(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 May 2014 (22.05.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/078696 A2

(51) International Patent Classification:
G06Q 20/42 (2012.01)

(21) International Application Number:
PCT/US2013/070369

(22) International Filing Date:
15 November 2013 (15.11.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/727,558 16 November 2012 (16.11.2012) US

(71) Applicants: **THE BOARD OF TRUSTEES OF THE UNIVERSITY OF ILLINOIS** [US/US]; 506 South Wright Street, Urbana, Illinois 61801 (US). **UNIVERSITY OF MICHIGAN** [US/US]; 4901 Evergreen Road, 220 ELB, Dearborn, Michigan 48128 (US).

(72) Inventors: **VENKATAKRISHNAN, Venkat**; 2020 N. Lincoln Park West, #30M, Chicago, Illinois 60614 (US). **ANSARI, Rashid**; 2104 Sand Island Court, Naperville, Illinois 60564 (US). **GJOMEMO, Rigel**; 1829 South May Street, Chicago, Illinois 60608 (US). **MALIK, Hafiz**; 1572 Brookline Street, Canton, Michigan 48187 (US). **SUMB, Nilesh**; 1926 West Harrison Street, Apt. #701, Chicago, Illinois 60612 (US).

(74) Agents: **MADDEN, Robert B.** et al.; Schwegman, Lundberg & Woessner, P.A., P.O. Box 2938, Minneapolis, Minnesota 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) **Title**: COUNTERING FRAUD IN REMOTE CHECK DEPOSIT

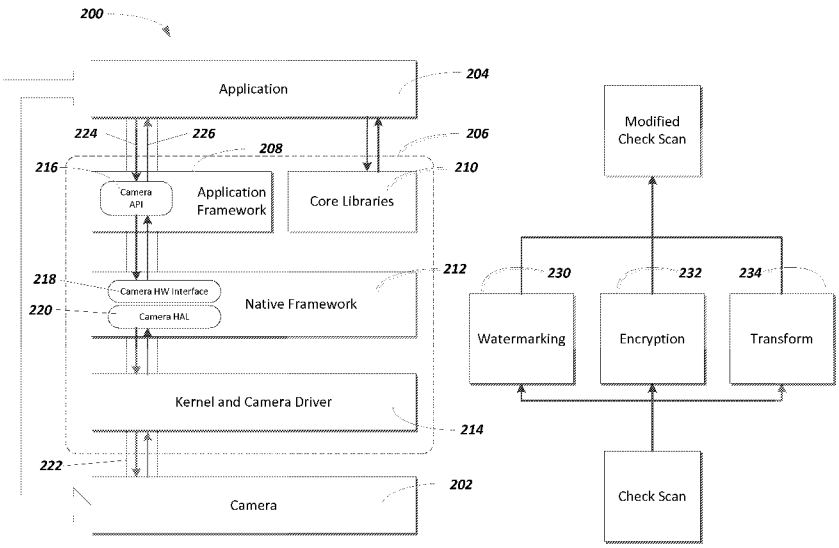

FIG. 2

(57) **Abstract**: Embodiments reduce the likelihood that an individual can successfully submit an altered scanned check image for payment to a bank, thus improving the security of mobile and remote banking. Some embodiments focus on reducing the ability to successfully doctor a scanned check image. Other embodiments focus on detecting a doctored scanned check image. Technologies employed in embodiments can include watermarking, encryption, transformation of scanned images, post processing consistency verification, spatial-domain consistency verification, transform-domain consistency verification, camera lens consistency verification, and camera signature verification.

WO 2014/078696 A2

COUNTERING FRAUD IN REMOTE CHECK DEPOSIT

CLAIM OF PRIORITY

[0001] This Application claims the benefit of U.S. Provisional Application No. 61/727,558, filed November 16, 2012, which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] Embodiments pertain to countering fraud in remote check deposit schemes. More particularly, embodiments pertain to identifying that a scanned check image has been altered and/or reducing the likelihood that an altered check will be accepted as valid.

BACKGROUND

[0003] Mobile banking is convenient for customers and banks and other institutions are rapidly adopting systems and infrastructure to support it. However, because at least part of the system resides fully within the control of the customer, security risks can exist.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The present technology is illustrated by way of example, and not by way of limitation in the figures of the accompanying drawings.

[0005] FIG. 1 illustrates the user end infrastructure to support mobile banking in some embodiments.

[0006] FIG. 2 illustrates an example embodiment to detect software tampering.

[0007] FIG. 3 illustrates an example embodiment to provide correspondence between an original check and a captured copy.

[0008] FIG. 4 illustrates an example embodiment to verify post-processing consistency.

[0009] FIG. 5 illustrates an example embodiment to verify spatial domain consistency.

[0010] FIG. 6 illustrates an example embodiment to verify transform domain consistency.

[0011] FIG. 7 illustrates an example embodiment to verify camera lens consistency.

[0012] FIG. 8 illustrates an example embodiment to verify camera signature.

[0013] FIG. 9 illustrates an example embodiment with multiple verification agents.

DETAILED DESCRIPTION

[0014] The description that follows includes systems, methods, techniques, instruction sequences, and computing machine program products that embody illustrative embodiments of the present disclosure. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide an understanding of various embodiments of the inventive subject matter. It will be evident, however, to those skilled in the art that embodiments of the disclosed subject matter may be practiced without these specific details. In general, well-known instruction instances, protocols, structures, and techniques have not been shown in detail.

[0015] Various modifications to the embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments and applications without departing from the scope of the

invention. Moreover, in the following description, numerous details are set forth for the purpose of explanation. However, one of ordinary skill in the art will realize that embodiments of the invention may be practiced without the use of these specific details. In other instances, well-known structures and processes are not shown in block diagram form in order not to obscure the description of the embodiments of the invention with unnecessary detail. Thus, the present disclosure is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

[0016] FIG. 1 illustrates a device 100, such as a smartphone, that is suitable for remote banking. Device 100 includes a camera 102 or other capture device, which can capture images of negotiable instruments, such as checks. As used herein, the term check is intended to cover all negotiable instruments that can be deposited or otherwise submitted for payment electronically using device 100. Device 100 generally includes one or more layers, sometimes referred to as a stack, which interacts with camera 102 and passes captured images to a suitable application 108. In FIG. 1, this stack is illustrated by driver 104 and operating system 106. In other embodiments, such a stack may include a virtual machine or other layers. Application 108 is designed to interact with a bank or other institution or infrastructure where a captured image of a check may be submitted for payment. For simplicity, the term bank will be used to encompass all such infrastructure, entities and institutions.

