 20A-B show disclosed reduction and comparison
15 techniques for creating a form digest and a font/point size digest for a new form or type of card.

Figure 21 shows an exemplary 16 x 16-pixel section of an image subject to bilinear reduction.

20 Figure 22 shows an exemplary image of a Texas driver's license after 16x16 bilinear reduction.

Figures 23A-B show exemplary database and form digest structures used with the disclosed reduction and comparison techniques.

Figures 24A-B show steps for determining the form or type of card presented using the disclosed reduction and comparison techniques.

25 Figure 24C shows steps for optical character recognition (OCR) of the presented card using the disclosed reduction and comparison techniques.

Figure 25 shows exemplary images undergoing filtering and enhancement techniques useful in OCR analysis.

Figure 26 shows edge detection schemes useful in OCR.

30 Figure 27 shows the internal structure of the Davis system server cluster 18 and the relationships between the various data structures therein.

- 7 -

Figure 28 shows a portion of the system disclosed in Figure 1.

Figure 29 shows a prior art vending machine.

Figure 30 shows a modification to the circuitry of the vending machine of Figure 29 to accompany an OSU.

5 Figure 31 shows a schematic of the circuitry of a vending machine modified to accompany an OSU.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

In the disclosure that follows, in the interest of clarity, not all features of actual implementations are described. It will of course be appreciated that in the development of any
10 such actual implementation, as in any such project, numerous engineering and design decisions must be made to achieve the developers' specific goals (e.g., compliance with technical- and business-related constraints), which will vary from one implementation to another. Moreover, attention will necessarily be paid to proper engineering and design practices for the environment in question. It will be appreciated that such a development effort might be
15 complex and time-consuming, but would nevertheless be a routine undertaking for those of skill in the art given the disclosure in the present specification.

I. System Overview

Disclosed herein is a transactional, multi-tiered, networked information system, referred to as the DavisTM system. ("Davis" is an acronym for the "Detsky Age Verification
20 Information System"). The system includes a broad range of technology and uses relating to the sale and distribution of products and/or services. Many of these uses are disclosed herein, but one skilled in the art should recognize that the system disclosed herein is capable of many uses, none of which detract from the spirit of the disclosed inventive concepts.

In a preferred embodiment of the system, the system includes a terminal accessible by a
25 consumer, such as a vending machine, an automatic teller machine (ATM), a gas pump, a public phone, etc. This terminal contains a means for determining certain information about the customer relevant to the desired purchase. In a preferred embodiment, the terminal is able to receive a piece of identification from the consumer, such as a driver's license or other identification (ID) card.

30 Preferably, but not exclusively, the consumer information is read from the ID card using optical scanning technology, the specifics of which will be disclosed later in this

- 8 -

specification. Thus, the terminal includes an optical scanning unit (OSU) for receiving the ID card and “reading” certain information from it. For example, assuming the terminal is a vending machine that vends age-restricted products such as cigarettes or alcohol, the consumer’s age can be read from the ID card and processed by the system to determine the consumer’s age and enable the purchase accordingly. If the terminal is a gas pump, the consumer’s driver’s license can be read and checked by the system to check its validity and enable the purchase of gas accordingly. If the terminal is an ATM, the consumer can use his ID card (as opposed to the more traditional, magnetic-strip debit cards issued by banks) to withdraw cash from his savings or checking account. Thus, the system allows a standard ID card, such as driver’s licenses, to act as a “smart card,” even if such card otherwise lacks the means for storing electronic data, such as on a magnetic strip or in integrated circuitry included on the card. These are just a few examples of the functionality of the system, all of which are made feasible by the OSU.

An overview of the components of the system 8 is shown in Figure 1. One skilled in the art will immediately recognize that the system is suitably flexible that certain components in the system can be combined, eliminated, or added based on the desired functionality as dictated by the product or service to be marketed.

A. The OSU Device 10

The terminal with which the consumer reacts, and which contains (preferably) the optical scanning unit (OSU) 6 (see next section), is referred to generally as OSU device 10. For example, OSU device 10 might constitute a vending machine, an ATM, a public phone, a gas pump, etc.

The system 8 is capable of interfacing with several OSU devices 10, which may be connected to the system (e.g., to the OSU connection server(s) 12) by any means known in the art to connect electronic devices, such as by fixed cable, modem, wireless, or other networking means. The OSU device 10’s primary function is to receive information from the consumer via its OSU 6 and to dispense products or services to the consumer (e.g., food, gas, money, etc.). Therefore, in accordance with the preferred embodiment, the consumer inserts his ID card into the OSU 6 on the OSU device 10, and a scanned image is taken of his ID card. This image may be sent to other parts of the system to be analyzed, such as the server cluster 18, using an optical character recognition scheme to be described in detail later, or the image data

- 9 -

may be locally processed at the OSU device 10. To avoid long transmission delays, it is currently preferable to process the image within the OSU device 10 itself. However, in the future, as higher bandwidth communication systems are made available, it is contemplated that it may be preferable to process image data remotely at the servers. The OSU device 10 also
5 performs other localized processing that need not be (or cannot be) performed by the remainder of the system.

An OSU device 10 is typically manufactured with certain factory standard functionality. For example, if the OSU device 10 is a vending machine, the machine will come pre-programmed to perform many of the functions standard to vending machines generally.
10 However, the OSU device 10 may also be remotely configured or periodically updated as necessary either by the system 8, or locally by a portable computer or personal data assistant (PDA) device capable of interfacing with the OSU device 10. Remote updating from system 8 is preferable due to its flexibility because it allows OSU device operators and owners to control updates via a web-based administration tool accessible over the Internet.

15 An OSU device 10 can be made to operate in "connection mode," "batch mode," or "disconnect mode," or may be attached to other non-Davis systems components if necessary or desirable. When operating in connection mode, the OSU device 10 constantly communicates with another portion of the system 8 to process certain consumer information. For example, analysis of the consumer's age, as determined optically and/or using magnetic strip data from
20 the consumer's driver's license, may be performed remotely by the system when operating in connected mode, although this is not presently preferred as previously mentioned. Connection mode is particularly useful for processing and validating consumer credit card information, which ideally should be performed during a consumer purchase transaction.

When operating in batch mode, the OSU device 10 is not in communication with other
25 portions of the system 8 during a consumer transaction. Instead, the OSU device 10 may be made to connect to the system 8 during off-hours to process consumer information, or to receive update instruction from the system. However, as mentioned previously, it is currently preferred that consumer information is processed directly by the OSU devices 10.

When operating in disconnect mode, the OSU device 10 is configured and updated only
30 when removed from service and attached to a PC or other device suitable for communicating with the OSU device 10 "off line," such as a personal data assistant (PDA). In this sense, one

- 10 -

skilled in the art should recognize that in a particular circumstance the OSU device 10 may be made to encompass all relevant functionality of the system 8, but without the benefit or necessity of communicating with a system or any other components. A good example of this would be an "age validation terminal" which could be installed in bars. In this embodiment, the consumer would simply insert his license into the terminal, most preferably in the presence of a bar attendant, at which point the terminal would perform an optical analysis of the license, and display a green light if the consumer's age is sufficient. In this embodiment, it may not be necessary to have the power of an entire networked system if the terminal itself is programmed off-hours to provide suitable functionality. In this scenario, the bar attendant is spared the potential discomfort of directly confronting the consumer about his age, and instead could rely on the age verification information provided by the terminal. Such a terminal may also prevent mistakes in age verification that otherwise might be made by the bar attendant, or may be able to determine validity concerns with the license that might not otherwise be discernable by the attendant.

The OSU device 10 may also be connected to other systems not normally included in system 8. For example, the OSU device 10 can be made to communicate with VisaNet (an on-line credit card service) to verify a consumer's credit card account information. Likewise, the OSU device 10 (or other parts of system 8) may be configured to dial into VisaNet during off-hours to reconcile transactions made during a specific day. Of course, should the OSU device 10 be made to connect directly with such third party systems, the method of communication may need to be programmed into the OSU device 10 and will not necessarily be the same as the connection, batch or disconnect modes generally contemplated with respect to system 8.

B. The OSU 6

1. Overview

A preferred embodiment for the OSU 6 is shown in Figures 2-6. As will be explained later in this disclosure, OSU 6 can be incorporated into a standard or custom-made OSU device 10, such as a vending machine.

The OSU 6 in a preferred embodiment is a dual-function card reader, capable of reading both the textual and graphical data printed on the face of an ID card, and (if present) a magnetic strip. Because the OSU 6 can read both optical and magnetic data, it is capable of receiving a wealth of important data concerning the consumer from a number of different

- 11 -

consumer ID cards, including driver's licenses and credit cards. In this regard, the OSU 6 can handle consumer transactions using ID cards that contain both optical information and magnetic information (which might be the case for some states' driver's licenses), or separate ID cards where one contains textual information and the other contains magnetic strip information. For example, the consumer's driver's license can be optically read to determine his age, and subsequently his credit card can be magnetically read to pay for a desired purchase. The preferred embodiment of the OSU 6 is therefore extremely flexible. However, it should be noted that an OSU may function according to the inventive concepts disclosed herein even if it does not perform both optical and magnetic reading functions. Thus, for a given application, only optical reading may be required (e.g., if age verification was performed using only a driver's license, but payment was to be made with cash or through debiting of an account established on the system 8), or only magnetic reading may be required. Additionally, an OSU 6 could also be easily modified by one of skill in the art to receive electrical data, e.g., as might reside in the integrated circuitry on a "smart card," in conjunction with any combination of optical and magnetic data.

2. Mechanical Structure of the OSU 6

Figures 2-5 disclose plan views of the OSU 6 as viewed from different vantage points. Figures 10, 11, 13, 14, 18 and 19, which are discussed later, disclose modifications that can be made to the OSU 6 to allow it to analyze more complex optical images, such as holograms and other similar devices useful in form determination and authentication.

In Figure 2, the face plate 200 is visible, which is the portion of the OSU 6 that a consumer would see from the outside of an OSU device 10, although this face plate 200 has been removed from the other drawings for clarity. Faceplate 200 contains a bezel 202, which is essentially a slot for receiving the consumer's ID card. Also present on the face plate 200 are LCD display 203, which provides the consumer operating instructions and status information, and a cancel button/indicator light 204. LCD display 203 is preferably a 16 by 2 character display, but could be larger, or could constitute any other suitable means for displaying information, such as a cathode ray tube, a TFT flat panel display, etc. The faceplate 200 also contains boltholes 206 for mounting the OSU 6 to the face of the OSU device 10.

Figures 2-5 show the internal structures of the OSU 6, including stepper motor 208 with its associated gear 209, gear train 210, front and rear drives 212 and 214, charge-coupled-

- 12 -

device (CCD) array 216, lamp 218, sensors 220, 221, and 223, magnetic head 225, and wires 219. Front and rear PC standoffs 222 and 224 are provided for mounting the printed circuit board (not shown for clarity) that contains the OSU 6's electronics, including microprocessor 230, Flash 231, and SRAM 232 (see Fig. 6). Although not shown, wires 219 are connected to
5 a mating connector on the printed circuit board supported by the standoffs 222 and 224. The printed circuit board also contains an additional connector for connecting to the preexisting circuitry within the OSU device 10 and for obtaining power.

In operation, motor 208 controls and drives the gear train 210, which in turn controls the rubber-coated front and rear drives 212 and 214 to move the ID card 4 passed the CCD
10 array 216 for optical reading and the magnetic head 225 for magnetic reading. A suitable motor for this purpose is part no. PF42T-48, which is manufactured by Nippon Pulse Motors and which has a full step angle of 7.5°. Lamp 218 extends through the entire width of the OSU 6, and acts to illuminate the textual and graphical information on the surface of the ID card 4 to create an image which is then picked up by the CCD array 216. A suitable lamp for use in
15 OSU 6 is part no. BF386-20B, manufactured by JKL Components Corporation. A suitable CCD array is a 768 pixel by 1 pixel linear array part no. TSL1406, manufactured by Texas Advanced Optoelectronics Solutions, Inc., (TAOS).

Also included within the OSU 6, but not visible in Figures 2-5, is the printed circuit board containing electronic control circuitry including microcontroller 230, flash memory 231,
20 and static random access memory (SRAM) 232. As previously mentioned, this printed circuit board is connected to the standoffs 222 and 224, but has been removed from the Figures for clarity. Although the memory chips 231 and 232 can be used in a particular embodiment to hold a variety of data, in a preferred embodiment flash 231 contains the configuration data for the OSU 6. Thus, flash 231 contains the program that defines the general operation of the
25 OSU as well as contains the templates used by this program to determine the validity of the license, and to locate, for example, the date of birth information on the license. Flash 231 also contains the programs or algorithms necessary to perform optical character recognition (OCR) on the received image data, e.g., to determine and interpret the "date of birth" field of the license. SRAM 232 provides temporary storage of data obtained from the license, both optical
30 and magnetic (if any), and provides general temporary storage for the microprocessor control system. An example of such temporary storage would be transaction information and batch

- 13 -

information stored at the OSU prior to communication with the OSU CS 12. A suitable component for the microcontroller 230 is part no. SABC161PILFT, a 16-bit microcontroller manufactured by Siemens AG Semiconductor Division. A suitable component for flash memory 231 is part no. SST39SF040-70-4C, a 4-Megabit, 55-ns flash manufactured by Silicon Storage Technology, Inc. (SST). A suitable component for SRAM 232 is part no. TC554001AF71(Y), a 4-Megabit, 55-ns SRAM manufactured by Toshiba Corporation.

While it is currently preferable to scan, in a line by line fashion, the ID card under analysis to receive an image thereof, other suitable means of receiving an image are contemplated. For example, the OSU 6 could be fitted with a digital camera device to take a “snap shot” of the ID card, instead of scanning line by line. As used herein, “scanning” should therefore be understood as referring to line by line scanning to procure an image, or to other technologies akin to taking a picture or image of the ID card.

3. Electrical Structure of the OSU 6

The relation of the components in the OSU 6 is shown in schematic form in Figure 6. Also shown are the microcontroller 230's connection to communication device 236 (such as a modem), which as previously explained communicates with an OSU CS 12, and its relation to the International Multi-Drop Bus 96, which is the bus internal to a standard vending machine, and which will be explained in further detail in a later portion of this disclosure. DEX (Direct Exchange) line 244 collects and communicates information about the vending machine in which OSU 6 is installed. DEX is well known in the vending machine arts and is based on a protocol published by the European Vending Association. In vending machines supporting DEX, DEX data stored within the vending machine may be shared with external devices such as hand held computers or the remainder of system 8. This protocol thus allows route operators or machine owners to access information such as inventory status of the vending machine, transaction data, metering data, and data pertaining to machine operation. An example of the latter would be temperature data for a machine supporting the vending of perishable food.

With reference to Figure 6, the sequence of events occurring in the OSU 6 is exemplified for a typical transaction. In this example, it is assumed that the consumer uses a driver's license containing a magnetic strip, and that the consumer's age must be verified prior to allowing the purchase of an age restricted product from the OSU device 10. It is also

- 14 -

assumed that payment might be made by a credit card. Of course, an actual transaction implemented with the OSU 6 need not be so limited to these assumptions.

When the consumer approaches the machine, display 203, which is under control of microcontroller 230, displays an instructional message, such as "please insert driver's license."

5 The consumer complies by inserting his driver's license 4 into the bezel 202. When the front edge of the license passes first optical sensor 220, microcontroller 230 starts motor 208, which engages front drive 212 through gear 209 and gear train 210. Front drive 212 then quickly pulls the license into the OSU until the front edge of the license reaches second optical sensor 221. During the transport of the license, the license is supported underneath by idler rollers
10 (not shown in the Figures).

Once the second sensor 221 is reached, the OSU prepares to optically scan the information on the face of the license. At this point, a light source, such as lamp 218 for this embodiment of the OSU having a single light source, is turned on to illuminate the face of the license, and the license is slowly advanced under CCD array 216 to capture an optical image of
15 the license. Alternatively, the OSU can have two or more light sources described below, in which case a first of the light sources (not shown) can be illuminated. Suitably slow forward motion of the license for scanning is achieved by advancing the license .125 mils (one one-thousandth of an inch) per pulse of the stepper motor. Each step of the motor denotes what will ultimately be a line of single pixels in the stored driver's license image. Stepping and
20 scanning the license occurs until the third optical sensor 223 is reached by the front edge of the license, at which point the license has been fully scanned. The line-by-line pixel data provided by the CCD array 216 is stored in SRAM 232 for further processing. The entire optical scanning process takes about 4.3 seconds, but a scanning time of 3.0 seconds is desired in a commercial embodiment. During scanning, display 203 could be made to display a message
25 such as "scanning license, please wait" to inform the consumer of the progress of the transaction.

After a slight delay, motor 208 is again activated, but in a reverse direction, i.e., such that the license is eventually ejected from bezel 202. For embodiments of the OSU having two or more light sources as described below, the second of the light sources can be illuminated
30 while the card 4 is moved in the reverse direction. During this ejection process, the information on the magnetic strip is read by magnetic head 225. Ejection and magnetic

- 15 -

reading of the license is preferably performed at the motor's maximum speed to provide a maximum magnetic signal detectable by magnetic head 225. If magnetic data is present on the license, microcontroller 230 stores this data in digital form in SRAM 232 along with the optical scanned data.

5 At this point, the stored optical and/or magnetic data is processed, either locally by microprocessor 230 or by other components of the system 8 through communication device 236. To the extent data is processed by other components of the system 8, the OSU 6 waits for a response from OSU CS 12. If no response is received, the display 203 might be made to state an appropriate response, such as "no server response, please try later," at which point the
10 OSU 6 reverts to its idle or start condition.

 The optical data is first compared with the templates residing in flash 231. The purpose of this comparison is to find a template match that would indicate to the microprocessor 230 in the OSU 6 that a valid driver's license has been presented for age verification and what issuing body (state or country) supplied the license. If no match is found, OSU 6 will interpret this
15 result to mean that no age verification can be accomplished using the optical data. If however a match is found, information associated with the matching template will indicate where on the scanned image to look for detailed information concerning the owner of the license, and more specifically, his date of birth, as will be explained in more detail later. Where the decision is to be made locally at the OSU 6, the OSU 6 need only to look at the date of birth and may not
20 need to determine other information about the consumer, such as name, driver's license number, etc. This date when compared to the current date (obtained from the real time clock in the OSU) will determine the age of the owner of the license. Preferably, optical character recognition of the name, address, driver's license number, and expiration date of the license will be sent to the server cluster 18 where additional checks can be made to further verify age,
25 license validity, and other necessary information. Additionally, where the driver's license contains magnetic stripe data, similar information may be sent to the server cluster 18 prior to age verification, or may be used to further verify the information determined by optical analysis by comparing the optical and magnetic data.

 If either the OSU 6 or other portions of the Davis system 8 determines that the
30 consumer's age is adequate, display 203 would display an appropriate message, such as "approved," and the display 203 would thereafter prompt the consumer to make payment to the

- 16 -

OSU device 10, such as, by displaying the message "insert cash or credit card." This step might not be necessary if the consumer has a pre-registered account on the system connected to his driver's license, in which case his account would be debited accordingly. If a pre-registered account is to be the basis for payment, the optical recognition data obtained from the
5 license will be sent to the server cluster 18 as a "key" to access the system account.

The consumer then makes the payment, and the vending proceeds as it would in a standard vending machine. If the consumer uses a credit card to pay for the purchase, the OSU 6 scans the magnetic data using magnetic head 225, stores it in SRAM 232, and sends it to the OSU CS 12 to be processed, as will be explained in more detail later. Assuming the credit
10 card is verified, the system will send an "approved" message to the OSU 6, which will then instruct the consumer via display 203 to "select product." If the credit card is not verified, or if insufficient credit remains on the card, the OSU 6 will be so notified by the system. In this case, the display 203 may state "not approved," and the OSU 6 will return to its idle or start condition. Additionally, the OSU 6 preferably reverts to its idle or start condition if any of the
15 steps in the process take an inordinate amount of time.

In any event, once payment has been made in a satisfactory manner, the OSU will generate a "vend enable" signal on "IMDB out" line 240 in the vending machine to enable the purchase. After distribution of the product, the IMDB 96 internal to the vending machine will send a "vend complete" signal to microcontroller 230 on "IMDB in" line 242. At this point,
20 the batch buffer in SRAM 232 is updated, and a message such as "thank you for your purchase" is displayed by display 203 for a time.

Later, for example, during off-hours, the OSU 6 will transmit the batch buffer to the OSU CS 12 for reconciliation, a process step which is particularly useful when dealing with a transaction where payment is made by a credit card. When a credit card is presented for
25 payment, it is presented before the product selection is made. The vending machine may have products being sold at various prices. Therefore, when the credit card is presented, the information on that card is sent to the server to obtain authorization for the purchase of unknown value. A preferable method to implement this credit authorization step is to request authorization for an amount that will allow the customer to select the highest priced item in the
30 vending machine. Once authorization is completed, and when the customer selects a product, the price of that product is recorded in the batch buffer. This buffer, which lists all of the

- 17 -

transactions occurring within the machine over some predetermined period of time, is transmitted to the OSU CS 12 at some time when the machine is not likely being used, say 2:00 AM. The server cluster 18 ultimately sends the batch to a credit card server (such as FSS 14 or other integrated system 24) for reconciliation, whereby the credit card processing
5 company compares the amount authorized to the amount of the actual purchase and charges the credit card account for the actual amount of the purchase. Information concerning cash transactions and DEX information, along with the credit card information, is also used by the server cluster 18 for the generation of system or OSU device 10 reports.

As mentioned earlier, the OSU device 10 can also operate in a batch or disconnect
10 mode, such that the OSU device is either temporarily or permanently disconnected from the system. Operation in these modes may be intentional or may be inadvertent, such as when the system is not functioning or if communication between the system and the OSU device 10 is compromised. In either of these modes, the above flow would be modified accordingly. First, age validation would have to occur locally within the OSU 6, which might increase the
15 processing power or amount of data storage that would be necessary in the OSU device 10. (As will be explained later, optical verification of a driver's license involves the use of algorithms and comparison with image templates, which generally increase the computing power needed for the verification function).

Second, the ability to verify the validity or creditworthiness of a credit card could not
20 be made during the process of the transaction. In this circumstance, and if the system is not responding, payment is preferably handled in two ways. First, the OSU 6 could be configured to receive only cash payments. Second, the OSU 6 could additionally be configured to receive a credit card. In this latter case, the OSU 6 is preferably configured to analyze as much information as is possible to try to validate the transaction. Thus, with the assistance of the
25 microcontroller 230 and information about correct credit card data format stored in memory within the OSU 6, the OSU 6 assesses the form of the credit card data and the expiration date. If acceptable in format, the credit card purchase can proceed. If not acceptable, the consumer may be instructed to pay for the purchase by cash. The transaction and credit card data would be stored in the OSU 6's memory to be later sent to the system or retrieved by an operator to be
30 processed.

- 18 -

C. The OSU Connection Server 12

OSU connection server (OSU CS) 12 communicates with OSU devices 10 using a bi-directional "Davis Terminal Protocol" (DTP) 26, the specifics of which are discussed later in this specification. Essentially, the OSU CS 12 acts as a bridge or proxy for OSU devices 10 with respect to their communication with server cluster 18. The OSU CS 12 can simultaneously handle bi-directions communication with one or many OSU devices over any transmission means capable of supporting DTP 26. One skilled will recognize that OSU CS 12 could constitute a cluster of several servers to prevent any particular server from becoming overworked and to provide redundancy in case a particular server fails. The OSU CS 12 can also be locally or geographically dispersed to enhance system reliability and robustness.

Every time an OSU device 10 queries the system, or the system provides information to the OSU device 10, an "OSU CS session" is created. In this manner, the OSU CS 12 handles communication between the OSU devices 10 and the remainder of the system. The OSU CS 12 can be any suitable server, but in a preferred embodiment constitutes any system that supports the Java 2 platform. Preferably, a commercial embodiment will use an x86-based server running linux 2.4 kernel with external modems connected through standard RS232 serial ports. Although several means of communication are possible between the OSU CS 12 and the remainder of the system (e.g., server cluster 18), it is presently preferred to use Java 2 Enterprise Edition (J2EE) over a TCP/IP connection to establish this communication link.

Depending on the application, OSU CSs 12 may not be necessary, and the OSU devices 10 could instead communicate with the server cluster 18 directly or by any other system using the Davis Terminal Protocol (DTP), which will be described later, or any other suitable protocol.

D. Server Cluster 18

Server cluster 18 essentially functions as the operating system of the Davis system 8. It provides, among other things (1) services to manage the OSU devices 10 and their associated OSU CSs 12, (2) storage for data used by the system, (3) web (internet) application functionality, (4) connectivity to off-system services like VisaNet, and (5) other integrated e-business systems.

One skilled in the art will recognize that server cluster 18 can include databases for storage of necessary system and consumer data, and that such databases can be integral with or

- 19 -

separate from the servers in the cluster. In a preferred commercial embodiment, server cluster 18 comprises (1) four Compaq Proliant systems running RedHat Linux 7.1 with the 2.4 Linux kernel, (2) two servers, each with 1GM of RAM and 50GB of mirrored disk storage provided hosting tasks utilizing JBOSS 3.0 J2EE protocol, and (3) two additional servers, each with 256MB RAM, 25GB mirrored disk storage, and dual external USRobotics modems, for providing hosting tasks to an OSU CS 12. In the preferred embodiment, the four modems are assigned to a single number call pool to which the OSU devices 10 connect. The modems preferably answer calls in a round robin fashion such that if one modem is busy another one in the pool answers the call. However, it should be recognized that while a cluster of networked servers is beneficial to handle overload and to provide redundancy in the event of server failure, server cluster 18 could constitute a single server in a given application.

E. Management Console 22

The management console 22 is essentially the terminal by which the Davis system's administrator accesses the network. In a preferred embodiment, management console 22 constitutes any suitable personal computer or workstation and provides the administrator a user interface for accessing the system. From this console 22, the administrator can list, group, and report information about the various OSU devices 10. For example, assuming the OSU devices 10 are vending machines, the administrator can determine if any of the machines are running low on product. Furthermore, console 22 can be used to configure and deploy software updates for the OSU devices 10 and/or other system components. For example, it is from this terminal that the administrator would deploy a new template specifying the configuration of a particular driver's license (e.g., the state of Texas), so that the system and the OSUs will know how to optically recognize and analyze such a license format.

In a preferred embodiment, limited system administration functionality is available to vending machine or other OSU device 10 operators. In this embodiment, each operator is assigned its own user profile and management console for logging into the system, from which they could add, edit, delete, inactivate, pulls reports on, etc., the OSU devices 10 under their control.

F. Monitor 16

Monitor 16 monitors and maintains communication with critical system functions to increase system reliability. Monitor 16 provides manual and automated means to observe

- 20 -

system functions and respond to system errors. For example, if an OSU CS 12 or OSU device 10 ceases to function properly, monitor 16 detects this error and responds appropriately. Thus, the monitor 16 may reroute communications to a working or redundant OSU CS 12, or page the system administrator. In the event of less critical system errors, monitor 16 may simply
5 record the systems error in a system log that may later be addressed by the administrator.

Monitor 16 registers when a component of the system has come on line. In this respect, system components may broadcast their presence on the system to be picked up by monitor 16, or the components may be configured to register themselves on monitor 16 without further assistance. Once registered and on line, components preferably “ping” monitor 16 at regular
10 intervals to provide a “heart beat” for the system. Monitor 16 may also request a ping or may request information about system functions. For example, the monitor may request an OSU CS 12 to provide the number of active connections with various OSU devices 10 and duration of each connection. In a preferred embodiment, monitor 16 constitutes a server similar to the OSU CSs 12 as described above.

15 **G. Financial Services System 14**

Financial Services System (FSS) 14 provides the system the ability to process account transactions, i.e., the ability for consumers to access their financial accounts in order to make purchases or receive other services on the system.

Several examples exist of financials services supportable by the system. For example,
20 FSS 14 could constitute a credit card payment service, such as VisaNet. In such an embodiment, the consumer would input their credit card into the OSU device 10 and credit for the consumer’s purchase would be effectuated and processed through VisaNet. If the system contains information linking a particular ID card (e.g., a license) to a credit card, such processing may also occur by simply having the consumer enter his ID card into the system,
25 which effectively allows the ID card to work as a credit card on the system.

Additionally, FSS 14 could constitute an aggregation of several accounts of the consumer, such as his credit/debit card accounts or checking or saving accounts. All of these accounts, if registered by the consumer on the system, may be accessible through the system 8 as part of FSS 14. This embodiment allows the system to function as an ATM, whereby a
30 consumer enters his ID card into an OSU device 10 and can withdraw money from his account or perform other financial transactions with his accounts without using his designated bank

- 21 -

debit card. In this embodiment, the OSU device 12 might constitute an ATM machine fitted with an OSU. Likewise, an OSU could be incorporated with cash registers or other point-of-sale machines to effectuate consumer purchases, and allow the consumer access to several of his accounts using a single ID card. Thus, by using his ID card at a point-of-sale terminal, the consumer can be presented with a list of accounts registered on the system, and may select an account to pay for the purchase.

In another embodiment, FSS 14 constitutes a Davis cash account set up by the consumer for use on the system 8. This embodiment is envisioned as being particularly useful in the marketing of low cost items such as candy bars. For such transactions, it may not be sensible to pay for the purchase with a credit card, as the credit transaction fees may be relatively expensive when compared to the cost of the item being purchased.

Using FSS 14, a consumer cash account can be established from which payment for purchases on the system will be drawn. Thus, the system could be used, again in conjunction with the FSS 14, to transfer funds from the consumer's bank account to the cash account, or the cash account could be established by other means, such as sending a check to the system administrator. Thereafter, when the consumer enters his ID card into the OSU device, credit for the purchase will be drawn from his cash account, or the OSU device 10 may present the consumer an option to either have the money so drawn or to provide cash payment to the OSU device 10. Such an embodiment is believed particularly suitable for vending machines, pay phones, slot machines, transportation ticket dispensers, stamp machines, etc. In this respect, it is important to note that the system has flexibility and utility beyond age verification. In other words, the system need not be used exclusively to vend age-restricted products, and whether age verification is required for a particular purchase transaction can be easily controlled by enabling or disabling such functionality using the system.

When dealing with consumer accounts on the Davis system, it is generally preferred that such accounts be accessible through the use of a personal identification number (PIN) to ensure security. In this regard, the OSU device 10 will contain a keyboard or other suitable means for allowing a PIN number to be entered after receipt and optical analysis of the ID card. Suitable PIN numbers may be distributed by traditional means by an administrator of the Davis system. Optionally, and more generally, a "private key" could be used to ensure security, which could comprise a PIN number, some sort of biometric input such as a finger

- 22 -

print, a code generation device containing an internal clock and encrypted algorithms for generating an access code, etc.

H. User Interface 20

User interface 20 generally constitutes a personal computer from which registered
5 consumers can access certain system features, and may be as numerous as the number of consumers that use the system. For example, using interface 20, a consumer can log onto the system (preferably via the web or Internet) to set up a system cash account, to transfer funds between registered accounts, or to check fund balances. Interface 20 can also be used to check product availability at a particular OSU device 10, to check their statuses, e.g., whether such
10 devices are functional at the present time, or to check for the location of OSU devices 10 connected to the system. For security reasons, it is contemplated that consumers be issued passwords and user names that enable them to log on to the system.

Suppose a consumer wishes to use his driver's license to purchase products for sale on a given Davis system. Using user interface 20, the consumer can log onto the Davis system
15 website and register her driver's license by inputting pertinent information from the face of the card, such as name, address, license number, date of birth, etc. (The system may thereafter be made to interface with an appropriate database or other integrated system 24, e.g., the Texas Department of Transportation, to ensure that the entered consumer information is correct). Thereafter, the consumer may be notified by e-mail that the license has been registered, and
20 may be issued a personal identification number (PIN) to use in conjunction with the license at the OSU device 10. At user interface 20, the consumer may also register certain bank accounts on the system, allow money to be transferred or deducted from those accounts, authorize payments for purchases to be made from their credit card (e.g., through FSS 14), or establish a cash account to pay for purchases made on the system. Once the service is activated in this
25 manner, the consumer can use their driver's license to purchase products from any OSU device 10. (It should be noted that registration of the license or ID card may not be necessary for all applications, such as applications in which the consumer will pay for the purchase by standard means, or for services not requiring payment, such as emergency phone calls).

Interface 20 also preferably allows access to others who are not necessarily consumers.
30 For example, interface 20 is contemplated as being accessible by registered operators who service and/or stock the OSU devices 10, such as vending machine product distributors. Such

- 23 -

operators should preferably have special passwords, and may have access to more detailed information in the system not available to the general consumer. Through interface 20, an operator can, for example, (1) add, edit, or remove OSU device 10 information, (2) create an OSU device 10 configuration for remote software updates, (3) provide or retrieve pertinent
5 system data, such as DEX data, (4) manage driver routes, (5) create financial reports, or (6) manage the inventory in the OSU devices 10.

Thus, a registered operator may essentially perform many of the same acts as system administrator, but only for the OSU devices 10 under his control. For example, suppose an operator purchases 20 OSU devices 10 to vend tobacco in a hotel or casino. After creating an
10 on-line account by logging into a user interface 20, he can access to his home page on the system and register each of the 20 vending machines. When the registered devices call into the Davis system, they can synchronize with the operator-configured settings. For example, the devices can be directed to dial in once a week to provide DEX, audit, or reporting data. From this information the operator is able to manage inventory, add optical scanning templates so
15 that the devices will recognize inserted ID cards, and generally control the functionality of his OSU device 10.

I. Integrated Systems 24

One skilled in the art will recognize that the system 8 could be made to interface with other integrated systems 24 to improve or enhance system performance. Examples of
20 integrated systems 24 include VisaNet, law enforcement agencies, etc., and enable the system to act as a subscriber (i.e., to receive information from other systems), a provider (i.e., to provide information to other systems), or a transaction partner (e.g., with VisaNet). Certain systems constituting FSSs 14 may also constitute examples of integrated systems 24.

J. System Geography

It is contemplated that Davis system 8 could be deployed on a nationwide or
25 international basis. Such flexibility is realizable because the system has the capability of recognizing ID cards issued from several different jurisdictions. In such an embodiment, it is preferred that the OSU devices 10 be located nationwide, that OSU CSs 12 be located in certain local regions (such as cities) such that they are capable of serving several different OSU
30 devices 10 within their locales, and that the server cluster 18, monitor 16, and management console 22 be located at a "headquarter" location in the vicinity of the Davis system

- 24 -

administrator. Of course, user interfaces 20, FSS 14, and integrated systems 22 will likely exist somewhere distant from headquarters. Smaller more regional systems are also possible, and the disclosed preferred geographical distribution of the system may be easily be modified depending on system requirements.

5 II. Davis Terminal Protocol (DTP)

As previously mentioned, a specialized protocol is used in the communication between the OSU devices 10 and the OSU cluster servers (OSU CS) 12 called the Davis Terminal Protocol (DTP) (see Figure 1, element 26). After researching several available communication protocols it was determined that none of them met the requirements for the Davis system 8,
10 such as:

- leverage existing communication layers: It is foreseen that OSU devices 10 may be plugged into existing communication infrastructures such as TCP/IP, PPP, etc. DTP provides a layer of abstraction that insulates OSU device development from current protocols and their evolution.
- 15 • reliable communications: DTP has proven highly reliable, an important feature as data loss during a transaction in a Davis system is unacceptable.
- full-duplex communication: DTP allows either side (i.e., the OSU device 10 or the OSU CS 12) to send or receive data.
- communication initiation: In a typical transaction, the OSU device 10 will
20 initiate communication. DTP, however, may also be used in future embodiments to allow the OSU CS 12 to initiate communication. Such two-way initiation was not well supported by existing protocols, but it achievable using DTP.
- byte stream oriented: Transactions must be capable of sending any type of data such as binary, string, numeric, etc. DTP supports such flexible streaming of data.
- 25 • packet oriented: Because DTP is a packet-oriented protocol, it provides flexibility with regard to the size of data to be transmitted. Therefore, larger or smaller pieces of data may be sent depending on the bandwidth available. Packet oriented communication also provides for reliable communication and assists in handling transmission errors. Thus, when smaller packets are used to transmit a single block of
30 data, if an error occurs, only the packet containing the error needs to be sent, increasing transmission efficiency.

- 25 -

- “lightweight”: DTP transmits data with little protocol-related overhead.
- simple and fast: Due to the variety of embodiments that an OSU device 10 might take, it is likely that any communication protocol to be used with it will need to be re-implemented many times and configured with different parameters depending on the data transmission requirements. It is therefore advantageous to be able to quickly
5 implement a new device that is able to communicate with the server. While TCP/IP was thought originally to be a suitable protocol candidate, it was determined that this protocol was not suitably “lightweight,” was not simple or fast to implement, and did not provide an important abstraction layer for OSU software development. DTP
10 squarely addresses these concerns, and was therefore determined to be a suitable candidate for use in the Davis system. One skilled in the art will notice however that DTP borrows certain technical concepts from TCP/IP, but tuned in such a way to make its implementation in the Davis system optimal. (Due to the limited resources of the modem-based communication channels that are preferably used in the system, it is not
15 feasible at this time to use the standard TCP/IP or TCP/PPP protocols that requires wider bandwidth than DHP/DMP, but this may change as technology progresses.)

In the current embodiment, the Davis system 8 uses the DTP protocol layered on top of the industry standard RS232 protocol for serial communications. DTP is itself composed of two layers: the Davis middle level protocol (DMP), and the Davis high level protocol (DHP).
20 Written together, communication protocol for the Davis system thus consists of a DHP/DMP/RS232 stack, although any lower level communication protocol could support the DHP/DMP stack disclosed herein. It is currently preferable in a commercial embodiment to use the V22 modem protocol, and thus the entire communication stack may be written as DHP/DMP/RS232/V22 or simply DTP/V22. Later, DTP can easily be upgraded in a
25 commercial embodiment to the DTP/TCP/IP or DTP/TCP/PPP combinations when technological advances allow.

The different layers in the DHP/DMP construction perform different functions independent of the other layers. Each layer of the protocol performs services for the layer above it and provides services to the layer below it. When two devices are communicating,
30 each layer of the protocol stack communicates with the same layer of the protocol stack on the other device. Figure 7 identifies three distinct communication phases that are utilized in DTP.

- 26 -

In Phase 1, an OSU device 10 communicates with the Davis server system (i.e., either OSU CS 12 or server cluster 18) and requests one of its services. It does so by calling one of the routines available in the DHP API (application programming interface). The DHP routine in turn forwards the request to the DMP layer. The DMP layer then forwards or repackages the request on to the native communication channel such as RS232 (and preferably V22). In Phase 2, the native communication channel relays the request from the OSU Device 10 to the Davis server system. In Phase 3, the Davis server system accepts the request and forwards it on to the receiving DMP layer. The DMP layer then passes the request on to the DHP layer, followed by the OSU CS 12 proxying the request on to the server cluster 18.

The three phases will repeat, now in the reverse direction, to allow the system to send a response to the OSU device 10. While this example assumes that the OSU device 10 has made the request to the system, the system may also make requests to the OSU device 10, thus allowing for asynchronous, bi-directional communication.

The DHP and DMP provide communication services independent of one another, and hence generally provide different functionality. Preferably, DHP provides APIs such as login requests, transaction requests, and update requests. By contrast, DMP provides for data packet field and segment definitions, handshaking, and other lower level tasks.

A. DMP

DMP provides reliable, full-duplex, byte stream-oriented service. It accepts data from the DHP layer, divides DHP segments into a set of DMP segments and adds a header to each segment, and sends individual segments to the modem. If the size of a segment exceeds the maximum payload capacity, DMP divides the segment into several segments and sets the sequence number field in the header for the benefit of the receiving system. The capacity of DMP data payload varies from 0 to 255 bytes per segment. DMP is also free to retransmit a single segment of 200 bytes as two segments each containing 100 bytes of data.

When a transmitted segment is received by the other system (e.g., OSU CS 12), DMP checks the sequence number in the header to verify that number of segments that carry a particular unit of data. When the expected number of segments is received, the receiving system retains the data for analysis or other processing and sends an acknowledgment back to the sending system (e.g., OSU device 10). The acknowledgment field in the header of the acknowledgment message contains the sequence number in the received data segment. To

- 27 -

verify that a segment was received without errors, DMP uses the checksum field, which contains the sum of all segment bytes, with the exception of the last two bytes containing the check sum.

The preferred format for the DMP data segments is shown in the below Table 1:

<i>Table 1: DMP Header Format</i>		
<i>Field</i>	<i>Size(bits)</i>	<i>Description</i>
Version	7	Specifies the protocol version and verifies that the sender, receivers are using a current version of the protocol. Preferably 0x01.
ACK Flag	1	1 if the previous segment was received without errors.
Sequence Number	8	Identifies the position of the data in the senders bit stream.
Acknowledge Number	8	The number of the last received sequence.
Length	8	Specifies the length of the data in bytes.
Data	Varies	
Checksum	16	The sum off all the bytes in the segment (used for error correction)

B. Handshaking

When an OSU device 10 and a server desire to communicate, they must first “handshake.” DMP uses a 2-way handshake to initiate a connection, a process that ensures that both devices are ready to transmit data, and that both devices know that the other is ready to receive data before the transfer actually starts. The procedure works as follows: sending device A (e.g., OSU device 10) sends a segment to device B (e.g., OSU CS 12) wherein Sequence Number = 0, and ACK_FLAG = 0. When device B receives the segment from device A, and if device B is ready to communicate with A, it sends a segment to A wherein Sequence Number = 0, Acknowledge Number = 0, and ACK_FLAG = 1. Thereafter, device A may transfer data to device B.

Note that a segment may be sent or received from either end at any time. If an acknowledgment (i.e., ACK_FLAG = 1) is not received for a non-zero length segment after a timeout of 2 seconds, the segment will be retransmitted. If the segment was retransmitted 3 times and the acknowledgment was not received, the connection is terminated.

- 28 -

C. DHP

Like DMP segments, every DHP segment has a structure that includes a header and associated data. With respect the DHP header, the first byte (i.e., eight bits) specifies the version of DHP protocol (4 bits) and type of data (4 bits). The next word (16 bits, or two bytes) specifies the length of the data within the segment, which preferably can be as large as 64K bytes. The rest of bytes in the segment constitute the data. This segment structure is shown in the below Table 2:

<i>Table 2: DHP Header Format</i>		
<i>Field</i>	<i>Size (bits)</i>	<i>Description</i>
Version	4	Version of the DTP protocol.
Type	4	Type of data: 0 – Login Request 1 – Login Response 2 – Transaction Request 3 – Transaction Response 4 – Transaction Commit 5 – Transaction Commit Response 6 – Update Request 7 – Update Response 8 – DEX Submit 9 – DEX Response 10 – Logoff Request
Length	24	Specifies the length of the data
Data	Varies	

There are two types of DHP segments, those that store payload data in an ASCII string format and those that store data in a binary format. Binary format is a sequence of bytes used to represent any type of data, such as numbers, bit-map images, or complex data structures. String data is a sequence of bytes used to represent ASCII characters, which is a more convenient way to represent some systems data such as birth date, person name, or an ID number. An example of a string format might be “propertyName1 = value1; propertyName2 = value2,” and a more specific example for a “Transaction Response” packet may look like “trn=1234567; time=09/27/01; err=0”, where different properties are separated by a semicolon character ‘;’ and a property name and property value are separated by an equal sign character

- 29 -

'='. Each of the eleven types of exemplary segments illustrated in the above table is summarized below, along with a description of their function. One skilled will realize that other segment types, carrying different forms of data for a variety of purposes, could easily be implemented, depending on the requirements of the application.

- 5 • Type 0 - Login Request (string packet): Before an OSU device 10 can commence a session with the system server (e.g., OSU CS 12) it must login by sending a Login Request segment. The data that accompanies this segment includes "sn," which denotes the serial number of the inquiring OSU device 10, and "rc6," which is a random number that is to be RC6 encrypted with the Davis system master key and
10 challenge.
- Type 1 - Login Response (string packet): After the server receives the login request from the OSU device 10, it sends a Login Response segment. The data accompanying this segment includes "busy," which equals '1' if the server is too busy to update the client, and "rc6."
- 15 • Type 2 - Transaction Request (string packet): This segment is used by the OSU device 10 to send the customer credit card and/or driver license information to the server. The data accompanying this segment includes "dln," the driver's license number, "dlname," the name on the license, "dldob," the date of birth on the driver license, "dlexp," the expiration date of the license, "dlst," the state in which the license was issued, "ccn,"
20 the credit card number, and "ccexp," the credit card expiration date.
- Type 3 - Transaction Response (string packet): When the server receives the Transaction Request segment, and assuming for example that this segment contains credit card data, the server checks the credit card information, sends the request to VisaNet or other FSS 14, and sends a Transaction Response segment to the OSU device
25 10, which includes "trn," a transaction number, "time," the current time, which can be used automatically by the OSU device 10 to update its clock, and "err," an error code (optional).
- Type 4 - Transaction Commit (string packet): After the OSU device 10 receives the Transaction Response segment, it vends the product to the customer and sends the
30 Transaction Commit segment to notify the server that the transaction has been committed. Data accompanying this segment includes "trn".

- 30 -

- Type 5 - Transaction Commit Response (string packet): The server sends the OSU device 10 this segment as confirmation of receipt of the Transaction Commit segment. If the OSU device 10 does not receive the Transaction Commit Response before terminating the connection to the server, it will resend the Transaction Commit again during the next session. No data accompanies the sending of this segment.
- Type 6 - Update Request (string packet): The OSU device 10 preferably sends this segment periodically (e.g., every 24 hours) to request configuration and software updates. Accompanying data includes "ver," which denotes the OSU configuration version.
- Type 7 - Update Response (binary packet): After the server receives the Update Request segment, it checks to see if the OSU device 10 needs to be updated, and if so, sends an Update Response segment containing the latest OCR templates and any other necessary OSU software. (OCR templates will be explained in a later section of this specification). Every Update Response segment constitutes a chain of one or more update units that add, update, or remove various parts of the OSU software. There are six types of units: "Font Update," which replaces the font template if it is already installed on the OSU device 10 or adds one if it doesn't exist, "Font Delete," "Header Image Update," which replaces or adds header templates, "Header Image Delete," "Form Update," which replaces or updates the form template, and "Form Delete".
- Type 8 - DEX Submit (binary packet): The OSU device uses this segment to periodically (e.g., every 24 hours) send DEX data to the server.
- Type 9 - DEX Response (string packet): The server sends this segment to the OSU device 10 to indicate that the DEX Submit segment was successfully received and saved in the database. Data accompanying this segment includes "saved," which equals '1' if the save was successful.
- Type 10 - Logoff Request Payload (string packet): The OSU device 10 sends this segment to notify the server that it wants to finish the current session. No data accompanies the sending of this segment.

Other segments are possible, such as segments used to update product information,

such as product pricing.

- 31 -

D. Example of DMP Communication Protocol

As an example of the operation of DTP, including DHP and DMP, the below Table 3 provides the data packet sequence to show how two devices (A and B, preferably OSU device 10 and OSU CS 12) login and logout using DTP. In this example, serial number for device A is 987654321.

<i>Table 3</i>		
<i>Bytes</i>	<i>Protocol</i>	<i>Description / Data</i>
<i>A sends Login Request packet</i>		
2	DMP	DMP Version 1, ACK_FLAG = 0
0	DMP	Sequence Number 0
0	DMP	Acknowledge Number 0
16	DMP	Size of payload DMP
16	DHP	DHP Version 1, Login Request Packet
0	DHP	higher bite of the offload size
0	DHP	middle bite of the offload size
12	DHP	lower byte of the offload size
115	DHP	S
110	DHP	N
61	DHP	=
57	DHP	9
56	DHP	8
55	DHP	7
54	DHP	6
53	DHP	5
52	DHP	4
51	DHP	3
50	DHP	2
49	DHP	1
3	DMP	higher bite of the checksum
41	DMP	lower bite of the checksum

<i>Table 3</i>		
<i>Bytes</i>	<i>Protocol</i>	<i>Description / Data</i>
<i>B sends Login Response packet</i>		
3	DMP	DMP Version 1, ACK_FLAG = 1
0	DMP	Sequence Number 0
0	DMP	Acknowledge Number 0
4	DMP	Size of payload DMP
17	DHP	DHP Version 1, Login Response Packet
0	DHP	higher bite of the offload size
0	DHP	middle bite of the offload size
0	DHP	lower byte of the offload size
0	DMP	higher bite of the checksum
24	DMP	lower bite of the checksum

- 32 -

<i>A sends Logoff Request packet</i>		
3	DMP	DMP Version 1, ACK_FLAG = 1
1	DMP	Sequence Number 1
0	DMP	Acknowledge Number 0
0	DMP	Size of payload DMP
0	DMP	higher bite of the checksum
4	DMP	lower bite of the checksum

III. Optical Character Recognition (OCR)

As noted previously, a desirable advantage of the disclosed system is its ability to receive data from a consumer through optical, non-electronic means, e.g., from the printed text on the face of an ID card such as a driver's license. This enables the consumer's driver's license, in conjunction with the OSU, to operate as would a standard credit card containing a magnetic strip or a "smart card" containing integrated circuitry. This is a desirable way of obtaining consumer information, such as birth date, driver's license number, social security number, or the consumer's name. Indeed, when dealing with driver's licenses, optical analysis of the license may be the only reasonable way to automate information procurement, as not all states' licenses contain magnetic strips, and the magnetic data on the various states' licenses are encoded in differing formats.

With this in mind, a focus of the disclosed system has been to provide an optical analysis algorithm capable of recognizing and analyzing the textual printing on the face of the driver's licenses of all fifty states. Of course, the system is not so limited, and could be configured to recognize other textual forms of consumer identification. An analysis of driver's license is disclosed merely as a preferred embodiment.

This section III discloses basic considerations of how template based character recognition can be performed in the disclosed Davis™ system. Section IV, which follows, discloses methods and systems for optical analysis of more complex optical elements, such as holograms. Section V, which follows further below, discloses additional methods for template based character recognition with special focus concerning schemes for reducing the processing power and storage requirements needed during analysis. One skilled in the art having the benefit of this disclosure will understand that the techniques disclosed in these sections III, IV, and V are not mutually exclusive and can be combined or modified as desirable for a given application. Accordingly, certain details appearing in one section may have applicability to another section, even if not expressly noted.

- 33 -

A. Background

Textual data are often arranged in forms. The consistent, regular organization of a form or report makes it easy to obtain desired information very quickly. For example, the organization of a phone book makes it easy to find a specific telephone number. Other
5 examples of forms include paycheck stubs, business cards, telephone bills, stock reports, insurance cards, credit cards, passports, visas, and driver's licenses. It is the consistency of the organization that makes the form useful.

It is often the case that a transaction involves or is conditioned upon an exchange of information between buyer and seller. One example has already been given. A liquor store
10 clerk must verify the age of the consumer prior to a transaction. The consumer's driver's license (a form) provides the necessary information. A transaction for medical services provides another example. When a consumer receives services from a doctor, she shows her insurance card (a form), which provides the needed information to the doctor to bill the insurance company.

In many transactions that involve an information exchange involving a form, a human
15 operator reads the information and either immediately acts upon it (by allowing the purchase of alcohol) or transfers the information from the customer's form (e.g., an insurance billing number) to a computer. This can be a laborious and error prone process. This function is normally performed by a human operator because humans can read forms and computers
20 typically cannot. There is therefore a need to enable computers with the ability to read forms, such as driver's licenses. This section describes methods believed to be novel for doing so. One skilled in the art will recognize that these methods are easily implementable on a computer, such as those provided in the disclosed system, and could be coded in any number of ways to perform the tasks described herein.

B. Template-Based Character Recognition

The preferred method for optically determining the textual information printed on the face of an ID card, such as a driver's license, employs the concept of template-based character recognition. According to this scheme, one starts with an unknown character or other image, such as a letter or a picture, and compares an optically scanned version of that character or
30 image to a series of templates. The templates are compared to the scanned character or image to determine the extent of the "overlap" of each template. The template with the smallest

- 34 -

degree of overlap, i.e., the one which “lines up” with the scanned image, is chosen as the template that matches, and therefore determines, the scanned image. Of course, because the template and the scanned image may be differently centered, the template may need to be slid (e.g., up and down, and from left to right) with respect to the scanned image to ensure that the degree of overlap is accurately assessed.

Template-based character recognition involves two tasks: the recognition task itself and the task of template creation. This disclosure improves upon both of these aspects of template-based character recognition in ways that are discussed below.

With respect to the recognition task, assume that a scanned test image, such as a scanned driver’s license, contains a two-dimensional array of M by N pixels, and that $D(i,j)$ represents the intensity of a particular pixel (i,j) , preferably a gray scale value ranging from 0 Hex to FF Hex (i.e., from 0 to 255). Assume further that there is an unknown character starting at coordinate (r,s) in the test image that represents one of K possible characters represented by K templates. (The procedure for generating the templates will be disclosed later). These templates are denoted $T_k(i,j)$, wherein $k = 1, 2, \dots K$. The vertical and horizontal dimensions of the k^{th} template are denoted by m_k and n_k respectively.

Template matching involves comparing each of the K templates to the test image and choosing the template that is “closest” to the test image to determine the unknown character at (r,s) . This is accomplished by calculating the least-squares “distance” between the test data $D(i,j)$ and the templates $T_k(i,j)$, which is a way of quantifying the extent of the overlap between the template and the unknown character. This distance $\text{dist}_k(r,s)$, can be defined as:

$$\text{dist}_k(r,s) = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} (D(r+i, s+j) - T_k(i,j))^2 \quad (\text{eq. 1})$$

For convenience, it has been assumed that $M \gg m_k$ and $N \gg n_k$. This is a reasonable assumption because the unknown character is typically embedded in a large scanned image (e.g., several inches in both dimensions) while the size of the template is equal to the actual character size (about one tenth of an inch in both dimensions).

- 35 -

As noted, the metric provided by Equation 1 gives the distance between the template and the test image starting at coordinate (r,s). The template that provides the minimum distance in this equation is the “winner” and is chosen as the template that represents the character under analysis. If the character under analysis is the k^{th} character, then $\text{dist}_k(r,s) = 0$; in other words, the character and the template match exactly, an ideal situation.

However, in practice, the test character as scanned will probably be corrupted by noise, sampling artifacts, or other distortion. Additionally, each of the pixels of the scanned characters will preferably be represented by a gray scale value, which may have poor contrast—i.e., the image may constitute just a few shades of gray. This will cause this distance metric to be non-zero for the matching template, but hopefully small, especially in comparison to the other $K-1$, non-matching (incorrect) templates. However, such discrepancies can lead to errors in the recognition process, and may cause the distance for a non-matching template to be smaller than the distance for the correct template, resulting in an error and incorrect recognition.

To relieve these problems, it has been discovered that it is desirable to vary equation 1 to reduce error that might be attributed to gray scale variations as follows:

$$\text{dist}_k(r,s) = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} (D(r+i, s+j) - [\alpha T_k(i,j) + \beta])^2$$

In this equation, fitting parameter α scales the intensity of the template while fitting parameter β denotes a constant intensity bias. This approach is believed to be novel in that these parameters adjust the contrast of the template to match the contrast of the test data. Convenient expressions for fitting parameters α and β which result in a minimal distance can be computed using ordinary calculus:

$$\alpha = \frac{m_k n_k A - BC}{\Delta}$$

$$\beta = \frac{\Omega C - AB}{\Delta}$$

- 36 -

where,

$$A = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} D(r+i, s+j) T_k(i, j)$$

$$B = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} T_k(i, j)$$

$$C = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} D(r+i, s+j)$$

$$\Omega = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} T_k^2(i, j)$$

$$\Pi = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} D^2(r+i, s+j)$$

$$\Delta = m_k n_k \Omega - B^2$$

Therefore, the minimum distance corresponding to the optimum α and β is

$$dist_k(r, s | \alpha, \beta) = \Pi - \alpha A - \beta C \quad (\text{eq. 2})$$

Significant advantages are achieved through the use of this modified distance metric. First, in comparison to a traditional least-squares formulation, the above formulation only requires one pass through the data to determine the optimal α and β using the above equations, resulting in significant computational savings. By contrast, in a traditional least squares formulation, two passes would be required to determine the fitting parameters α and β . In the first pass, the average value of the image data $D(i, j)$ would be calculated. In the second pass, the variance of that data would be calculated. Because the variance calculation depends upon the average value, these two calculations must be done in sequence, and cannot be done simultaneously in one pass.

Second, because this formulation, via fitting parameters α and β , adjusts the intensity levels of the template to match the test image, the intensity of a stored template is of no importance. In other words, the templates do not have to be stored with gray scale values, and can instead be more efficiently stored, such that every pixel in a template $T_k(i, j)$ is denoted by either a logic '0' or a '1' (representing completely dark and light pixels). In other words, the templates can be stored as black and white images, without the need for storing gray scale values, typically eight bits per pixel (i.e., from 0 Hex to FF Hex). Additionally, "quantization"

- 37 -

of the templates results in significant computational advantages because it turns many of the necessary multiplication steps into logical operations that are more efficient. Consider for example the calculation of parameter "A" above, which represents the sum of products of $D(r+i,s+j)$ and $T(i,j)$. Although the values for $D(r+i,s+j)$ represent grayscale values, e.g., from 0 to 255, $T(i,j)$ represent either ones or zeros. Therefore, "A" is really just the sum of all $D(r+i,s+j)$ when $T(i,j)$ is equal to one. No multiplies are required, except in the calculation of "II." (Note that parameters "B" and "Ω" depend only on the template, $T_k(i,j)$, and are computed in advance and stored in the template data structure for use during recognition). Some loss of accuracy results from this template "quantization" step. However, for images sampled at 400 dots-per-inch (dpi), this loss of accuracy should not lead to an intolerable error rate.

As mentioned earlier, the procedure for matching a template in the vicinity of the test character at coordinate (r,s) is to "slide" the templates horizontally and vertically with respect to the test image until a best fit is found, preferably pixel by pixel although other prescribed offsets could also be used such as every other pixel. At each offset for a given template, the fitting parameters α and β are calculated according to the formulas given above, and the distance is calculated for each offset. This yields several distance calculations for each template, corresponding to each slide of the template, and the smallest of these distances is kept as the minimum distance for each template. Each of the minimum distances for each template are then compared to determine the template with the smallest minimum distance, such template being determined as the matching template which determines the character at (r,s).

For larger templates, the template matching algorithms can become computationally demanding and a less computationally-demanding algorithm may be required. For this purpose, a modified distance metric can be used which only compares a subset of the pixels $T_k(i,j)$ in the template with the pixels $D(i,j)$ in the test image. This modified distance metric is represented as

$$dist_k(r,s|\alpha,\beta) = \sum_{p=1}^P \left(D(r+i_p, s+j_p) - [\alpha T_k(i_p, j_p) + \beta] \right)^2$$

This reduces any given distance measurement down to an assessment of P terms. The set of points (i_p, j_p) at which the distance is calculated is determined in advance and is optimized for

- 38 -

best performance. This procedure is called "fast" template matching and is preferably only used for large templates. These "fast" templates can be stored more efficiently than the full test image.

C. Template Training

5 To be able to optically "read" pertinent information on, for example, a driver's license, it has been discovered that it is beneficial to allow the system to "learn" the template corresponding to a driver's license of a particular state, rather than "feeding" the template into the computer in the first instance. This procedure can increase the accuracy with which optical recognition of characters on the license is determined when compared with pre-fed templates, 10 which may or may not accurately reflect the true structure of the "form," and which may not be able to handle variations in the elements on the license. However, while this training approach is believed novel, template training is not specifically necessary to the implementation of the disclosed invention, and pre-fed templates (i.e., templates generated off-line and in advance of recognition) may work adequately.

15 Template training involves using example characters to generate a character template $T(i,j)$. Throughout the training process, it is assumed that a set of scanned forms is available. For example, if the problem presented is character recognition for a Texas driver's license, then we will assume that several, e.g., 30, different Texas driver's licenses have been scanned into the computer. This driver's license image data will be used during the training process. 20 During template training, the driver's license data will be used to obtain examples of each character. For example, if we wanted to create a template for the character "5," we would look through the inventory of 30 scanned Texas drivers licenses and extract all the examples of the character "5" to form the template. Note that an operator must review the scanned license to isolate those portions of the larger image that contain the image for the number "5" in order to 25 provide the examples necessary to "train" the "5" template. This is a time consuming process which can be automated somewhat by a computer or workstation.

As the generation of only a single template is referred to, the index ("k") has been dropped from the notation. Let $A_1(i,j)$, $A_2(i,j)$, . . . $A_N(i,j)$ represent examples of a particular character isolated from the set of sample licenses. The template $T(i,j)$ will preferably 30 recognize all of the given examples as if they were actually embedded in a test image. Therefore, the template is chosen to minimize the distance between the template and each of

- 39 -

the examples. Due to uncertainty in the sampling phase and other anomalies, the examples must be shifted until they are all aligned. The total error or distance between the template and the examples is expressly mathematically as

$$\sum_{k=1}^N \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} (T(i, j) - A_k(r_k + i, s_k + j))^2$$

5 The offsets (r_k, s_k) are adjusted until a minimum of the total error is reached. At the minimum, the template is given by the average of all the examples, which is expressed mathematically as

$$T(i, j) = \frac{1}{N} \sum_{k=1}^N A_k(r_k + i, s_k + j)$$

This formula can be updated recursively as new examples are found. Thus, suppose $A_N(i, j)$ represents a new example. When this new example is shifted until a best fit (i.e., minimum distance) is achieved, a new offset (r_N, s_N) is provided. The template formula can then be
10 updated as follows:

$$T(i, j) = \frac{N-1}{N} T(i, j) + \frac{1}{N} A_N(r_N + i, s_N + j)$$

D. Sequence Estimation

Information in a form is typically represented by more than just a single character. The
15 information of interest may be a date, a number or amount, a word, a name, etc. These types of information are represented by a sequence of characters. A sequence estimation algorithm uses the character recognition algorithm of the previous sections to recognize the individual characters of a word, number, or other string of characters. The sequence estimation algorithm must also be able to detect the end of a variable length string of characters.

20 Sequence estimation takes as its input a pattern specification. The pattern specification defines specific characters, or more generally types of characters, that are present in a string of characters. Various different characters include numbers, capital letters, lower-case letters, punctuation, delimiters, and symbols. Character types include "wild cards" (designating any particular character), letter type (whether upper or lower case), and alphanumeric type (any
25 letter or number). Character types may also include other symbols, for example, the seal appearing on a driver's license. A pattern specification also contains information on the minimum and maximum number of characters that can occur within a particular test image.

- 40 -

Take for example the birth date on a Texas driver's license, which takes the following format: MM-DD-YY, where MM represents the month, DD represents the day, and YY represents a year, and where each of these designator is separated by a dash "-". In this format, both the month and the day may be either one or two characters in length (compare 9-1-75 with
5 11-12-75). Thus, a pattern specification for the date would look like

N[1:2] "-" N[1:2] "-" N[2:2]

The "N" denotes that the particular field contains numbers, and [1:2] denotes a sequence with either one or two characters. Together, N[1:2] denotes that the computer should look for a variable length sequence of numbers of either one or two characters in length (the month).

10 Continuing through the pattern specification, the computer next looks for one dash character "-", followed by another variable length sequence of numbers of either one or two characters in length (the day), followed by yet another dash. Finally, the computer looks for the last sequence, which necessarily constitutes a two-character numerical sequence (the year). This exemplary pattern specification consists of five elements, referred to as pattern characters,
15 although two of these pattern characters (N[1:2] and "d") are repeated for this particular pattern specification.

Consider as another example the consumer's name as printed on the face of Texas driver's license, and assume that the name is written in all capital letters with the first name first and the last name last. A suitable pattern specification should be able to describe the name
20 "ALEXANDER PEABODY" as well as "JON DOE," even though these names are different in length. Such a pattern specification might look like

A[1:64] " " A[1:64]

Here, the "A" designates a capital letter. So, this pattern tells the computer to look for between one and sixty-four capital letters in the first name, followed by a space, followed by between
25 one and sixty-four capital letters in the last name. Again, this pattern specification consists of three pattern characters.

If lower case letters were used then the letter "a" could be used to designate the lower case alphabetic character type. Thus, if a name were printed using capital letters for only the

- 41 -

first letter of each name, and if the last name were printed first and separated from the first name by a comma and a space (e.g., "Lewis, Terril"), a suitable pattern specification might look like

A[1:1] a[1:63] " , " " A[1:1] a[1:63]

5 As noted earlier, the sequence estimation algorithm uses the pattern specification to determine what sets of templates to use when performing character recognition. Therefore, in the last given example above, sequence estimation will utilize 54 different templates to assess the consumer's name: 26 Texas license "capital letter" templates, 26 Texas license "lower case letter" templates, and Texas license templates designating the comma and space. For this
10 example, the pattern specification contains four pattern characters.

There are two methods for sequence estimation: maximum likelihood sequence estimation (MLSE) and symbol by symbol detection. MLSE essentially builds a tree of all possible patterns allowed by the pattern specification. Every combined pattern is tried and the best matching pattern is the winner. Performing this comprehensive search is time consuming
15 but can be efficiently implemented in a given application if necessary.

As an example of MLSE, suppose the computer is provided a pattern specification "N[2:3]," denoting the analysis of a sequence of numbers that is either two or three numbers long. There are 1100 different sequences that fit this specification: 00, 01, ..., 09, 10, 11, ..., 19, ..., 99 (i.e., 100 two-number sequences), and 000, 001, ..., 009, 010, 011, ..., 019, ..., 099,
20 100, 101, ..., 999 (i.e., 1000 three-number sequences). In MLSE, the computer would concatenate together the image templates for each of these 1100 sequences, would compare each of these concatenated templates with the single test images of the characters under analysis, and would choose the one with the best match using the template matching algorithm disclosed above. In each case, the whole sequence of characters is compared as if it were one
25 image as opposed to comparison of the individual characters.

While not as comprehensive, symbol by symbol detection generally performs as well as does MLSE when the distortion in the given image is low, which is usually the case in form processing, and therefore is the preferred approach employed by the OSU. In symbol by symbol detection, character recognition proceeds in a linear sequential fashion through the
30 character string under analysis. Consider again the pattern specification of N[2:3]. Employing symbol by symbol detection, the computer would look at the specification and would see that

- 42 -

the first character must be a number. The computer would perform template matching, as disclosed above, using the templates for the characters 0 through 9, and choose the best match. Suppose that the best matching template for the first character was "5". The computer would then again consult the specification and see that the next character must also be a number. It would therefore again perform template matching and choose the best match. Suppose that the best matching template for the second character was "4," so that, thus far, the sequence "54" has been recognized. Next the computer would look at the specification and see that the next character may be a number, but may also be a space (" ") because the specification indicates that the sequence may be either two or three numbers in length. Accordingly, when performing pattern matching, the computer consults the templates for 0 through 9, and additionally consults a space template (which would be a blank template). Suppose that the best matching character was " ". Then the computer ultimately determine that the sequence under analysis was "54". Suppose, on the other hand, that the best matching character was "3". Then the computer would ultimately determine that the sequence under analysis was "543."

Representing a particular element pursuant to a pattern specification is beneficial in that it reduces the number of character (or symbol) template comparisons that need to be used in the analysis of a given element. Take, for example, the "lastname, firstname" pattern specification of A[1:1] a[1:63] ",", " " A[1:1] a[1:63] discussed earlier. As noted, this pattern specification requires the use of 54 templates to perform an analysis of the alphabetical string "lastname, firstname." Were a pattern specification not used to assist in the analysis, each character in the name under analysis would potentially need to be compared against each of the 54 templates. For even a short name, like "Li, Tan," consisting of five letters, a space, and a comma, this exercise could involve $54 * 7$ template comparisons, which could be very computationally demanding and slow. By providing the algorithm, through the pattern specification, information concerning the expected characters in the element, the number of comparisons is greatly reduced. For example, determination of the first letter in the name requires comparison to only 26 templates, i.e., the upper case templates, and the sequence estimation algorithm may ignore the lower case letter templates, the space template, and the dash template. By the time the analysis is completed for this example, the number of comparisons is approximately cut in half. This results because the pattern specification references only a particular subset of templates to be used at certain points in the analysis.

E. Form Decomposition

Although the disclosed character recognition techniques may be used with a variety of forms, a driver's license is used as the example form in the following discussion due to its utility in the disclosed Davis system.

5 As shown in Figure 8, a driver's license contains many different pieces of information, including: the license (form) header 50, which identifies the state in which the license was issued (e.g., "Texas"), (2) data 52, such as the holder's name, address, date of birth, driver's license number, and expiration date, (3) a holder ID photo 54, and (4) a validation seal 56, used to verify the genuineness of the license. For a particular state, the information is arranged on
10 the card at various known locations. The date of birth, for example, is always located in the same general vicinity on a Texas driver's license.

To process the driver's license, the license is decomposed into three hierarchical levels, called "form," "cluster," and "element." An element 58 denotes a single piece of information, such as the date of birth. A cluster 60 denotes a group of elements, or possibly a single
15 element, that occur near each other on the license. For example, the license class, the date of birth (DOB), the expiration date, license restrictions (REST), and "END," may all represent elements 58 within a single cluster 60. A form 62 denotes a group of clusters, and typically represents the entire image under analysis.

The form and each cluster typically have "headers" with which they are associated. For
20 example, form header 50 on the Texas driver's license reads as "TEXAS." Several pieces of graphical information within cluster 60 could operate as cluster header 61, such as "CLASS:", "DOB:", or even possibly the graphic of the Texas flag above these elements, were this graphic to be contained within cluster 60. For simplicity, and unless otherwise noted, it will be assumed that "DOB:" operates as the cluster header 61 for the cluster 60 illustrated in Figure 8.

25 The form header and the cluster headers contain, respectively, a form header origin and cluster header origins. The form header origin 63 and the cluster header origins (e.g., 65) are represented, respectively and preferably, by the upper-left-most pixel in the form header and the given cluster header. The form header origin is determined with reference to the upper-left most pixel in the entire scanned image, which is referred to for convenience as the image
30 origin 67. Thus, if the image origin has horizontal and vertical coordinates of (0,0), and if, for example, the entire image is 1000 pixels in the horizontal direction and 768 pixels in the

- 44 -

vertical direction, the form header origin 63 for the form header 50 in the exemplary Texas driver's license shown in Figure 8 might be approximately (400,20).

The cluster header origins are determined with respect to form header origin. In this respect, once the form header origin is known, that origin operates as the "master" origin from which the clusters are located. Relating the cluster header origins to the form header origin, as opposed to the image origin, assists in the subsequent optical analysis of the clusters in the event that the printing on the license has been uniformly shifted in a given direction. Thus, if the form header origin 63 is "reset" to (0*,0*), the cluster header origin 65 for the "date of birth" cluster might be at approximately (-350*,180*) with respect thereto, or approximately (50,200) with respect to the image origin. Of course, in a given application, the image origin can be used as the reference point for location of both the form header origin and the cluster header origins.

The location of each element, as defined by element origin 69, can be known with reasonable precision within a given cluster, and is determined with reference to the cluster header origin. An analysis of driver's licenses shows that there is a high variability (plus or minus 15 pixels) in the position of clusters relative to the form header origin but very small variability (plus or minus 1 pixel) in the position of elements relative to its cluster header origin. This provides the motivation for decomposing the form as described above.

F. Template Training in Practice And Exemplary File Structures

Figures 9A and 9B show a simplified illustration of the organization of the various computer files or data structures that are used by the disclosed OCR algorithms and a description of their contents. One skilled in the art will recognize that the files necessary for OCR as disclosed herein could be organized and structured in numerous ways. Indeed, Figure 17 represents a broader disclosure of the organization of the data structures as they are preferably applied in a commercial system. Figures 9A and 9B are thus merely illustrative to describe possible relations between the various data structures referred to in the foregoing paragraphs in this section.

Referring to Figures 9A and 9B, a file or database 300, contains the basic structure for the analysis of particular forms (*e.g.*, driver's licenses). File 300 in this embodiment is split into two sections that comprises form and cluster information files 302 (see Figure 9A) and font information files 304 (see Figure 9B). In a preferred embodiment, each state or form

- 45 -

would have entries in both of information files 302 and 304, although only the state of Texas is shown as an example.

Generally speaking, form and cluster information file 302 contains a plurality of form digests 310, such as that shown for Texas. Each form digest 310 contains information used in processing a particular form, such as the form name, the name of the file containing the form header template, and the form header origin. Each form digest 310 also contains information concerning the various clusters of interest in the form, such the cluster name, the names of the file containing the cluster header template, the cluster header origin, the element origin for each element in the cluster, the pattern specification for each element, and the font associated with each element. Optionally, each form digest 310 may also contain information such as the sizes of the form header and the cluster header specified relative to the form header origin and the cluster header origin respectively. For example, if it is known that the form header is 300 pixels in the horizontal direction and 80 pixels in the vertical direction relative to the form header origin, these offset may also be specified in form digest 310, and may be of assistance in further defining the location of the form header in the image under analysis.

Generally speaking, font information file 304 contains all of the necessary font information for a particular form or forms. What fonts are required for a particular form is determined by the pattern specifications specified in the corresponding form and cluster information file 302. Thus, in the simple example shown in Figures 9A and 9B, which contains the file structures necessary for determining the date of birth and expiration date on a Texas driver's license, the font information file 304 contains information concerning the fonts necessary to implement the pattern specification for these elements. In this case, the same pattern specification, N[1:2] "-" N[1:2] "-" N[2:2], is used to decipher both the date of birth and the expiration date because both of these elements on a Texas driver's license have the same format. However, for exemplary purposes, assume the date of birth is written in courier 12 point font, while the expiration date is written in courier italic 12 point font. Both of these font types are specified for each element, as shown in Figure 9A.

As noted, both the form and cluster information file 302 and the font information file 304 specify and reference certain template file names, which are respectively referred to as form and cluster templates files 306 and character templates files 308. Form and cluster template files 306 contain the form header template and the cluster header templates for a

- 46 -

given state. Thus, and for example, the Texas form and cluster template files in Figures 9A-B include the form header template (e.g., "Texas"), which as previously noted is the first template that will be considered when determining the state corresponding to the license under analysis. Also included are the cluster header template files. In this example, "DOB:" is used
5 as the cluster header, although other headers within this cluster could be used as well, such as "CLASS:" or even a graphic such as the picture of the Texas flag (see Figure 8). Of course, and depending on the information desired from the license, other headers may exist for a particular license form.

Font template files 308 include all the font/point size digests 320 from all the forms.
10 Each of the font/point digests 320 includes all of the necessary character digests or templates 330 referenced by the pattern specification during sequence estimation. Thus, for the date of birth pattern specification, which references Font 1, a total of eleven templates 330 are used, each written in Courier 12 point font as specified. Thus, ten of these templates correspond to font name N, which constitutes (preferably learned) templates for the numbers 0, 1, 2, . . . 9 as
15 they appear in the date of birth field on a Texas drivers license. Together these 10 templates constitute a character template family. The eleventh template corresponds to font name dash (-), and is the template for the dash that separates the month, day, and year. Because, as assumed, the expiration date is written in Courier italic 12 point font, referencing Font 2, a different set of eleven templates are referenced, and which correspond to italicized versions of
20 the eleven templates referenced with respect to the analysis of date of birth.

Of course, other font/point size digests 320 and character digests or templates 330 may be required for a given application. Additionally, and as mentioned earlier, letter fonts may be required for word or name analysis, such as capital letters and lower case letters, and which are designated respectively by "A" and "a" in the pattern specification. In this case, the font
25 template file 308 would additionally contain 52 template files, corresponding to the 26 capital and lower case letters, for both the italics and non-italic Courier fonts. Further, each license form will probably require its own unique font templates, as it is unlikely that the fonts used between two different state's licenses will be suitably similar for analysis purposes, although this is possible.

30 Of course, an operative embodiment need not structure the files in the exact manner specified in Figures 9A and 9B. For example, the form header origin, or the size of the form

- 47 -

header template, could be stored in file 304 instead of in 302. Furthermore, the form and cluster information file 302 could be hierarchically split into two separate form header and cluster files. Other variations are possible, such as those disclosed in Section V, as one skilled in the art will immediately recognize.

5 A suitable file structure such as that shown in Figures 9A and 9B must be set up in advance of analyzing a particular license. This preferably requires template training and other manual and computer-assisted analysis of the example licenses. Thus, the form header 50 and cluster headers 61 are preferably trained as discussed above, and their origins 63 and 65 (and, if necessary, sizes) determined. Element origins with a particular cluster must also be
10 determined. Additionally, the font templates for the elements are preferably trained, again as discussed above. Finally, the pattern specification is determined. Such training is preferably formed on each state's license to be analyzed, again using preferably at least a minimum of thirty exemplary licenses. With such information pulled from the exemplary driver's licenses, files may then be structured and linked as shown in Figure 9A-B, (or more preferably, as in
15 Figure 17), and analysis of a license may now begin.

G. Form Processing

Form processing begins by taking a test image of the form under analysis, preferably by scanning with the OSU 6, wherein each pixel of the test image is associated with a black-and-white (grayscale) intensity (i.e., $D(i,j)$). (Color information could also be stored, but is not
20 expected to be necessary for the analysis of driver's licenses. If color information is desired, the lamps 218 in the OSU 6 would preferably be modified to illuminate the license in red, blue, and green light, as one skilled in the art would recognize.) This image is preferably initially stored in the SRAM memory 232 on the OSU 6, and processed locally using the necessary template information stored in Flash 231.

25 The first step in the analysis is to determine the state of the license at issue. In this regard, each state's header template file is compared to the relevant pixels on the test images, using the stored form header origin to determine the suspected location of the header. Therefore, when attempting to match the Texas header template, the form header origin (e.g., 400,20) specified in Texas' form digest 310 in file 302 is located in the test image, and the
30 characters present at that position on the image are template matched. Because the form headers (e.g., "Texas") are typically printed in a large type face on the license, the "fast"

- 48 -

template matching technique disclosed earlier preferably used for identifying the license type. Additionally, if information about the size of the form header has been stored in the form and cluster information file 302 as well as the form header origin, a particular rectangular field of pixels on the test image may be extracted, which may quicken analysis and better define the
5 pixels on the test image to be analyzed.

Once the license type is determined and a template is chosen (e.g., the Texas template), cluster processing begins on each cluster of interest. For example, if it is desired to extract only the date of birth from a Texas driver's license, which would be necessary in an application requiring age verification, then there is only one cluster 60 to process. In this
10 example, the cluster header origin is read from Texas' form digest 310 in file 302, which as noted earlier corresponds to a pixel offset (x*,y*) with respect to the form header origin. However, because the location of the cluster may vary by plus-or-minus 15 pixels, the cluster header template 61 is preferably "slid" horizontally and vertically within this variable range to locate and "set" the cluster origin 65 as a particular pixel on the test image. This sliding and
15 setting process involves assessment of the minimal distance as discussed above.

The analysis would be more complicated, and perhaps more time consuming, for an assessment of clusters that did not contain a cluster header, which would be the case if, for example, it was desirable to determine the name of the consumer from the face of the license. In this case, the cluster template would still have a pre-determined cluster origin, but would
20 lack information about content. In this case, sequence estimation would begin immediately at the location of the cluster origin. Otherwise, a black rectangle the size of one capital letter could be used as a dummy cluster header template to assist in determining the location of the cluster or the elements within it.

Once the cluster header origin (or more generally, the cluster origin) has been
25 determined, sequence estimation is performed for each element in the cluster as described above. The first step is to apply the element origin provided in Texas' form digest 310 of file 302 to determine the location of the elements and the pixels at which sequence estimation analysis should begin. As noted previously, because the locations of the elements are known very precisely relative to the cluster origin (usually plus or minus one pixel), sequence
30 estimation preferably begins immediately at this point without the need for template shifting and distance determinations. However, these extra steps may be useful in a given application

- 49 -

to further ensure the accuracy of the analysis. Thereafter, the pattern specification (e.g., N[1:2] "-" N[1:2] "-" N[2:2]) is retrieved from form digest 310. Each portion of the pattern specification is linked to a font name in file 304, which in turn specifies the requisite character template files in file 308. These character template files in file 308 may then be used during
5 sequence estimation as discussed above to determine the textual content of the element under analysis, in this case, the date of birth. As mentioned earlier, the templates consulted by the sequence estimation algorithm are preferably binary templates, which provides for efficient use of memory in the system and which speeds up the analysis.

H. Form Validation

10 As noted above, the test image of the driver's license is an optical image of the license that has been converted to grayscale. However, it might be easy to tamper with the license, e.g., by changing the date of birth with a marker, to fool the system. Or, a completely false form might be generated, e.g., using a computer and a printer. For this reason, it is preferred that a commercial system employ further analysis measures to verify the validity of the form
15 being analyzed.

Several different methods of validation are possible. For example, most states' driver's licenses use a seal or hologram somewhere on the face of the license that can also be detected and analyzed using character recognition techniques. (The hologram can be detected as it will cast a shadow upon optical illumination within the OSU). This is preferably performed by
20 training a template to represent the seal or hologram. Recognition of the seal or holographic image after recognizing the date of birth provides the needed verification, and helps to ensure that the form under analysis is not wholly false. Further details concerning the detection and analysis of holograms and other optically variable devices are provided below with reference to Figures 10-25. For identification forms having a bar code, templates of the bar codes could
25 also be stored and optically compared with the bar code on the form to further verify form validity using the disclosed techniques, which might be simpler in some applications than actually reading and interpreting the bar code in the standard manners known in the prior art.

Additional validation measures can be accomplished by comparing OCR data with magnetic stripe data. In this case, the OSU would also be fitted with a magnetic head, as in
30 OSU 6, and the system configured to compare the optical data and the magnetic data to compare the retrieved information to ensure that tampering has not occurred. Further security

- 50 -

could be added by encrypting the magnetic data. Of course, such a scheme would not be possible if the license under analysis does not contain a magnetic stripe, which is the situation in some states. Additionally, validation could be compared through a comparison of optical data with the consumer's credit card data to compare, for example, the consumer's name.

5 Other types of verification may be used with licenses that could provide higher levels of security, and which could be easily handled with the disclosed techniques. For example, images could be encoded in the license, which are only visible using an infrared detector or ultraviolet detector. Such a security measure would be difficult to forge. If the OSU were fitted with an infra-red or ultraviolet light source and detector, validation of the license could
10 be performed with great confidence that the license is authentic and has not been tampered with. Further details concerning the detection and analysis of infra-red and ultraviolet encoded images and other optically variable devices are provided below with reference to Figures 10-25.

I. Handling of ID Cards Not Yet Having a Template on the System

15 It would be expected in a commercial system that a consumer might try to enter an ID card for which a template has not yet been created and stored in the system. In this instance, it is presently preferred that the ID card be scanned by the system, saved, e.g., in database 70, and that the following message be displayed to the consumer:

20 "The ID card you have inserted is not currently supported by the Davis system at this time. However, if you return within X hours, our system administrators will try to ensure that your ID card will be useable in the system. Please wait a few seconds while we scan your ID card. Thank you for your patience. We look forwarding to approving your ID card within X hours."