[0017] A typical use scenario would include a user launching application 108 and capturing an image of a check made out to her. The user then uses application 108 to remotely deposit the check to her account. Application 108 transmits the image securely to the user's bank using suitable encryption. Application 108 offers many conveniences such as avoiding the need for transportation to banks and standing in lines, being available 24-7, and so on. However, without adequate safeguards, the check image may be vulnerable to fraudulent manipulations by a user.

[0018] Application 108 is typically produced by (or under the direction of) a bank and hence is a trusted entity. However, the stack between camera 102 and application 108 is typically not produced by the bank and can be subject to manipulation. For example, a captured image from camera 102 may be intercepted before it reaches application 108. The intercepted image can then be manipulated in order to modify such critical information as the amount due. An image modified in this way will be referred to herein as either modified or doctored (e.g. modified image or doctored image). The doctored image can then be submitted to application 108 as if it had come from camera 102.

[0019] In order to reduce the likelihood that a doctored image will be accepted as authentic, embodiments herein will be described in terms of two broad categories: 1) embodiments that build in tamper resistance to make it hard for users to create a doctored image; 2) embodiments that focus on detection of doctored images. While these categories may be thought of as separate, they are typically interrelated. For example, incorporating a watermark having certain characteristics may result in increased effectiveness of a particular detection strategy (or any detection strategy). So although the embodiments may be described independently, employment of an embodiment in one category will typically have an effect in the other category and the categories are often blurred.

[0020] FIG. 2 illustrates an example embodiment to detect software tampering. A basic device, such as a smartphone, tablet or other portable device, suitable for remote deposit is shown in FIG. 2 at 200. The device has a capture device, such as camera 202, to capture an image of a check. The device also has application 204, which interacts with a bank to submit the captured image for payment. Between camera 202 and application 204 is a stack 206. Stack 206 can include a variety of layers, depending on the exact implementation and environment. In one example (not shown), stack 206 may have as few layers as a camera driver and general operating system (such as the layers illustrated in FIG. 1). Alternatively, stack 206 may have more (or fewer) layers. In an example illustrated

in FIG. 2, stack 206 may comprise some or all of application framework 208, core libraries 210, native framework 212 and/or kernel and camera driver 214.

[0021] Application framework 208 may be a framework such as the Java application framework found in the Android operating system. In one example, such a framework can contain the camera Application Programming Interfaces (APIs) 216 that are called by application 204 and the like to access camera 202 functionality.

[0022] Many operating systems provide a set of core libraries or interfaces, which are available to provide services or access functionality provided by the operating system and other layers. In addition, some operating systems allow libraries of functionality to be added thereto. In the example of FIG. 2, such libraries or interfaces are illustrated as core libraries 210.

[0023] Native framework 212 can be a lower-level framework such as C/C++ Android Framework provided in the Android operating system. Other operating systems may have other examples. Native framework 212 may provide additional functionality or interfaces that relate to camera functionality. In FIG. 2, native framework 212 contains camera hardware interface 218 and camera Hardware Abstraction Layer (HAL) 220. Camera hardware interface 218 provides the functionality and interface of camera 202 at the hardware level. Camera HAL 220 provides an adaptation layer that allows changes in the underlying hardware (e.g. between different phones, tablet devices, etc.) without requiring changes in the higher level layers. In other words, camera HAL 220 provides a consistent set of interfaces and functionality to higher level layers even though the exact hardware details may vary somewhat from platform to platform.

[0024] Kernel and camera driver 214 illustrate further layers that may reside between the native framework and the camera.

[0025] All the layers described are simply examples and different operating systems and platforms will have different layers. However, the functions described above (allowing application 200 to access camera 202 functionality) will be performed.

[0026] Unlike device 100 of FIG. 1, device 200 includes a trusted path to reduce the likelihood of a user gaining access to the image in order to doctor the image prior to submission, increase the likelihood of detecting tampering with the image, the path itself, components along the path (e.g. application framework 208, native framework 212, kernel and/or camera driver 214), or any combination thereof. In FIG. 2, two alternatives are illustrated (although only one need be implemented in any give embodiment). The first protected path is illustrated by protected path 222, which encircles the arrows (e.g. 224, 226) from camera 202 all the way to application 204. The arrows between each of the layers represent possible data exchanges between the layers.

[0027] An alternative protected path implementation is illustrated in FIG. 2 as protected path 228. In the figure, protected path 228 runs directly from application 204 to camera 202, bypassing stack 206. In this implementation, if the data does not flow through stack 206, it cannot be intercepted in the event one or more of the layers within stack 206 are compromised. Although protected path 228 is shown as bypassing the entire stack 206, it is possible to implement a protected path that only bypasses some of the layers within the stack and not all of them as illustrated by 228.

[0028] There are numerous hardware and/or software mechanisms and strategies to create a trusted path, which may depend on the chosen hardware and software platforms.

[0029] In FIG. 2, application 204 can ensure that a trusted path is established between all the components of the application that are involved in image processing all the way to the root of trust that includes most (if not all) components of the stack that are involved in the image capture. This ensures that this path is tamper-resistant and reduces the likelihood of any interposing attack.

[0030] With trusted path 222 or 228 in place, additional defensive measures can be employed. Examples of such defensive measures are illustrated in FIG. 2 and can be implemented by the application, or by various layers within the stack. In

FIG. 2, measures that can be employed include watermarking 230, encryption 232 and transformation 234.

[0031] Watermarking 230 indicates that regions of interest in a check can be watermarked. Regions of interest (ROIs) can include the courtesy area where the amount of the check is written numerically, the legal area where the amount of the check is written in words, and the area containing the name of the person or entity the check is made out to (e.g. the payee). Other areas can also be included in the ROIs, depending on particular strategy employed. Watermarking generally embeds robust and secure watermark(s) in image captured. In one example, a secret key can be used to select watermarking locations and watermark generation. Alternatively, watermarking location or parameters may be selected in a predetermined manner, be selected based on other information or perhaps received from the bank or other entity. A variety of methods are available for watermarking. One example that can be used is described in S. Ababneh, R. Ansari, A. Khokhar, *Compensated Signature Embedding for Multimedia Content Authentication*, ACM Journal of Data and Information Quality (JDIQ), Volume 1 Issue 3, December 2009, which is incorporated herein by reference.

[0032] Encryption 232 indicates that raw image or raw image ROIs can be encrypted prior to manipulation. The encrypted ROIs can be sent as side information or otherwise included in communications between application 204 and a bank. The bank can then check the integrity of the encrypted ROIs and check the consistency of the received information with the received image. Alternatively, rather than encrypt ROIs, cryptographically strong hash functions can be used to create a digital signature that can then be employed to check for alteration in the captured image. Again, a secret key can be used to select ROIs or specific predetermined ROIs can be used. In some embodiments the encryption would take place at the hardware level and asymmetric (e.g. public key) or symmetric (e.g. secret key) encryption can be used. In some instances, encryption keys should be shared between the banks and hardware manufacturers in some trusted way (e.g. not through the software stack, which is under the control of a potential attacker).

[0033] Transformation 234 indicates that the raw image (or ROIs) can be transformed in some fashion. For example, various parameters can be varied according to a server (e.g. bank) sent key prior to capturing the image. In one example, different parts of the image may be captured with different quality parameters. In such a representative example, the captured image may be divided into a number of blocks (or squares) and the image may be compressed differently for every block, with the compression qualities selected by the key sent by the bank. Alternatively, the bank may send the parameters themselves.

[0034] Although the above description indicated that watermarking 230, encryption 232 and transformation 234 can be applied to ROIs, it is also possible to apply one or more of these to the entire image. Furthermore, the various mechanisms of watermarking, encryption, and transformation may be used alone or in any combination. Thus, although FIG. 2 illustrates watermarking 230, encryption 232 and transformation 234 as parallel operations, the output of one block may be used as the input to another block, in any combination. So the captured image may be transformed in accordance with a set of parameters, watermarked in accordance with a set of parameters, and then encrypted in accordance with a set of parameters. This order may be rearranged as well.

[0035] As an example of how the protected path may be used in conjunction with watermarking 230, encryption 232 and transformation 234, application 204 may receive information such as one or more keys from the bank that allow selection of various parameters. Based on the key(s), application 204 selects a set of watermarking parameters (including the locations of the watermarks), encryption parameters (including, perhaps, encryption parameters that vary in different blocks or regions of the captured image), and transformation parameters. These parameters are sent along protected path 222 through stack 206 (or alternatively through protected path 228 bypassing some or all of the layers in stack 206) to camera 202 and perhaps other hardware such as a hardware encryption module (not shown) perhaps using a function call such as takePicture(tr_watermark, region_specific_parameters). The picture is taken using the

region_specific_parameters, the watermark is added according its parameters (tr_watermark), and all data is encrypted at the hardware level in accordance with its parameters. The encrypted, transformed, watermarked image is then passed back through protected path 222 (or through protected path 228) where application 204 submits it to the bank.

[0036] In some embodiments, mechanisms are put in place to establish a correspondence between the hard check (e.g. the paper check itself) and the captured image. FIG. 3 illustrates an example embodiment to provide correspondence between an original check and a captured copy.

[0037] In FIG. 3, at least one feature of the check is captured at the time the check is written (e.g. before it is given to the payee or contemporaneously with the payee receiving the check). In this way, some form of "carbon copy" of the check can be captured and transmitted to the bank to compare against later received information and images. This capture/carbon copy is illustrated in FIG. 3 by 300. Various features that can be captured are key check information, an image of the check or a portion of the check (e.g. ROIs) or information derived therefrom. This can include, for example, a hash of the check image or ROIs, an encrypted or otherwise transformed copy of the check image or ROIs, the amount of the check and other identifying information (such as the check number) or any other feature that will help detect a doctored image. An appropriately configured device can be used to accomplish this. For example, a smartphone, tablet, or other device with a capture device (camera, etc.) can be used. It is also possible to have a device that prints the check and captures information to be sent to the bank at the same time.

[0038] Additionally, or alternatively, the paper checks can be designed with patterns that aid tamper detection. Such patterns preserve the detection capability even after they are mapped to a camera image under different and non-ideal lighting conditions. Check alterations would likely disturb the pattern enough for the alteration to be detected. One option is to use a pattern that is designed so that the spatial autocorrelation function of the check approximates a 2-D impulse function.

[0039] After features of the check are captured, they are sent to the bank (or other location accessible to the bank). This is illustrated in FIG. 3, by check feature store 302. The check is also given to the payee as indicated by 304. When the check image is captured for deposit as illustrated by 306, the same features can be extracted as indicated by 308. Note that this can be done either by a local device or by the bank itself when it receives the captured image. The extracted features can then be compared with the features previously stored in 302 as indicated by 310. The received image can then be declared as doctored or original.

[0040] When images are processed after capture, such as when an image is doctored, there are often various artifacts left behind that can be discovered by various techniques. All these techniques fall into the category of detecting a doctored image. These techniques can be used either signally or in combination with each other (or in combination with other mechanisms illustrated herein). Such techniques can include post processing consistency verification, spatial domain consistency verification, transform domain consistency verification, chromatic aberration consistency verification and camera signature consistency verification. In each case, the captured image is analyzed to detect traces of a specific forgery.

[0041] FIG. 4 illustrates an example embodiment to verify post-processing consistency. In FIG. 4, an image is captured or other wise obtained as illustrated in 400. From there, ROIs of the image can be extracted at 402. This is particularly useful when particular regions of the captured image are more likely to be doctored than others (like the amount fields). The output is then checked to determine whether the captured image has been subjected to post-processing such as double JPEG compression in 404. This is accomplished by examining the output for fingerprints or traces left by the post-processing. For example, traces are left behind when an image is doubly compressed, that is, it is first compressed with a quality factor Q1 followed by quality factor Q2, where $Q1 \neq Q2$. Such traces in the resulting image can be used for double compression detection. Other post-processing detection that may be done include splicing, cut and paste, and so on. In general, traces of almost all common image post-processing operations are detectable.

[0042] Although good forgeries may leave no clues in the doctored image which are visible to the human eye, these forgeries may however disturb the underlying statistics of the image. In general, digital cameras such as those used in smartphones, tablets or other devices employ a single sensor per pixel in conjunction with a color filter array (CFA), and then interpolate the missing color samples to obtain a three-channel color image. As a result, only one third of the samples, in a color image, are captured directly by the camera and remaining two thirds are interpolated. The CFA interpolation process introduces specific correlations, which are likely to be altered in the tampered image.

[0043] FIG. 5 illustrates an example embodiment to verify spatial domain consistency. The embodiment in FIG. 5 uses the local-correlation masks estimated from RGB color channels for detecting forgery in the captured image. To reduce the computational cost and improve detection performance the embodiment in FIG. 5 focuses on ROIs such as the courtesy amount and the legal amount areas of the bank checks, for forgery detection. However, this is only exemplary and embodiments that consider the entire captured image can be used.

[0044] The embodiment in FIG. 5 is designed to identify traces of splicing in ROIs of the check. To achieve this, the CFA for each color channel is estimated in 502 using the image captured in 500. Next, the ROIs are extracted from the CFA estimates in 504. The ROIs are filtered using a high-pass filter in 506. In general a good high-pass filter will retain high frequency image content using a cut-off normalized radian frequency greater than $\frac{\pi}{2}$ in each of the horizontal and vertical direction.