During the X hour timeframe, the system administrator will ideally have time to assess
25 the stored image and create a template for it recognizable by the system. Otherwise, the image itself could be used as a specialized template, with systems assistants during this time working off line to verify the information on the card with appropriate officials, and then storing the contents of the ID card in a specialized file in the system associated with that specialized template. Thereafter, when the consumer returns to the system, his ID card will be recognized,
30 but not necessarily subjected to analysis using a pattern specification. Instead, the ID card would be template matched, and information for that specialized template would be pulled from the specialized file created for that ID card and verified accordingly.

IV. PROCESSING OPTICALLY VARIABLE DEVICES ON A FORM

In the art of document security, optically variable devices or elements can be used to authenticate a document and to prevent tampering, duplication, or counterfeiting of the document. A number of optically variable devices or elements exist in the art of document security. An optically variable device or element can include, but is not limited to, an optical variable device (OVD), a diffractive optical variable image device (DOVID), a hologram, a colorgram, a holomatrix, a stereogram, a transparent or semi-transparent element, a watermark, an element or ink reflecting ultraviolet or infra-red light, an element having bi-dimensional luminescent effects, an element embossed on a surface, an element etched on a substantially smooth surface, or an element having an uneven surface. In addition, an optically variable device can include a holographic silver foil having demetallized portions, which can be produced by techniques similar to those used to create micro-text threads for integration inside banknotes. In addition, an optically variable device can include metallized areas of transparent foils having high-resolution shapes, micro-texts, or "metallic watermarks" formed thereon.

In other words, an optically variable device can include devices or elements having one or more characteristic or response to a particular aspect, orientation, or type of light. For example, such variable responses can include definition or illumination of the device at a particular wavelength or polarization of light, can include changes in definition or illumination when the device is viewed at different angles, and can include changes in definition or illumination due to reflective, refractive, dispersive, or diffractive properties of the device.

As disclosed above, the OSU can employ a single light source and a linear CCD array to produce one scan of the document or card. To detect an optically variable device, such as a hologram or ultraviolet lettering for example, the optical components of the OSU can include one or more CCD arrays and/or one or more light sources to produce multiple scans of the card.

In Figure 10, an embodiment of an optical scanning unit 250 is schematically illustrated in cross-section adjacent a card 4. The optical scanning unit 250 can be used to process optically variable devices on the card 4. The optical scanning unit 250 includes two CCD arrays 216a-b and a single light source 218. The CCD arrays 216a-b and light source 218 are housed within a mounting block 260, which is milled to accommodate the CCD arrays 216a-b and light source 218. Although not shown, it is understood that the optical scanning unit 250

- 52 -

can include a housing, a stepper motor, gears, and drive rollers, among other components described in detail above.

The CCD arrays 216a-b are oriented at different angles ϕ_a and ϕ_b with respect to the card 4. In a preferred embodiment, the first CCD array 216a is used to produce a first scan of the card 4, and the second CCD array 216b is used to produce a second scan of the card 4. The scans can be produced at the same time as the card 4 is passed in one direction. Alternatively, the scans can be produced separately as the card 4 is passed in opposing directions. The two scans produced by the CCD arrays 216a-b at different angles ϕ_a and ϕ_b can be used to analyze and process an optically variable device, such as a hologram, using methods and techniques described below.

In Figure 11, another embodiment of an optical scanning unit 250 is schematically illustrated in cross-section adjacent a card 4. The optical scanning unit 250 can be used to process optically variable devices, such as a hologram or ultraviolet lettering for example. The optical scanning unit 250 includes two light sources 218a-b and a CCD array 216, although more light sources or CCD arrays could be utilized. Each light source 218a-b is used to produce a scan of the card 4. A first scan of the card 4 is produced with the first light source 218a when the card 4 is advanced in a first direction A (*i.e.*, into the OSU), and a second scan of the card 4 is produced with the second light source 218b when the card 4 is advanced in a second direction B (*i.e.*, into the OSU). The two light sources 218a-b emit different wavelengths of light. For example, at least one of the light sources 218a or 218b can preferably emit green light having a wavelength of approximately 575-nm, which is suitable for the linear CCD array 216 preferably employed in the disclosed optical scanning unit 250. The character definition and image contrast of driver's licenses and ID cards are enhanced by the use of green light. Therefore, a main image scan of the card 4 suitable for optical character recognition can preferably be made with the first green light source 218a.

In other documents or cards, elements or lettering composed of a material reflecting in a particular bandwidth or at a wavelength is purposely used on the document or card. An exemplary image in Figure 12 shows a card 4, which has been scanned with an ultraviolet light source. The surface of the card 4 contains ultraviolet lettering of "TEXAS" repeated thereon. The actual printed information and other graphics on the card 4, which would be visible under white or green light, do not reflect ultraviolet light and are therefore not visible on the

- 53 -

ultraviolet scanned image. The ultraviolet letter may be only faintly visible under white light, for example. It is understood that, depending on the wavelengths of light to be detected, the linear CCD array employed is necessarily capable of detecting the wavelengths that the lettering reflects, be it ultraviolet, infra-red, or other portions of the electromagnetic spectrum.

5 Accordingly, the second light source 218b can emit light of a different wavelength, such as ultraviolet or infra-red light, to pick up the optically variable device that, in this example, constitutes the word "TEXAS." The second light source 218b, therefore, can be used for the second scan of the card 4 to exploit the particular reflective properties of the ink or other element, which can be used for further identifying or authenticating the document or
10 card. Beyond the intentional use of ultraviolet or infra-red elements, the ink used for printing information on the card can reflect at a particular bandwidth and can be used for determination and authentication of the card, even though the ink was not intentionally selected for that purpose when the card was designed.

Referring to Figures 13, another embodiment of an optical scanning unit 250 is
15 schematically illustrated in cross-section adjacent a card 4. The optical scanning unit 250 includes a mounting block or body 260, a first light source 270, a second light source 272, a CCD array 216, and a plurality of focusing lenses 280, 282, and 284. It is understood that the optical scanning unit 250 can include additional components that are not shown for mounting the light sources 270, 272 and the CCD array 216 in the block 260, for connecting the light
20 sources 270, 272 and the CCD array 216 to other electronics (not shown), and for securing the mounting block 260 within a host device, such as the OSU device of a vending machine.

The mounting block 260, which is shown alone in the isometric view of Figure 14, is mounted adjacent to where the card or document 4 is passed in the host device. For example, the mounting block 260 can be positioned substantially where the CCD array 216 and lamp
25 218 are indicated in the OSU device of Figure 5. The mounting block 260 has a first side 261 and a second side 262. A plurality of cavities or housings are defined in the first side 261 of the block 260. A first cavity 264 houses the CCD array 216, an offset cavity 266 houses the first light source 270, and another offset cavity 268 houses the second light source 272.

To communicate reflected light from the card 4 to the CCD array 216, a first light
30 channel 265 is defined in the second side 262 of the mounting block 260. To communicate emitted light from the light sources 270 and 272 to the surface of the card 4, offset channels

- 54 -

267 and 269 for light are respectively defined from the light sources 270 and 272 to the side 262. As indicated by line P, the CCD array 216 is substantially perpendicular to the surface of the card 4. As indicated by lines L₁ and L₂, the two light sources 270 and 272 are offset at acute angles θ_1 and θ_2 with respect to the perpendicular line P for the CCD array 216. In one
5 embodiment, the angles θ_1 and θ_2 can preferably be 40 and 48-degrees, respectively.

As best shown in Figure 14, the mounting block 260 is approximately 2.75-inches in width W, 1.60-inches in depth D₁, and 0.60-inch in height H. The cavities 264, 266, and 268 span nearly the entire width of the block 260 and are approximately 2.225-inches in width. As best shown in Figure 13, the cavity 264 for the CCD array 216 is approximately 0.50-inch in
10 depth D₂, and the cavities 266 and 268 are both approximately 0.35-inch in depth D₃. All of the channels 265, 267, and 269 are approximately 0.10-inches in depth D₄. It is understood that these values are only exemplary for scanning driver's licenses and ID cards and can be changed depending on the size of the document to be scanned and the type and number of CCD arrays or light sources used for the disclosed optical scanning unit 250.

15 Selection of suitable sources of light can be a factor requiring consideration. For example, Cold Cathode Fluorescent Lamp (CCFL) tubes of various colors can be used for the light sources in the disclosed optical scanning unit 250. In addition, dual colored and multi-colored CCFL tubes are known in the art and may be advantageous for use as light sources in the disclosed optical scanning unit 250. The time required to bring the CCFL tubes to proper
20 brightness, however, can be longer than desired. Furthermore, CCFL tubes can require high voltages, a problem which is compounded when multiple light sources are to be used. As an alternative to CCFL tubes, Light Emitting Diodes (LED) strips require small voltages, illuminate quickly, come in a variety of wavelengths, and are readily available. The optical scanning unit 250, therefore, preferably uses LED strips for the light sources 270 and 272. The
25 LED strips can include a plurality of standard surface mount LEDs.

Light emitted from LEDs is typically focused. Consequently, lenses 280 and 282 are preferably used to diffuse the light from the LED light sources 270 and 272 to ensure even light distribution across the surface of the card 4. The first lens 280 is disposed in the light channel 267 for the first light source 270, and the second lens 282 is disposed in the light
30 channel 269 for the second light source 272. A third lens 284 is disposed in the light channel 265 of the CCD array 216 and is used to focus the light from the surface of the card 4 to the

- 55 -

CCD array 216. The lenses 280, 282, and 284 are preferably SELFOC lenses by Nippon Sheet Glass Company, which are graded index fiber lenses. As one skilled in the art will recognize, the lenses utilized will preferably match and will be effective to bend radiation of wavelengths provided by the LEDs. In one embodiment, however, it may be advantageous to have focused
5 light from one of the LED strip light sources, in which case one of the light sources 270 or 272 can lack a lens.

When used, the first and second lenses 280 and 282 are accurately positioned in the mounting block 260 to diffuse the light from the LED light sources 270 and 272. Likewise, the focusing lens 284 is accurately positioned in the mounting block 260 at the proper focal
10 distances with respect to the card 4 and the CCD array 216. To achieve the required accuracy and precision, the mounting block 260 is preferably constructed from milled plastic.

For security and authentication purposes, a document or card 4 can have one or more optically variable devices or elements. Therefore, the selection of one of the light sources 270 or 272 can vary depending on a target device and the types of documents or cards 4 to be
15 processed. If the cards 4 to be analyzed include elements having characteristic or variable responses to a particular aspect or type of light, then at least one of the light sources 270 or 272 can be a light source capable of providing that particular aspect or type of light. For example, the reflective properties of white light can be advantageous for documents or cards 4 having holograms. Therefore, one of the light sources 270 or 272 can preferably be a strip of white
20 LEDs. In another example, a common type of security measure used on driver's licenses or ID cards involves the use of ultraviolet elements or ink on the cards. Consequently, one of the light sources 270 or 272 can preferably be a strip of LEDs capable of illuminating ultraviolet elements or ink.

As evidenced herein, the disclosed optical scanning unit 250 can use two or more lights
25 having different characteristics and/or different angular orientations to make multiple scans of an optically variable device or element on a card. The ultimate selection of the characteristics of the light sources, the number of light sources, and angular orientations of the light sources depends on the types of cards to be processed and the types of optically variable devices to be detected. Below, use of the disclosed optical scanning unit 250 of Figure 13 is discussed in
30 connection with five examples of optically variable devices or elements that may be present on

- 56 -

the card 4, including a hologram I, an ultraviolet element or ink U, embossed element E, an etched or recessed element R, and a transparent element T.

In Figure 13, the card 4 can have a hologram I, for example. As is known in the art, a hologram is a three-dimensional image reproduced from a pattern of interference created by a split coherent beam of light from a laser. In standard industry usage, the term hologram is used to describe any optical security image, whether of a diffractive optical variable image device (DOVID) or optical variable device (OVD) type. A unique feature of holograms is a "parallax" effect, which is the ability to see an image and colors from many angles and depths within the hologram. The "parallax" effect is produced by microstructures within the hologram, which cause diffraction of the light hitting them. Another feature of holograms is the complex optical patterns that they can contain. These patterns can encode information about the depth and photographic appearance of the image. Creating a hologram requires precise optical instruments, lasers, and special photosensitive materials, which makes counterfeiting a hologram difficult. Holograms can also be designed to contain "covert" information imbedded within the image, which may only be visible under certain types of light, allowing for an additional level of security and authentication.

To detect the hologram I on the card 4 for later processing, the card 4 is preferably scanned twice. A first scanned image is created with the first light source 270, and a second scanned image is created with the second light source 272. As the card 4 is advanced in direction A, the CCD array 216 receives reflected light from the first light source 270 at angle θ_1 . The card 4 is advanced by a stepper motor as disclosed above. With each successive step of the motor, the linear CCD array 216 receives a column of light from the first light source 270 reflected off the card 4 and hologram I. Thus, the first scanned image of the hologram I produced with the first light source 270 is unique to the angle θ_1 . In a similar fashion, the second scanned image is made using the second light source 272. As the card 4 is advanced in a reverse direction B, the CCD array 216 receives light from the second light source 272 reflected off the card 4 at angle θ_2 . Both the first and second scanned images can be recorded in memory and can undergo analysis and processing as described herein.

As an alternative to producing the two scanned images of the card 4 as it is respectively moved in opposite directions A and B, the first and second scans can be made as the card 4 is passed in one direction A or B. For example, the card 4 can be advanced by the stepper motor

- 57 -

in direction A so that a first line of the card can be exposed to the CCD array 216. The first light source 270 can then be enabled, and the first line of the card 4 can be stored in memory under a first scanned image. Before the card 4 is advanced, the second light source can then be enabled, and the first line of the card 4 can be stored in memory under a second scanned image. Subsequently, the card can be advanced one step by the stepper motor to expose another line of the card 4 to the CCD array 216. Again, the light sources can be alternately enabled, and the alternate scans can be stored under the separate scanned images in memory. One benefit, then, of using LED light sources 270 and 272 is the ability to quickly energize and de-energize them in this alternating fashion. By alternating the energizations of the light sources 270 and 272, the card 4 only has to be moved once in front of the linear CCD array 216 for both images to be captured.

Referring to Figures 15A-B, an exemplary Indiana driver license 4 having a holographic overlay is shown. The Indiana license, which contains fictional information, is a test image released by the 'Bureau of Motor Vehicles' in Indiana and is included in many publications that are available to liquor stores to show how to identify a fake license. The holographic overlay on the license 4 includes holographic emblems repeated over the surface of the license 4. The driver's license 4 has been scanned twice using two light sources at different angles as disclosed above. In Figure 15A, a first scanned image 290 of the license 4 is produced with a first light source angled at 5-degrees from a line normal to the surface of the license 4. In Figure 15B, a second scanned image 292 of the license 4 is produced with a second light source angled at 17-degrees from a line normal to the surface of the license 4. The first scanned image 290 shows a first aspect I_1 of the hologram, and the second scanned image 292 shows a second aspect I_2 of the hologram. Because the angles are different, the light sources exploit the parallax effect of the holographic overlay. Thus, the first and second aspects I_{1-2} are different and unique to the angles of the light sources that produced them. For example, the first aspect I_1 of the emblems in Figure 15A is faint. However, the second aspect I_2 of the emblems in Figure 15B is considerably more pronounced.

In processing the images 290 and 292, the gray scale value of each corresponding pixel of the two scanned images 290 and 292 recorded in memory are subtracted from one another, and the absolute values of each subtraction are stored in memory. For example, the first scanned image 290 of Figure 15A can represent an array or matrix of m by n pixel values (S_1),

- 58 -

and the second scanned image 292 of Figure 15B can represent an array or matrix of m by n pixel values (S_2). For pixel values of the two arrays S_1 and S_2 that are substantially identical, the resulting subtraction will be zero or nearly zero (*i.e.*, a black or dark pixel). For pixel values of the two arrays S_1 and S_2 that are substantially different, the resulting subtraction will be greater than zero (*i.e.*, a white or light pixel).

Referring to Figure 16A, a third image 294 of the license results from the above subtraction technique. The portions of the first and second images 290 and 292 of Figures 15A-B that do not vary optically at the different angles will produce black or near black portions. However, the holograms are apparent as white portions or aspects I_3 in the resulting, subtractive image 294. The subtraction of the two scanned images 290 and 292 can also leave other artifacts in the resulting image 294. For example, the plastic that contains the overlay and some of the inks on the license 4 can reflect differently to the angled light sources so that some optical variance will exist between the two scanned images 290 and 292 even in areas outside of the holograms.

Aspect I_3 results from the unique and different aspects I_{1-2} of the holograms in the scanned images 290 and 292. Although the present example of the holographic overlay is relatively simple, larger or more three-dimensional holograms, such as used on credit cards for example, can produce similar effects. For illustrative purposes, the resulting, subtracted image 294 has been inverted in image 294' of Figure 16B so that the artifacts that also come from the subtraction process can be seen more clearly. Gray scale inversion is not a necessary step in processing the image 294, although inversion could be performed. It should be noted that the aspects I_3 of the holographic images that are obtained can typically be blurred. Using techniques disclosed below, the blur may not effect pattern matching with templates stored in memory. In addition, filtering techniques, such as disclosed below with reference to Figure 25, can be used to enhance the holographic images.

To complete preparation of the resulting image 294, filtering techniques are used to remove background noise and to enhance the holographic portions of the image. Preferably, a median filter is applied to the image 294 or 294', which leaves only areas of the image that have a very significant color shift. Figure 17 shows a filtered image 296 of the inverted image 294'. It is understood that the filtering could be performed on the image 294 of Figure 16A just as well. The filtered image 296 can then be analyzed and processed using the methods and

- 59 -

techniques disclosed herein. Many different types of gray scale filters can be used, depending on the expected or permissible statistical variance of the pixels after subtraction.

In addition to standard holograms of arbitrary colors, colorgrams are holograms where the colors of the image are more realistic or true. Colorgrams can also be characterized by a chromatic scale of color such that a specific color, magenta for example, appears from a certain angle. The angle and appearance of the specific color can be fixed in advance and can be used to further substantiate the authenticity of the hologram. Consequently, if a colorgram is present on a document or card, a light source and/or a CCD array in one embodiment of the disclosed optical scanning unit 250 can be situated so as to source and/or detect a specific color on the colorgram appearing from a certain angle.

In another example, the card 4 in Figure 13 can have an element or ink U reflecting ultraviolet light. As before, the card 4 is scanned twice. A first scanned image is produced with the first light source 270, which can be white or green light. A second scanned image is produced with the second light source 272, which is capable of illuminating the ultraviolet element or ink U. For example, Figure 12 described above illustrates an exemplary scanned image of the card 4 having ultraviolet lettering on its surface. The first and second scanned images can be stored in memory. For documents having watermarks, the ink used may reflect ultraviolet light, making the watermark detectable. The two scanned images are preferably not subtracted from one another, such as done with holograms, and instead are separately analyzed and processed using the methods and techniques disclosed herein.

In another example, the card 4 of Figure 13 can have an embossed and/or relieved element E formed thereon. As before, the card 4 is scanned twice to produce first and second scanned images. Because the light sources 270 and 272 are at different angles and directions with respect to the card 4, raised and/or lowered portions of the element E can produce different shadows. As discussed in detail above, the corresponding pixel values of the first and second scanned images can be subtracted from one another to pronounce the embossed area for further analysis. Because the shadows of the element E are different between these scanned images, the resulting, subtractive image will contain whiter portions representing the presence of the shadows. The resulting, subtracted image can then be processed using the methods and techniques disclosed herein.

- 60 -

In detecting the element E, it may be advantageous to have a focused light source. Thus, in one embodiment of the optical scanning unit 250, an LED strip light source 270 or 272 can lack a dispersive lens. It may also be advantageous to have a light source oriented at a substantially greater angle to the CCD array 216. Thus, in one embodiment of the optical scanning unit 250, a light source can be positioned at an angle greater than 40 to 48-degrees with respect to the CCD array 216. It is understood that differences in shadows may occur only if the embossed/relieved element E includes sufficient raised and/or lowered portions on the surface of the card or document 4. Typically, a driver's license or ID card may not include such well-defined embossed/relieved elements. Other documents, such as passports and credit cards, which fall within the purview of the present invention, can include punched seals embossed on the document. Such documents may be more suitable for use with the above technique.

In another example, the card 4 of Figure 13 can have an unreflective element R on the surface of the card 4. For example, some documents, such as driver's licenses and ID cards, have transparent plastic surfaces that are substantially smooth. One or more unreflective elements R, such as a repeating pattern of state emblems, can be etched or pressed on the smooth surface of the card. When viewed at an angle to a light source, the smooth surface of the card can reflect most of the light, whereas the unreflective element R will not reflect light as efficiently. To achieve this effect, it is understood that the light source and the CCD array may need to be at an obtuse angle with respect to one another. Thus, in one embodiment of the disclosed optical scanning unit 250 of Figure 18, a light source 276 and a CCD array 216c are oriented at a substantially obtuse angle α with respect to one another and at substantially equivalent acute angles with respect to the surface of the card 4. Such an orientation can be used to detect unreflective elements R on the card 4 and without the dual scans that need to be subtracted. By contrast, if the card is largely rough across its surface, elements R can be relatively smooth and reflective and can be processed according to the same technique. This might be the case where card 4 is paper having a plastic seal or label placed thereon.

In yet another example, the card 4 of Figure 13 can have a transparent or semi-transparent element or portion T of the card 4 through which some light can pass. The element T can have a specific, identifiable pattern or shape. A common transparent element T used in document security is a watermark. Although not necessarily common on driver's licenses or

- 61 -

ID cards, other documents or forms falling within the purview of the present invention can have transparent or semi-transparent elements or portions T. A first scanned image of the card 4 can be produced for processing textual and graphical information using the conventional manner of receiving light reflected from the card 4 from a light source 270 or 272 to the CCD array 216. To detect and identify the element T, the optical scanning unit 250 can include a light source 274 on the opposite side of the card 4 from the CCD array 216. The presence of a transparent or semi-transparent element or portion T of the card 4 allows light from the opposing light source 274 to pass through the element T on card 4 to the CCD array 216. Therefore, a scanned image produced with the opposing light source 274 will contain the specific pattern or design of the element T, which may not be evident from the light reflected from the card 4. For example, a watermark can produce a faint, shaded image through the card 4, which is recorded in the scanned image produced with the opposing light source 274. Such a scanned image can then be analyzed and processed using the methods and techniques disclosed herein.

Referring to Figure 19, another embodiment of an optical scanning unit 250 is schematically illustrated in cross-section adjacent a card 4. The optical scanning unit 250 can be used to process optically variable devices or elements and to scan both sides of the card. The optical scanning unit 250 includes first and second opposing mounting blocks 260a-b. The first mounting block 260a has a central CCD array 216a and offset light sources 270a and 272a. The second mounting block 260b has an offset CCD array 216b, a central light source 270b, and an offset light source 272b.

The card 4 is passed between the mounting blocks 260a-b with front and back rollers 212 and 214, such as discussed above with reference to Figures 2-6. The card 4 has a top surface 7a proximate to the first block 260a and has a bottom surface 7b proximate to the second block 260b. A typical card 4 is approximately 3 or 3.2-inches in length. As noted above with reference to Figure 14, a mounting block having two LED light sources and a CCD array for use with the disclosed optical scanning unit 250 can be about 1.6-inches in depth. Therefore, the mounting blocks 260a-b in the present embodiment can easily fit between the front and back rollers 212 and 214 and still allow the card 4 to pass therebetween.

In the position shown in Figure 19, the card 4 is centrally positioned between the blocks 260a-b, and the ends of the card 4 engage the front and back rollers 212 and 214.

- 62 -

Because there is no surface on which the free ends of the card 4 can rest when passing between the blocks 260a-b, a dual set of front rollers 212 and a dual set of back rollers 214 can be used to stabilize the card 4 as it is advanced between the blocks 260a-b, but use of dual sets of rollers 212 and 214 is not strictly necessary. Moreover, additional mechanism (not shown),
5 such as guide rails, can be provided to support the sides of the card 4 as they are passed between the blocks 260a-b.

In the disclosed unit 250 of Figure 19, the central light source 270b on the second block 260b opposes the central CCD array 216a on the first block 260a. As noted above, this can be advantageously used to detect and analyze transparent or semi-transparent portions of the card
10 4, if present. In addition, the second block 260a has the second light source 272b and the CCD array 216b offset from each other by an obtuse angle, which can be advantageously used to detect and analyze etched, reflective, unreflective or embossed elements on the bottom surface of the card 4, if present. Of course, these arrangements may or may not be used, depending on what information and optically variable devices are to be detected and analyzed on the sides
15 7a-b of the card 4.

With the benefit of the present disclosure and the above examples, one of ordinary skill in the art will appreciate that the disclosed optical scanning unit 250 can have one or more CCD arrays and one or more light sources to detect a number of optically variable devices or elements known in the art on various documents and cards. For brevity, not all examples can
20 be elucidated herein. Therefore, it is understood that the disclosed optical scanning unit 250 is not limited to the selection and arrangement of light sources and CCD arrays illustrated in the above-examples. Rather, the selection and arrangement of the light sources and CCD arrays can be altered to suit the optically variable devices that may be present on a given card or document.

The disclosed techniques and systems for producing images of optically variable
25 devices allow other techniques disclosed herein to be used in analyzing the resulting images for determination and authentication purposes. For example, hologram templates for forms or documents of interest can be "trained" and stored in the system. As a result, forms input into the system can be assessed as to what type of form is presented (*e.g.*, is the presented form a
30 Texas license?) or can be assessed as to the authentication of the form once its type has been verified (*e.g.*, the presented form is a Texas license, but is it authentic?).

- 63 -

V. Optimized Optical Character Recognition (OCR) Using Reduction and Comparison Techniques

As noted in section III above, processing of a card (*i.e.*, ID card or driver's license in the present examples) includes scanning the presented card with the Optical Scanning Unit (OSU), then determining the form or type (*i.e.*, State or style) of card presented, subsequently performing cluster processing on each cluster of interest, and finally processing an element of a cluster of interest. For the OSU to perform a timely analysis of the data of the presented card, the processes of determining the form or type of card presented and translating the data of the scanned image into character data preferably uses a number of reduction and comparison techniques disclosed in detail below.

The reduction techniques disclosed below include compressing the entire scanned image or major portions thereof and compressing relevant areas of the scanned image to reduce the size or amount of data required for analysis. Using the reduction techniques, the speed of the analysis can be increased while maintaining a sufficient amount of data and detail. The comparison techniques disclosed below include evaluating the entire compressed image or portions thereof for changes in contrast across the compressed image and creating a data structure of "comparative" bytes for the compressed image. The comparative bytes are then compared to known information stored in memory. The comparison technique can also speed up the analysis, while maintaining a sufficient amount of data and detail. These techniques can be useful in conjunction with techniques disclosed in section III, which are not repeated here for simplicity.

In the sections that follow below, methods and techniques are first disclosed in Figures 20-23 to create a form digest for a new driver's license or ID card. Then, methods and techniques are disclosed in Figures 24A-B to process a scanned image of a presented card using the information created in the form digests. Finally, methods and techniques are disclosed in Figures 24C-31 to optically read a date of birth on a presented card, as is useful for vending of age-restricted products. It is understood that the disclosed methods and techniques can be applied to forms, instruments, documents or cards containing standardized graphics, typography, and characters, such as social security cards, passports, money, signature cards, institutional documents, etc., as well as ID cards, such as drivers licenses. It is also to be

- 64 -

understood that the disclosed methods and techniques can be used for other purposes than validating the age on a driver's license or ID card.

For brevity, the disclosed methods and techniques are represented graphically in terms of arrays or matrices of pixels having certain values. For simplicity and because they are within the purview of those of ordinary skill in the art, mathematical equations or examples of programming code useful in processing the images are not disclosed.

A. Creating a Form Digest for a New Form or Card

Depending on the target market and the number of cards to be accepted by the device, the optical scanning unit (OSU) may need to store an extensive file of forms, types, or templates. All governing bodies design their own style of ID cards and driver's licenses, for example, which results in an endless variety of fonts, sizes, and positions of data. To quickly and efficiently create and update template information for new OSUs and those already in the field, a software program is used to process a scanned image of a new card and to create a new form digest. As one skilled in the art will recognize, the software program can be implemented on a personal computer.

Referring to Figures 20A-B, a process 100 performed by the software program and an operator using the software program is illustrated for developing a form digest 310 for a new card. In Figures 20A-B, the data structures of the form digest 310 created by the process 100 are shown next to the process steps. Although only one card is described as being used to create the new form digest 310, it is understood that a number of cards of the new form or type are preferably used to create the new form digest 310 so that suitable levels of statistical variance can be assessed and compensated for.

The new card is scanned to create a template image (102). The software program can accept the new image from a flat bed scanner, from an image file, or directly from an OSU using communication methods known in the art or disclosed herein. The software program then compresses the template image using bilinear reduction (104). Efficiently determining the form or type of card presented in later analysis depends on how large of a pixel area is used in the bilinear reduction. If too much information is maintained, speed of the analysis can be compromised. For example, scanning the card can render a scanned image having 1200 x 768 pixels or 921 kb of data. During analysis of a presented card, it would be inefficient to

- 65 -

compare every pixel of the scanned image against the 961 kb stored images of over a hundred different card types in memory of an OSU.

For some forms, documents, or cards; only a small portion of each needs to be analyzed to determine the type of form, document, or card presented. However, state licenses and ID cards lack a common physical area that can be used to uniquely identify the licenses or cards. Most states have a banner with a state name or other static information on the licenses or ID cards. Many states, however, have no banner at all. Still other states have pictures or other variable information that may overlap the banner. For state licenses or ID cards, the preferred solution is to compress the entire scanned image or a major portion thereof in the bilinear reduction step (104).

Bilinear reduction is an averaging technique where sections of the scanned image are evaluated to produce a compressed image. Using sections having a size of 16 x 16 pixels represents a good compromise between size and detail when performing bilinear reduction on State licenses or ID cards. For example, Figure 21 illustrates an exemplary, 16 x 16-pixel section 130 of a scanned image. Each cell in the section 130 represents one pixel having an 8-bit number representative of the gray scale value of the pixel. Uncompressed, this section 130 represents 256 bytes of data. A bilinear value of the 16x16 section 130 is calculated by taking the sum of all pixels in the section 130 and dividing that sum by the total number of pixels in the section 130 (*i.e.*, Bilinear Value for Section = (Sum(1,1:16,16))/256). Therefore, the bilinear value of the exemplary section 130 in Figure 21 is an 8-bit number equal to 108.

The above-technique of bilinear reduction described above is applied to each 16 x 16-pixel section of the entire scanned image or major portion thereof for the new card stored in memory. After bilinear reduction is applied to each pixel section, a compressed image is produced and stored in memory. For example, the card in Figure 8 can represent an uncompressed image of a Texas driver's license stored in memory. The scanned card in Figure 8 would have an initial scanning resolution of 1200 x 768 pixels. During bilinear reduction, the 3,600 sections of 16 x 16 pixels in the uncompressed image are evaluated using the reduction technique described above. After bilinear reduction, the same Texas driver's license will resemble a compressed image 132 shown in Figure 22. The compressed image 132 has a resolution of 75 x 48 pixels. To the human eye, the compressed image 132 may appear illegible. However, the compressed image 132 retains aspects of the graphics, lettering, and

- 66 -

other unique qualities of the original scanned image, making it suitable for analysis and processing as described below.

Once the scanned image has been compressed with 16x16 bilinear reduction in step (104) of Figure 20A, comparison steps (106) are then performed on the compressed image.

The comparison steps (106) evaluate differences in contrast or color between individual bilinear reduction values for the pixel sections of the compressed image with other bilinear reduction values of the compressed image to determine whether the individual bilinear reduction values for the pixel sections are darker or lighter than the other bilinear reduction values for the pixel sections. Preferably, each individual bilinear reduction value for the pixel sections of the compressed image is compared to the bilinear reduction values of the eight pixel sections surrounding it.

Using Tables 4 and 5 below, an example of the comparison steps will be discussed. Table 4 represents a 3 X 3-pixel section of a compressed image. Each cell in Table 4 contains a gray scale value representative of the bilinear reduction value of an original 16 X 16-pixel section of the uncompressed image. During the comparison steps, the bilinear reduction value “108” in cell B2 is subtracted from the values in each neighboring cell. If a subtraction results in a value less than or equal to zero, then “0” is recorded in the comparison steps. If the subtraction results in a value greater than zero, then “1” is recorded in the comparison steps.

<i>Table 4: Example of Bilinear Reduction Values in a Compressed Image</i>			
	<i>A</i>	<i>B</i>	<i>C</i>
<i>1</i>	102	106	104
<i>2</i>	108	108	112
<i>3</i>	107	111	115

Table 5 illustrates results of the subtraction process performed on the bilinear reduction values in Table 4. In the example, the subtractions begin with cell A1 and continue clockwise around cell B2. A resulting byte “00011100” created by this process represents a “comparative” byte for the pixel B2.

- 67 -

Table 5: Comparative Byte Calculation for Example in Table 4

$A1 - B2 = 102 - 108 = 0$
$B1 - B2 = 106 - 108 = 0$
$C1 - B2 = 104 - 108 = 0$
$C2 - B2 = 112 - 108 = 1$
$C3 - B2 = 115 - 108 = 1$
$B3 - B2 = 111 - 108 = 1$
$A3 - B2 = 108 - 108 = 0$
$A2 - B2 = 107 - 108 = 0$

The comparative byte “00011100” in the above example shows that B2 is lighter (0) than A1, B1, C1, A3, and A2 and is darker (1) than C2, C3, and B3, for example. The actual differences in gray scale values are not used, which simplifies later analysis and processing. The comparison steps are performed on each individual pixel (A1-3, B1-3, and C1-3) of Table 4. For corner values, such as A1, that lack eight neighbors, values of “255” (i.e., white) can be used where a neighboring value is missing from the compressed image. After the comparison steps are performed on each individual value of Table 4, a comparative byte data structure is created for the compressed image. In the present example, the comparative byte data structure has the same size as the compressed image.

It is understood that each individual value can be compared with fewer than eight neighbors, for example with five neighboring value, or with more those that are immediately adjacent the individual value. In addition, it is understood that each value in the compressed image need not be compared. Selection of the appropriate amount for comparison to be performed in the comparison steps (106) depends on a number of variables, including the detail of the forms being analyzed, the contrast present in the images, the desired processing speed, or the amount of available storage space, among other variables. To determine the form of driver’s license or ID card in the present example, it is preferred in the comparison step (106) that each individual value of the compressed image or portion thereof be compared to the eight values surrounding it, as illustrated.

When the comparison steps (106) are completed in Figure 20A, a comparative byte data structure (CBDS) 312 is created and stored in the form digest 310 for the new card. The

- 68 -

CBDS 312 represents the differences in contrast of the pixels in the compressed image. For example, if each pixel in the entire compressed image 132 of Figure 22 has undergone comparison steps, the CBDS 312 will contain 3,600 comparative bytes of eight bytes each. In general, however, the file for CBDS 312 can be larger or smaller than the file for the entire
5 compressed image, as noted above.

It may be beneficial that the reduction and comparison steps (104) and (106) described above are performed on only a portion of the card. For example, driver's licenses and ID cards typically have a photograph, which can vary significantly in color and contrast from one card to another and may not contain relevant information for analysis. Therefore, the reduction and
10 comparison steps (104) and (106) may be selectively performed on the card such that the portion of the card having the photograph is not processed. Alternatively, the portion of the card with photograph may be reduced and compressed in the steps (104) and (106) but not used in later analysis.

In another alternative, the photograph may be used in later analysis if it contains
15 relevant details in verifying the authenticity of a presented card. For example, if the scanned image is to be subtracted from another scanned image to detect an optically variable device or element on the card as disclosed above, then the photograph may be used in later analysis. In another example, techniques and software programs are known in the art that are capable of mapping features of the human face and matching the identity of an analyzed image to one
20 stored in memory. Such techniques and software programs may be used to analyze the photograph in the scanned image.

Using the software program in Figure 20A, an operator then selects a field on the scanned, uncompressed image (108). With the assistance of the software program, the operator identifies pertinent data fields, field designators, optically variable devices or elements,
25 emblems, graphics, designs, insignia, or headers on the uncompressed image obtained in step (102). The software program then determines the size and coordinates for the selected field and stores an entry 314 in the form digest 310 containing the coordinate and size information for the selected field. Alternatively, the operator by viewing the image can map out the field of interest and input its size and coordinate into the software program.

30 In the uncompressed image of Figure 8, for example, the operator can use the software program to set up a field designator 61 containing the title of a desired field on the new card.

- 69 -

Typically, the size, lettering, and position of the field designator 61 will be substantially consistent between individual cards of a particular type or form. For example, on newer Texas driver's license, the field designator 61 for the date of birth on the card is "DOB:" (see Figure 8) and can have a starting horizontal position of 690 pixels and a starting vertical position of 460 pixels. The software program can store a size or dimension in pixels and a reference coordinate, such as the corner pixel 65, for the selected field.

It should be noted that if a field of the card has micro-printing or other fine detail to be used in determining or validating the presented form, a high resolution image may be required on at least a portion of the form at this stage. In the present example, the preferred resolution used on driver's licenses or ID cards is approximately 400 dpi generating an image having 1200 x 768 pixels, which is suitable for resolving even the smallest text or graphics on the card.

When the field selected in step (108) of Figure 20A is a field that is consistently used on all of the cards of the new form or type, the software program performs a 2X2 bilinear reduction of the selected field of the original, uncompressed image (110). The bilinear reduction in step (110) is similar to that discussed above in step (104) with the exception that smaller 2 X 2 sections of the selected field are used. In this way, the selected field (e.g., "DOB:") is compressed to reduce the amount of data that the system must store and analyze. The software program then performs comparison steps (112) on the selected field and stores a comparative byte data structure 318 for the selected field in the form digest 310. The comparison steps (112) are similar to those discussed above in step (106), albeit with a smaller portion of the scanned image. The operator can then set up/select additional designator fields, graphics, designs, emblems, insignia, form headers, and cluster headers by returning to step (108).

In embodiments where the licenses, cards, or forms have holograms or other optically variable devices or elements to be analyzed, more than one scanned image of the card can be made, as discussed in detail above. After the multiple scanned images have been recorded in memory, the operator can use the software program to process the images and store information in the form digest 310. For example, in the case of a card having a hologram, the software program can subtract one scanned image of the card from another scanned image of the card and can record the absolute values of the differences as a resulting, subtractive image.

- 70 -

The presence of the hologram on the card will be apparent as white portion on the resulting, subtractive image. The operator can then select all or part of the resulting, subtractive image for processing in step (108). The operator can use the software program to perform a 2X2 bilinear reduction step (110) and comparative steps (112) on the subtractive image or portion thereof resulting from the hologram and can store a coordinate/size entry 314 and CBDS 318.

Once fields for determining the form or type of card are stored in memory, the operator also uses the software program to set up particular fields for optical character recognition in step (114) of Figure 20B. For example, the operator can select a field, such as field 58 in Figure 8 containing the date of birth on the card. The size of the field 58 is assessed and given a size and a coordinate in pixels, such as the corner pixel 69. In Figure 20B, the software program then creates a field coordinate/size entry 314 in the form digest 310 for the selected OCR field. In addition, with the assistance of the software program, the operator inputs the font type and point size used in the OCR field of the new card (116). For example, the selected OCR field may use an 18-point, Arial font. The software program creates an entry 316 in the form digest 310 indicating the font/point size of the selected OCR field.

The font/point size entry 316 merely indicates the font and point size of the selected OCR field. The actual data of the font and point size are stored in font-point size digests 320, which hold specific data for the types and sizes of fonts known to the software program. If the font/point size digests 320 contain the font and point size combination for the selected OCR field in step (116), the font/point size entry 316 is merely recorded in the form digest 310 and the operator can return to step (114) to select another field for optical character recognition.

If a new font-point size is discovered in step (116) that does not have a font/point size digest 320 associated with it, the operator uses the software program to create a new font point/size digest 320 for the particular font and point size in step (118). For example, the selected OCR field may contain characters having an Arial font and a point size of 14. A font/point size digest 320 corresponding to the 14-point Arial characters may not be stored in memory, in which case the operator and software program create a new font/point size digest 320 for the 14-point Arial characters to be used in the analysis of this particular form and possibly others that contain 14-point Arial text.

To create a new font/point size digest 320, the software program first filters and enhances the scanned OCR field in step (120), using techniques described in more detail

- 71 -

below. The operator then selects a new character from the filtered OCR field under selection (122). The software program then detects edges of the character to form fitted boundaries by a process disclosed in more detail below (124). Knowing the edges of the character, the software program stores dimensional data 332 of the character in a character digest 330. For example, dimensional data 332 can include the width and height in pixels between the fitted boundaries of the character. The software program then performs a 2 x 2 bilinear reduction of the fitted character (126) and performs comparison steps on the compressed character (128), using methods and techniques disclosed in detail above. A comparative byte data structure (CBDS) 314 is created for the new character and is stored in the character digest 330. The operator returns to step (122) and repeats the steps (122-128) to create character digests 330 representative of all of the new characters found on the new card. Typically, several of the new card types are used to create character digests 330 of all of the representative characters of the new font and point size used in the new card type, because a single card will probably not contain all potential characters of interest. The operator can also return to step (114) to select another field of the new card requiring optical character recognition.