[0045] Each ROI is then processed in both horizontal 510 and vertical 508 scans. Next 512 segments the output into blocks and 514 computes the block-based randomness using approximate entropy for each scan. The approximate entropy may be estimated by computing the log-likelihood of the similarity between adjacent data blocks. More details of how to estimate approximate entropy can be found in Pincus SM. *Approximate entropy as a measure of system complexity*, Proc Natl Acad Sci USA 1991;**88**:2297-2301, which is incorporated herein by reference.

[0046] Any block (or group of blocks) that has a randomness value higher than the predefined threshold is (or are) labeled as doctored. Finally 516 determines whether the image is doctored based on this information.

[0047] Cues for tampering often become more evident in the transform domain, for example Fourier, Wavelet, and other transforms. FIG. 6 illustrates an example embodiment to verify transform domain consistency. For each color channel, the embodiment of FIG. 6 performs higher-order spectral analysis on CFA estimates. In one example, the bicoherence spectrum, a third-order statistics in the Fourier transformed domain, can be used to detect traces of splicing. Again, to reduce the computational cost and improve detection performance embodiments can focus on the ROIs, although this is not required.

[0048] In FIG. 6, the CFA for each color channel at 602 from the image captured in 600. ROIs are extracted from the CFA estimates in 604. Next 606 filters the ROIs with a high-pass filter. Each ROI is processed in both horizontal 610 and vertical 608 scan. Next 612 segments the output into blocks and 612 computes the block-based bicoherence spectrum for each scan. Any block (or group of blocks) that has significant energy in the high frequency bands is (or are) labeled as doctored or forged. Finally 612 determines whether the image is doctored based on this information.

[0049] Most images contain a variety of aberrations that result from imperfections and artifacts of the optical imaging system. In an ideal imaging system, light passes through the lens and is focused to a single point on the sensor. Optical systems, however, deviate from such ideal models in that they fail to perfectly focus light of all wavelengths. The resulting effect is known as chromatic aberration, which occurs in two forms: longitudinal and lateral. Longitudinal aberration manifests itself as differences in the focal planes for different wavelengths of light. Lateral aberration manifests itself as a spatial shift in the locations where light of different wavelengths reach the sensor. This shift is proportional to the distance from the optical center. In both cases, chromatic aberration leads to various forms of color imperfections in the image. When an

image is doctored, these aberrations are often disturbed and fail to be consistent across the image.

[0050] FIG. 7 illustrates an example embodiment to verify camera lens consistency. The embodiment of FIG. 7 focuses on the 2D chromatic aberration consistency in the ROIs of the check. However, as noted elsewhere, the entire captured image may be used.

[0051] In FIG. 7, the global 2-D lateral chromatic aberration model parameters that bring the color channels back into alignment are estimated for the captured image. This is illustrated in 700 and 702. One algorithm to accomplish this is described in *M.K. Johnson and H. Farid,* "Exposing Digital Forgeries Through Chromatic Aberration" ACM Multimedia and Security Workshop, Geneva, Switzerland, 2006, incorporated herein by reference.

[0052] The ROIs are extracted 704 and segmented 706 into non-overlapping blocks or size N1xN2. The local 2-D lateral chromatic aberration model parameters for each block are then estimated in 708. Finally, the global 2D lateral chromatic aberration estimate is compared to estimates from local blocks. Any block that deviates significantly from the global estimate is labeled as doctored or forged and a final decision is then reached 710.

[0053] A camera signature can be generated from the noise pattern of the sensor. Doctoring an image can often disturb the noise pattern. The example embodiment of FIG. 8 focuses on detecting forgery traces using camera signature (or camera noise pattern). In the example embodiment, a set of reference images will be used to generate reference noise pattern image. The captured noise pattern will be extracted from the captured image. Local similarity measure, e.g., cross correlation, between the captured noise pattern and the reference noise pattern is estimated for integrity verification. Again, the embodiment in FIG. 8 uses ROIs rather than the entire captured image, but that is only exemplary and the entire image can also be used.

[0054] In FIG. 8, a set of captured references images 800 is used to estimate a reference noise pattern 802.

[0055] This reference noise pattern is stored in a database accessible to the bank (or device if this is performed locally), such as 804. This process can be accomplished during a registration process when a user downloads an appropriate application and enters information needed to allow functioning of the application. In this sense, the registration process ties the application and registration information to a particular device. The set of reference images may comprise some number, five for example, taken using a "normal" setting that allows the noise pattern to be estimated. The captured reference images can be updated by adding (supplementing the existing reference images or replacing existing reference images) new authenticated images.

[0056] When a check is deposited, the captured noise pattern is estimated 808 from the captured image 806. ROIs are extracted from the captured noise pattern as indicated in 810. The ROIs are segmented into blocks in 812. A similarity measure between ROIs of the reference noise pattern and the captured noise pattern is computed in 814. Any block (or group of blocks) with low similarity measure is (or are) labeled as doctored or forged. Finally a decision is made based on the information as indicated by 816.

[0057] As previously indicated, the individual embodiments and mechanisms may be used individually or in various combinations. FIG. 9 illustrates an example embodiment with multiple verification agents. In FIG. 9, all the various consistency checks illustrated by FIGs. 4-8 are included. However, such agents can be used in any combination.

[0058] In FIG. 9, a captured image 900 is made available to multiple agents. The term agents is generic and should be read to encompass a particular embodiment that accomplishes the verification indicated and is not limited to a particular implementation. The agents illustrated in FIG. 9 include a post-processing verification agent 902, a spatial domain consistency verification agent 904, a transform-domain consistency verification agent 906, a chromatic aberration consistency verification agent 908, and a camera consistency verification agent 910. Post processing verification agent 902 may be an embodiment such as that

illustrated in FIG. 4. Spatial domain consistency verification agent 904 can be an embodiment such as that illustrated in FIG. 5. Transform domain consistency verification agent 906 can be an embodiment such as that illustrated in FIG. 6. Chromatic aberration consistency verification agent 908 can be an embodiment such as that illustrated in FIG. 7. Camera consistency verification agent 910 can be an embodiment such as that illustrated in FIG. 8.

[0059] As illustrated in FIG. 9, the outputs of the various agents can be combined as illustrated by decision fusion unit 912. Such a decision fusion unit can employ a variety of mechanisms to fuse the individual decisions into a single decision. In one example, an expert system may be created based on a weighted combination of the output of each verification agent. The weights can be decided based on image data such as estimated lighting conditions, image resolution and so forth.

[0060] Embodiments described herein may be implemented in a variety of hardware and/or software configurations. An example embodiment extends to a machine in the example form of a computer system within which instructions for causing the machine to perform any one or more of the methodologies discussed herein may be executed. In alternative example embodiments, the machine operates as a standalone device or may be connected (e.g., networked) to other machines. In a networked deployment, the machine may operate in the capacity of a server or a client machine in server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine may be a personal computer (PC), a tablet device, a Personal Digital Assistant (PDA), a cellular telephone or smartphone, a web appliance, etc. Further, while only a single machine is illustrated, the term "machine" shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[0061] An example computer system may include a processor (e.g., a central processing unit (CPU), a graphics processing unit (GPU) or both), and memory of various forms. The computer system may further include a display and an input

device such as keyboard, touch screen, various user interfaces such as on screen keyboards, gesture input, voice input, etc.

[0062] *Machine-Readable Medium*

[0063] Embodiments also may include machine-readable storage medium on which is stored one or more sets of instructions and data structures (e.g., software instructions) embodying or used by any one or more of the methodologies or functions described herein. The instructions may also reside, completely or at least partially, within the memory or within the processor during execution thereof by the computer system, with the memory and the processor also constituting machine-readable media.

[0064] While the machine-readable storage medium is shown in an example embodiment to be a single medium, the term "machine-readable storage medium" may include a single storage medium or multiple storage media (e.g., a centralized or distributed database, or associated caches and servers) that store the one or more instructions. The term "machine-readable storage medium" shall also be taken to include any tangible medium that is capable of storing, encoding, or carrying instructions for execution by the machine and that cause the machine to perform any one or more of the methodologies of embodiments of the present application, or that is capable of storing, encoding, or carrying data structures used by or associated with such instructions. The term "machine-readable storage medium" shall accordingly be taken to include, but not be limited to, solid-state memories and optical and magnetic media. Specific examples of machine-readable storage media include non-volatile memory, including by way of example semiconductor memory devices (e.g., Erasable Programmable Read-Only Memory (EPROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), and flash memory devices); magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. Machine-readable storage medium excludes transmission medium including signals per se.

[0065] *Transmission Medium*

[0066] The instructions may further be transmitted or received over a communications network using a transmission medium via the network interface device and utilizing any one of a number of well-known transfer protocols. Examples of communication networks include a local area network (LAN), a wide area network (WAN), the Internet, mobile telephone networks, Plain Old Telephone Service (POTS) networks, and wireless data networks (e.g., WiFi and WiMax networks). The term "transmission medium" shall be taken to include any intangible medium that is capable of storing, encoding, or carrying instructions for execution by the machine, and includes digital or analog communications signals or other intangible medium to facilitate communication of such software.

[0067] Although an overview of the inventive subject matter has been described with reference to specific example embodiments, various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of embodiments of the present application. Such embodiments of the inventive subject matter may be referred to herein, individually or collectively, by the term "invention" merely for convenience and without intending to voluntarily limit the scope of this application to any single invention or inventive concept if more than one is, in fact, disclosed.

[0068] The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived there from, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

[0069] Moreover, plural instances may be provided for resources, operations, or structures described herein as a single instance. Additionally, boundaries between various resources, operations, modules, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned

and may fall within a scope of various embodiments of the present application. In general, structures and functionality presented as separate resources in the example configurations may be implemented as a combined structure or resource. Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements fall within a scope of embodiments of the present application as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[0070] The Abstract is provided to comply with 37 C.F.R. Section 1.72(b) requiring an abstract that will allow the reader to ascertain the nature and gist of the technical disclosure. It is submitted with the understanding that it will not be used to limit or interpret the scope or meaning of the claims. The following claims are hereby incorporated into the detailed description, with each claim standing on its own as a separate embodiment.

[0071] It will be appreciated that, for clarity purposes, the above description describes some embodiments with reference to different functional units or processors. However, it will be apparent that any suitable distribution of functionality between different functional units, processors or domains may be used without detracting from embodiments of the invention. For example, functionality illustrated to be performed by separate processors or controllers may be performed by the same processor or controller. Hence, references to specific functional units are only to be seen as references to suitable means for providing the described functionality, rather than indicative of a strict logical or physical structure or organization.

[0072] Although the present invention has been described in connection with some embodiments, it is not intended to be limited to the specific form set forth herein. One skilled in the art would recognize that various features of the described embodiments may be combined in accordance with the invention. Moreover, it will be appreciated that various modifications and alterations may be made by those skilled in the art without departing from the scope of the invention.

CLAIMS

What is claimed is:

1. A device for reducing check fraud comprising:

a capture device to capture a check images;

an application designed to receive the captured check image;

a protected path between the capture device and the application; and

a defensive protection mechanism to alter the captured check image, the defensive protection mechanism consisting of an alteration to increase the likelihood of detecting a modified captured check image or an alteration to increase the difficulty of modifying a captured check image, or both.

2. The device of claim 1 wherein the defensive protection mechanism comprises watermarking at least a region of interest in the captured check image.

3. The device of claim 1 wherein the defensive protection mechanism comprises encrypting at least a portion of the captured check image.

4. The device of claim 1 wherein the defensive protection mechanism comprises altering the captured check image based on information received by the application from a bank.

5. A system for reducing check fraud comprising:

a feature capture device to capture at least one feature of a check prior to or contemporaneously with a recipient receiving the check and to send the captured features to a bank;

a check deposit device comprising:

a capture device to capture a scanned copy of the check when the recipient submits the check for payment; and

a feature extractor to extract at least one feature from the scanned copy and to send the at least one feature to the bank.

6. The system of claim 5 wherein the extracted at least one feature comprises at least one region of interest for further examination.

7. The system of claim 5 wherein the extracted at least one feature comprises at least one region of interest extracted from a color filter array estimate created from a color filter array for each color channel in the check image.

8. The system of claim 5 wherein the extracted at least one feature comprises at least one region of interest extracted a noise pattern of the check.

9. A method to detect check fraud comprising:

obtaining a check image;

performing, using at least one processor, at least one consistency check comprising at least one of:

a post-processing consistency check;

a spatial domain consistency check;

a transform domain consistency check;

a camera lens consistency check; and

a camera signature consistency check.

10. The method of claim 9 wherein the post-processing consistency check comprises:

extracting at least one region of interest from the check image; and

examining the at least one region of interest for compression artifacts in order to detect double JPEG compression.

11. The method of claim 9 wherein the spatial domain consistency check comprises:

estimating a color filter array for each color channel in the check image to produce a color filter array estimate for each channel;

extracting at least one region of interest from the color filter array estimates;

filtering the at least one region of interest with a high pass filter;

processing the output of the high pass filter in both horizontal scan and vertical scan;

computing the block-based randomness for each block using approximate entropy for each scan;

comparing the block-based randomness to a predetermined threshold; and

identifying the corresponding block as tampered when the block based randomness exceeds the predetermined threshold.

12. The method of claim 9 wherein the transform domain consistency check comprises:

estimating a color filter array for each color channel in the check image to produce a color filter array estimate for each channel;

extracting at least one region of interest from the color filter array estimates;

filtering the at least one region of interest with a high pass filter;

process the output of the high pass filter in both horizontal scan and vertical scan;

computing a block-based biocoherence spectrum for each scan; and

identifying the corresponding block as tampered when the block-based biocoherence spectrum has energy exceeding a predetermined threshold in the high frequency bands.