Once a new form of driver's license or ID card has been analyzed, the new form digests 310, font/point size digests 320, and character digests 330 created for the new card are then loaded onto new or existing OSUs using communication techniques disclosed herein. Consequently, each OSU can maintain a database of pertinent form digests 310, font/point size digests 320, and character digests 330 for the types of cards that the OSU needs to validate and analyze.

Referring to Figure 23A, a database 300 for the OSU is schematically depicted. The database 300 includes the form information file 302 and the font information file 308 that have been described above with reference to Figures 9A-B. The form information file 302 includes all of the form digests 310-1 to 310-N for each type or form of card (e.g., state or style) to be analyzed by the OSU. Because font types and point sizes can be common among several different forms or types of cards, the font information file 308 can be maintained separate from the form information file 302 and can be repeatedly referenced by the form digests 310-1 to 310-N where appropriate.

The font information file 308 includes the fonts/point size digests 320 (1-k) used in all the fields for the plurality of cards. A particular font/point size digests 320 for a given card is

- 72 -

referenced in the card's form digest 310. Each font/point size digest 320 includes a plurality of character digests 330 (1-g). Each character digest 330 includes dimension data 332 and a comparative byte data structure 334 for a particular character of the font/point size.

Referring to Figure 23B, a form digest 310(N) is schematically illustrated. Each form
5 digest 310(N) contains the following data:

312 – a comparative byte data structure for the entire image or a major portion thereof;

314 - coordinates and sizes identifying the location and dimensions of each data field (1
through i) on the form, such as field designators, pictures, designs, graphics,
emblems, insignia, headers, banners, and optically variable elements or devices;

10 316 - entries of the font/point size type for each of the data fields for OCR (1-i) on the
form, if necessary. For example, pictures and emblems would not require this
type of entry; and

318 - comparative byte data structures for each data field (1-i) on the form.

The entries and data structures of the form digests 310, font/point size digests 320, and
15 character digests 330 described above can be arranged in the architecture of the database as
described herein with reference to Figures 9A-B and Figure 27. For simplicity, the
correspondence between entries, files, data structures, tables, relationships, etc. of Figures
23A-B with Figures 9A-B and 27 is not enumerated herein.

One of ordinary skill in the art will recognize that the database 300, files 302 and 308,
20 and digests 310, 320 and 330 are only schematically depicted in Figures 23A-B. It would be a
routine undertaking by one of ordinary skill in the arts of data processing and programming to
arrange the information of the database and digests to improve the processing, searching, and
storing of the information. For example, it may be beneficial to store the information in data
tables of a relational database management system. Alternatively, it may be beneficial to store
25 the information in sequentially accessible files stored in ROM, which facilitates searching at
the expense of storage space.

B. Form Determination of Presented Card

Once the OSU contains the files and digests described above, the OSU is capable of
processing a presented document or card of the particular form or type. Referring to Figure
30 24A-C, steps performed by the OSU in processing a presented card are discussed. In Figures
24A-B, the OSU first determines the form of card presented. In Figure 24C, the OSU

- 73 -

enhances a specific field of the card and performs optical character recognition using the disclosed reduction and comparison techniques.

Referring to Figure 24A, the OSU scans the unknown card presented by the consumer during a transaction, for example, and stores its entire image to RAM (140). In one embodiment, a user interface on the device may prompt the consumer to indicate the state or other origin of the presented card, if applicable. For example, if a driver's license or state ID is to be used in the transaction, the consumer may be prompted to enter a state code. In this way, the OSU can use the state code to determine the proper form digest to access in the database. In a further aspect, the user interface may prompt the consumer to enter the year of expiration of the card for determining the proper style of the card being presented if a state has altered their cards in a given time period. Although these steps may be used to initiate a search for the type or form of card presented, it is preferred that the form of card is determined independently by the OSU for validation and security and without intervention by the consumer.

To abbreviate processing and searching, it may be beneficial to determine where a photo on a driver's license or ID card is on the presented card. For example, a photo may typically be darker than other portions of the card. By averaging pixel values in the rows and columns of the scanned image, the OSU can determine where the darkest portion of the image is present. The darkest portion can then be ignored during later steps, if the darkest portion is assumed to be a photo that offers no relevant detail for processing.

Alternatively, the OSU can determine on which side of a central column that the darkest portion lies on the scanned image. Because the darkest portion of the image is typically the photo and because most driver's licenses or ID cards have a photo either to the left or the right of a central column, the OSU can summarily eliminate driver's licenses or ID cards stored in memory having photos on an opposite side than that found for the presented card. For example, Texas licenses can have photos on the right-hand side, while Rhode Island driver's licenses can have photos on the left-hand side. If the OSU determines from the scanned image that the photo lies on the left-hand side of the presented card, Texas can be easily eliminated. Although the abbreviated steps may improve later processing, the capability or viability of performing such steps depends on a number of factors, including the typical contrast of photos against the surrounding background on the licenses, the sizes of the photos, and the existence of relevant details on or adjacent the photos, among other factors. Thus, it

- 74 -

may be found that the differences between driver's licenses and ID cards are too great to attempt such abbreviated steps.

To narrow the search for the possible form of card presented, the entire image or a major portion thereof is first compressed by the OSU in step (142), for example, by a 16x16
5 bilinear reduction, such as described above. Then, a comparative byte data structure (CBDS) of the compressed image for use in later comparison is produced in step (144).

In step (144), each pixel or a predetermined set of pixels of the compressed image, is evaluated in terms of its relationship to surrounding pixels. Consequently, any differences in color or contrast between the presented card and a corresponding comparative byte data
10 structure of a form digest stored in memory will be diminished. For example, the presented card may be faded due to age or wear. Comparing pixels of the scanned image directly with pixels of a stored image would result in large false errors, because the stored image would probably be much darker than the faded image. However, if a presented card is severely faded, each pixel can be generally faded a substantially equivalent degree. Thus, the comparative
15 relationship of each pixel to its neighbors on the scanned, faded image should remain constant or nearly constant so that the CBDS of the faded image will not be substantially different from a stored CBDS of an unfaded image stored in memory. Furthermore, any differences between two different forms or types of cards (*e.g.*, where one card has a state emblem, while the other does not) are accentuated with this technique of comparison.

20 The OSU then performs first evaluation steps (150), where the CBDS for the presented card obtained in step (144) is compared to each of the CBDS stored in the form digests 310-1 to 310-N, if not already eliminated. For example, the CBDS for the presented card in step (144) is first compared to the CBDS for the first form digest 310-1. A bitwise XOR operation is preferably performed on each comparative byte of the CBDS in step (144) against a
25 corresponding comparative byte in the form digest's CBDS 310-1. One of ordinary skill in the art will recognize that a number of known methods and techniques can be used to compare the CBDS for the presented card to the CBDS for the first form digest 310-1. The bitwise XOR operation is preferred because it requires reduced processing time.

The two CBDSs need not be the same size, but they at least contain a majority of
30 corresponding comparative bytes. Furthermore, if the CBDS for the form digest 310-1 contains comparative bytes based on comparisons of each pixel to eight neighboring pixels,

- 75 -

then the CBDS in step (144) must be structured in an equivalent fashion. For simplicity, all of the form digests 310 are preferably structured consistently with one another, and the information of the presented card is similarly structured.

Table 6 below illustrates an example of such a bitwise XOR operation used in the evaluation steps (150). In the bitwise XOR operation, a first comparative byte, 00011100, is evaluated against a second comparative byte, 01001100, in the example of Table 6.

<i>Table 6: Example Match Error Calculation</i>								
First Comparative Byte	0	0	0	1	1	1	0	0
Second Comparative Byte	0	1	0	0	1	1	0	0
Match Error Byte	0	1	0	1	0	0	0	0

The result of the bitwise XOR operation is a match error byte, 01010000. In the example of Table 6, the comparison of the two elements resulted in a match error of two, where a maximum possible error between compared bytes is eight.

In the first evaluation steps (150), such a bitwise XOR operation represented in Table 6, is repeated for each comparative byte of the CBDS of step (144) against a corresponding comparative byte in the CBDS of the digest 310-1. A running summation of the match error is tracked as the corresponding bytes are compared. After evaluating the entire CBDS of the first form digest 310-1 and the CBDS of step (144), the total calculated match error is stored (152). In general, the total match error represents the presented cards similarity with the first form digest 310-1.

Once the match error is calculated for the first form digest 310-1, the OSU then selects the next form digest for evaluation (154). Performing the first evaluation (150), storing of the match error (152), and selecting the next form (154) are repeated until all of the forms have been evaluated. Ultimately, the OSU analyzes the match errors calculated for the form digests (156), and the software program identifies the form digest(s) producing the smallest match error (158).

Analyzing the match errors in step (156) and identifying the form digest(s) with the least error in step (158) involves first determining which form digest has the smallest error and then establishing the match error of that form digest as a baseline error level. Using the baseline error, the software program determines which match errors of the other form digests

- 76 -

lie within a predetermined threshold. For example, only form digests that have a match error within a 20% threshold of the baseline match error may require additional evaluation.

In Table 7, for example, a comparative byte data structure of a presented driver's license has been compared with comparative byte data structures in form digests for Texas, Rhode Island, Arizona, and Indiana driver's licenses. The OSU determines that the CBDS for Texas has the lowest match error of 4,235 and makes 4,235 the baseline match error. Using Texas' match error as the baseline, the OSU calculates the percentage above the baseline match error to determine which licenses other than Texas are logically suitable for further analysis.

<i>Table 7: Example Comparative Error Analysis</i>		
Form Digest	Match Error	% above Baseline
Texas	4,235	0%
Indiana	4,965	15%
Arizona	5,043	19%
Rhode Island	7,325	73%

In this example, only three form digests have match errors that are less than a predetermined threshold of 20% from the baseline match error and possibly need further consideration. In the example of Table 7, the presented card will thus be further analyzed to determine if it constitutes a Texas, Indiana, or Arizona license; consideration of the presented card as a Rhode Island license ceases, saving subsequent processing and analysis time. The threshold above the baseline match error is predetermined to account for differences that may exist between the presented card and the stored information in its corresponding form digest. In addition, the threshold above the baseline match error is predetermined to account for the possibility that two or more forms or types of cards may be substantially similar to one another and may require detailed comparison to distinguish them from one another. Sampling and statistical analysis can be used to arrive at a suitable threshold level, which can be different than 20%.

When identifying the form digests with the least error in step (158), the match error for one form digest may be more than 20% below the next closest match error. In that case, the form digest that produced the exceptionally low match error is determined as the presented card's form digest. This is not the case in Table 7, as both Indiana and Arizona are reasonably

- 77 -

close (<20%) to the match error for Texas. Thus, once the form digests with the least errors are identified (*e.g.*, Texas, Indiana, and Arizona), the final step in determining the form of the presented card involves repeating the techniques of reduction, comparison, and evaluation using more detailed information.

5 Referring to Figure 24B, detailed steps for form determination are discussed when two or more form digests have match errors falling within the predetermined threshold described above. Each form digest 310 contains pertinent information unique to that form or type of card. The form digest 310 includes an entry 314 having the coordinate/size of a field designator, graphic, picture, hologram, banner, header, etc. expected to be present in all cards
10 of that form. Furthermore, the form digest 310 includes an entry 318 having the comparative byte data structure of the designator field as reduced by the 2x2 bilinear reduction. These entries 314 and 318 are used to make a more detailed comparison to determine with improved certainty the form of the presented card.

Starting with a first form digest 310-N for the preliminarily selected forms (*e.g.*, Texas,
15 Indiana, and Arizona), the OSU obtains the coordinate/size information of the designator field 314 from the first form digest (160). The OSU then performs a bilinear reduction of an area of the scanned image at a location having the same coordinates and size. A finer, 2x2 pixel bilinear reduction is preferably performed on this area (162). The OSU then creates a comparative byte data structure of the compressed area of the presented card, using the
20 comparison techniques described above (164).

In step (166), the OSU calculates a match error between the CBDS of the field designator 318 in memory and the CBDS obtained in step (164) using an XOR operation as described in detail above. The OSU then stores the match error (168). For example, the first form digest 310 with a low match error can be for a Texas driver's license and can include the
25 coordinate/size information of a field designator used to denote the date of birth on the card. For example, the Texas driver's license in Figure 8 uses "DOB:" to designate the date of birth information on the license. If the card presented for analysis is an Indiana driver's license, it will not contain the same information at the designated location on the Texas driver's license, resulting in a large match error. If the presented card is a Texas driver's license, then it will
30 have "DOB:" at substantially the same coordinates and having substantially the same size as

- 78 -

indicated in the form digest 310 for the Texas driver's license, resulting in a smaller match error with the form digest for Texas.

After calculating the match error of the designated field, the OSU selects the next form digest (e.g., the digest for Indiana) for comparison (170). The OSU repeats the steps (160-168) for the other form digests that had low match errors in Figure 24A. Once match errors are calculated for all the preliminarily-selected form digests requiring detailed analysis, the OSU compares the match errors of the areas on the presented card (172). From the comparison of the match errors, the OSU determines the form of card presented (174) as the form digest 310 producing the lowest match error. For validation, a predetermined value may be set for indicating a match between the form and the presented card. The predetermined value may be calculated, for example, based on sampling and statistical analysis of the various cards analyzed to create the form digest and may be calculated with recognition to the fact that other forms might be somewhat similar, which again may be determined by sampling and statistical analysis.

As noted above with reference to Figures 13-19, the OSU can make multiple scans of the presented card using one or more CCD arrays and/or one or more light sources. Among the number of uses disclosed above, the multiple scans of the presented card can be used to process an optically variable element or device and determine the form of card presented in steps (160-174). In addition, after the form of the presented card is determined, the multiple scans of the presented card can be processed to assess the presence of an optically variable device or element on the presented card and to authenticate the card for security and validation reasons.

For example, to determine if a hologram is authentic on the card, the disclosed reduction and comparison techniques can be used to detect whether the actual pattern of the image corresponds to a template image stored in memory for the determined card type. When authenticating the hologram, the OSU already knows what state has issued the license or card. Consequently, the OSU will contain the holographic pattern(s) for the state stored in memory. For holograms that are substantially blurred when detected with the process described above in Figures 15A-17, the disclosed reduction and comparison techniques preferably use a larger pixel section to ensure that the blurring of the holograms does not effect pattern matching with template images stored in memory. For less blurred holograms, for example, edge detection

- 79 -

and more refined reduction and comparison techniques can be used, such as those techniques disclosed below for optical character recognition.

Alternatively, using the techniques just described, the alternative scans designed to pick-up optically variable devices can themselves be used to determine the form at issue instead of merely authenticating an already determined form.

C. Optical Character Recognition (OCR) of Presented Card

Once the correct form digest of the presented card is determined and authenticated if necessary, the OSU can then perform optical character recognition (OCR) on a field of the presented card designated for OCR. The OSU employs the reduction and comparison techniques discussed above to obtain and process character information from the presented card, such as the date of birth.

Referring to Figure 24C, steps (180-189) for determining characters in an OCR field of the presented card are discussed. The correct form digest 310 of the presented card is known, and the OSU obtains the coordinate and size of the OCR field 314 from the form digest 310 (180). From the form digest for Texas, for example, the OSU can obtain the location of the date of birth information on a Texas driver's license, which can have a starting horizontal position of 690 pixels and a starting vertical position of 460 pixels and can have an average width of 90 pixels and an average height of 30 pixels, for example.

1. Image Filtering and Enhancement

Using the above coordinate and size information for the OCR field, the OSU filters and enhances that area of the scanned image (181). If multiple areas of the image are to be investigated, the filtering and enhancement can be performed on the entire image or major portions thereof as well. The OSU initiates a number of image filters and enhancements, exemplified in Figure 25.

Scanned field 400 in Figure 25 shows an example of the unaltered OCR field having the scanned date of birth from a presented card. The scanned field 400 shown in Figure 25 comes from a typical license laminated in plastic. The plastic of this type of license is prone to wear, damage, and contamination. Consequently, scanned images of such cards can contain substantial noise or artifacts and can lack distinct contrasts. These forms of licenses differ from "newer" types of licenses being used that lack plastic lamination. The "newer" licenses, however, can also be prone to many of the same problems. Therefore, it is preferable that at

- 80 -

least the portions of scanned images intended for OCR undergo the following enhancement and filtering steps to remove noise and to accentuate the text at issue. These filtering techniques are also useful to enhance images of graphics or optically variable devices as well, and use of these techniques to “clean up” textual information is only exemplary.

5 Enhancement begins by conditioning the unaltered field 400 through the sequential application of software filters that improve the image quality with each process. It is to be understood that other forms may not require all of the filters and enhancements or may not require filtering and enhancing in the sequence disclosed below. Moreover, one of ordinary skill in the art will appreciate that a number of filtering techniques are known and can be used
10 to achieve similar results. The filter sequence preferred for driver’s licenses and ID cards is as follows:

1. Histogram Level Adjustment
2. SUSAN Algorithm
3. High-Pass Filter
- 15 4. Convolution Filter
5. SUSAN Algorithm and Color Reduction

The first image conditioning applied to the unaltered field 400 is a histogram level adjustment. Sometimes, an image is over- or under-exposed during the acquisition stage. In such a case, its quality may be improved by adjusting the gray scale levels. The histogram level
20 adjustment determines the darkest and lightest pixels in the image and scales all the pixels proportionately to gain a maximum image contrast with pixel values between 0 and 255. The histogram level adjustment creates a better differentiation between the background and the graphical element or characters as shown in field 401.

The next step is to apply the Smallest Univalued Segment Assimilating Nucleus
25 (SUSAN) algorithm to field 401. The SUSAN algorithm is a general noise and outline filter known to those of ordinary skill in the art, which performs edge and corner detection and structure preserving noise reduction. The SUSAN algorithm traces the primary points present in the image and then removes the remaining, gaussian noise. One source for more information on the SUSAN algorithm can be obtained from “SUSAN- A new approach to low
30 level image processing,” *International Journal of Computer Vision*, Stephen M. Smith and J. Michael Brady, Vol. 23(1), 1997. Field 402 illustrates the results of the SUSAN algorithm.

- 81 -

The field 402 is next conditioned with a high-pass filter, which looks for large non-linear variations in the image. The high-pass filter removes gaussian noise remnants around each character improving character edge definition. Field 403 illustrates the results of the high pass filter.

5 A convolution filter is then used, which samples a block of 8 by 8 pixels and evaluates each pixel relative to its surrounding pixels. The filter then adjusts pixel values to a common, weighted value within the evaluation block. Computer graphics textbooks having a more complete explanation of convolution filters include *High Performance Computer Imaging*, by Ihtisham Kabir, Manning Publications, 1996, ISBN 1-884777-26-0 and include *Computer*
10 *Graphics: Principles and Practice*, Foley J.D., van Dam A., Feiner S.K., and Hughes J.F., Addison-Wesley, 2nd. Edition, 1990, pp. 629-636, ISBN 0-201-12110-7. The convolution filter accentuates the edges of the characters, as shown by field 404.

The field 404 is then passed through another SUSAN algorithm to further highlight character edges and is reduced to two colors (black and white) to produce the final field ready
15 for character recognition, as shown by field 405.

2. Character Identification

Returning to Figure 24C, once filtering and enhancement of the OCR field is complete, the OSU continues processing the OCR field by detecting the edges of the characters in the OCR field (182). Figure 26 illustrates the technique used by the OSU to detect the edges of the
20 first character, "0" in the example, of the OCR field. In Figure 26, a portion of the OCR field 410 is shown after the filtering and enhancement described above. The size of the OCR field 410, which is known to the OSU and obtained from the form digest 310, is represented as dotted boundary 412. In addition, a starting coordinate of the OCR field 410, which is also known to the OSU and obtained from the form digest 310, is represented as corner pixel 414.
25 The OCR field 410 is made of a plurality of columns 416 and rows 418 of pixels contained within the boundary 412.

To perform optical character recognition of the first character "0" in the field 410, a fitted box or area 440 is established around the first character. To accomplish this, the OSU averages the pixel values of all the pixels in the first column 416 of the field 410. Because the
30 field 410 has been reduced to two colors (i.e., black and white), any column-averaged values less than 255 indicate that black is present and represents a left edge of the first character. If

- 82 -

the first column 416 produces a value of 255, then the first column 416 contains only white pixels, and the OSU moves to the second column 416. The OSU then averages the values of the pixels in the next column 416. The process is continued until a leading edge of the first character is detected, and the OSU determines a first boundary line 420 of the first character.

5 After the first boundary line 420 has been determined, the OSU continues averaging the following columns 416 to determine a right edge of the first character. The right edge is detected when the average pixel value for a column 416 switches back to white or 255. The OSU determines a second boundary line 422 of the first character at this column 416.

 Some error detection may be employed to improve the accuracy of the edge detection.
10 If the second column 416 produces an average value of 200 but the third and fourth columns produce averages of 255, for example, then the second column 416 may appear to be both a leading and trailing edge of the first character. Since the column width of one pixel is clearly not wide enough to represent a character, the value of the second column 416 can be ignored as noise.

15 Once the first and second boundaries 420 and 422 have been positively identified for the first character, the OSU uses the same averaging procedures described above on the rows 418 between these boundaries 420 and 422. The OSU thereby determines third and fourth boundaries 430 and 432 of the first character.

 Now that the exact position and size of the first character is determined, the next step in
20 the OCR process begins in Figure 24C. In step (183), the OSU obtains the font and point size information of the OCR field 316, which includes the types of characters to be encountered within the field of interest. The OSU can abbreviate the recognition of the first character based on the font-point size information. Because the field may contain only numeric characters, the first character may only be between "0" and "9," for example, and other characters (e.g.,
25 letters, special symbols, etc.) can be ignored. Furthermore, because a fitted box having boundaries has been established around the first character of the OCR field and the pixel width is known, character templates within the font/point size digest 320 that are larger or smaller than that size by a predetermined value, say 20%, can be summarily eliminated from further analysis.

30 For example, it may be determined that the first character is not a "1," because the fitted boundaries indicate the first character of the OCR field has a larger pixel width. Of

- 83 -

course, it is understood that abbreviating the character recognition in the field depends on how the characters are particularly arranged in the field and on what types of characters are used. For example, the field 410 in the example of Figure 26 has only numerical characters. A field of another card, however, may include alphanumeric characters or even hyphens, commas, or colons, for example. Abbreviating the character recognition will depend on the known arrangement, sizes, and types of characters to be encountered in such a field and their spacing. The known arrangement of characters in the OCR field can be supplied from a pattern specification, as described in detail in Section III of the present disclosure. The dimensions and types of characters in the OCR field can be obtained from the form digest 310 and the font/point size digest 320.

In Figure 24C, the OSU then scales the fitted character with the size of a first template character in the character digest 330-1 (183), if the first template character has not been eliminated through another process. The OSU temporarily scales either the fitted character or the first template character in the character digest 330-1 to the size of whichever is the largest. For example, if the fitted character is 28 pixels by 47 pixels, but the template character in digest 330-1 is only 30 pixels by 50 pixels, then rows and columns of white pixels are inserted around the fitted character to produce an equivalent image size of 30 pixels by 50 pixels. The reverse may also be true, wherein the template character requires rows and columns to produce an equivalent image size with the fitted character. Scaling enables the same amount of information to be processed between the fitted character and the template character. Scaling is necessary for a number of reasons, including misalignment of the scanned character, poor edges on the scanned character, or loss of information due to noise and filtering.

At this point, the OSU performs a bilinear reduction of the fitted character under analysis, which may or may not have additional rows or columns added during scaling (184). Preferably, a 2x2-pixel bilinear reduction is performed on the fitted character. For example, in Figure 26, the fitted box or area 440 between the boundaries 420, 422, 430, and 432, if to scale, undergoes the 2x2 bilinear reduction.

Next, the OSU creates a comparative byte data structure (CBDS) of the compressed, fitted character (185). The CBDS of the compressed, fitted character is compared against a stored comparative data structure (CBDS) for the template character within the first character digest 330-1, using the XOR operations described above. The OSU calculates and stores a first

- 84 -

match error for the comparison (186). Then, the OSU continues comparing the CBDS of the compressed, fitted character with the CBDSs of the remaining template characters in the character digests 330-2 to 330-j. Match errors for these comparisons are calculated and stored for each of these character digests 330-2 to 330-j.

5 Not all of the characters in the character digests 330 of the font/point size digest 320 may have the same pixel width and height. Therefore, the fitted character and template characters of the remaining digest 330-2 to 330-j may need to be re-scaled relative to one another, and the process may need to return to steps (183-185) when calculating the match errors for the characters in the other digests 330-2 to 330-j in step (186). It is possible, 10 however, that the dimensions for all the template characters are made the same in the digests 330 for a particular font and point size. This will eliminate the need to re-scale the fitted character and the next template characters to the same dimensions. Of course, if the dimensions of all the template characters in the digests 330 are the same, then abbreviating the selection process by comparing the width of the fitted character with the widths of the template 15 characters cannot be used. Depending on the circumstances, it may be more or less beneficial to have the dimensions of all the template characters in the digest 330 to be the same. One of ordinary skilled in the art would find it a routine undertaking to abbreviate, eliminate, or repeat steps disclosed herein to improve processing of the information.

Once all the match errors are calculated for the template characters in the digests 330-1 20 to 330-j, the OSU determines which template character produces the smallest match error with the fitted character under analysis (187). The determination may depend on a predetermined threshold of the match error as discussed above. The determined character 330 is then recorded in memory as the first character in the OCR field.

It should be noted that the scanned fitted character for the presented card is compared 25 to template information obtained from what is known and expected consistently on the cards of that form. Therefore, if the scanned, fitted character from the card is not of the same font and point size as what is expected, then recognition of the character will fail. Such failure is beneficial for validation and verification purposes and assists in rejecting forms that have altered or changed characters.

30 With the first character of the OCR field determined, the OSU then returns to detecting the edges of the next character in the OCR field (182). The steps (182-187) are repeated until

- 85 -

all of the characters in the OCR field are determined. Ultimately, the OSU determines all of the characters in the OCR field of the card (189) and thus can determine, for example, the consumer's date of birth. The OSU can also perform the steps in Figure 24C on another field of interest that may require optical character recognition. Based on the information contained
5 in the OCR fields, the OSU can determine whether to complete or decline the transaction, for example.

VI. System Configuration

A. Arrangement of Data Structures In The Database

Periodically, it may be necessary to provide updates usable by the OSU devices 10 in
10 the Davis system. For example, in a system placed in service on a nationwide scale, and capable of receiving several different drivers' licenses, the system's templates may need to be periodically updated if a given state changes the structure of its license. Additionally, it may be possible to add new functionality to pre-existing OSU devices 10. Thus, an OSU device initially structured as a vending machine could be updated to also act as a change machine, or
15 may be modified to allow age verified vending. Likewise, the OSU devices 10 may have to report data to the system. Such data can take many forms and could constitute, for example, the sending of the image data taken from the license or provide information relevant to the servicing of the OSU device 10.

Figure 28 shows a subset of the larger Davis system 8 and explains the way in which
20 the central components in the system are managed. This figure shows an OSU device 10, the OSU connection server(s) 12 (OSU CS 12), the server cluster 18, and the management console 22. In this figure, the OSU CS 12 and the server cluster 18 are combined into one logical block in recognition of the similarity in function that these two components may provide. This combination in Figure 28 notwithstanding, in a preferred embodiment, the OSU CSs 12
25 preferably merely act as communication points for the OSU devices 10, while the server cluster 18 stores important system data (such as consumer files and template files), performs necessary computations and interfaces with other non-OSU systems (such as user interface 20, FSS 14 or other integrated systems 24). Of course, one skilled in the art will recognize that these functions could be split between the servers 12 and 18 in any number of ways.

30 Important system data is preferably stored in database 70, including the configuration data for each OSU device 10 present on the system. The configuration of the various data

- 86 -

components necessary to run the system and which are preferably stored in database 70 are shown in Figure 27. Figure 27 illustrates the various data tables and files (more generally, data structures) that are stored in the database, and shows their relationships in an "Entity Relationship Diagram" (ERD) format that is well known to those of skill in the art of database architectures. Pursuant to this format, the various tables within database 70 have relationships structured in a one-to-one (1-1) format, a one-to-many (1-m) format, or a many-to-many (m-m) format. Of course, the database could be structured in a variety of different ways to achieve suitable system performance as disclosed herein. Thus, Figure 27 is merely exemplary of a commercial embodiment.

The contents of each table in Figure 27 are described in the following paragraphs. It is important to note that the database structure supports more than one version of a template. For example, the state of Texas may have three different versions of its driver's license that have been issued and are active, and the system should be able to comprehend all three types. Accordingly, the system stores various versions of the templates and other supporting information relevant to the version, as shown in the disclosed "[Name]_version" tables below.

Consider, for example, tables "Header" and "Header_version" below. The "Header" table has only a few fields, including header name, description, and status. By contrast, the "Header_version" table contains a significant number of fields that apply to OCR analysis, including the form header templates that are used during OCR analysis. If an ID card authority like the State of Texas decides to issue a new license, a new form header version record is created and updated with the latest information. Such an organization scheme is similar to assigning a new model number to a product when just a few features in the product have been changed. In short, through this organizational scheme, a catalog of all versions of licenses issued in the State of Texas can be maintained and referenced in the database.

- Geo: The "Geo" table stores information about the geographical locations of OSU device 10.

<i>Geo</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every geo record
Name	String	Name of the geographical location
Note	String	Description of the geographical location
Creator_id	String	The user id that created this record
Editor_id	String	The user id that edited this record last

- 87 -

Created	String	The date it was created
Edited	String	The date it was last edited
Active	String	Flag representing if the record is active or not

- OSU: The “OSU” table represents information for a particular OSU device 10.

<i>OSU</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every OSU record
Osu_config_id	String	Osu config id that links this OSU record to its configuration record in osu_config table
Serial_no	String	OSU unit serial number
Time_zone	String	Time zone for the OSU unit
Line1	String	Address line 1 for the OSU unit
Line2	String	Address line 2 for the OSU unit
City	String	City in which OSU is located
State	String	State in which OSU is present
Zip	String	Zip code of the OSU location
Directions	String	Directions if any to get to that OSU unit
Cert	String	Certification of OSU unit
Creator_id	String	The user id that created this record
Editor_id	String	The user id that edited this record last
Created	String	The date it was created
Edited	String	The date it was last edited
Active	String	Flag representing if the record is active or not
Acquirer_bin	Integer	Visa assigned Bank Identification number issued by the merchant's member bank or processor
Merchant_number	Integer	A unique number assigned by the signing merchant's bank or processor used to identify the merchant within the VisaNet system.
Store_number	Integer	Number assigned by the signing member, processor to identify a specific merchant store within the VisaNet system
Terminal_number	Integer	Number assigned to identify a unique terminal within a merchant location
Device_code	Character	Device type of the merchant submitting the authorization request
Industry_code	Character	Industry type of the merchant submitting the authorization request
Language	String	Language to be used in formatting the authorization response text message
Merchant_category	Character	Number assigned by the signing member or processor to identify a merchant industry classification
Merchant_name	String	Merchant Name

- Osu_config: This table contains configuration information for each OSU device 10.

It has 1-m relation with “Osu” so that a single configuration can be applied to multiple OSU devices 10. “Osu_config” is linked with “Ocr_form,” “Header” and “Ocr_font_set,” and is

5 related to each with a m-m relation. As will be explained later, each of these three tables is

- 88 -

associated with a corresponding version table. At one time, only one version of each will be active for a particular configurable OSU device 10.

<i>Osu_config</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every OSU record
Name	String	Name of the OSU configuration
Version	Integer	Version of the OSU configuration
Creator_id	String	The user id that created this record
Editor_id	String	The user id that edited this record last
Created	String	The date it was created
Edited	String	The date it was last edited
Active	String	Flag representing if the record is active or not

• Header: This table contains information about the various form headers, and has a 1-m relation with “Header_version” table.

<i>Header</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every header record
Name	String	Name of the header
Description	String	Description of the header
Status	Integer	Status of header record to indicate if this header is the current (indicated by 0), added (indicated by 1) or removed (indicated by 2) one from the configuration

5 • Header_version: This table provides information for the headers, like their form header origin coordinates, and possibly their bottom right coordinates. It also stores multiple versions of the form header templates for the relevant states.

<i>Header_version</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every header version record
Template_package_id	String	Template package record id that this header version is a part of
Header_id	String	Header record id which it is a version of
Version	Integer	Version number of this header version
Image_name	String	Image name used by this header version record
Top_left_id	String	Top left corner of the header region structure
Right_bottom_id	String	Right bottom corner of the header region structure
Creator_id	String	The user id that created this record
Editor_id	String	The user id that edited this record last
Created	String	The date it was created
Edited	String	The date it was last edited
Header_template	Binary	Scanned image of the header version
Active	String	Flag representing if this version is active for its parent header

- 89 -

• **Ocr_font_set**: As mentioned previously, elements in a given form can be written using various fonts, such as Courier font, and these may be printed in different sizes. Basic font information for the elements is provided in the “Ocr_font_set” table. This table has 1-m relation with the “Ocr_font_set_version” table.

<i>Ocr_font_set</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every font set record
Name	String	Name of the OCR Font Set
Description	String	Description if any, for the font set
Status	Integer	Status of font set record to indicate if this font set is the current (indicated by 0), added (indicated by 1) or removed (indicated by 2) one from the configuration

5 • **Ocr_font_set_version**: This table is dependent on “Ocr_font_set” and provides information for any “Ocr_font_set.” The basic information for each of the fonts is stored within this table. Thus, “Family” represents the basic font type (e.g., Arial or Courier), “Font_size” represents the size of the font (e.g., 10 point or 12 point), and “Style” represents modifications of the font, such as italicized or bolded. It has 1-m relation with “Font_type”
10 table.

<i>Ocr_font_set_version</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every font set version record
Template_package_id	String	Template package record id that this font set version is a part of
Ocr_font_set_id	String	Font set record id which it is a version of
Version	Integer	Version number of this font set version
Family	String	Family of the font set (e.g., Arial or Courier)
Font_size	String	Size of the font set (e.g., 10 point or 12 point)
Style	String	Style of the font set (e.g., bold or italic)
Creator_id	String	The user id that created this record
Editor_id	String	The user id that edited this record last
Created	String	The date it was created
Edited	String	The date it was last edited
Active	String	Flag representing if this version is active for its parent font set

• **Font_type**: This table stores the various types of characters recognizable by the system, such as “A” for upper case letters A-Z, “a” for lower case letters a-z, “N” for numbers 0-9, “P” for punctuation and symbols (such as .,“”-/:!~?[]{}%\$), “Z” for any upper or lower case letter, “X” for any letter or number, “*” for a wildcard representing any character, and “S”
15 for a space. It has a 1-m relation with “Font_pattern” table.

- 90 -

<i>Font_type</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every font type record
Ocr_font_set_version_id	String	Font set version record id which it is a type of
Font_type	String	Specifies the type of character stored in the associated font type (e.g., "A," "a," "N," "P," etc.)
Description	String	Description of the character that font type has

• **Font_pattern:** This table stores the character templates for a given font. For example, there would be twenty six templates stored within the "Font_pattern" table for each upper case letter and for each font type. Thus, assuming two fonts (e.g., arial or courier), there would be a total of 52 templates stored for each font type "A," representing upper case letters.

<i>Font_pattern</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every font pattern record
Font_type_id	String	Font type record id which this pattern is a part of
Name	String	Name of the pattern
Font_data	Binary	Image of the font pattern

5 • **Ocr_form:** This table stores information for a form template. It is related to the "Ocr_cluster" table by a 1-m relation because a form template can have many clusters defined within it. It is associated with the "Header" table by a 1-1 relation that links the header belonging to a particular form. It is also related to the "Ocr_form_version" table. If any information is changed for an existing form template, a new version of it is created and a
10 record is set for it in "Ocr_form_version" table.

<i>Ocr_form</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every form record
Geo_id	String	Link to the Geo table for associated state information for a form record
Header_id	String	Header id for the form header
Name	String	Name of the form (e.g., Texas driver's license)
Description	String	Description if any of the form
Status	Integer	Status of form record to indicate if this form template is the current (indicated by 0), added (indicated by 1) or removed (indicated by 2) one from the configuration

• **Ocr_form_version:** This table is dependent on the "Ocr_form" table and stores version information for each form. Included within this table is the X and Y coordinates for the starting position of the image under analysis. Thus, if it is known that the first ten pixels of a given form image contains information not indicative of the content of the form (e.g.,

- 91 -

because of the rounded corners that exist on the form), these first ten pixels can be ignored during OCR.

<i>Ocr_form_version</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every form version record
Template_package_id	String	Template package record id that this form version is a part of
Ocr_form_id	String	Form record id which it is a version of
Version	Integer	Version number of this form version
Xpos	Integer	X coordinate of the starting point of the form template
Ypos	Integer	Y coordinate of the starting point of the form template
Creator_id	String	The user id that created this record
Editor_id	String	The user id that edited this record last
Created	String	The date it was created
Edited	String	The date it was last edited
Active	String	Flag representing if this version is active for its parent OCR form

• Ocr_cluster: This table is dependent on “Ocr_form” table and provides a list of clusters for a particular form. It has 1-m relation with the “Ocr_cluster_version” table that provides versioning support. As discussed earlier, a cluster is a group of several elements. Therefore, “Ocr_cluster” is associated with the “Ocr_element” table to provide a list of necessary elements.

<i>Ocr_cluster</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every cluster record
Ocr_form_id	String	Form id which this cluster is a part of
Header_id	String	Header id for this cluster
Name	String	Name of the cluster
Description	String	Description of the cluster

• Ocr_cluster_version: “Ocr_cluster_version” stores the top left and right bottom coordinates for the cluster header origin and also stores the cluster header template images. Thus, for example, this table is where the cluster header image for the cluster containing the date of birth (such as “CLASS:”, “DOB:”, or the image of the Texas flag) would be stored.

<i>Ocr_cluster_version</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every cluster version record
Template_package_id	String	Template package record id that this cluster version is a part of
Ocr_cluster_id	String	Cluster record id which it is a version of
Version	Integer	Version number of this cluster version
Name	String	Name of the cluster version
Point_id	String	Starting point (X,Y) for the cluster version template

- 92 -

Cluster_template	Binary	Cluster image for this version
Creator_id	String	The user id that created this record
Editor_id	String	The user id that edited this record last
Created	String	The date it was created
Edited	String	The date it was last edited
Active	String	Flag representing if this version is active for its parent OCR cluster

• **Ocr_element**: This table stores the name and description of particular elements, such as date of birth, expiration date, name, etc. It also is related with “Ocr_element_version” table through a 1-m relation that provides versioning support.

<i>Ocr_element</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every element record
Ocr_cluster_id	String	Cluster id which this element is a part of
Name	String	Name of the element
Description	String	Description of the element

• **Ocr_element_version**: The “Ocr_element_version” in effect stores the element origins for the various elements within a cluster. Thus, this table stores top left and right bottom coordinates (“top_left_id” and “bottom_right_id”) for sliding a character template during OCR analysis, and preferably defines a small rectangle at the upper left corner of the character under analysis. In this regard and as disclosed earlier, it has been noted that the location of an element within a cluster varies approximately plus-or-minus one pixel within the cluster. Therefore, and for example, a small rectangle, perhaps 3-by-3 pixels in dimension, is set at the element origin in the test image where it is expected that the first character in the element is located. In other words, the small rectangle defines the element origin in the test image as a variable region. The upper left pixel of the character template is then moved or slid to correspond to one of the nine pixels within the 3-by-3 pixel rectangle, and a distance metric is calculated for each position. The minimum of these nine distance metrics will define the location of the first character of the element under analysis. This procedure is then repeated as the sequence estimation algorithm sequentially identifies each character in an element.

Also referenced in this table are the various fonts and pattern specification that are to be used for the various elements during OCR analysis.

<i>Ocr_element_version</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every element version record
Template_package_id	String	Template package record id that this element version is a part of

- 93 -

Ocr_font_set_id	String	Font set record id for this element version record
Element_pattern_id	String	Element pattern id for this element version record
Ocr_element_id	String	Element record id which it is a version of
Version	Integer	Version number of this cluster version
Top_left_id	String	Top left corner of the element region structure
Right_bottom_id	String	Right Bottom corner of the element region structure
Creator_id	String	The user id that created this record
Editor_id	String	The user id that edited this record last
Created	String	The date it was created
Edited	String	The date it was last edited
Active	String	Flag representing if this version is active for its parent ocr element

• **Element_pattern:** The “Element_pattern” table is linked to “Element_pattern_character” table with a 1-m relation and is linked to the “Ocr_element_version” table with a 1-m relation. The purposes of the “Element_pattern” and “Element_pattern_character” tables are to specify information about the pattern specification. For example, in the aforementioned
 5 pattern specification representing the six-digit date of birth (i.e., N[1:2] “-” N[1:2] “-” N[2:2]), there are five pattern characters in the pattern specification, three denoting the month, day, and year (N[1:2] and N[2:2]), and two denoting the dashes that separate them (“-”). Thus to create a database representation of a six-digit date of birth, one would create a record in the “Element_pattern” table with the name of “6-digit date” and then create pattern character
 10 entries in the “Element_pattern_character” table, each linking back to the newly created “Element_pattern” record.

<i>Element_pattern</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every element
Name	String	Description of pattern specification (e.g., “6-digit date,” “social security number,” etc.)
Creator_id	String	The user id that created this record
Editor_id	String	The user id that edited this record last
Created	String	The date it was created
Edited	String	The date it was last edited
Active	String	Flag representing if this pattern specification is active

• **Element_pattern_character:** The “Element_pattern_character” table stores information concerning each pattern character in the pattern specification. Thus, stored here are information for each pattern character’s character type (e.g., “N” representing numbers, or
 15 the dash symbol) and length of the pattern character, represented by minimum and maximum number of occurrences of the character of that type (e.g., a minimum of 1 for the month, and a maximum of 2 for the month). “Seq” stands for sequence and denotes the order of the pattern

- 94 -

characters within the pattern specification. Thus, "Seq" equals 1 for the first pattern character (i.e., N[1:2]), 2 for the second pattern character (i.e., "-"), and eventually would equal 5 for the last pattern character (i.e., N[2:2]).

<i>Element_pattern_character</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every element pattern character record
Seq	Integer	Identifies the place of a pattern character in it pattern specification.
Element_pattern_id	String	Element pattern id for this pattern character record
Character_type	String	Describe the type of character (e.g., "N" for numbers, "A" for upper case letter, etc.).
Min	Integer	Minimum character length of the element pattern character (e.g., 1 for month or day, or 2 for year).
Max	Integer	Maximum character length of the element pattern character (e.g., 2 for month, day, or year)

• **Template_package:** This table provides versioning support for all OSU configuration components. It stores the version number of latest configuration and also the lists for "Header_version," "Ocr_font_set_version," "Ocr_cluster_version" and "Ocr_element_version." Note the various tables contain a field called "template_package_id" that provides the link or relationship to the "Template_package" table. This table is associated with each of these other version tables by a 1-m relation.

<i>Template_package</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every template package record
Version	Integer	Version number of the template package
Created	String	The date it was created
Active	String	Flag representing if this template package is active or not

• **Trn:** This table stores the Visa Net transactions performed for the OSU device 10. It is linked to "Osu" table through a 1-m relation.

<i>Trn</i>		
<i>Name</i>	<i>Type</i>	<i>Description</i>
Id	String	Unique identifier for every transaction record
Osu_id	Integer	Osu id to which this transaction record is linked
Returned_aci	String	Returned requested authorization characteristics indicator
Store_number	String	Number assigned by the signing member, processor to identify a specific merchant store within the VisaNet system
Terminal_number	Integer	Number assigned to identify a unique terminal within a merchant location
Auth_source	Character	Source of the authorization code
Trans_sequence	Integer	Terminal generated transaction sequence number
Response_code	String	Code indicating the status of the authorization request

- 95 -

Approval_code	String	Authorization code when a transaction has been approved
Local_trans_date_time	String	Date and time when the transaction took place
Auth_response	String	Response or display text message
Avs_result	Character	Avs Result
Retrieval_ref_number	String	Transaction retrieval reference number returned by the authorizing system
Market_data_identifier	Character	Industry specific data being submitted
Trans_id	Integer	Visa transaction identifier or Master Card reference number
Validation_code	String	Specific information generated by the card issuer
Group_ver	Integer	Addendum data group version number
Committed	Character	Flag representing if the transaction has been committed or not

Whenever the configuration of an OSU device 10 is changed by the vending machine operator or Davis system administrator executing an update, a new version for that device is created and is added to its version table(s). At the same time, the "Template_package" table is updated. When an OSU device 10 connects to the system, its current configuration version number is supplied and is checked against the version number present in "Template_package" table. If the number present in the table is greater than the one sent by the device, that device requires an update. The latest configuration data is then retrieved from the database 70 by reviewing all the version tables discussed above. An update package is then created for and sent to the device. If the version numbers match, meaning no change is necessary in the configuration of the device, the server cluster 18 checks to see if a (new) template needs to be added to or deleted from that device's configuration file, and again an update package is created and sent accordingly. Update packages are created and sent to the devices in a format specified by the DTP protocol as explained earlier.

Control and management of the system occurs at management console 22, which was explained in some detail earlier. It is from this console that new data is entered onto the system, such as new or improved templates for the OSU devices 10, or new configuration updates for the OSU devices 10. Console 22 may also be used to add new OSU devices 10 to the system. System information may also be retrieved to console 22. For example, console 22 can obtain updates sent from the OSU devices 10, retrieve a template list supported by any OSU device 10, or delete templates from an existing OSU device 10.

Of course, database 70 also preferably includes data files for each of the consumers who have either pre-registered to use the system, or who have used the system. Such consumer files may contain important information about the consumer, such as their names and

- 96 -

addresses, information regarding their registered accounts or actual storage of the accounts, and may also contain information that might make subsequent processing of the consumer's information easier. For example, once a consumer's date of birth has been assessed, it can be stored. Thereafter, if the system determines (through OCR) that a particular customer has requested to make a purchase, that consumer's file can be analyzed and, for example, the date of birth retrieved, making further OCR processing of the license potentially unnecessary.

In addition to the table definitions described above, in a commercial system, there may be over 100 tables in database 70 that are used to support and collect audit data, often referred to in the art as DEX or EVA data, and which was briefly discussed earlier. For more information concerning these DEX related data constructs, the reader is referred to "European Vending Association Data Transfer Standard," European Vending Association, Release 5.0, July 1999, which is hereby incorporated herein by reference for all that it teaches.

B. Update Payload Information

As has been discussed previously, it may be necessary to update the templates or other configuration information resident in the OSU 6 for optically analyzing a given license. Below is shown an example of the payload information that is sent by DTP to an OSU 6 to provide an update to such information, i.e., a Type 7 "Update Response" packet. This example shows only a representation of the payload of data and does not otherwise show DTP header or other supporting information that will also be sent with the Update Response packet. As mentioned earlier, the payload will ultimately be stored in the OSU 6, preferably in Flash 231.

As one skilled in the art will recognize, the payload information is organized in hierarchical data structure, as opposed to the relational database format used to organize the data structures in the database 70 associated with server cluster 18 (see Figure 27). In other words, the payload information is organized in a "tree format," with one data structure referencing another and so on. Thus, the "Form" data structure references a data structure called "Point," which has its own data structure, and so on. It will also be appreciated by those of ordinary skill that multiple versions of the data structures listed below will like be sent in a given application. For example, "Cluster N" will be sent for every N clusters of interest in the form, and "Element N" will be sent for every N elements of interest within a particular cluster.

- 97 -

Preferably, each Update Response packet contains configuration information for a single form. Thus, if it were necessary to fully update the OSU 6 to provide image templates and other supporting OCR data for the states of Texas and Louisiana, at least two Update Response packets would be sent.

5 One or more update responses may be sent to the OSU device to complete a full upgrade. The first byte of the update response payload determines the extent of the update that is to occur at the OSU 6. If the first byte equals "00," the font information will be updated or added if it doesn't presently exist. If the first byte equals "01," this is interpreted by the OSU 6 as an instruction to delete a font residing in the OSU 6. If the first byte equals "1X," the form
10 header templates stored in the OSU 6 are implicated, with "10" designating the replacing of an old template or the additional of a new header template, and "11" designating the deletion of a header template. If the first byte equals "2X," other form structure information is implicated, such as the cluster information, the cluster headers, the character templates for the elements, the pattern specifications, etc. Specifically, "20" designates and update or addition to such
15 information, while a "21" designates a deletion.

Some of the data in the payload is variable in length. For example, the cluster list may contain several clusters or only a single cluster. For this reason, the cluster list data structure contains at its end an End Of Line ("EOL") marker to denote the boundary between that data structure and the next data structure in the payload.

20 Keeping the foregoing in mind, the Update Response payload is preferably represented as follows. Parenthetical description are used to tie the various data structures to concepts introduced earlier in this specification:

"Form":

- 25 • name (e.g., Texas driver's license)
- "Header"
- "Point" (The reference point for the form. It is the pixel location where (x,y) = (0,0))
- "Clusterlist" (a list of the clusters within the form)

30 "Header": (represent both form and cluster headers)

- name (e.g., "Texas" or "Texas date of birth")
- "Region" (defines the expected location of the header on the form as a rectangle, thus providing the header origin)
- 35 • header image name (identifies the name of the "Header Image" data structure. I.e., this name points to the correct header image data structure)

- 98 -

“Region”:

- “Point” (top left corner)
- “Point” (right bottom corner)

5

“Point”: (specifies a particular pixel)

- X (16 bits)
- Y (16 bits)

10 “Clusterlist”: (a list of clusters associated with the form)

- “Cluster 1”
- “Cluster 2”
- “Cluster N”, etc.
- "EOL"

15

“Cluster N”:

- “Header”
- “Point” (i.e., the cluster header origin or pixel locate that is remapped to be (x,y)=(0,0) for the cluster reference point. Offset values for OCR Elements are calculated relative to this point.)
- “Elementlist” (a list of elements associated with the cluster)

20

“Elementlist”: (a list of elements associated with each cluster)

- “Element 1”
- “Element 2”
- “Element N”, etc.
- "EOL"

25

“Element N”:

- name (e.g., 6-digit date of birth)
- “Region” (defines expected location of the element, i.e., the element origin, with necessary variance as explained earlier)
- “Pattern”
- “Font” (specifies the font type for the element)

30

35

“Pattern”: (defines the pattern specification)

- pattern length (16 bits) (defines the number of pattern characters in the pattern specification, e.g., 5 for a 6-digit date of birth. Note: this length obviates the need for an “EOL” marker)
- “Pattern Character 1” (e.g., the month for the date of birth, i.e., N[1:2])
- “Pattern Character 2” (e.g., “-“)
- “Pattern Character N”, etc.

40

“Pattern Character N”:

- 99 -

- “Character Type” (a one byte variable that specifies a particular character type, e.g., “N” for numbers, “A” for capital letters, etc.)
- number (a one-byte variable that tells the minimum and maximum number of characters to look in the particular pattern character)

5

“Header Image”: (i.e., the header templates)

- name
- colnum (16 bits) (specifies the number of columns in the template)
- rownum (16 bits) (specifies the number of rows in the template)
- data (pixel data)

10

“Font”:

- name
- “Font Type List”

15

“Font Type List”: (lists the various types of fonts, e.g., Courier 12 pt., Arial 10 pt., etc.)

- “Font Type 1”
- “Font Type 2”
- “Font Type N”, etc.
- "EOL"

20

“Character Type N”:

- Type (a byte that specifies the type of characters stored in the associated list. E.g., A = upper case letters (A-Z), a = lower case letters (a-z), N = numbers (0-9), P = punctuation and symbols e.g., "-/;:!?()[]{}%\$", Z = any upper or lower case letter, X = any letter or number, * = wildcard (any character, S = space)
- “Character Template List”

25

30 “Character Template List”:

- “Character Template 1” (e.g., may represent the template for the number “0” or the letter “A”)
- “Character Template 2” (e.g., may represent the template for the number “1” or the letter “B”)
- “Character Template N”, etc.
- "EOL" (null terminated string)

35

“Character Template N”: (i.e., the character templates)

- name
- colnum (16 bits) (specifies the number of columns in the template)
- rownum (16 bits) (specifies the number of rows in the template)
- data (pixel data)

40

- 100 -

VII. Modifying a Preexisting Vending Machine to Incorporate an OSU/OSU Architecture

One of the advantages of the disclosed system is its ability to work with preexisting vending hardware. Only slight modifications are needed to retrofit such pieces of equipment with the OSU 6 disclosed herein. How such modifications are made to a standard vending machine is disclosed as illustrative of this process, but similar techniques would be used to modify other pieces of equipment, as one skilled in the art will recognize. The structure, functionality, and operation of such standard vending machines is also discussed in U.S. patent applications 09/836,805 and 09/851,198, which are incorporated by reference herein in their entirety, and which are assigned to the present assignee.

Figure 29 shows a typical vending machine 79, including a display 81, a keypad 87 for making selections in the machine, a product dispensing mechanism 84 (typically a hinged door), a product selection window 75, and (internally) a microprocessor 82. Also present are a currency acceptor 88, which receives paper or coin money, and a credit card acceptor 89. (One skilled will realize that in an actual embodiment, the currency acceptor 88 would likely constitute two separate acceptors to handle the two different types of currency. Furthermore, other standard vending machine components, not necessary to facilitate the discussion of the invention, are not disclosed). Also shown in Figure 29 is the International Multi Drop Bus (IMDB) 96, which supports a communication protocol usable in standard vending machines. In the standard vending machine, microprocessor 82, through the IMDB 96, polls payment devices 88 and 89 to see if payment has been made. In a standard vending machine, once payment has been made, the product may be vended.

However, in a vending machine retrofitted to include a OSU 6, vending of the product is, at least in some embodiments, only to be made upon verification of certain consumer information, such as his age or the validity of his driver's license. For this reason, the vending machine logic is preferably modified as shown in Figure 30. In this figure, microcontroller 230 (see also Figure 6) has been inserted between the microprocessor 82 and the payment devices 88 and 89. (In this disclosure, for convenience and clarity, the control device within the OSU 6 is referred to as a "microcontroller," while the control device on the vending machine is referred to as a "microprocessor." However, as one skilled in the art will realize, these two devices are interchangeable as both microcontrollers and microprocessors perform

- 101 -

similar functions). The microcontroller 230 is also in communication with a communication device 236, such as a modem, which in turn is connected to an OSU CS 12 as described above. (Of course, a modem is merely an exemplary way of providing communication, and any other suitable form of communication, e.g., wireless or optical cable, is also contemplated). Before
5 allowing a purchase, the microcontroller 230 waits for the results of the OSU analysis to determine if the condition for purchase (e.g., age) has been met. If this condition is met, and if payment has been made at either of payment devices 88 or 89, then the vend will be made via the product dispensing mechanism 84. The microcontroller 230 must be operated in a fashion such that the microprocessor 82 doesn't know that the IMDB bus 96 has been disconnected
10 from the payment devices 88 and 89, and therefore will send mock polling data on bus 96 for this purpose. Additionally, the microcontroller 230 must now perform the function of polling the payment devices 88 and 89, just as did the microprocessor 82 before the retrofit.

Figure 31 shows further details concerning the relation of microcontroller 230 in a vending machine retrofitted with an OSU 6. Other desirable features integrated with the OSU
15 6 are also shown, such as a cancel button 233, which allows the consumer to cancel a transaction already in progress, and a printer 234, capable of printing truncation receipts. Also present in Figure 31 are IMDB control circuits 235 that handle the disconnection and polling of the IMDB 96 as explained earlier, and which are present on the printed circuit board along with microcontroller 230, flash 231, and SRAM 232.

20 In the retrofitted device, the microcontroller 230 essentially becomes the master processor and largely takes control over the native microprocessor 82 already present in the vending machine, although the microprocessor 82 will continue to run other normal vending machine functions, such as running the machine's dispensing motors and display 81. The microcontroller 230 directly communicates with the communication device 236, the cancel
25 button 233, the payment devices 88 and 89, and the printer 234. The microcontroller 230 also controls the IMDB bus 96 through IMDB control circuits 235. To the extent that the microprocessor continues to run local systems functions like enabling the vending of products, it does so through receipt of commands from microcontroller 230. Microcontroller 230, when operating in an "off-line" mode, can make decisions concerning allowing or not allowing a
30 purchase. When operating in an "on-line" mode, microcontroller 230 receives instructions

- 102 -

from the server cluster 18 through the communications device 236 concerning allowing or not allowing a purchase.

While the disclosed embodiment shows a traditional vending machine retrofitted with an OSU, one could of course build in accordance with the teachings in this disclosure an OSU device 10 from scratch containing an OSU. In such OSU original models, the architecture and circuitry could be arranged in any number of ways to achieve suitable functionality, as one skilled in the art will immediately recognize. For example, it would probably be beneficial in an OSU device 10 designed from scratch to combine the functionality of the verification controller 93 and the microprocessor 82 into a single microprocessor, and perhaps to dispense with the use of the microcontroller 230 IMOB bus 96 altogether. Likewise, it may be desirable for the microcontroller 230 to be positioned outside the OSU, or to reprogram an existing microprocessor 82 to perform the functions of the microcontroller 230 as disclosed herein.

VIII. System Installation and Initialization

Suppose a vending machine operator, Bob's Beverages ("Bob"), purchases a Davis system enabled beverage vending machine equipped with an OSU 6. Bob desires to sell alcoholic beverages from the machine in a hotel/casino in Las Vegas, Nevada. Bob, using a web browser on the public internet, e.g., from his interface 20, goes to the Davis system 8 website and "logs in" to a secure portion of the site using the user name and password that he received either when earlier registering with the system on-line or when he purchased the machine. Bob then creates a vending machine pool on the website and adds one machine to it—the machine scheduled for delivery to the hotel. He enters data about the new vending machine to register it with the system, such as its unique identification number, machine type, location, etc.

Bob may then uses the on-line machine configuration editor to set machine and OSU 6 operation parameters, i.e., Bob creates a configuration file for his machine on-line. For example, Bob may review what types of ID card templates are currently supported by the system and may select which of those will be accepted by his machine. Thus, if the system currently supports 100 ID types, including 50 state driver's licenses type, Bob may choose all ID types or some subset of these to be supported by his machine. This ID type selection process will allow the templates for the selected ID card types to eventually be sent by the system to the OSU 6 in Bob's machine. With the configuration editor, Bob may also configure

- 103 -

other functional aspects of his machine. For example, Bob may specify that periodic audits be scheduled for his machine, e.g., that DEX/EVA information be sent daily at 2:00am. He may also specify that only certain product selection window 75 rows will be used to sell age restricted alcoholic beverages, and therefore that the consumer's age will need to be verified by the system to vend products from these rows. He may further configure the system to accept either cash, coin, and credit card payment methods, and may require credit card information to be supplied by the consumer to provide further validation of the consumer's identity. After setting the relevant machine configuration parameters, Bob may now "log out" from the site.

When the machine arrives at the hotel/casino location, Bob plugs it in and connects it to a telephone jack. At this point, the OSU 6 in the machine begins an initialization phase that preferably is factory pre-programmed into the machine, preferably in Flash 231. The machine accordingly dials a stored phone number to connect to the Davis system 8, and more specifically and preferably to a designated initialization computer connected to the system 8. That computer receives the call by modem, answers it, and notifies a relevant OSU-CS 12 on the system (e.g., one in the vicinity of the machine) that a connection is being attempted. The OSU-CS 12 attaches to the connection and requests security credentials from the OSU 6, again which are pre-programmed. The OSU-CS 12 then in secure fashion authenticates the OSU 6 as a new vending machine for the Bob's Beverages account, e.g., by verifying the ID code for the machine. Thereafter, a connection is established with the server cluster 18, thereby establishing a "session" as described earlier. The Davis session is responsible for maintaining dialogue with the OSU 6, via the OSU-CS 12, and for performing services on behalf of the OSU 6. In this case, i.e., during the initialization phase, the OSU 6 needs to be updated with the latest software and ID card support.

The OSU 6 makes an "Update Request" to the server cluster 18, which is initially transmitted to the OSU-CS 12 using the DTP protocol described earlier. The OSU-CS 12 receives the packet and accordingly requests the server cluster 18 to provide a data structure for the updates. The server cluster 18 in turn creates an EJB (Enterprise Java Bean, per the Java 2 Enterprise Edition platform defined by Sun Microsystems) to perform the service. This EJB then accesses system data to create an "Update Response" packet. During initialization, Bob's previously created configuration file is consulted to understand the functionality that is needed at Bob's machine. For example, in accordance with the configuration file, Bob may

- 104 -

receive the necessary templates to perform template matching and identification for all 50 states, and may receive further template data for these states to read and interpret the date of birth on the license to verify the consumer's age. The "Update Response" is returned to the OSU-CS 12, which in turn repackages the data into a DTP packet and sends the data to the OSU 6 as described earlier. The OSU 6 then updates itself with the new data, preferably by storing it in Flash 231. The server cluster 18 then receives notification from the OSU 6 that the upgrade completed successfully. Optionally, the server cluster 18 may send an e-mail to Bob's user interface 24 to confirm the completion of the update procedure.

At this point Bob is ready to stock his machine and put it into operation. Suppose a 43-year-old hotel guest from Texas passes by the machine and decides to purchase a beer. He makes his selection and is prompted by the display 81 to swipe his credit card into credit card acceptor 88 or insert cash into currency acceptor 88. The consumer chooses to insert his credit card and then is prompted to insert his driver's license into OSU 6. He does so and in a few seconds receives his license back. A few seconds later, after locally performing the license and birth identification procedures outlined earlier, the display 81 states "purchase successful" and his can of beer is dispensed. By contrast, a 17-year-old hotel guest from Colorado passes by the machine and tries to purchase a beer. He makes his selection and inserts a five-dollar bill when prompted. He then insert his drivers license. After failing the age verification procedure, the display 81 may state "Purchase denied. Must be 21 for purchase. Please make another selection." That consumer then may select a soda for purchase, or may receive his five dollars back by aborting the transaction and selecting a change return option.

Assume many other purchases are made throughout the day. Then, at 2:00 am the next morning, and pursuant to Bob's desires as reflected in his downloaded configuration file, the machine dials the server cluster 18 via OSU CS 12 and uploads its DEX information. In the morning, Bob checks his e-mail and may find a received message from the system 8 saying that his machine was successfully audited. The message also preferably provides a link to the audit information. Bob may then click on the link and log into the Davis system where he may view the audit report for his new machine. From this report, Bob may review detailed information concerning each information field collected by the DEX feature. For example, he can view information about each transaction, he can determine his inventory in the machine, see what product is most popular, "when" it is most popular, at what price, etc. After one

- 105 -

week, Bob generates a location report to show hotel management how successful the machine has been with consumers. Based on its success, he receives approval to place one machine on each of the hotel's 50 floors plus 9 additional units throughout other areas of the hotel and casino.

5 Bob then purchases, configures, installs, and stocks the new machines as outlined above, bringing the total of Bob's machines at the hotel to 60. Ultimately Bob may expand his presence into other regions with many other machines, all of which can be easily managed and tracked using the disclosed system 8. Importantly, Bob may also have his machines automatically updated with the latest software and image templates to further improve the
10 functionality of his machine.

IX. Other Embodiments

While this disclosure has primarily focused on the vending of age-restricted products as an illustrative embodiment, the technology disclosed in the system is capable of vending other products and services in a reliable and efficient manner, and performing other useful tasks.

15 An important advantage of the system stems from its ability to treat ordinary ID cards, such as driver's licenses, as "smart cards," even when those cards do not contain means for electronically holding consumer information, such as magnetic strips or integrated circuits. In conjunction with the use of a personal identification (PIN) number, the ordinary driver's license, or any other ID card issued in any jurisdiction, opens the consumer to an enhanced
20 ability to electronically purchase items and services, and without the need for vendors to issue specialized and expensive smart cards, which are usually only useful for purchasing a particular vendor's product.

Thus, the Davis system provides a convenient, low-cost, platform that provides "smart card" functionality. Furthermore, OSU devices 10 can easily be present at or incorporated in
25 merchant point-of-sale equipment, building entrances, vending machines, cars, pay phones, personal computers, gas pumps, and personal data assistants (PDAs), enabling the consumer to use such devices with only his driver's license or other ID card. Indeed, a Davis system may contain several of these types of terminals (e.g., vending machines and gas pumps) in one network.

- 106 -

Here are some examples where the disclosed technology is expected to be useful:

- Law Enforcement: A police vehicle equipped with an OSU allows a driver's license to be scanned. If the system includes or is connectable to a law enforcement system, information concerning the driver's record could be verified on the spot, and without the necessity of keying drive license data into a computer.
- Vehicle Rental: Cars equipped with OSU devices could be rented, perhaps without the assistance of a rental car attendant. In one embodiment, cars could be directly equipped with OSUs which communicate with the Davis system by wireless means. The license could then be verified as valid. Additionally, and if the consumer does not already have an account on the system capable of paying for the rental, the consumer could be asked to insert his credit card, either as an extra validation measure or to pay for the rental or both. Approval of both the license and the credit card would then allow the car to be started, either automatically or by turning the ignition key. (After payment has been arranged, only insertion of the driver's license would thereafter be necessary to start the car). Such a system is particularly advantageous because it allows validation of the driver's license, ensures that the license was not suspended or revoked (if linked to a law enforcement system), and allows a means for payment via the ID card, making the rental process a quick and fully automated procedure. As an alternative, an OSU-equipped vending machine could be used to dispense keys after license (and perhaps credit card) validation in much the same way.
- Automated Forms Processing: Standard forms, such as insurance cards, could be scanned in order to automate data entry of the information contained thereon. Manual data entry is by comparison slow and error prone.
- Security Card: High security areas, such as building entrances, parking garages, certain rooms within a building, etc., when equipped with an OSU, would allow or disallow access (e.g., by locking or unlocking doors or gates) merely upon an assessment of a driver's license, and without the need to issue special access cards.
- Check Cashing/Credit Card Transactions: OSU devices connected to the Davis system could be used as an extra security check to verify the identity of those presenting licenses to support the cashing of a check or those using credit cards to make a purchase.

- 107 -

- Gas Pumps: A gas pump equipped with an OSU could not only be used to vend the gas and pay for the purchase, but could also allow the license to be checked for validity if interfaced with an appropriate law enforcement system. If the consumer's license is determined not to be valid or has been suspended or revoked, the purchase could be disabled, with the tangential benefit of keeping unlicensed drivers from driving. Additionally, the system could be programmed to receive periodic updates (e.g., daily) from the law enforcement system concerning license status (suspended, revoked, valid), which could then be stored in database 70. In this embodiment, the system would not need to query the law enforcement system each time a consumer made a purchase request, but could instead maintain a file on database 70.
- Validation of Passports and Visas: If the OSU devices 10 were fitted with flat bed scanners, or a modified version of the OSU 6, they could be used to allow passports and visas to function in much the same way as driver's licenses or other ID cards as disclosed herein. Thus, customs or other officials could employ such OSU devices 10 to verify such information on the spot for travelers and other persons. Thus, the Davis system to which the OSU devices are connected could be connected to government agency databases to verify that the passport or visa is valid, contains the correct information, and has not been tampered with. In such an application, the OSU could be used to determine, by OCR, the traveler's name, and this name could be sent by the system to a government agency database, to pull other desired information for the individual, such as his immigration status. Additionally, and in conjunction with the proper database, background or criminal checks could be run. If necessary, the photo on the passport or visa (if any) could be sent in real time to personnel at these agencies for a manual photo check. This may be useful in the apprehension of terrorists, missing persons, and criminals. Additionally, future technologies may allow for passport photos to be pre-scanned and stored, possibly allowing template matching of the faces stored on the scanned and stored ID photos.
- Locating Criminals: In another embodiment, certain drivers containing suspended or revoked licenses, or which have criminal records, could have their licenses "hot listed" in the system by law enforcement officials. When such licenses were used at any OSU device 10 connected to the system, special procedures could be put in place by the

- 108 -

system which would immediately notify law enforcement agencies of the time, date and location in which the purchase was completed or attempted, with the hope that such persons could more easily be brought to justice. Additionally, such information could be stored in the system and made accessible for law enforcement officers to review at their leisure. Such an OSU device 10 might be especially well-suited for installation at airports, where passenger identities could be verified. Thus, when a passenger checks in at the airport he would be required to insert his drivers license into an OSU device 10, which in turn could be connected to a national database to check if the person was on a "watch list."

- License plate capture: In another embodiment, an automobile license plate image could be captured and processed. An OSU 6 can be embedded with or connected to an image capturing device, such as a video camera or motion-sensitive still image camera. Using such a device, license plates could be optically captured by law enforcement officers (e.g., to nab speeding drivers) and automatically processed to tap into information databases containing, for example, vehicle registration information. Such a device could also be used in parking garages to capture information about who is entering and exiting the garage, or to authorize access.
- Tamper-proofing Photos: In another embodiment, an ID card photo image can be compared with the original photo when the ID card was created. After a person is issued an ID card, the image is stored in a database connected to a Davis system. As the person uses the card in an OSU device, the two are compared. Specific points or the entire image can be compared to determine if the image has been significantly altered.

As well as having other uses, the disclosed system may be implemented in a number of different ways depending on the desired system functionality. Databases and/or servers could be combined with OSU devices. Other components disclosed herein as being integrated could also be separated if desirable. The specific hardware components could be easily changed or altered by those of ordinary skill. Furthermore, the system may be used to vend a wide array of products and services. For example, some of the OSU devices 10 could be configured to vend age-restricted products, while other OSU devices 10 on the system could be configured to

- 109 -

act as ATMs, security monitors, gas pumps, etc. The disclosed system therefore has great flexibility.

Moreover, the use an OSU is not strictly necessary to realize some of the benefits that are disclosed herein. Other suitable means for receiving consumer information, e.g., such as
5 by computer or keypad, or through electronic means such as by credit cards containing magnetic strips or smart cards containing integrated circuitry, may be useful in certain novel aspects as disclosed herein. In this vein, it should be noted that the disclosed systems and associated methods are believed to be patentable in several different respects, and with respect to several of its components and/or subcomponents, even if the benefits of these other
10 inventive aspects have not been specifically touted in this specification.

The concept of storage of data within a memory refers to storage in any suitable means for retaining digital data, such as in a memory chip or on a magnetic disk. References to multiple memories in the appended claims, such as a first memory and a second memory, should be understood as referring generally to storage in separate discrete memory devices, or
15 storage on a single device in different memory locations, registers, or blocks within the same memory device.

From the foregoing detailed description of specific embodiments of the invention, it should be apparent that a system and associated methods for vending products and services using an identification card has been disclosed. Although specific embodiments of the
20 invention have been disclosed herein in some detail, this has been done solely for the purposes of illustrating various aspects and features of the invention, and is not intended to be limiting with respect to the scope of the invention. It is contemplated that various substitutions, alterations, and/or modifications, including but not limited to those design alternatives which might have been specifically noted in this disclosure, may be made to the disclosed
25 embodiment without departing from the spirit and scope of the invention as defined in the appended claims.

- 110 -

CLAIM:

1. A system, comprising:

(a) at least one terminal containing a form reader capable of taking an optical image of a consumer identification form; and

(b) at least one memory device within the at least one terminal for storing templates to assist in the analysis of the optical image to determine consumer information therefrom.

2. The system of claim 1, further comprising a server in communication with the at least one terminal.

3. The system of claim 2, wherein the server provides the templates to the memory device.

4. The system of claim 2, wherein the server receives data from the terminal.

5. The system of claim 4, wherein the data is selected from the group consisting of DEX information, information concerning the contents of the terminal, consumer account information, and consumer credit card information.

6. The system of claim 1, wherein the at least one terminal is a vending machine, and further comprising an enabling circuit for receiving the consumer information to enabling the vending of goods or services from the terminal.

7. The system of claim 1, wherein the consumer information is selected from the groups consisting of the consumer's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

8. The system of claim 1, wherein the at least one terminal is a gas pump, and further comprising an enabling circuit for receiving the consumer information and enabling the vending of gasoline from the terminal accordingly.

9. The system of claim 8, wherein the consumer information comprises information indicative of the validity of the consumer's driver's license.

10. The system of claim 2, further comprising at least one integrated system in communication with the server.

11. The system of claim 10, wherein the integrated system is selected from the group consisting of credit card databases, governmental law enforcement databases, consumer reporting agency databases, and financial services system databases.

- 111 -

12. The system of claim 2, wherein the server is capable of communicating with a plurality of consumer accounts accessible in accordance with the consumer information.

13. The system of claim 2, wherein the system comprises at least two different types of terminals.

5 14. The system of claim 13, wherein the types of terminals are selected from the group consisting of a vending machine, an automatic teller machine, a cash register, and a gas pump.

15. A method for determining information about a consumer prior to enabling the vending of a good or service from a machine, comprising:

(a) receiving a form containing information about the consumer at the machine;

10 (b) optically analyzing the form to electronically determine information about the consumer; and

(c) enabling the vend on the basis of the information.

16. The method of claim 15, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

15 17. The method of claim 15, wherein optically analyzing the form comprises scanning the form to produce an image and comparing the image to image templates.

18. The method of claim 17, wherein the image templates are transmitted to the machine by a system.

19. The method of claim 15, wherein the determined information is selected from the group
20 consisting of the consumer's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

20. The method of claim 19, wherein vending is enabled if the consumer is of a suitable age to purchase the good or service.

21. The method of claim 15, wherein the machine is selected from the group consisting of a
25 vending machine, an automatic teller machine, a cash register, and a gas pump.

22. A machine for distributing goods or services to consumers, comprising an optical scanning unit, wherein the optical scanning unit receives and optically scans a form containing information about the consumer to electronically determine the information about the consumer contained on the form and to enable the vending of goods or services on the basis of
30 the information.

- 112 -

23. The machine of claim 22, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

24. The machine of claim 22, further comprising image templates stored within the optical scanning unit, and wherein determining information about the consumer comprises producing
5 an image of the form and comparing the image to the image templates.

25. The machine of claim 24, wherein the image templates are transmitted to the machine by a system.

26. The machine of claim 22, wherein the determined information is selected from the group consisting of the consumer's age, date of birth, name, address, identification number,
10 driver's license number, social security number, and passport number

27. The machine of claim 26, wherein vending is enabled if the consumer is of a suitable age to purchase the good or service.

28. The machine of claim 22, wherein the machine is selected from the group consisting of a vending machine, an automatic teller machine, a cash register, and a gas pump.

15 29. An optical scanning unit, comprising:

(a) a form reader for optically producing an image of a form containing information about a person; and

(b) stored templates to assist in analyzing the image to electronically determine information about the person.

20 30. The optical scanning unit of claim 29, wherein the optical scanning unit is connectable to a machine, and wherein the information determined about the person is used to enable the vending of goods or services from the machine.

31. The optical scanning unit of claim 30, wherein the machine is selected from the group consisting of a vending machine, an automatic teller machine, a cash register, and a gas pump.

25 32. The optical scanning unit of claim 30, wherein the information is selected from the group consisting of the person's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

33. The optical scanning unit of claim 29, wherein the optical scanning unit is connectable to a system, and wherein the information determined about the person is used by the system to
30 provide further information about the person.

- 113 -

34. The optical scanning unit of claim 33, wherein the further information is selected from the group consisting of credit information, information regarding honoring of checks, driver's license validity, criminal record information, immigration status, or fugitive status.

35. The optical scanning unit of claim 29, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

36. The optical scanning unit of claim 29, wherein the form reader further comprises a magnetic head for reading magnetically encoded information on a form.

37. The optical scanning unit of claim 29, wherein the form reader includes a charge coupled device for optically producing the image.

38. The optical scanning unit of claim 29, wherein the image includes a bar code.

39. An optical scanning unit, comprising:

(a) a form reader for optically producing an image of a form containing security indicia for verifying the validity of the form; and

(b) stored templates to assist in analyzing the security indicia to electronically determine information indicative of the validity of the form.

40. The optical scanning unit of claim 39, wherein the optical scanning unit is connectable to a machine, and wherein the information determined about the person is used to enable the vending of goods or services from the machine.

41. The optical scanning unit of claim 40, wherein the machine is selected from the group consisting of a vending machine, an automatic teller machine, a cash register, and a gas pump.

42. The optical scanning unit of claim 39, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

43. The optical scanning unit of claim 39, wherein the form reader further comprises a magnetic head for reading magnetically encoded information on a form.

44. The optical scanning unit of claim 39, wherein the form reader includes a charge coupled device for optically producing the image.

45. The optical scanning unit of claim 39, wherein the security indicia comprises a hologram.

46. The optical scanning unit of claim 39, wherein the security indicia comprises a bar code.

- 114 -

47. The optical scanning unit of claim 39, wherein the security indicia comprises a validation seal.

48. A method for accessing at least one consumer account using a system, comprising:

(a) receiving a form containing information about a consumer into the system;

(b) optically analyzing the form to electronically determine information about the consumer; and

(c) using the information to electronically access at least one consumer account in communication with the system.

49. The method of claim 48, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

50. The method of claim 48, wherein optically analyzing the form comprises scanning the form to produce an image and comparing the image to image templates.

51. The method of claim 48, wherein the determined information is selected from the group consisting of the consumer's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

52. The method of claim 48, further comprising charging a purchase price of a good or service provided by the system to the accessed account.

53. The method of claim 48, wherein the information is used to access a plurality of consumer accounts, and further comprising allowing the consumer to select one of the plurality of accounts.

54. The method of claim 53, further comprising charging a purchase price of a good or service provided by the system to the selected account.

55. The method of claim 53, wherein at least one of the plurality of account comprises a credit card account.

56. The method of claim 48, further comprising enabling the consumer to enter a private key prior to accessing the at least one consumer account.

57. The method of claim 48, wherein the account resides on an integrated system in communication with the system.

58. A method for allowing a consumer to pay for a good or service having a purchase price at a vending machine using a system, the method comprising:

- 115 -

- (a) receiving at the system consumer account registration information to establish at least one electronic consumer account accessible by the system;
- (b) receiving a form containing information about the consumer into the vending machine;
- 5 (c) optically analyzing the form to electrically determine information about the consumer; and
- (d) using the information to electronically charge the purchase price from the at least one consumer account.

59. The method of claim 58, wherein establishing an electronic consumer account
10 comprises communicating with the system using a computerized user interface.

60. The method of claim 58, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

61. The method of claim 58, wherein optically analyzing the form comprises scanning the form to produce an image and comparing the image to image templates.

15 62. The method of claim 58, wherein the determined information is selected from the group consisting of the consumer's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

63. The method of claim 58, wherein the at least one consumer account comprises a credit card account.

20 64. The method of claim 58, wherein the at least one account resides on an integrated system in communication with the system.

65. The method of claim 58, wherein the at least one account comprises a plurality of accounts, and further comprising allowing the consumer to select one of the plurality of accounts prior to step (d).

25 66. The method of claim 58, further comprising enabling the consumer to enter a private key prior to charging the at least one consumer account.

67. A method, implementable on a system, for making a plurality of electronic consumer accounts accessible by a single consumer identification form, comprising:

- (a) associating each account with information about the consumer;
- 30 (b) enabling the receipt of the form at a terminal in the system;

- 116 -

(c) optically analyzing the form to electrically determine the information about the consumer; and

(d) using the determined information to access the plurality of consumer accounts.

68. The method of claim 67, further comprising registering the plurality of consumer
5 accounts with the system.

69. The method of claim 68, wherein registering the plurality of consumer accounts comprises communicating with the system using a computerized user interface.

70. The method of claim 67, further comprising enabling the consumer to enter a private key prior to accessing the plurality of consumer accounts.

10 71. The method of claim 67, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

72. The method of claim 67, wherein optically analyzing the form comprises scanning the form to produce an image and comparing the image to image templates.

73. The method of claim 67, wherein the determined information is selected from the group
15 consisting of the consumer's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

74. The method of claim 67, wherein the at least one consumer account comprises a credit card account.

75. The method of claim 67, wherein the at least one account resides on an integrated
20 system in communication with the system.

76. The method of claim 67, further comprising allowing the consumer to select one of the plurality of accounts.

77. The method of claim 67, further comprising enabling the consumer to enter a private key to charge at least one consumer account.

25 78. A system for accessing at least one consumer account registered with a system, comprising:

(a) at least one terminal for receiving a form containing information about a consumer and for producing an optical image of the form;

(b) a program for analyzing the optical image and determining consumer
30 information therefrom; and

- 117 -

(c) at least one integrated system in communication with the system which contains at least one consumer account, wherein the at least one consumer account is accessible using the determined consumer information.

79. The system of claim 78, further comprising a user interface to allow the at least one consumer account to be preregistered with the system.

80. The system of claim 78, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

81. The system of claim 78, wherein the program compares the image to image templates.

82. The system of claim 78, wherein the determined information is selected from the group consisting of the consumer's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

83. The system of claim 78, wherein the at least one consumer account comprises a credit card account.

84. The system of claim 78, further comprising a server disposed between and in communication with the at least one terminal and the at least one integrated system.

85. The system of claim 78, wherein the system comprises at least two different types of terminals.

86. The system of claim 85, wherein the types of terminals are selected from the group consisting of a vending machine, an automatic teller machine, a cash register, and a gas pump.

87. A method for determining information about an individual using a form, comprising:

(a) receiving the form at a first system;

(b) optically analyzing the form to determine first information about the individual;

(c) transmitting the first information to a second system containing second information about the individual;

(d) using the first information to access the second information; and

(e) receiving the second information.

88. The method of claim 87, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

89. The method of claim 87, wherein optically analyzing the form comprises scanning the form to produce an image and comparing the image to image templates.

- 118 -

90. The method of claim 87, wherein the first information is selected from the group consisting of the individual's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

91. The method of claim 87, wherein the second information is selected from the group consisting of credit information, information regarding honoring of checks, driver's license validity, criminal record information, immigration status, and fugitive status.

92. The method of claim 87, wherein the second system is an integrated system in communication with the first system.

93. The method of claim 87, wherein the first system includes a terminal for receiving the form.

94. A method for verifying the identity of a person using a terminal, comprising:

- (a) receiving optical image data from a first form at the terminal;
- (b) analyzing the optical image data to determine first information about the person;
- (c) receiving magnetic data from a magnetic strip on a second form at the terminal;
- (d) analyzing the magnetic data to determine second information about the person;
- and
- (e) comparing the first information and the second information to verify the identity of the person.

95. The method of claim 94, wherein the first form or second form is selected from the group consisting of an identification card, a driver's license, a credit card, a social security card, and a passport.

96. The method of claim 94, wherein the first information or the second information is selected from the group consisting of the person's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

97. The method of claim 94, further comprising electronically enabling a purchase at the terminal if a match occurs between the first information and the second information when compared.

98. The method of claim 94, wherein the optical image data comprises a bar code.

99. A method for verifying the identity of a person using a terminal and a form, the form including a magnetic strip, comprising:

- 119 -

- (a) receiving both optical image data from the form and magnetic data from the magnetic strip on the form at the terminal;
- (b) analyzing the optical image data to determine first information about the person;
- (c) analyzing the magnetic data to determine second information about the person;
- 5 and
- (d) comparing the first information and the second information to verify the identity of the person.