13. The method of claim 9 wherein the camera lens consistency check comprises:

estimating global 2D chromatic aberration model parameters that bring color channels of the check image into alignment;

extracting at least one region of interest from the check image;

segment the at least one region of interest into non-overlapping blocks;

estimate local 2D chromatic aberration model parameters for each block;

compare the global 2D chromatic aberration model parameters to the local 2D chromatic aberration model parameters; and

identifying the corresponding block as tampered when the local and global model parameters diverge by more than a predetermined amount.

14. The method of claim 9 wherein the camera signature consistency check comprises:

extracting a noise pattern from the check image;

extracting at least one region of interest from the noise pattern;

compute a block similarity score by comparing blocks of the at least one region of interest to blocks of at least one stored reference noise pattern; and

identifying the corresponding block as tampered when the similarity score falls below a predetermined threshold.

15. The method of claim 9 wherein multiple consistency checks are performed and wherein the method further comprises:

computing fraud scores from the multiple consistency checks;

combining the fraud scores into an ultimate fraud score; and

determining the check has been tampered with when the ultimate fraud score exceeds a predetermined threshold.

16. A computer-readable storage media having executable instructions embodied thereon that, when executed, cause a system to perform operations comprising:

obtaining a check image;

performing, using at least one processor, at least one consistency check comprising at least one of:

a post-processing consistency check;

a spatial domain consistency check;

a transform domain consistency check;

a camera lens consistency check; and

a camera signature consistency check.

17. The storage media of claim 16 wherein the post-processing consistency check comprises:

extracting at least one region of interest from the check image; and

examining the at least one region of interest for compression artifacts in order to detect double JPEG compression.

18. The storage media of claim 16 wherein the spatial domain consistency check comprises:

estimating a color filter array for each color channel in the check image to produce a color filter array estimate for each channel;

extracting at least one region of interest from the color filter array estimates;

filtering the at least one region of interest with a high pass filter;

processing the output of the high pass filter in both horizontal scan and vertical scan;

computing the block-based randomness for each block using approximate entropy for each scan;

comparing the block-based randomness to a predetermined threshold; and

identifying the corresponding block as tampered when the block based randomness exceeds the predetermined threshold.

19. The storage media of claim 16 wherein the transform domain consistency check comprises:

estimating a color filter array for each color channel in the check image to produce a color filter array estimate for each channel;

extracting at least one region of interest from the color filter array estimates;

filtering the at least one region of interest with a high pass filter;

process the output of the high pass filter in both horizontal scan and vertical scan;

computing a block-based biocoherence spectrum for each scan; and

identifying the corresponding block as tampered when the block-based biocoherence spectrum has energy exceeding a predetermined threshold in the high frequency bands.

20. The storage media of claim 16 wherein the camera lens consistency check comprises:

estimating global 2D chromatic aberration model parameters that bring color channels of the check image into alignment;

extracting at least one region of interest from the check image;

segment the at least one region of interest into non-overlapping blocks;

estimate local 2D chromatic aberration model parameters for each block;

compare the global 2D chromatic aberration model parameters to the local 2D chromatic aberration model parameters; and

identifying the corresponding block as tampered when the local and global model parameters diverge by more than a predetermined amount.

21. The storage media of claim 16 wherein the camera signature consistency check comprises:

extracting a noise pattern from the check image;

extracting at least one region of interest from the noise pattern;

compute a block similarity score by comparing blocks of the at least one region of interest to blocks of at least one stored reference noise pattern; and

identifying the corresponding block as tampered when the similarity score falls below a predetermined threshold.

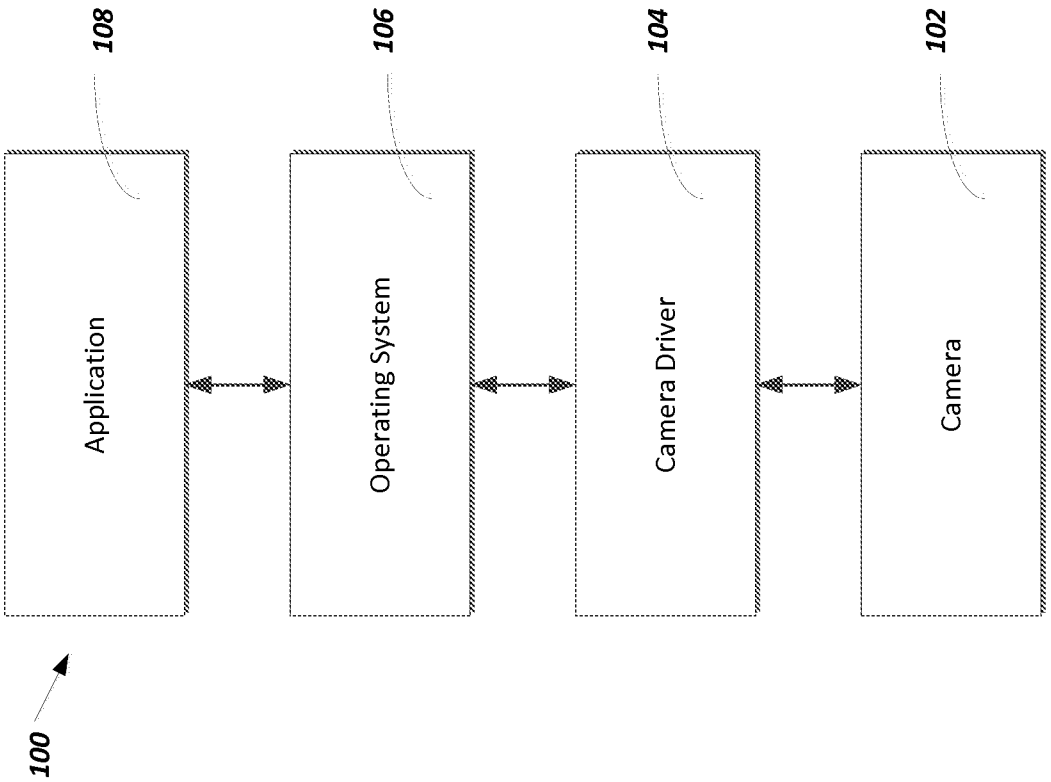
108
106
104
102
Application
Operating System
Camera Driver
Camera
100