100. The method of claim 99, wherein the form is selected from the group consisting of an identification card, a driver's license, a credit card, a social security card, and a passport.

101. The method of claim 99, wherein the first information or the second information is selected from the group consisting of the person's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

102. The method of claim 99, further comprising electronically enabling a purchase at the terminal if a match occurs between the first information and the second information when compared.

103. The method of claim 99, wherein the optical image data comprises a bar code.

104. A system, comprising:

- (a) at least one terminal containing a form reader capable of taking an optical image of a form containing personal information to determine indicia indicative of the identity of the person;
- (b) a server in communication with the at least one terminal for receiving the indicia; and
- (c) an integrated system in communication with the server for receiving the indicia from the server and providing in return information concerning the person.

105. The system of claim 104, wherein the at least one terminal is a vending machine, and wherein the returned information is used to enable vending from the vending machine.

106. The system of claim 104, wherein the at least one terminal is a gas pump, and wherein the returned information includes information concerning the validity of the person's drivers license.

107. The system of claim 106, wherein the information concerning the validity of the person's driver's license is used to enable the vending of gasoline from the gas pumps.

- 120 -

108. The system of claim 104, wherein the integrated system comprises a database having immigration data, and wherein the returned information includes information concerning the person's immigration status.

109. The system of claim 104, wherein the integrated system comprises a database
5 containing information about the person's credit, and wherein the returned information includes information concerning the person's credit status.

110. The system of claim 104, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

111. The system of claim 104, wherein the indicia is selected from the group consisting of
10 the person's age, date of birth, name, address, identification number, driver's license number, social security number, and passport number.

112. The system of claim 104, wherein the integrated system is selected from the group consisting of credit card databases, governmental law enforcement databases, consumer reporting agency databases, and financial services system databases.

113. The system of claim 104, further comprising image templates to be used in determining
15 the indicia.

114. A device for receiving a form, comprising:

- (a) a magnetic head for reading magnetically encoded information on the form; and
- (b) an optical receiver for receiving an image of the form.

115. The device of claim 114, wherein the device is connectable to a machine for vending of
20 goods or services.

116. The device of claim 114, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

117. The device of claim 114, wherein the device includes a charge coupled device for
25 receiving the image.

118. The device of claim 114, wherein the image is selected from the group consisting of a hologram, a validation seal, and a bar code.

119. The device of claim 114, further comprising a memory for storing image template used in analyzing the image of the form.

120. A method for optically analyzing a test image in a system containing memory,
30 comprising:

- 121 -

- (a) storing a test image $D(i,j)$ in a first memory;
- (b) storing K templates $T_k(i,j)$, each representative of a character, in a second memory;
- (c) adjusting the contrast of the K templates to match the contrast of the test image $D(i,j)$;
- (d) electronically positioning the test image relative to each template to calculate a minimum least squares difference between the test image and each template;
- (e) storing the minimum least squares difference for each template in a third memory; and
- (f) selecting the template with the smallest minimum least squares difference to determine the test image.

121. The method of claim 120, wherein $T_k(i,j)$ is equal to either a logical '1' or a logical '0'.

122. A method for optically analyzing a test image in a system containing memory, comprising:

- (a) storing a test image $D(i,j)$ in a first memory;
- (b) storing K templates $T_k(i,j)$, each representative of a character, in a second memory, each template having respective vertical and horizontal dimensions of m_k and n_k ;
- (c) electronically positioning the test image relative to each template by offsets r and s to calculate a minimum least squares difference $dist_k(r,s)$ between the test image and each template in accordance with the following equation:

$$dist_k(r,s) = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} (D(r+i, s+j) - [\alpha T_k(i,j) + \beta])^2 ;$$

where α and β are dependent upon both $D(i,j)$ and $T_k(i,j)$;

- (d) storing the minimum least squares difference for each template in a third memory; and
- (e) selecting the template with the smallest minimum least squares difference to determine the test image.

123. The method of claim 122, wherein α and β are calculated in accordance with the following equations:

- 122 -

$$\alpha = \frac{m_k n_k A - BC}{\Delta}$$

$$\beta = \frac{\Omega C - AB}{\Delta}$$

where

$$A = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} D(r+i, s+j) T_k(i, j)$$

$$B = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} T_k(i, j)$$

$$C = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} D(r+i, s+j)$$

$$\Omega = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} T_k^2(i, j)$$

$$\Pi = \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} D^2(r+i, s+j)$$

$$\Delta = m_k n_k \Omega - B^2$$

124. The method of claim 122, wherein $T_k(i, j)$ is equal to either a logical '1' or a logical '0'.

5 125. The method of claim 122, wherein only P elements of the templates are used when calculating the minimum least squares difference in accordance with the following equation:

$$dist_k(r, s | \alpha, \beta) = \sum_{p=1}^P \left(D(r+i_p, s+j_p) - [\alpha T_k(i_p, j_p) + \beta] \right)^2.$$

126. A method for generating a template $T(i, j)$ representative of a character in a system containing memory, comprising:

- 10 (a) scanning N of examples of the character to produce a plurality of example images $A_N(i, j)$ representative of the character;
- (b) storing the example images in a first memory;
- (c) determining an offset (r_k, s_k) for each example image to bring the images into alignment with each other;
- 15 (d) calculating the template in accordance with the following equation:

$$T(i, j) = \frac{1}{N} \sum_{k=1}^N A_k(r_k + i, s_k + j); \text{ and}$$

- (e) storing the template in a second memory.

- 123 -

127. The method of claim 126, wherein determining the offset for each template involves computation of the minimum distance in accordance with the following equation:

$$dist_k(r,s) = \sum_{k=1}^N \sum_{i=1}^{m_k} \sum_{j=1}^{n_k} (T(i,j) - A_k(r_k + i, s_k + j))^2.$$

128. The method of claim 126, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

129. A method, implementable on a system containing data, for analyzing a form containing a form header and at least one cluster which contains at least one element, comprising:

- (a) scanning the form to create a computerized optical image of the form;
- (b) electronically determining the form type by comparing the optical image of the form header with form header template data associated with the form type;
- (c) using offset data associated with the form type to determine the location of a cluster on the form; and
- (d) assessing at least one element within the cluster by comparing the element to character template data associated with the form type.

130. The method of claim 129, wherein step (b) further comprising using form header origin data associated with the form type when determining the form type.

131. The method of claim 129, further comprising, after step (b), determining a form header origin for the optical image.

132. The method of claim 131, wherein the form header origin is used in association with the offset data to determine the location of the cluster in step (c).

133. The method claim 129, wherein step (d) includes using a pattern specification associated with the form type, and wherein the pattern specification is indicative of the structure of the elements and references the character template data.

134. The method of claim 129, wherein the cluster includes a cluster header, and wherein determining the location of the cluster comprises comparing the optical image of the cluster header with cluster header template data associated with the form type.

135. The method of claim 129, wherein the character template data is selected from the group consisting of alphabetical template data and numerical template data.

136. The method of claim 129, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

- 124 -

137. A method, implementable on a system containing data, of analyzing a form to create an optical template file for the form, the form containing a form header, at least one cluster, and at least one element within the cluster, comprising:

- (a) optically scanning a form to create an image file;
- 5 (b) storing a portion of the image file containing the form header as form header template data associated with the optical template file;
- (c) determining, from the image file, the origin of the form header, and storing the form header origin data in association with the optical template file;
- (d) determining, from the image file, the cluster origin for at least one cluster, and
10 storing the cluster origin data in association with the optical template file; and
- (e) determining, from the image file, the element origin of at least one element within the cluster and storing the element origin data in association with the optical template file.

138. The method of claim 137, wherein steps (a) through (e) are repeated for several forms,
15 and wherein the data stored with the optical template file represents the average of the data for each of the forms.

139. The method of claim 137, further comprising storing a pattern specification in association with the optical template file, wherein the pattern specification is indicative of the structure of the elements within the cluster.

20 140. The method of claim 137, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

141. The method of claim 137, further comprising storing the portion of the image file containing a cluster header as cluster header template data associated with the optical template file.

25 142. A method for optically analyzing a sequence of symbols using a system containing memory, wherein the set of symbols comprises a plurality of different symbol types, comprising:

- (a) storing a pattern specification in a first memory, wherein the pattern specification is comprised of a sequence of pattern characters, wherein each
30 pattern character corresponds to a symbol type in the sequence of symbols, and wherein each pattern character references data stored in a second memory;

- 125 -

- (b) optically scanning the sequence of symbols to form an image comprising images of each symbol type, and storing the image in a third memory; and
- (c) analyzing the image by assessing the image of each symbol type with respect to the data referenced by the corresponding pattern character of the symbol type.

5 143. The method of claim 142, wherein the sequence of pattern characters is indicative of the arrangement of the sequence of symbols.

144. The method of claim 142, wherein each pattern character sequentially corresponds to a symbol type.

145. The method of claim 142, wherein the referenced data comprises template images.

10 146. The method of claim 145, wherein analyzing the image further comprises comparing the template images to the image of each symbol type.

147. The method of claim 142, wherein at least one pattern character specifies the expected number of symbols in its corresponding symbol type.

148. The method of claim 142, wherein the symbols are characters.

15 149. The method of claim 142, wherein the symbol types are selected from the group consisting of numbers, upper case letter, lower case letters, and punctuation symbols.

150. A method for optically analyzing a continuous sequence of symbols using a system containing memory containing a plurality of optical templates, comprising:

- 20 (a) optically scanning the sequence of symbols to form an image comprised of a plurality of images of each symbols;
- (b) determining a first subset of the plurality of templates; and
- (c) comparing a first subset of the plurality of templates to at least one symbol image to identify that symbol.

25 151. The method of claim 150, further comprising comparing a second subset of the plurality of templates to a different symbol image to identify that symbol.

152. The method of claim 150, wherein determining a first subset of the plurality of templates includes the use of a pattern character.

153. The method of claim 152, wherein the pattern character is contained within a pattern specification.

30 154. The method of claim 152, wherein the pattern character specifies the expected number of symbols in its corresponding symbol type.

- 126 -

155. The method of claim 152, wherein the pattern character specifies a certain symbol type and wherein the first subset of the plurality of templates correspond to templates for that symbol type.

156. The method of claim 150, wherein the symbols are characters.

5 157. The method of claim 150, wherein the symbol are selected from the group consisting of numbers, upper case letter, lower case letters, and punctuation symbols.

158. A method for allowing a user to initialize a machine to be connected to a network, comprising in order:

(a) accessing the network;

10 (b) inputting configuration data for the machine at the network;

(c) connecting the machine to the network, whereby the machine automatically establishes a communication channel with the network; and

(d) transmitting the configuration data to the machine through the communication channel.

15 159. The method of claim 158, wherein accessing the network comprises use of a user interface in communication with the network.

160. The method of claim 158, wherein the configuration data enables image templates to be sent to the machine from the network.

161. The method of claim 158, further comprising:

20 (a) receiving at the network information concerning the status of the machine via the communication channel; and

(b) comparing at the network the received status information with the inputted configuration data, wherein the transmitted configuration data is dependent on the received status information.

25 162. The method of claim 161, wherein received status information represents software for controlling the functionality of the machine, and wherein the transmitted configuration data adds to, updates, or deletes at least a portion of the software.

163. The method of claim 158, wherein the configuration data enables audit data to be sent from the machine to the network.

30 164. The method of claim 158, wherein the machine contains an optical scanning unit for receiving a form.

- 127 -

165. The method of claim 164, wherein the form is selected from the group consisting of a identification card, a driver's license, a social security card, and a passport.

166. The method of claim 164, wherein the transmitted configuration data is stored in the optical scanning unit.

5 167. The method of claim 158, wherein the machine is selected from the group consisting of a vending machine, an automatic teller machine, cash register, and a gas pump.

168. A method for configuring the functionality of a machine containing an optical scanning unit connected to a network, comprising:

(a) accessing the network using a user interface;

10 (b) selecting configuration options for the machine using a graphical user interface;
and

(c) transmitting the configuration options to the optical scanning unit in the machine.

169. The method of claim 168, wherein accessing the network comprises use of a user
15 interface in communication with the network.

170. The method of claim 168, wherein the configuration options enable image templates to be sent to the machine from the network.

171. The method of claim 168, wherein the configuration options enable audit data to be sent from the machine to the network.

20 172. The method of claim 168, wherein the machine contains an optical scanning unit for receiving a form.

173. The method of claim 172, wherein the form is selected from the group consisting of an identification card, a driver's license, a social security card, and a passport.

25 174. The method of claim 172, wherein the optical scanning unit includes memory for storing the transmitted configuration options.

175. The method of claim 168, wherein the machine is selected from the group consisting of a vending machine, an automatic teller machine, a cash register, and a gas pump.

30 176. A method for determining a potential match between a target image and one or more of a plurality of template images, wherein the target image and the plurality of template images comprise a plurality of pixels, the method comprising in no particular order:

- 128 -

- (a) creating a first electronic file from at least a portion of the target image, the first file being indicative of the contrast between different pixels in the target image;
- (b) creating a plurality of second electronic files from at least a portion of the plurality of template images, each of the second files being indicative of the contrast between different pixels in the corresponding template image; and
- (c) electrically comparing the first file with each of the plurality of second files to determine a plurality of match errors between the first file and the plurality of second files, and storing the match errors in memory.

177. The method of claim 176, further comprising, prior to step (a), scanning a form to create the target image.

178. The method of claim 176, further comprising, after step (c), comparing the match errors to a threshold value, wherein the match errors below the threshold value determine which of the plurality of template images potentially match the target image.

179. The method of claim 178, wherein the template image with the lowest match error is determined to be the template image matching the target image.

180. The method of claim 178, wherein the plurality of template images potentially matching the target image are further processed to determine which of the potential template images matches the target image.

181. The method of claim 176, further comprising, prior to step (a), compressing the target image.

182. The method of claim 181, wherein compressing the target image comprises performing bilinear reduction.

183. The method of claim 181, further comprising, prior to step (b), compressing the plurality of template images.

184. The method of claim 183, wherein compressing the plurality of template images comprises performing bilinear reduction.

185. The method of claim 176, wherein determining the plurality of match errors comprises exclusively ORing pixels of the first file with corresponding pixels of the plurality of second files.

186. The method of claim 185, wherein determining the plurality of match errors further comprises summing the results of the exclusive OR step.

- 129 -

187. The method of claim 176, wherein creating either the first file or the plurality of second files comprises assigning to at least one pixel in either the first file or the plurality of second files a value indicative of the relative contrast between that pixel and a plurality of neighboring pixels.

5 188. The method of claim 176, further comprising, prior to steps (a) and (b), determining the edges of a target element in the target image.

189. The method of claim 188, further comprising scaling either the target element or the plurality of template images so that they are equal in size.

10 190. The method of claim 188, wherein the target element comprises a character, symbol, graphic, or an optically variable device.

191. The method of claim 176, further comprising, prior to step (a), filtering the target image.

15 192. The method of claim 191, wherein filtering the target image comprises performing a histogram analysis, a SUSAN algorithm, a high pass filter, a convolution filter, or a color reduction technique on the pixels of the target image.

193. A method of creating a data structure for an electronic image having pixels, where each pixel represents an intensity level, the method comprising:

- (a) providing a target pixel within the image, wherein the target pixel has a coordinate;
- 20 (b) calculating at least one value indicative of the relationship of the intensity level between the target pixel and at least one neighboring pixel;
- (c) electronically storing the at least one value in the data structure at the same coordinate as the target pixel; and
- (d) choosing another target pixel within the image and repeating steps (a) through
- 25 (c) for the another target pixel.

194. The method of claim 193, wherein the image is compressed prior to step (a).

195. The method of claim 194, wherein the image is compressed by bilinear reduction.

196. The method of claim 193, wherein the value represents at least one logical '0' or '1.'

30 197. The method of claim 196, wherein a logical '1' represents that the at least one neighboring pixel has a higher intensity value than the target pixel.

- 130 -

198. The method of claim 193, wherein step (b) comprises calculating a plurality of values indicative of the relationship of the intensity level between the target pixel and a plurality of neighboring pixels.

199. The method of claim 198, wherein the values represent a plurality of binary numbers.

5 200. The method of claim 193, wherein the image is scanned prior to step (a).

201. The method of claim 193, wherein the image comprises a character, a symbol, a graphic, or an optically variable device.

202. The method of claim 193, further comprising, prior to step (a), filtering the image.

10 203. The method of claim 202, wherein filtering the image comprises performing a histogram analysis, a SUSAN algorithm, a high pass filter, a convolution filter, or a color reduction technique on the pixels of the target image.

204. A method for scanning a form containing an optically variable element, comprising:

(a) illuminating incident first radiation onto the form;

15 (b) detecting reflected first radiation from the form to create a first image, wherein the incident first radiation and the reflected first radiation define a first angle;

(c) illuminating incident second radiation onto the form;

(d) detecting reflected second radiation from the form to create a second image, wherein the incident second radiation and the reflected second radiation define a second angle different from the first angle; and

20 (e) comparing the first and second images to create a third image of the optically variable element.

205. The method of claim 204, wherein the first radiation and second radiation are of a common wavelength.

25 206. The method of claim 204, wherein steps (a) and (c) comprises using a common radiation source.

207. The method of claim 204, wherein steps (b) and (d) comprise using a common radiation detector.

208. The method of claim 204, wherein comparing the first and second images comprises subtracting the first and second images.

30 209. The method of claim 204, wherein steps (b) and (d) comprise using charge coupled devices.

- 131 -

210. The method of claim 204, wherein steps (a) or (c) comprise using LEDs or lamps.

211. The unit of claim 204, wherein the first and second angles are acute.

212. The unit of claim 204, wherein the optically variable element comprises a hologram or an uneven surface on the form.

5 213. The method of claim 204, wherein steps (a) and (c) comprise illuminating on opposite sides of the form, and wherein the optically variable element comprises a substantially transparent element.

214. A method for scanning a form containing an optically variable element, comprising:

(a) illuminating incident first radiation onto the form;

10 (b) detecting reflected first radiation from the form to create a first image;

(c) illuminating incident second radiation onto the form, wherein the second radiation comprises a wavelength different from the first radiation; and

(d) detecting reflected second radiation from the form to create a second image,

15 wherein the optically variable element is substantially imaged only in the second image.

215. The method of claim 214, wherein the first radiation comprises visible radiation and the second radiation comprises either ultra-violet or infra-red radiation.

216. The method of claim 214, wherein steps (b) and (d) comprise using a common radiation detector.

20 217. The method of claim 214, wherein steps (b) and (d) comprise using charge coupled devices.

218. The method of claim 214, wherein steps (a) or (c) comprise using LEDs or lamps.

219. The method of claim 214, wherein the optically variable element comprises ultra-violet or infra-red sensitive ink.

25

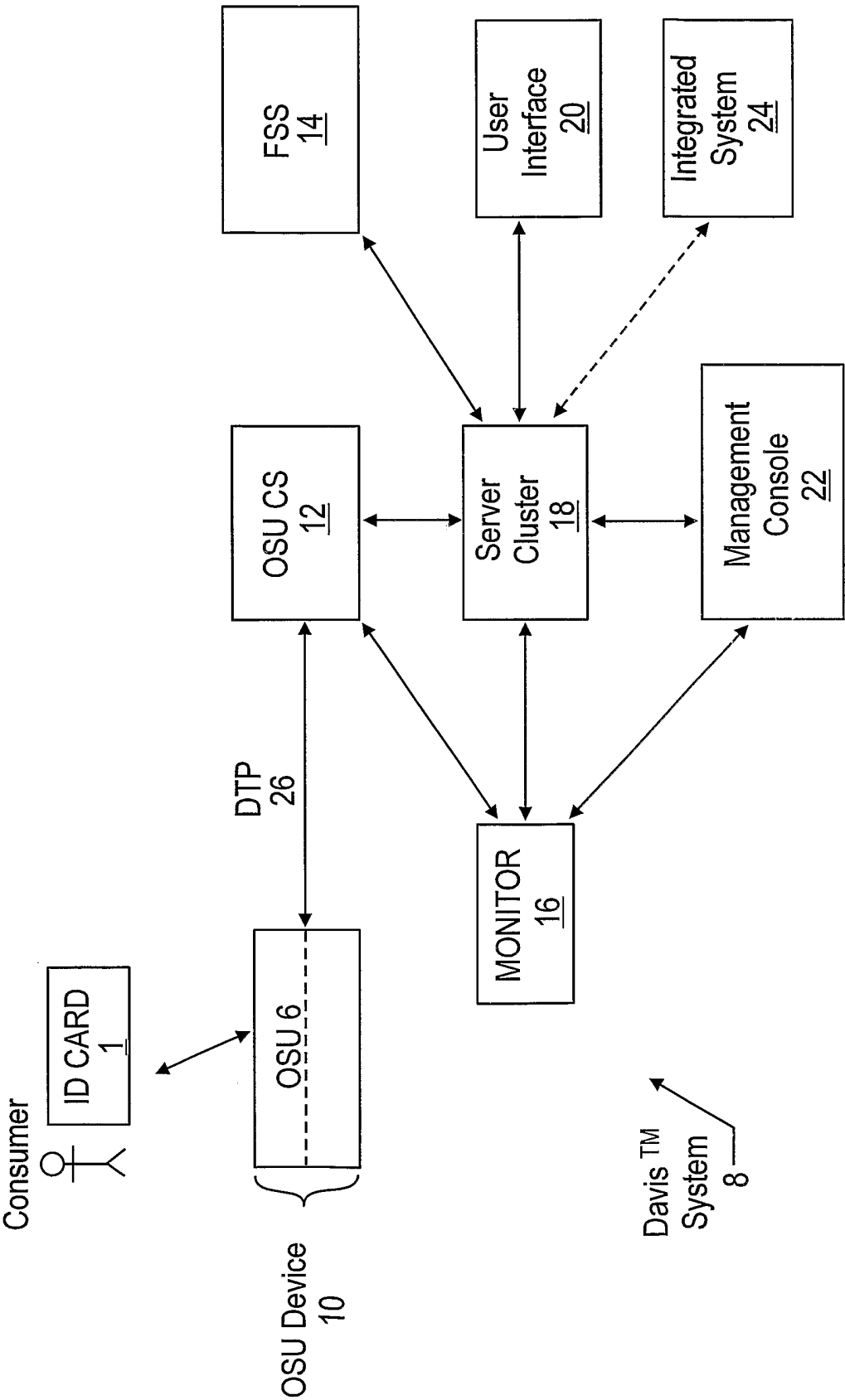
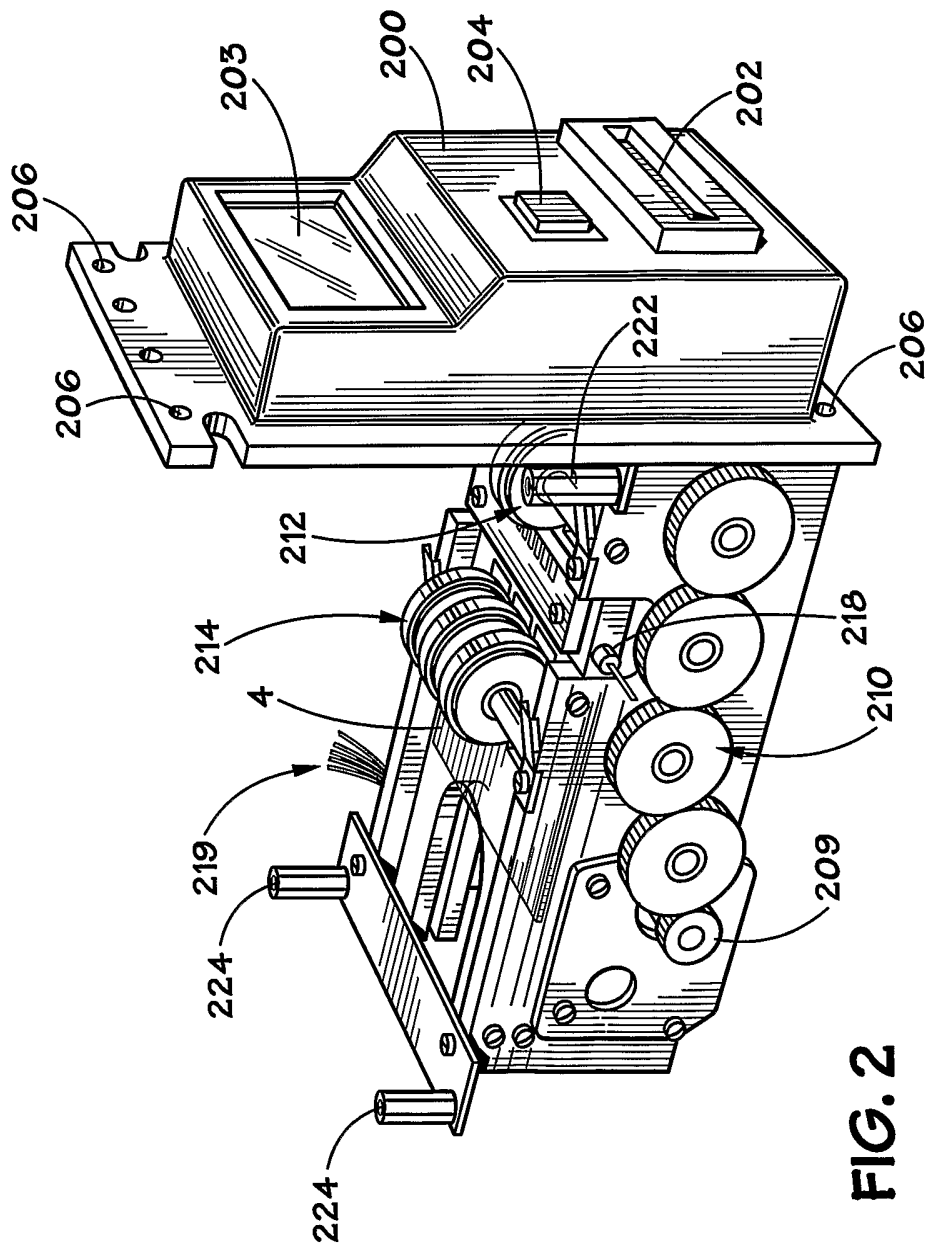


FIG. 1



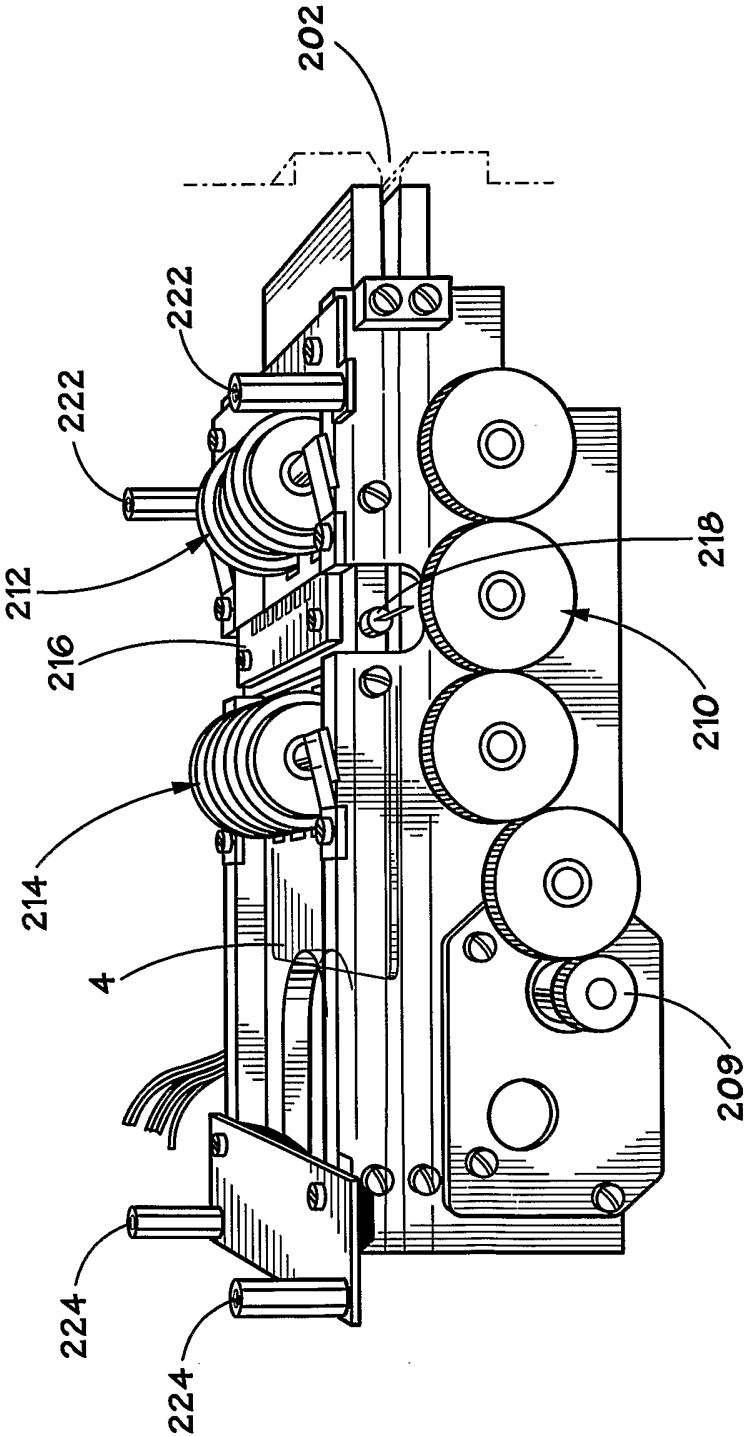
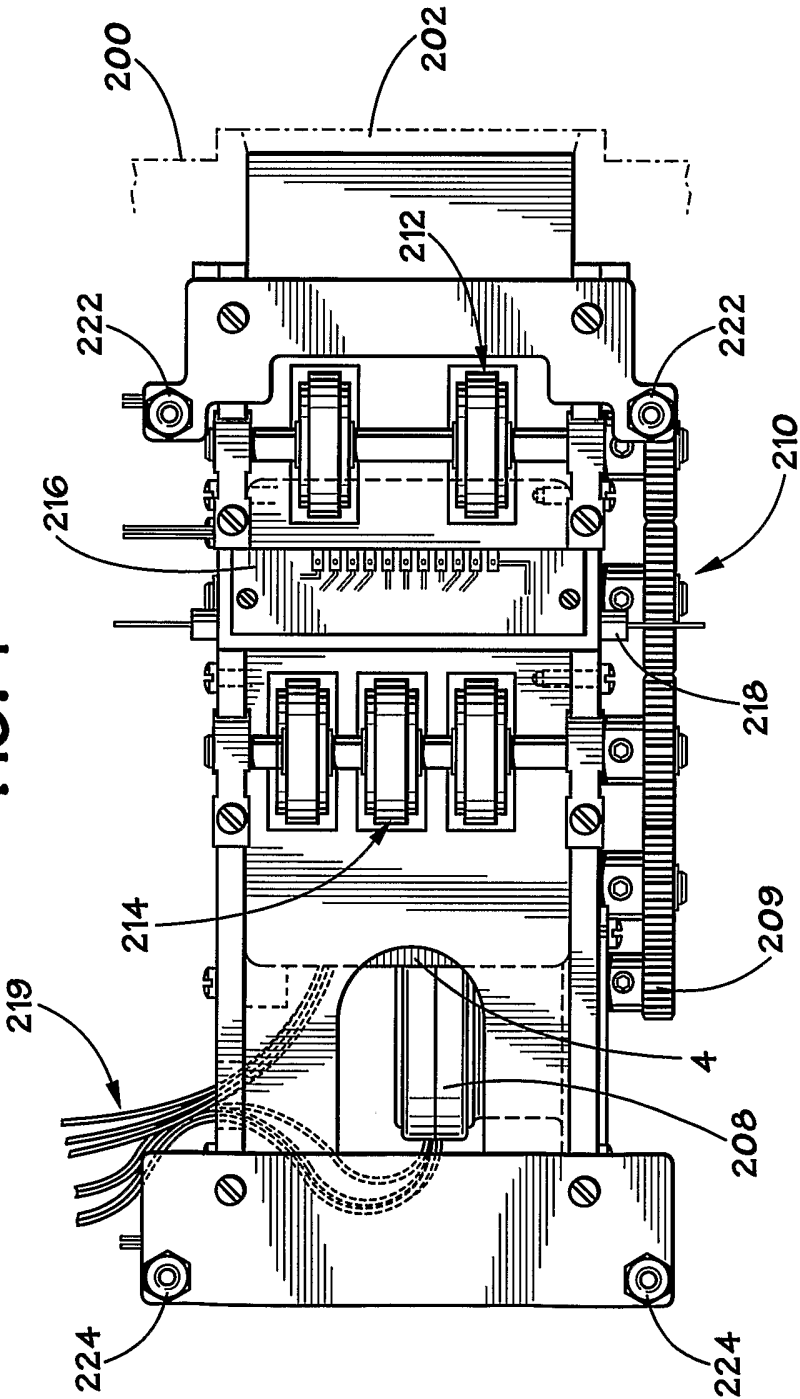