FIG. 1

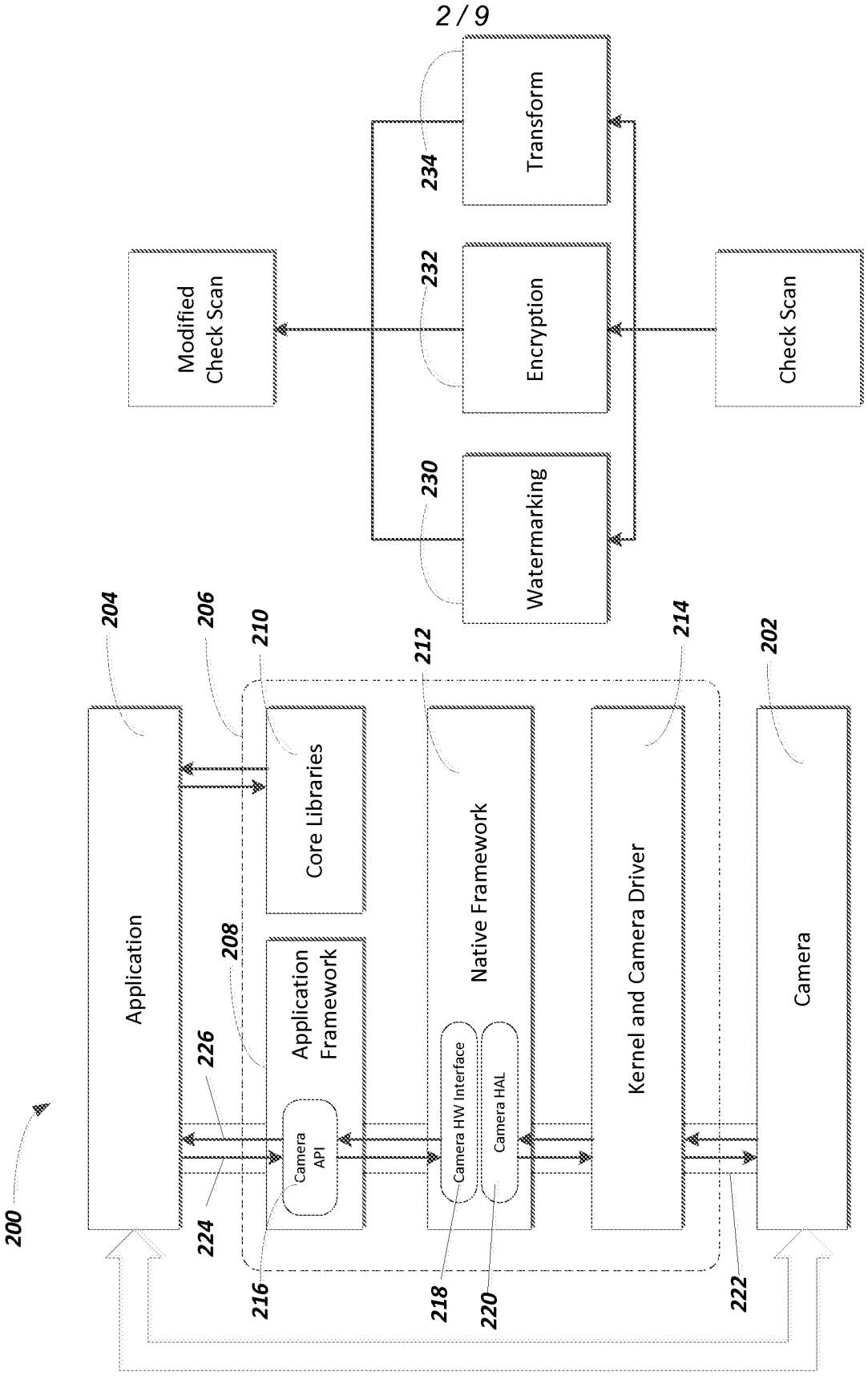
200
Application
204
206
Core Libraries
210
208
Application Framework
226
224
Camera API
216
212
Native Framework
Camera HW Interface
218
Camera HAL
220
214
Kernel and Camera Driver
202
Camera
222
Transform
234
Encryption
232
Watermarking
230
Modified Check Scan
Check Scan

FIG. 2

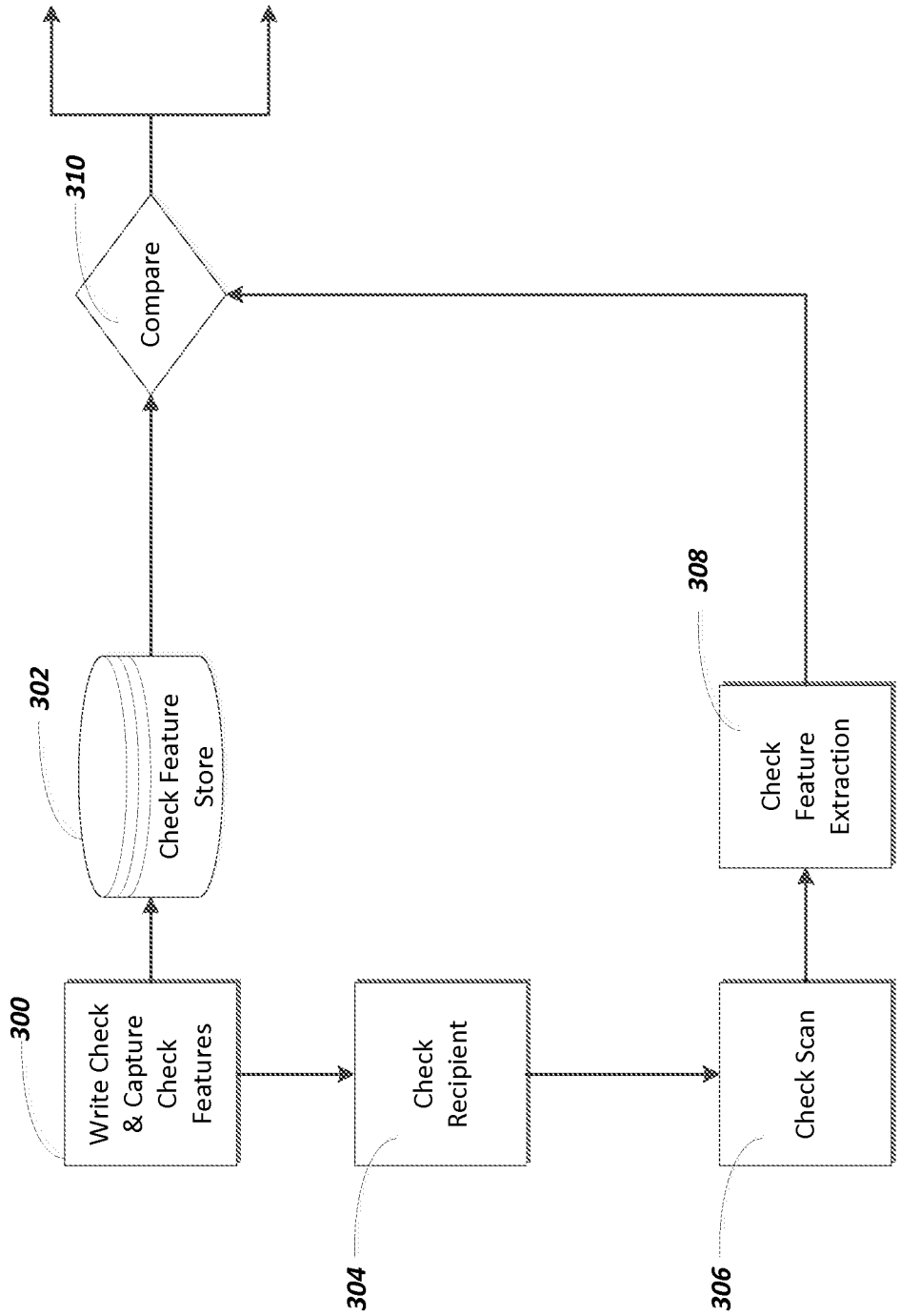
310
Compare
302
Check Feature Store
308
Check Feature Extraction
300
Write Check & Capture Check Features
Check Recipient
Check Scan
304
306