FIG. 3

FIG. 4



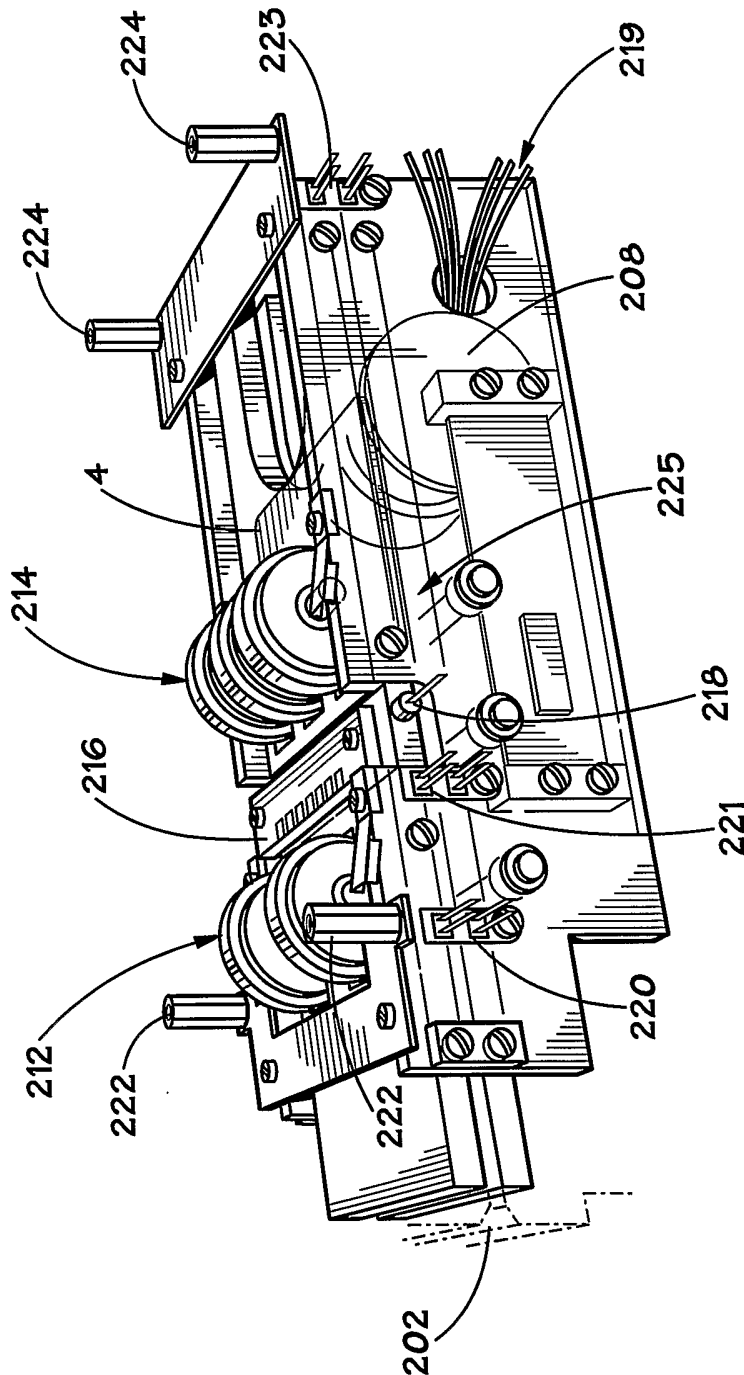


FIG. 5

6/28

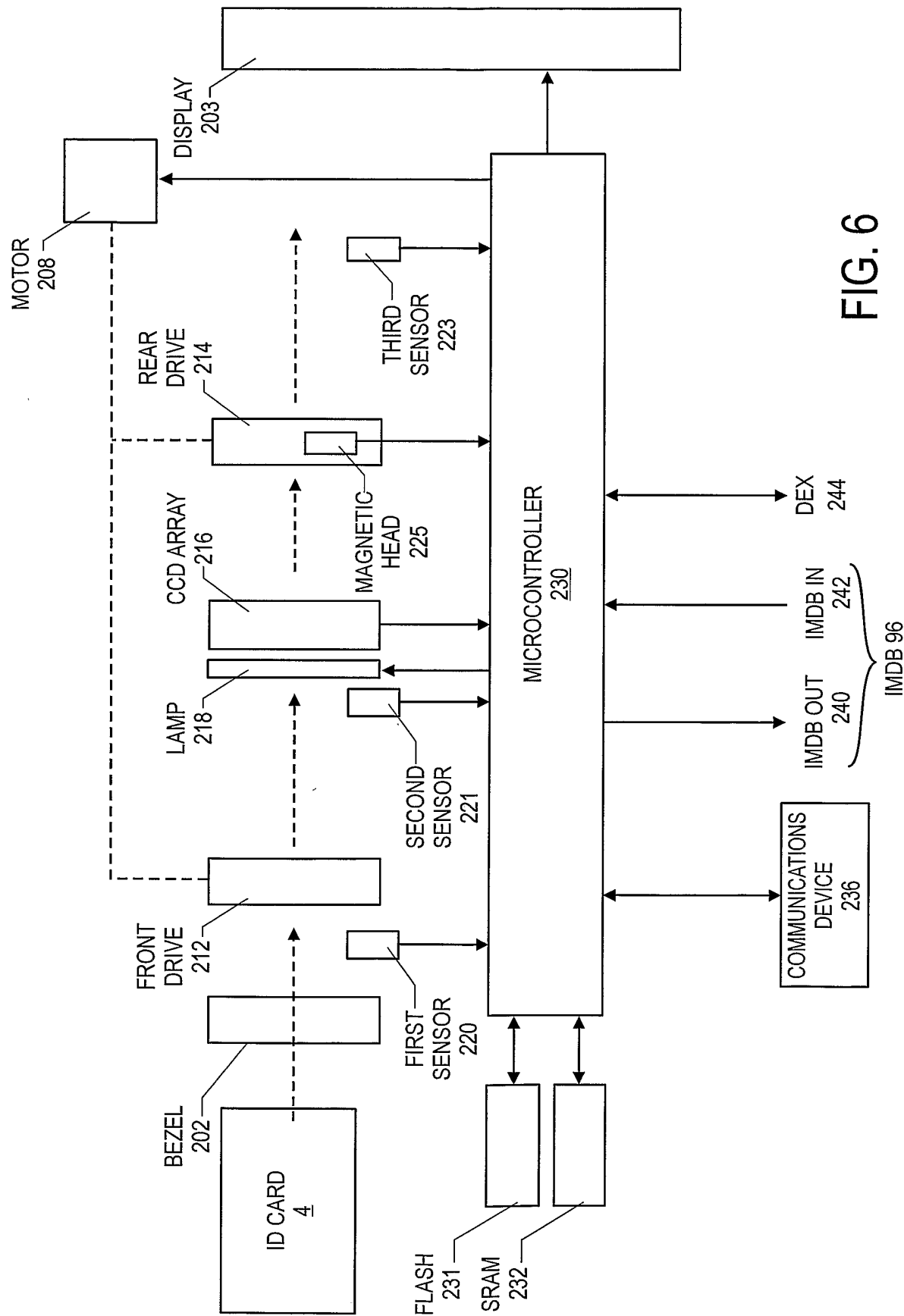


FIG. 6

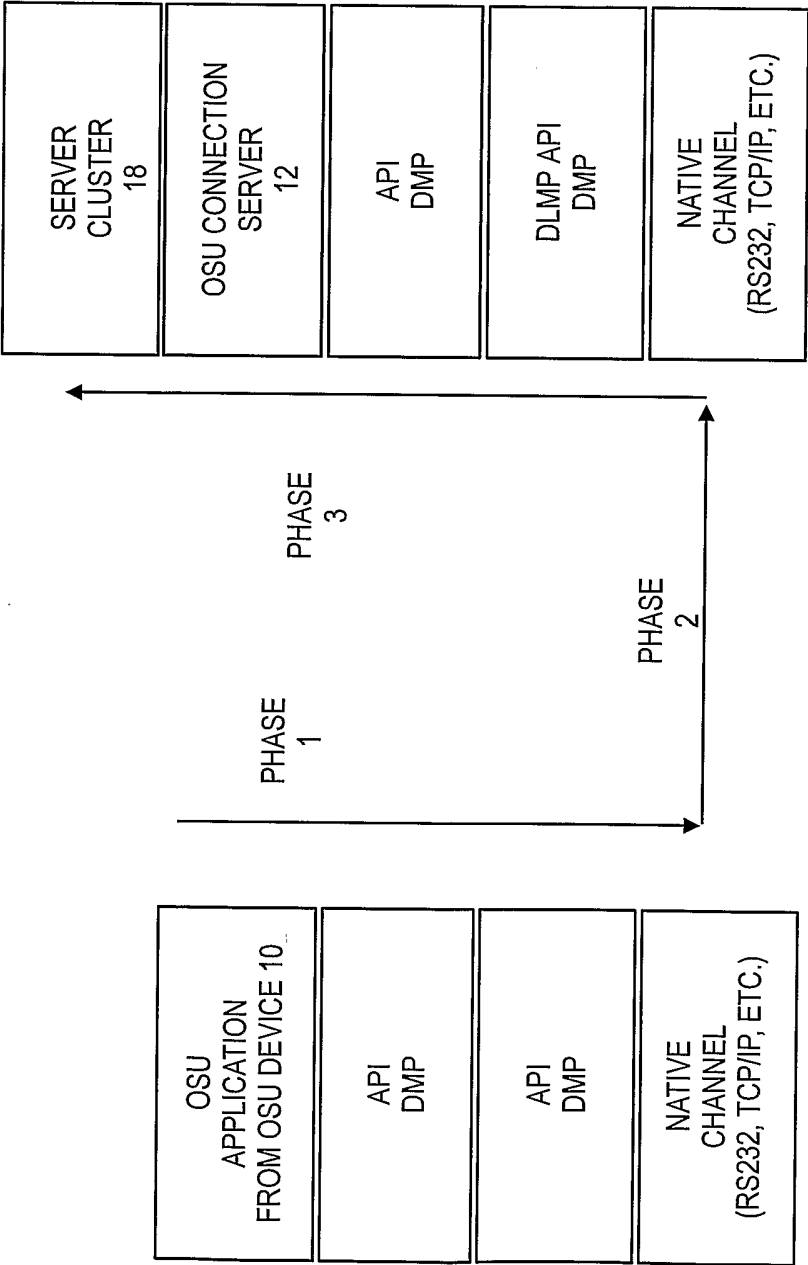


FIG. 7

8/28

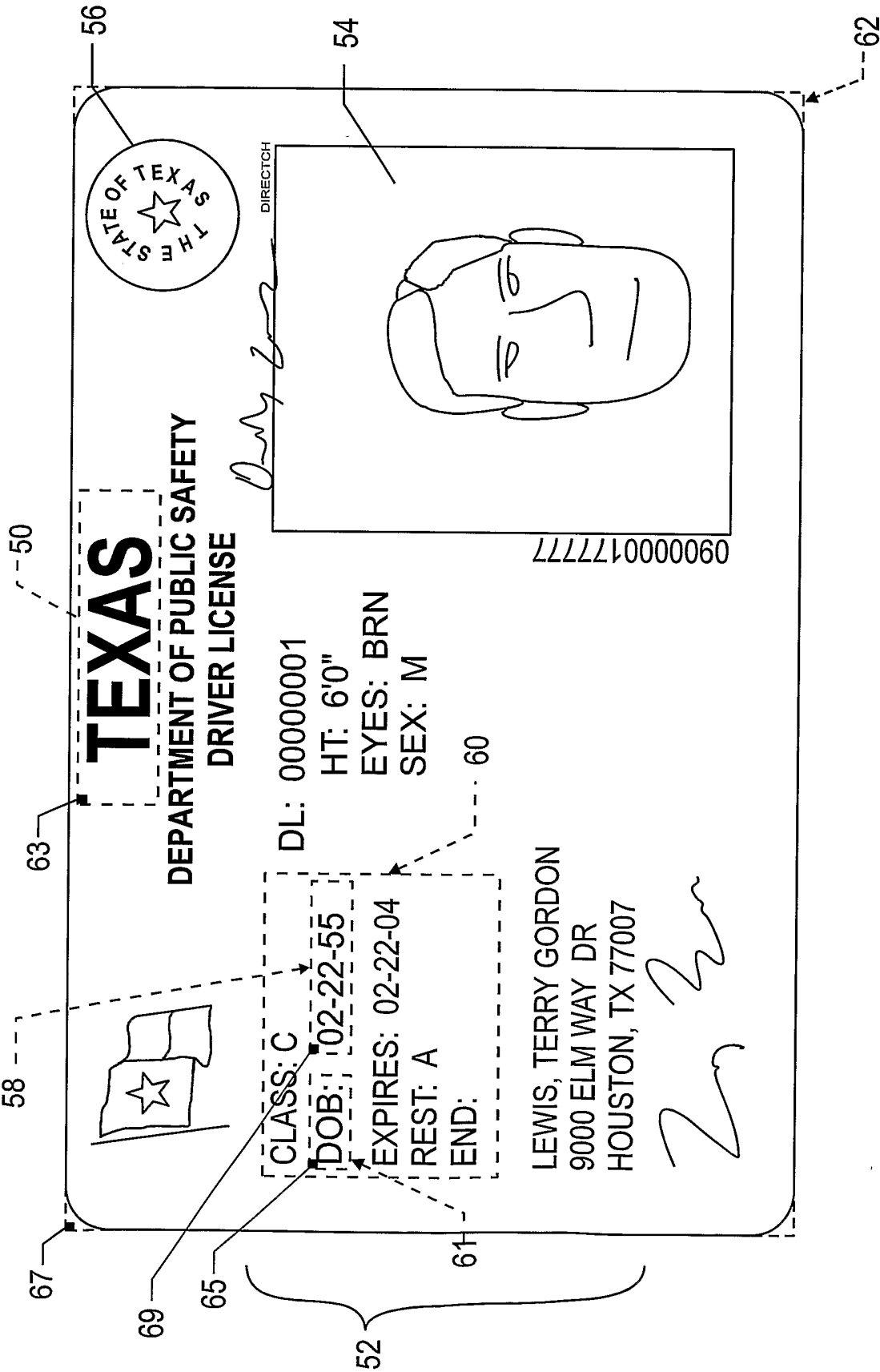


FIG. 8

9/28

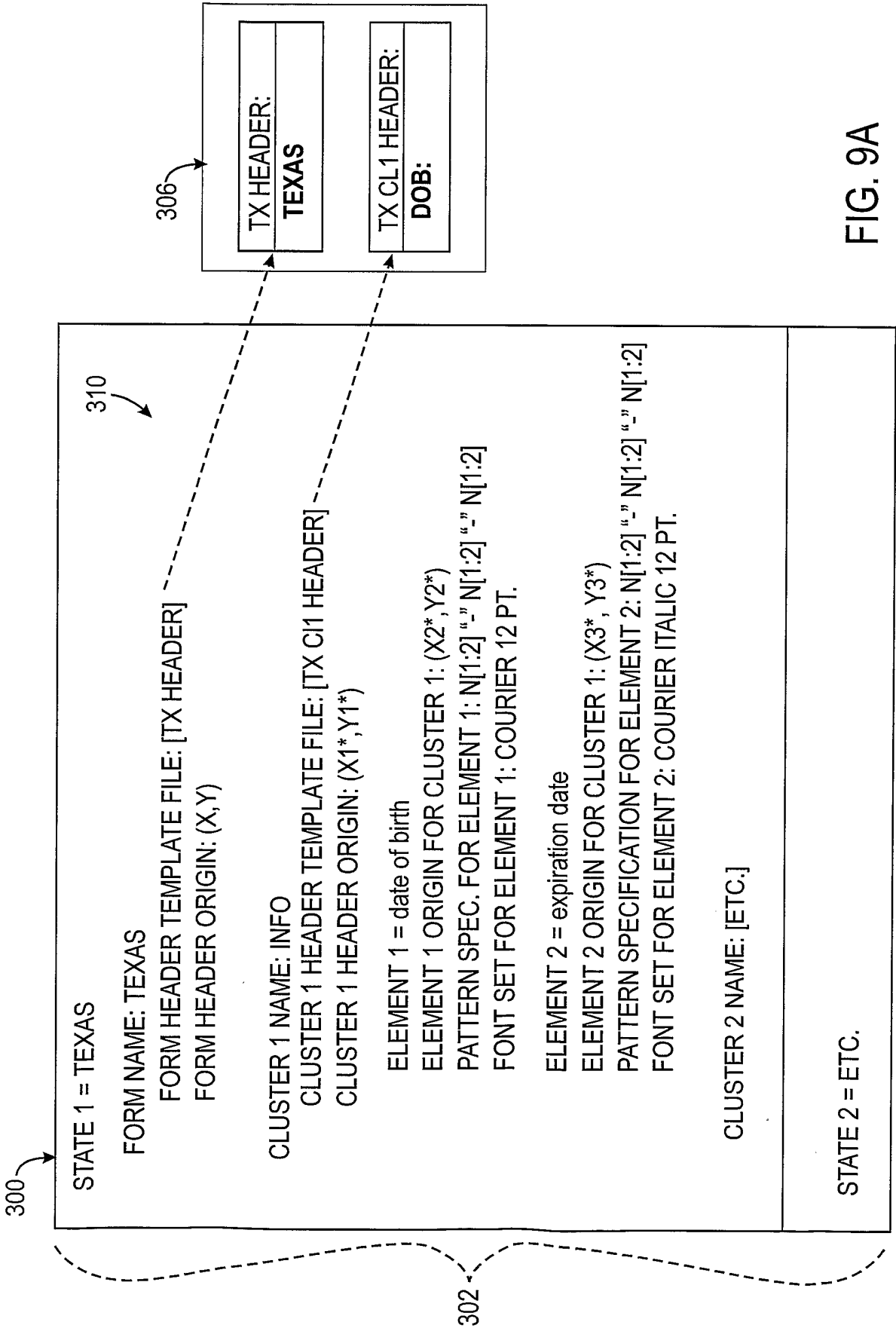


FIG. 9A

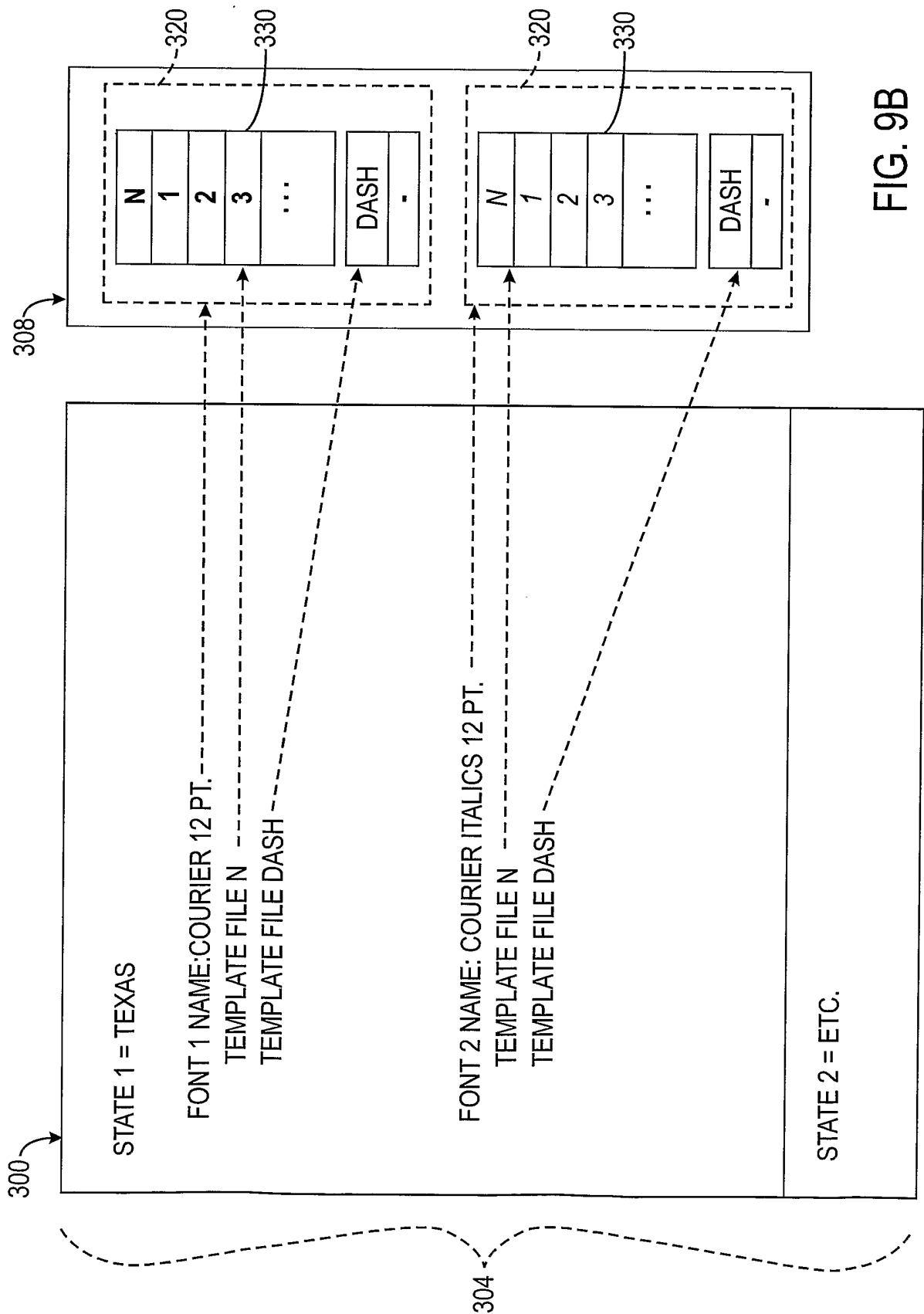


FIG. 9B

11/28

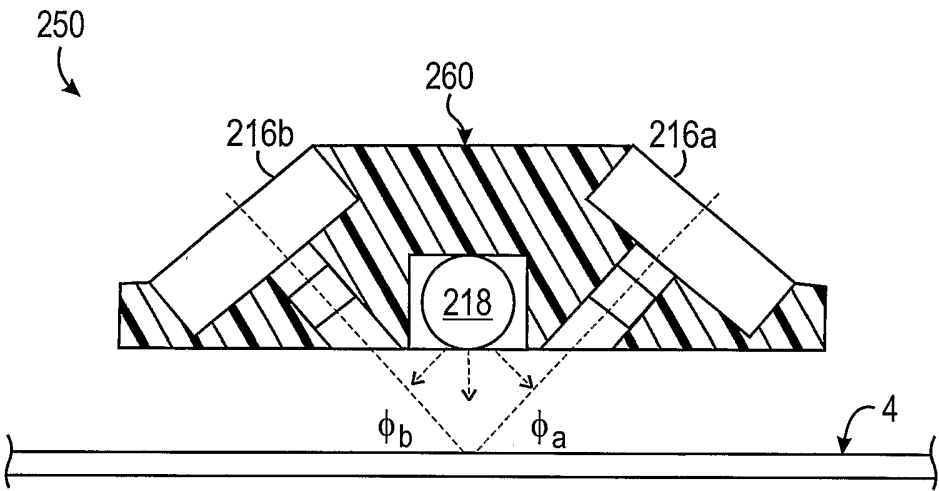


FIG. 10

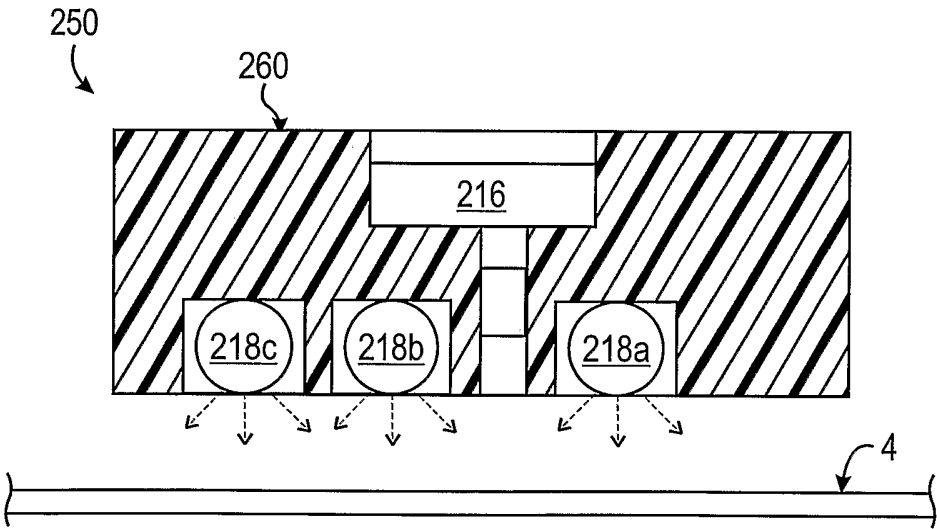


FIG. 11

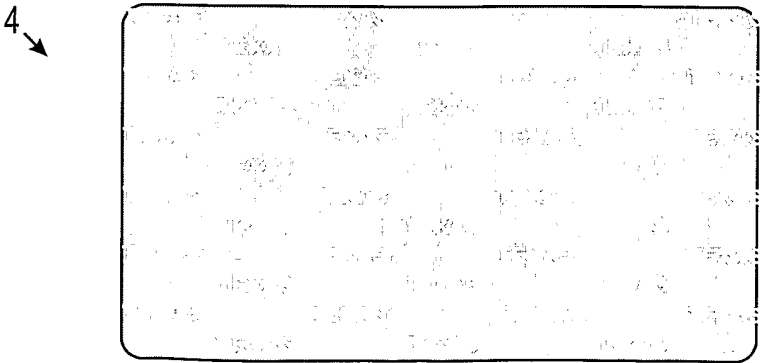


FIG.12

12/28

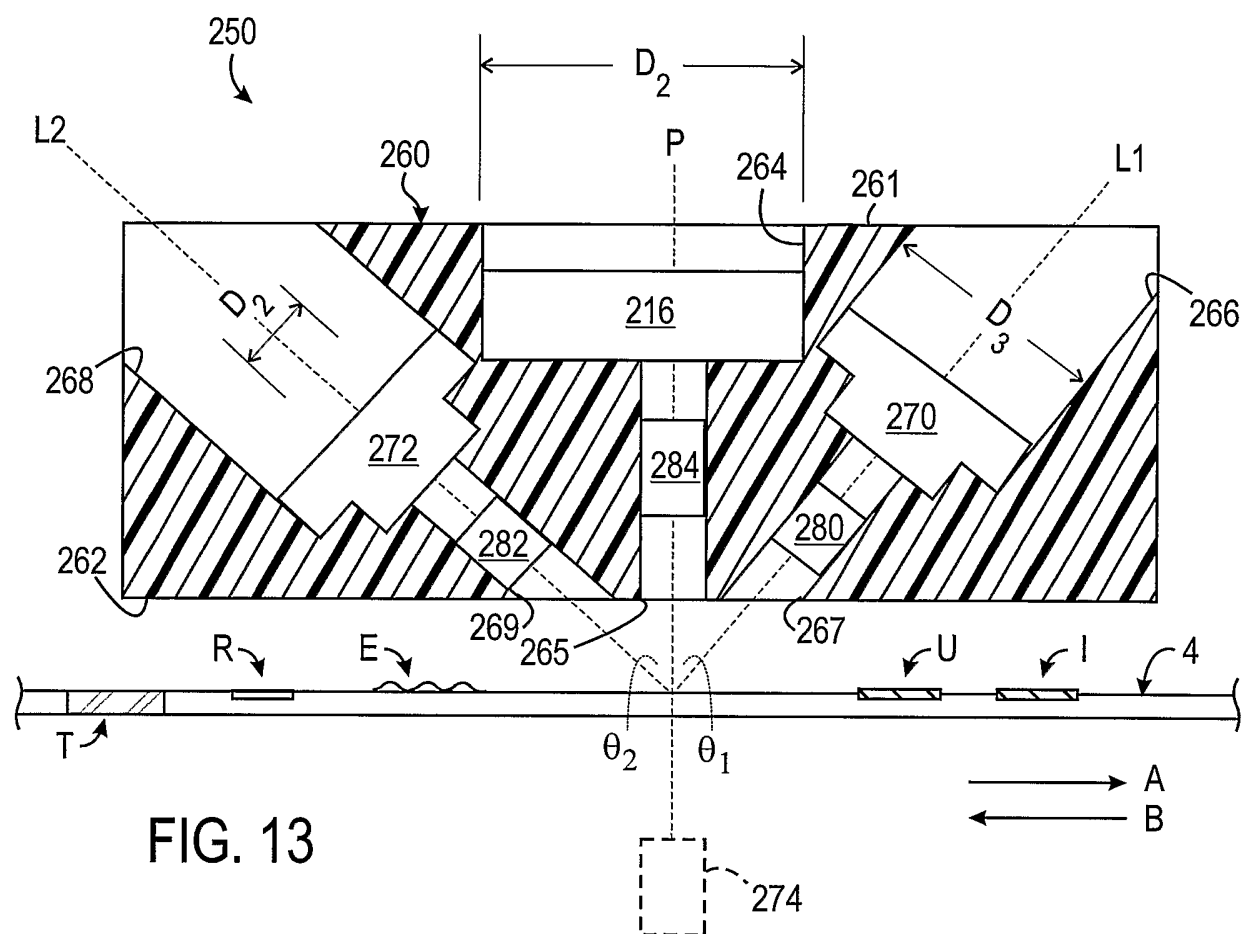


FIG. 13

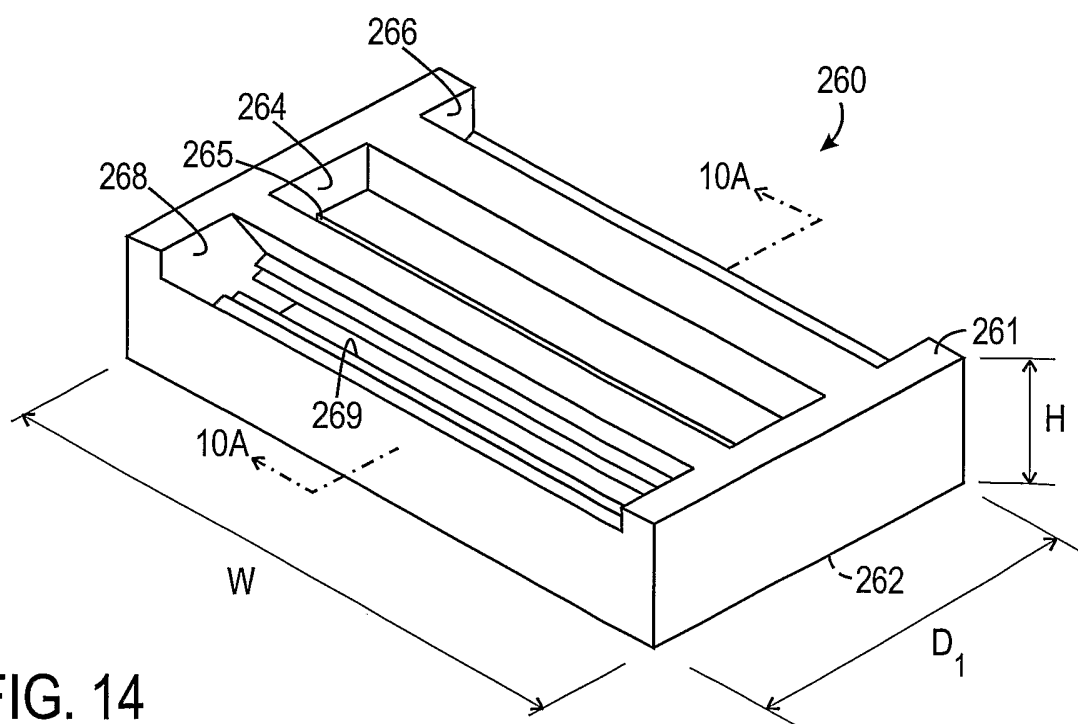


FIG. 14

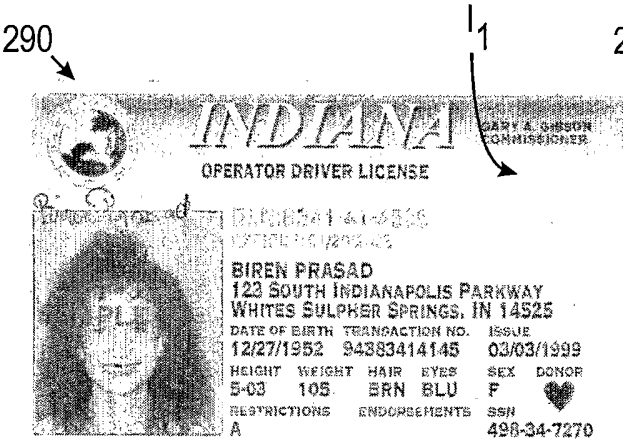


FIG.15A

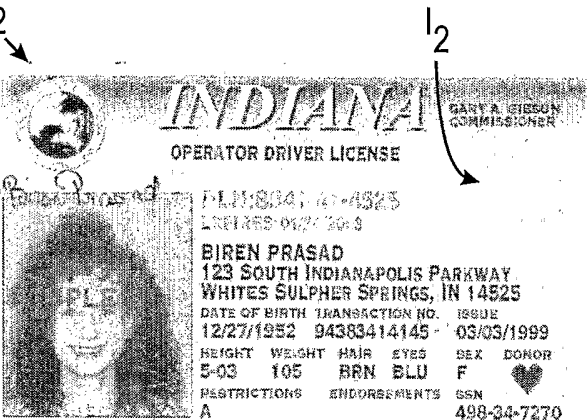


FIG.15B

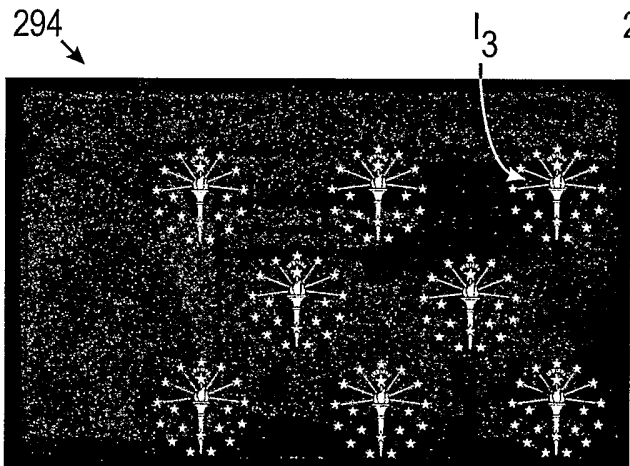


FIG.16A

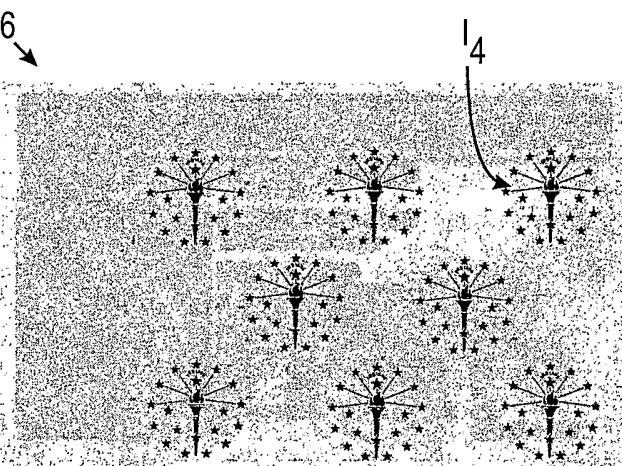


FIG.16B

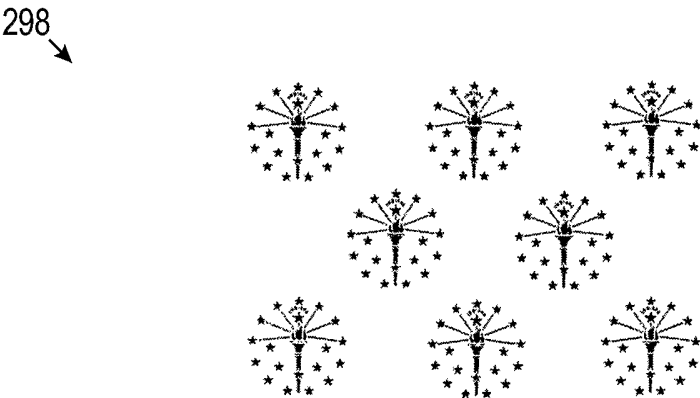
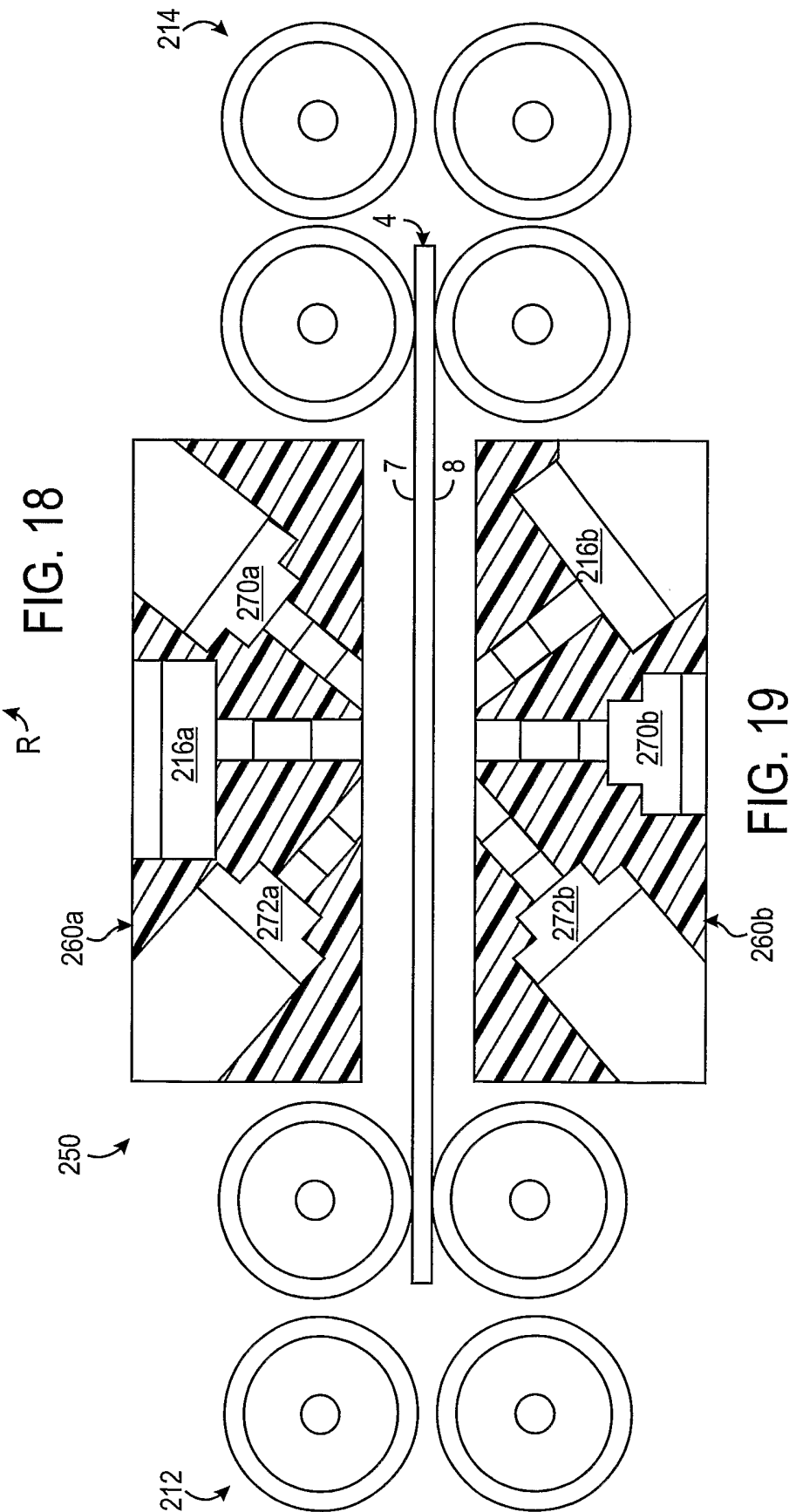
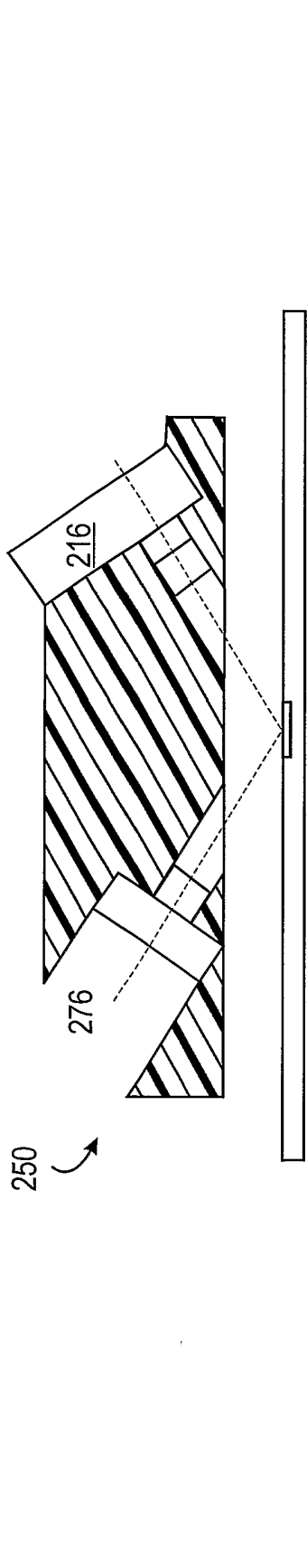


FIG.17



15/28

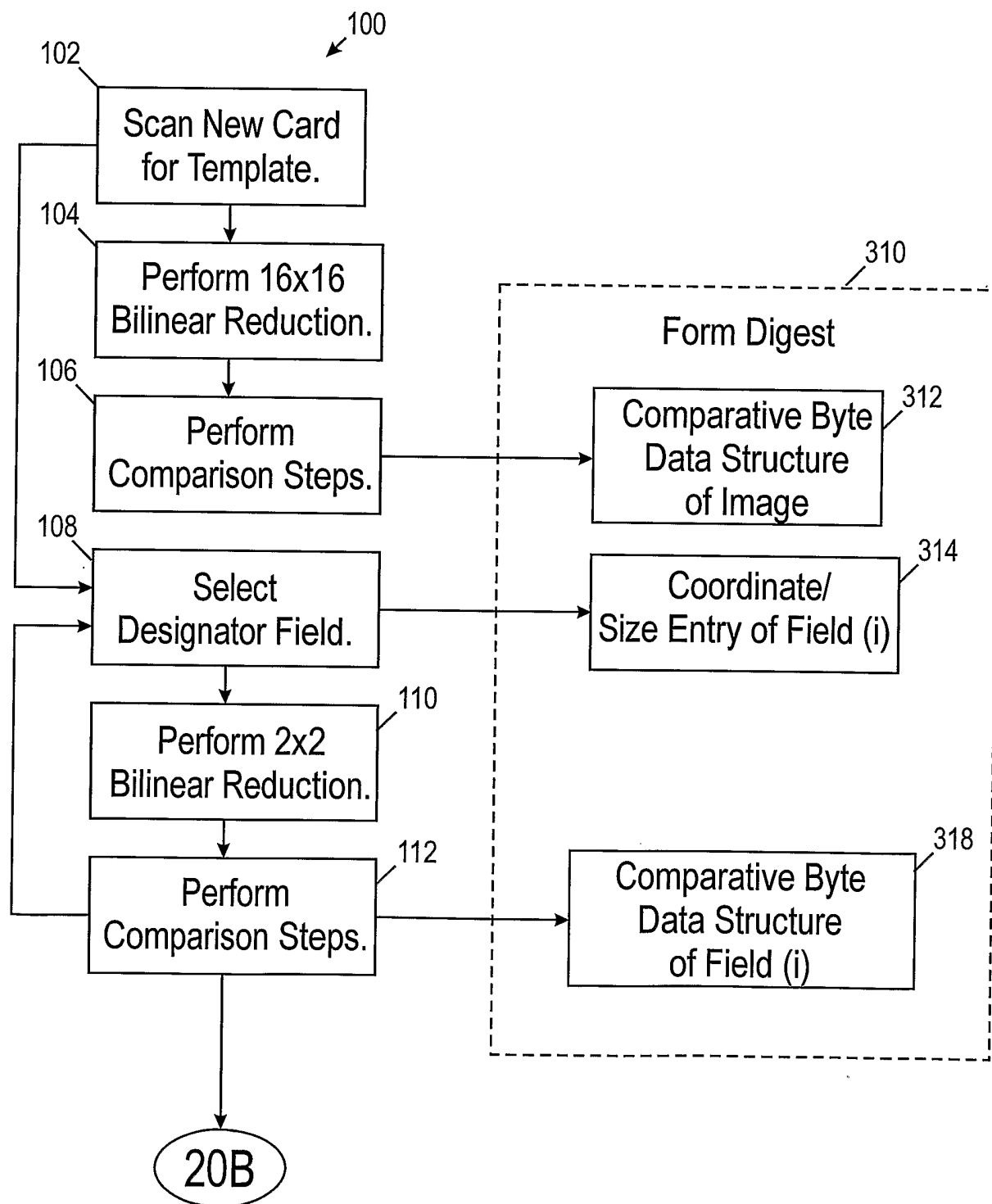


FIG.20A

16/28

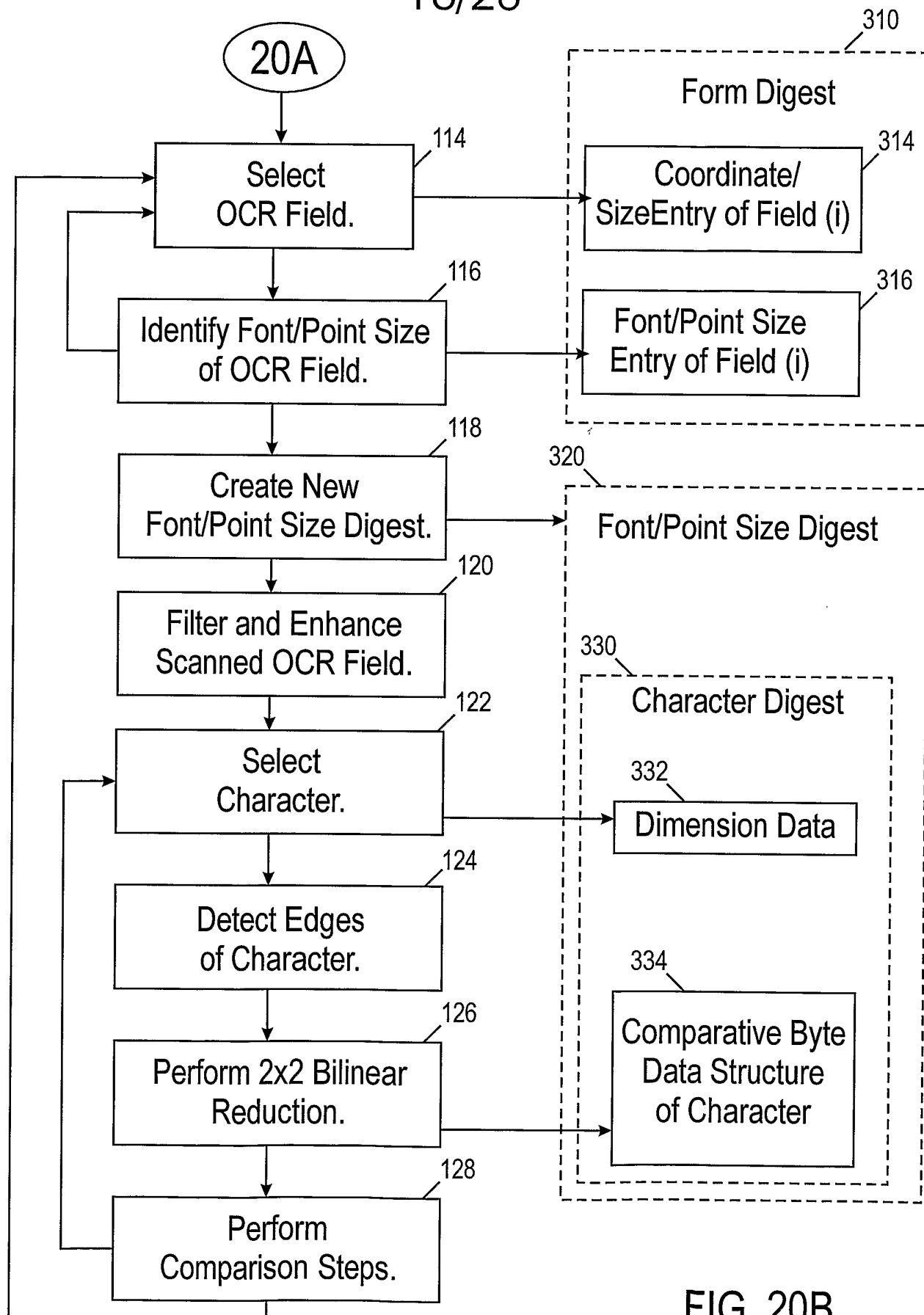


FIG. 20B

17/28

130 ↘

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0
4	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0
5	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0
6	0	0	255	255	255	255	255	255	255	255	255	255	255	255	0	0
7	0	0	255	255	255	255	255	0	0	255	255	255	255	255	0	0
8	0	0	255	255	255	255	0	0	0	0	255	255	255	255	0	0
9	0	0	255	255	255	255	0	0	0	0	255	255	255	255	0	0
10	0	0	255	255	255	255	255	0	0	255	255	255	255	255	0	0
11	0	0	255	255	255	255	255	255	255	255	255	255	255	255	0	0
12	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0
13	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0
14	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIG.21

132 ↘

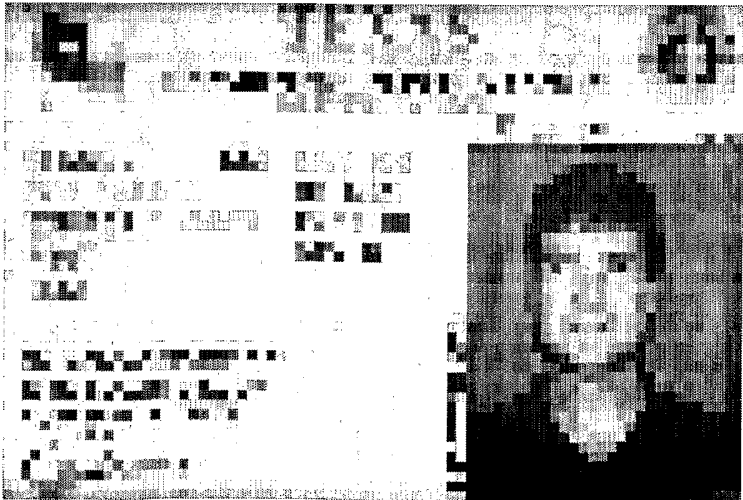


FIG.22

18/28

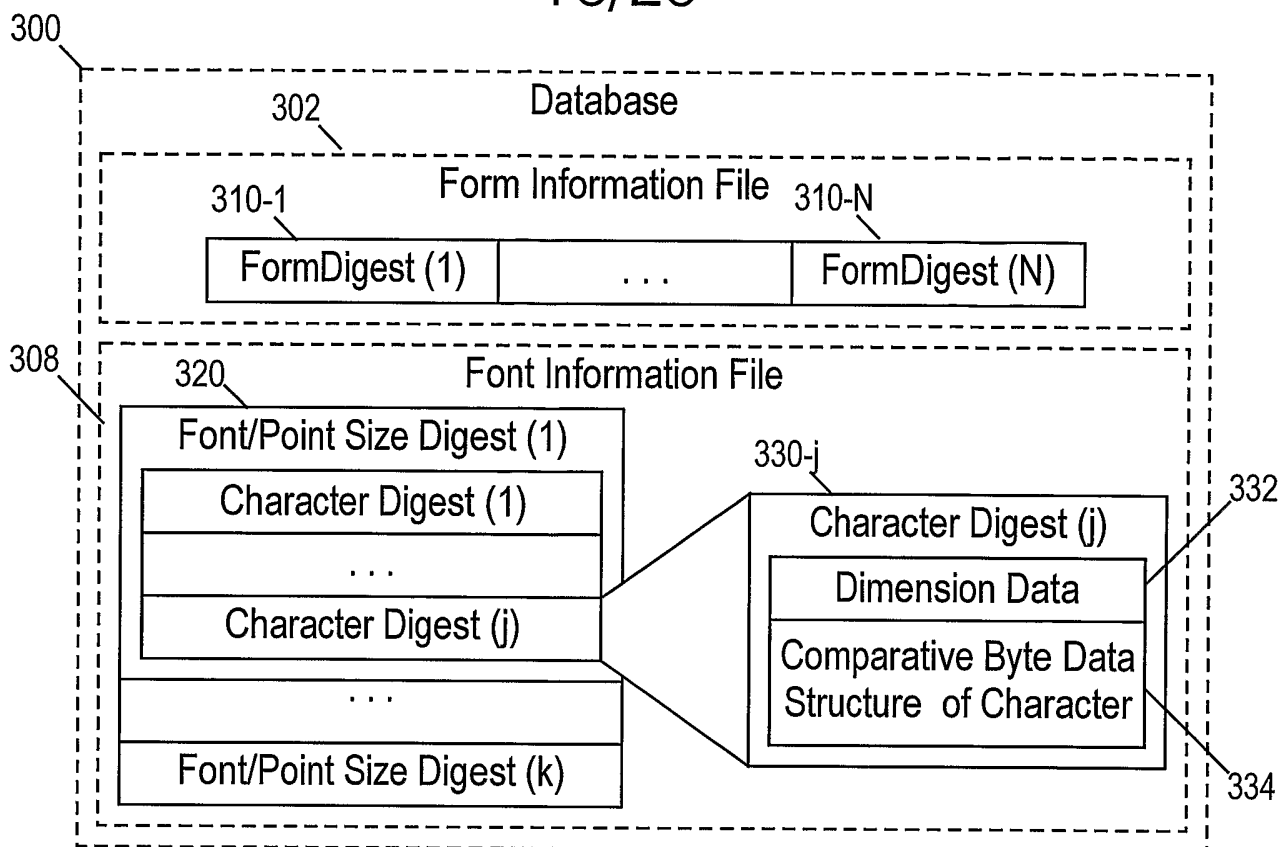


FIG. 23A

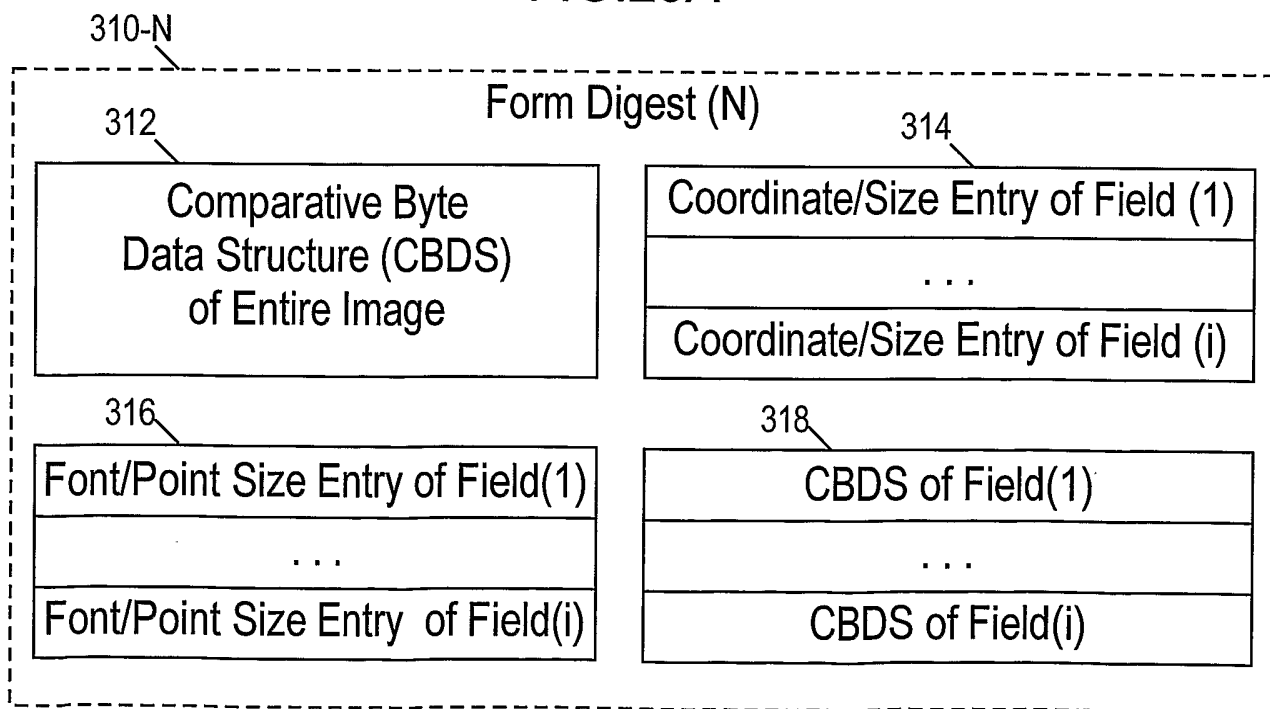


FIG. 23B

19/28

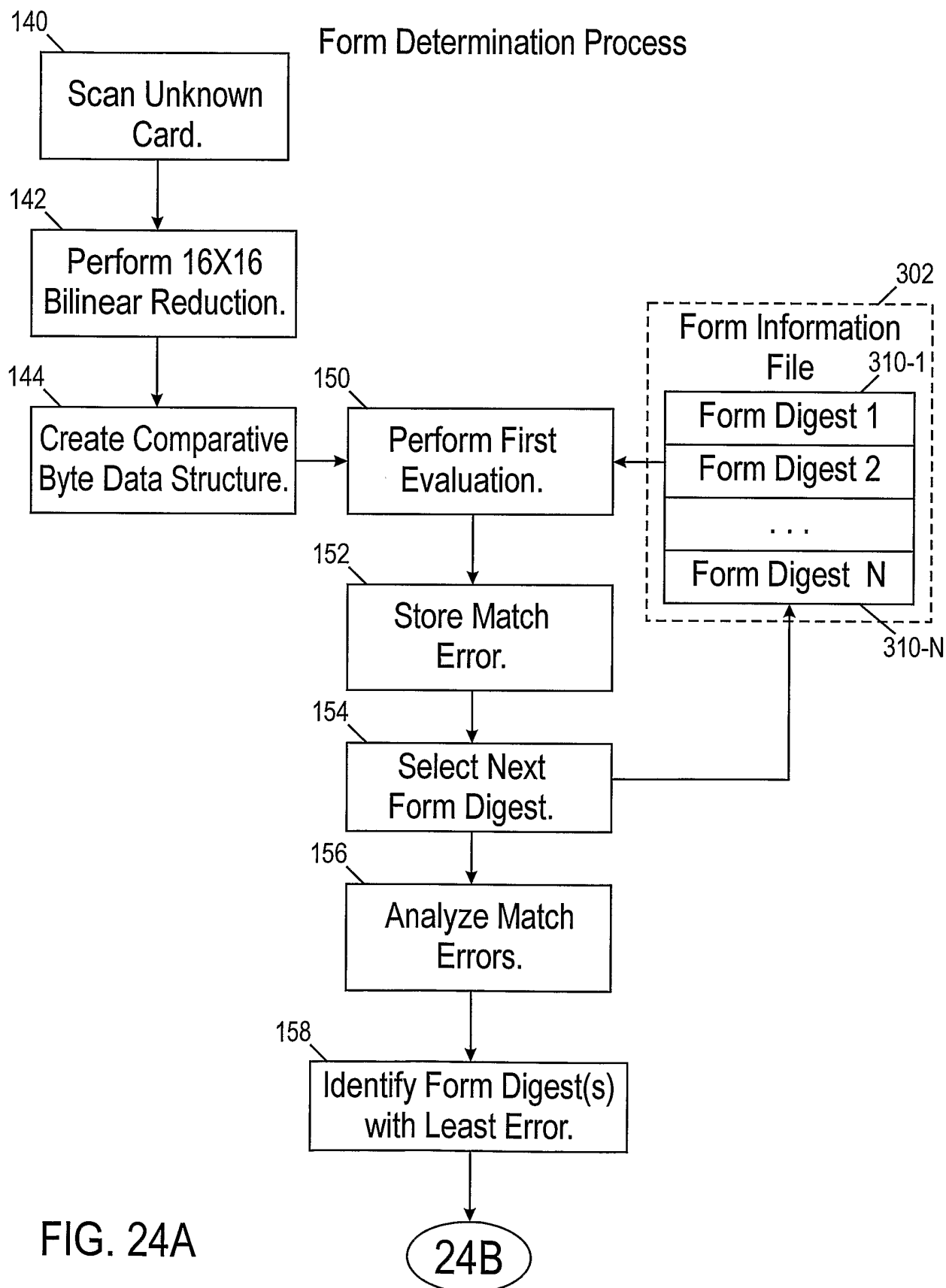


FIG. 24A

20/28

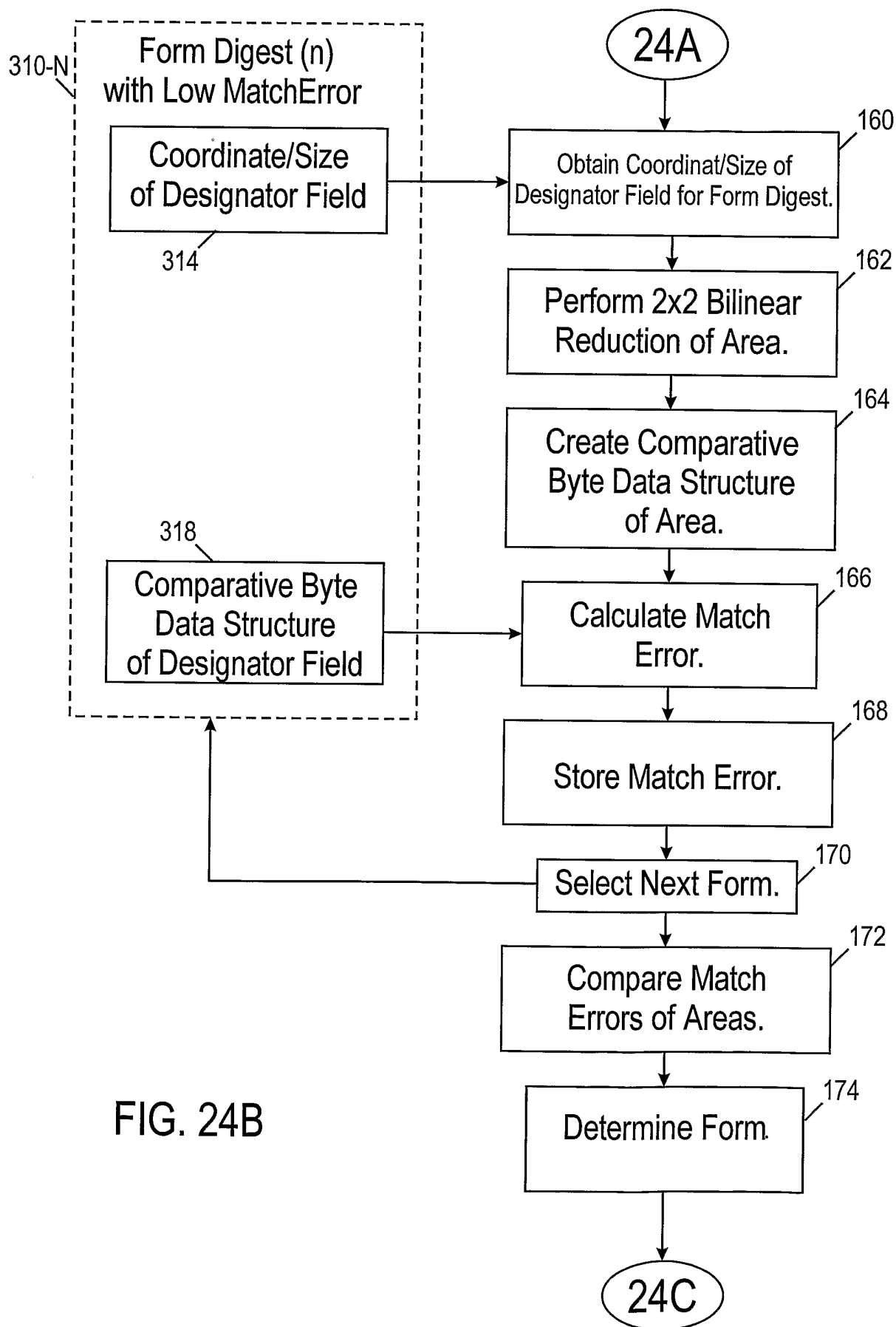
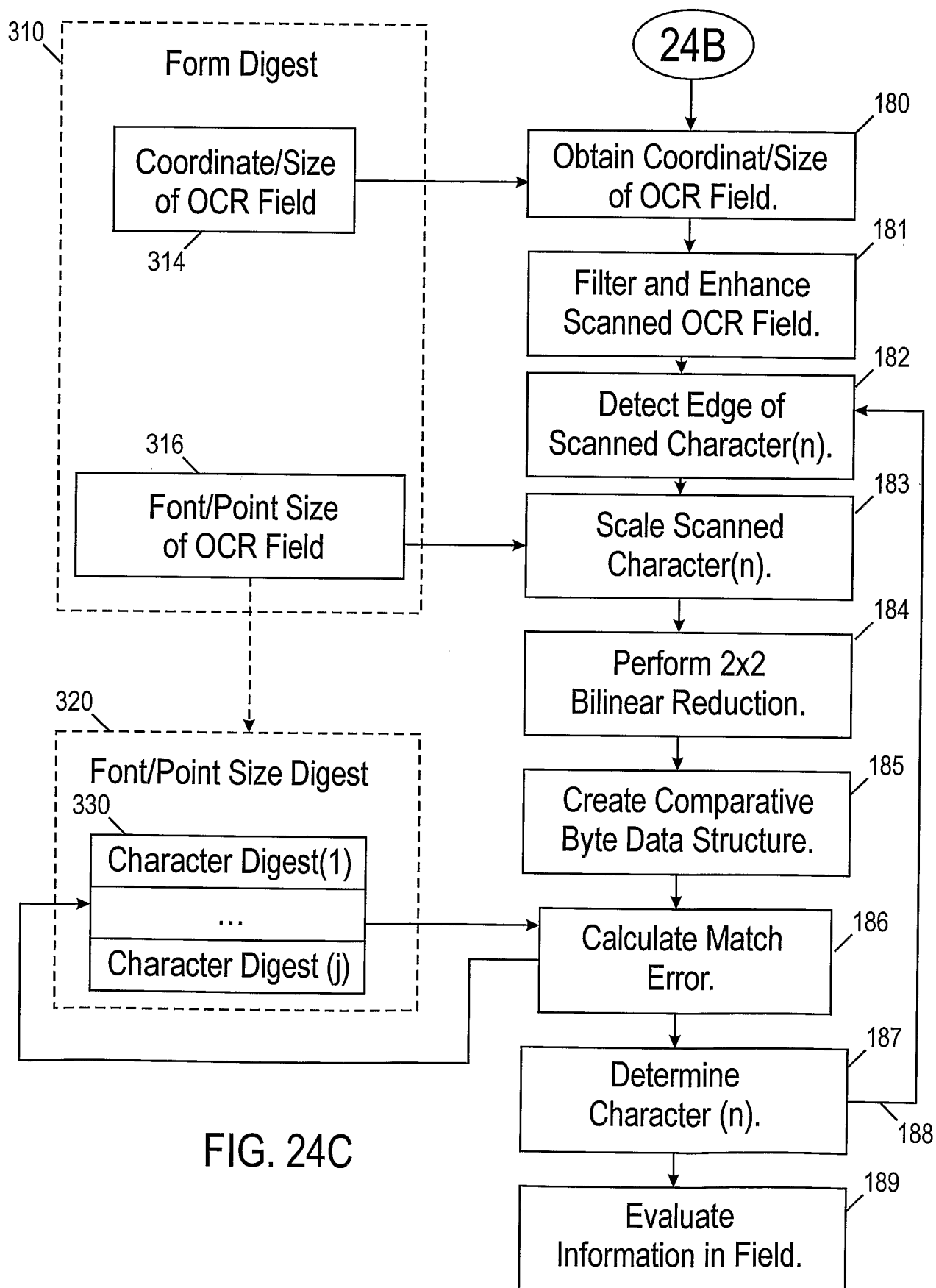


FIG. 24B

21/28



22/28

400
052269

401
052269

402
052269

403
052269

404
052269

405
052269

FIG. 25

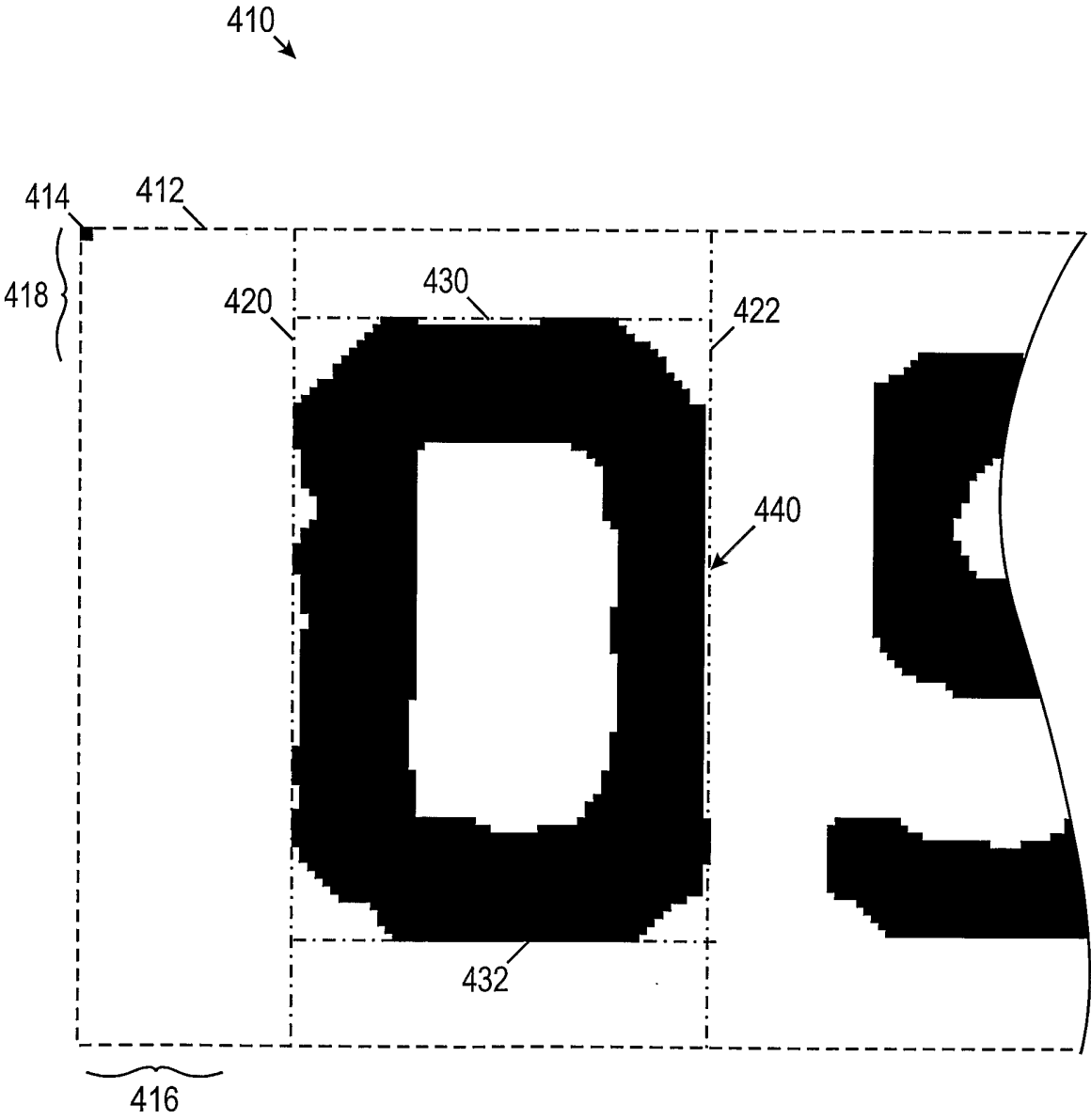


FIG. 26

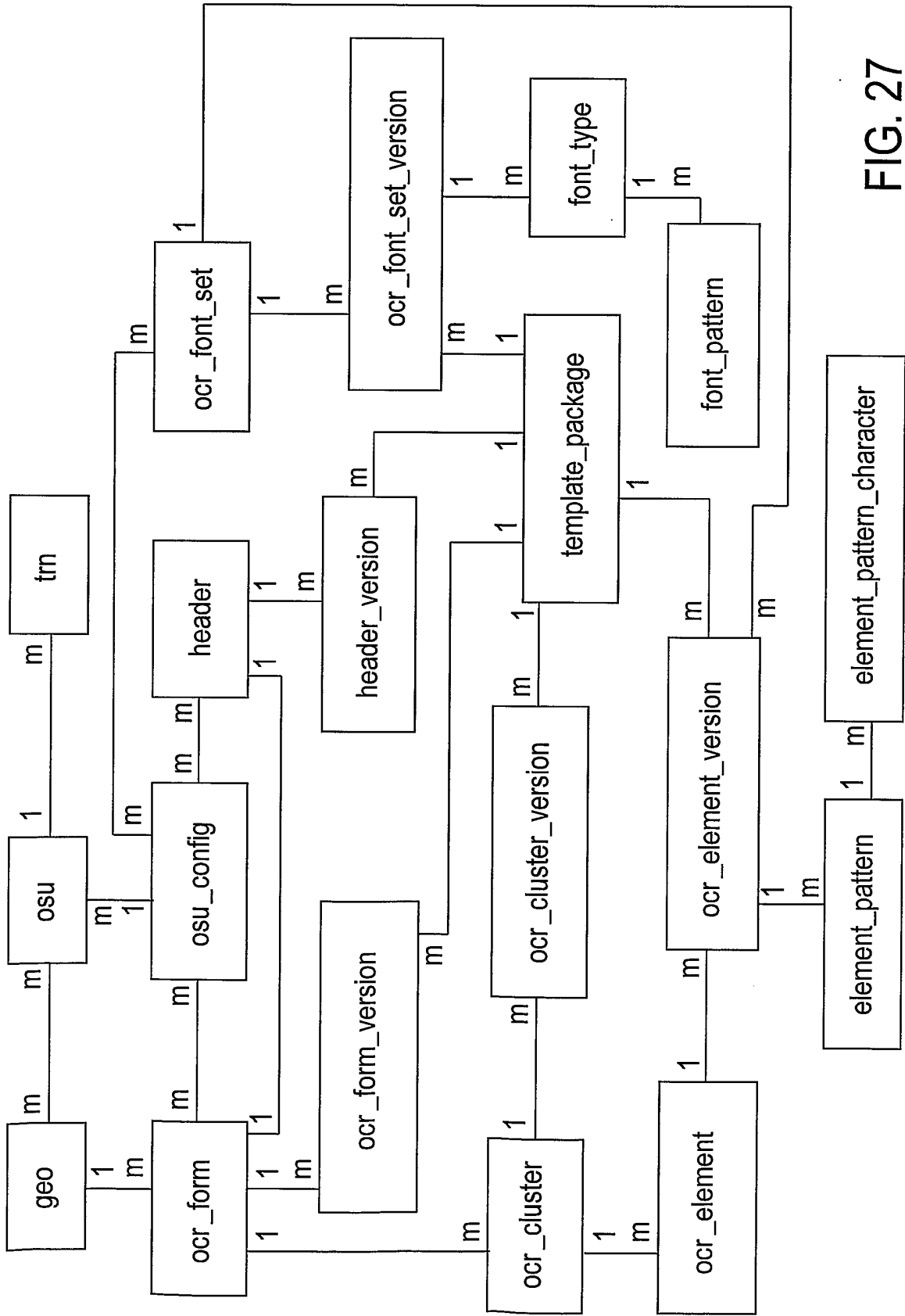


FIG. 27

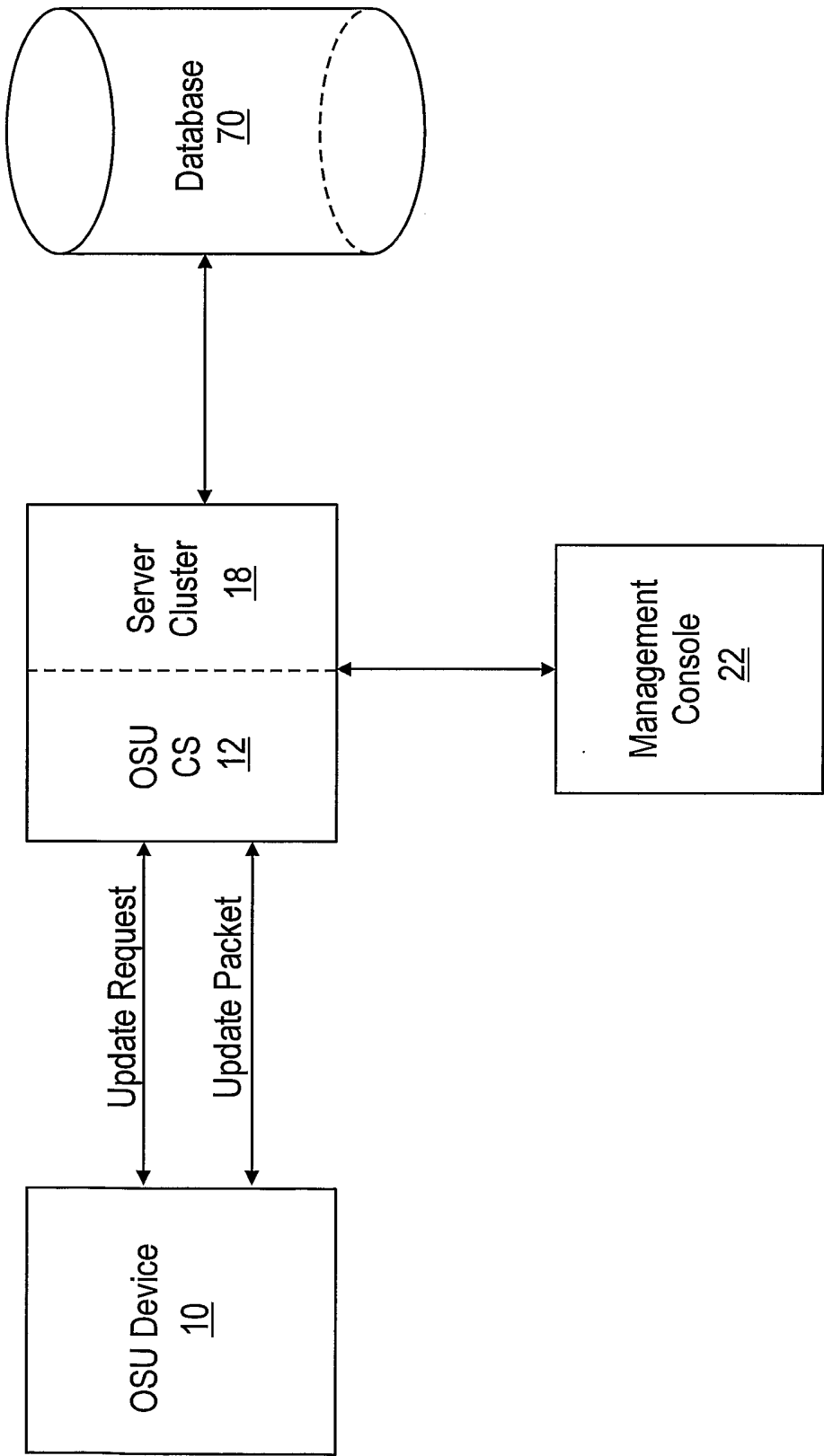


FIG. 28

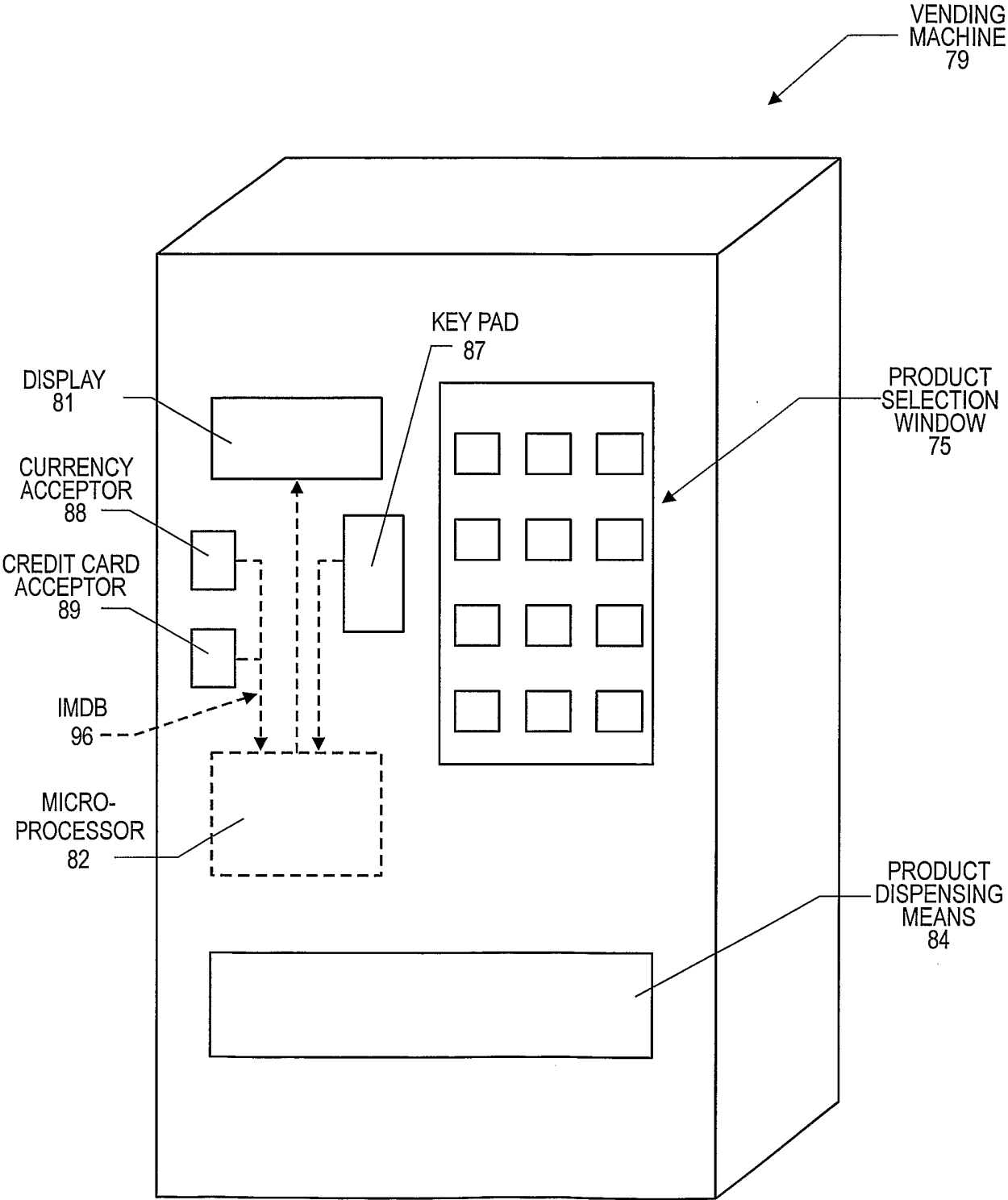


FIG. 29
(PRIOR ART)

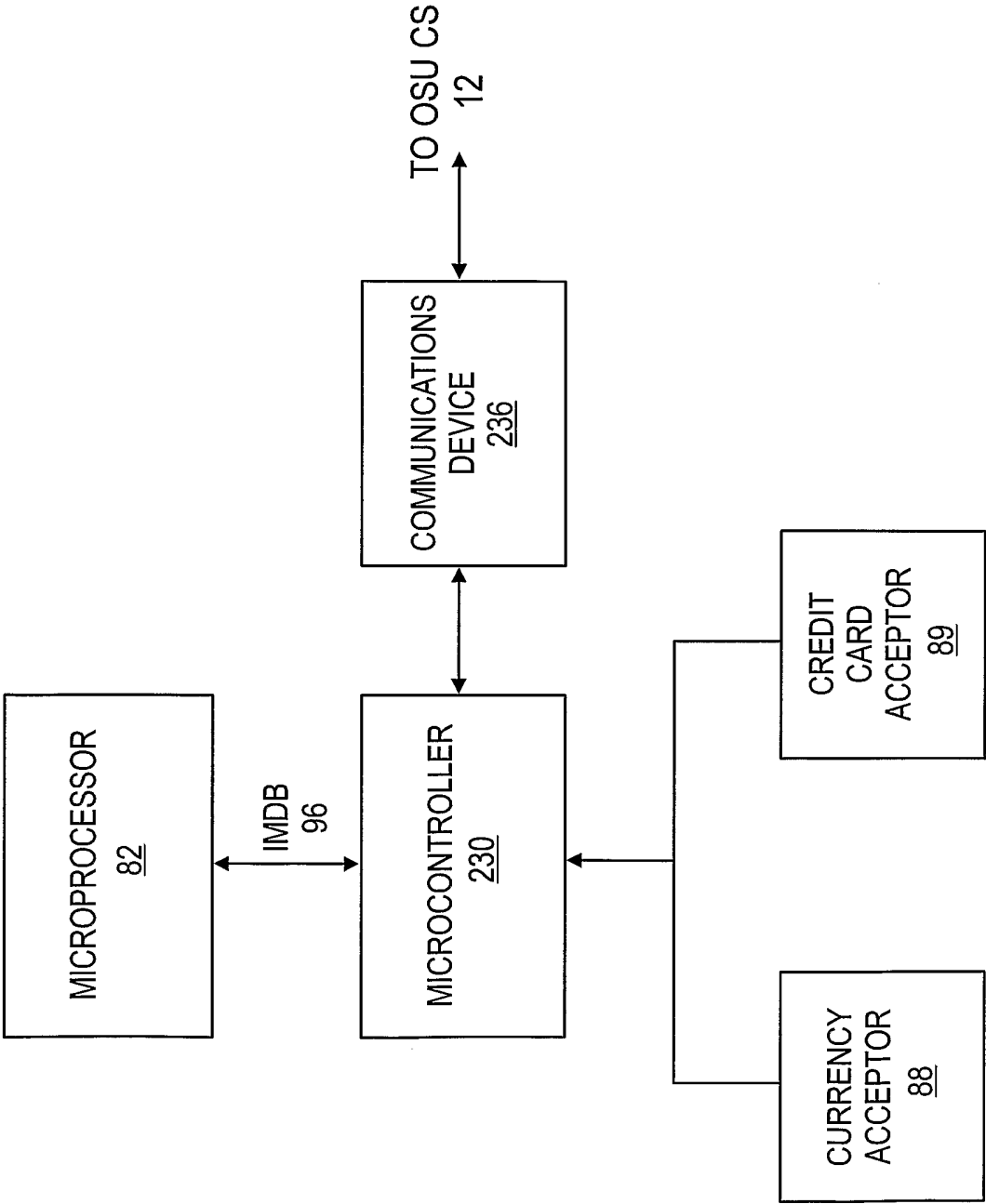


FIG. 30

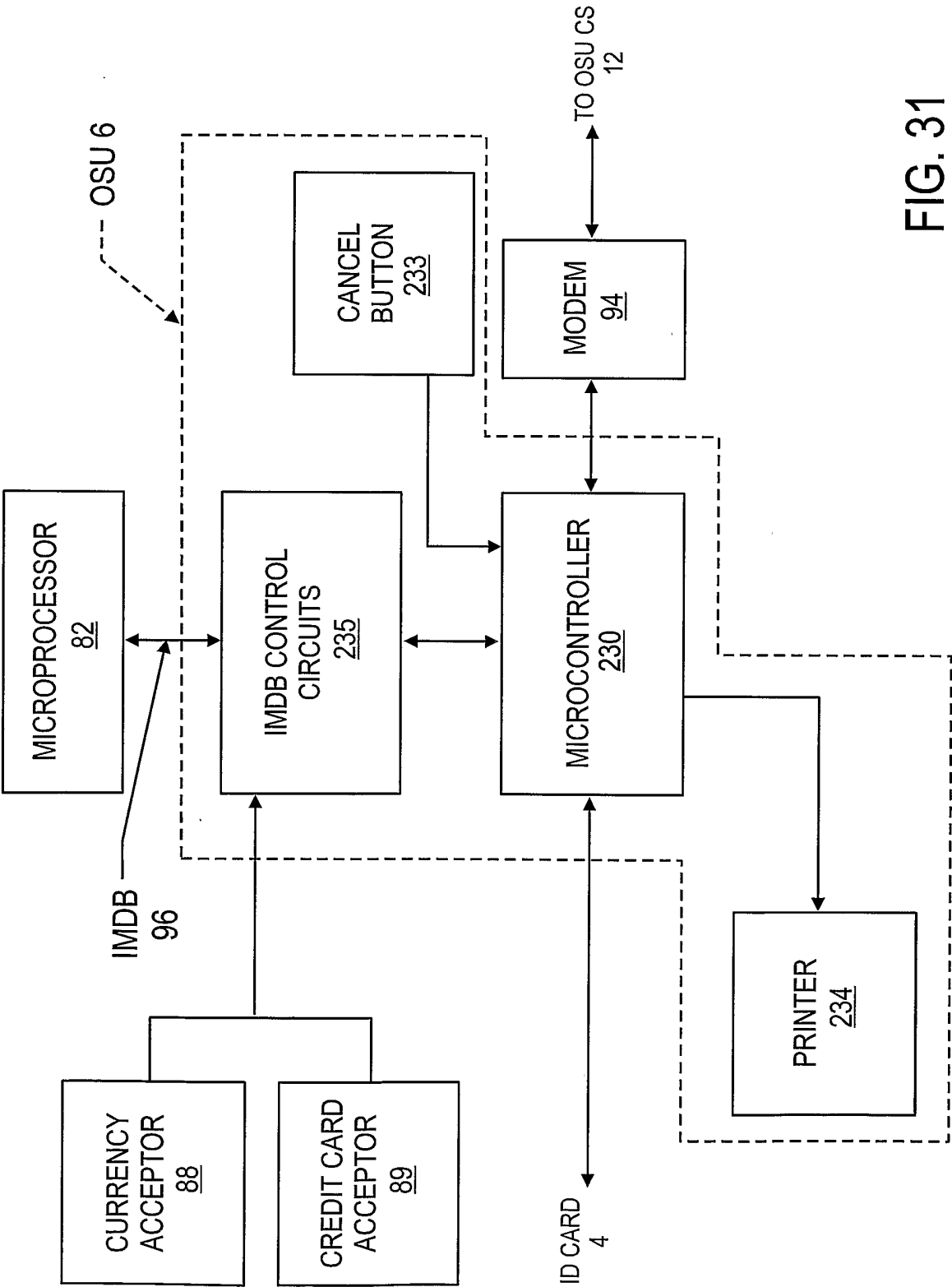


FIG. 31