FIG. 3

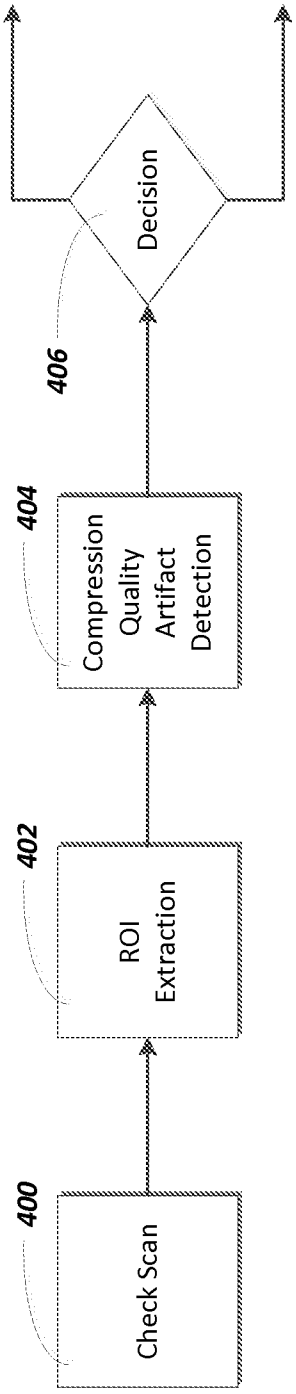
400
Check Scan
402
ROI Extraction
404
Compression Quality Artifact Detection
406
Decision

FIG. 4

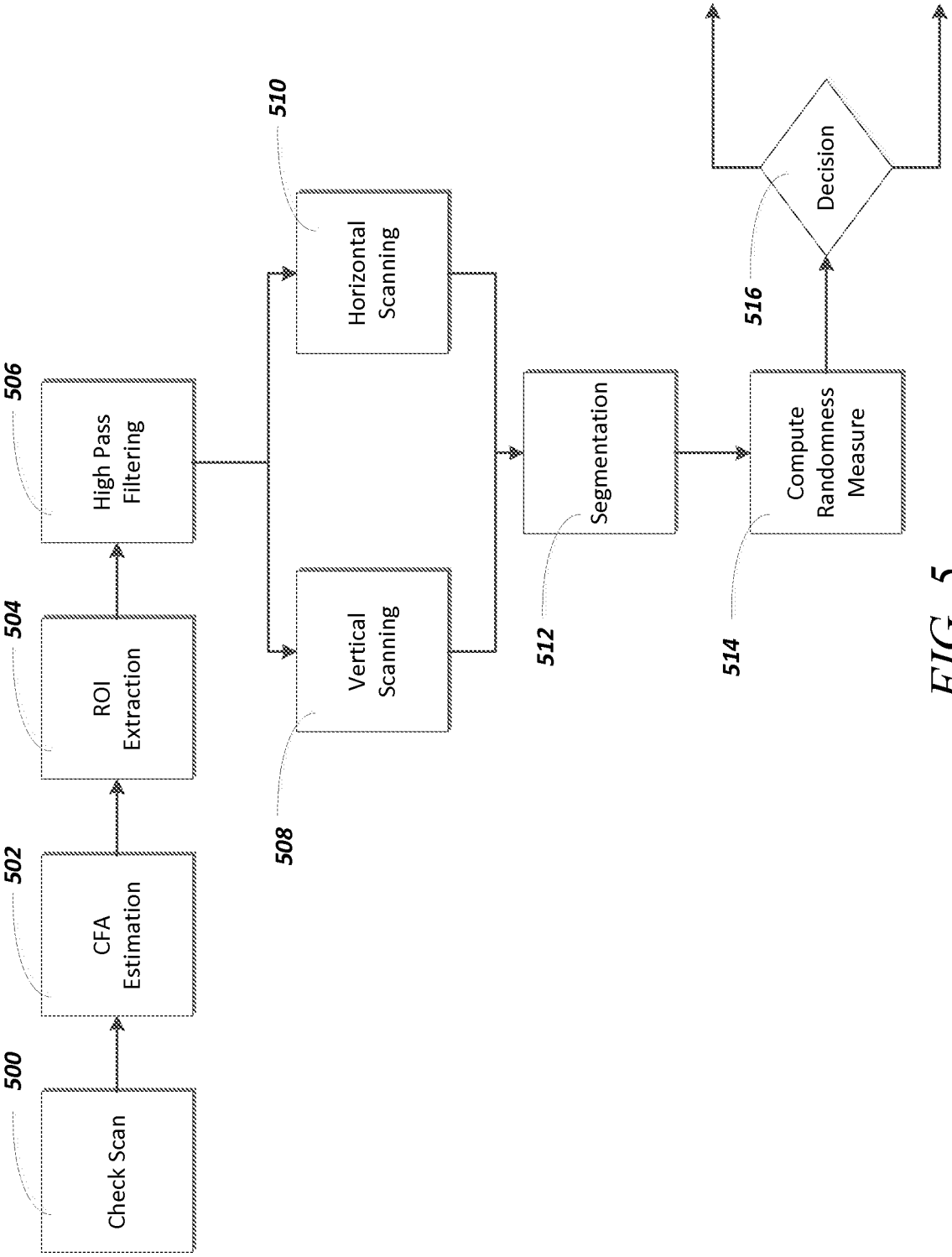
Check Scan
500
CFA Estimation
502
ROI Extraction
504
High Pass Filtering
506
Horizontal Scanning
510
Vertical Scanning
508
Segmentation
512
Compute Randomness Measure
514
Decision
516

FIG. 5

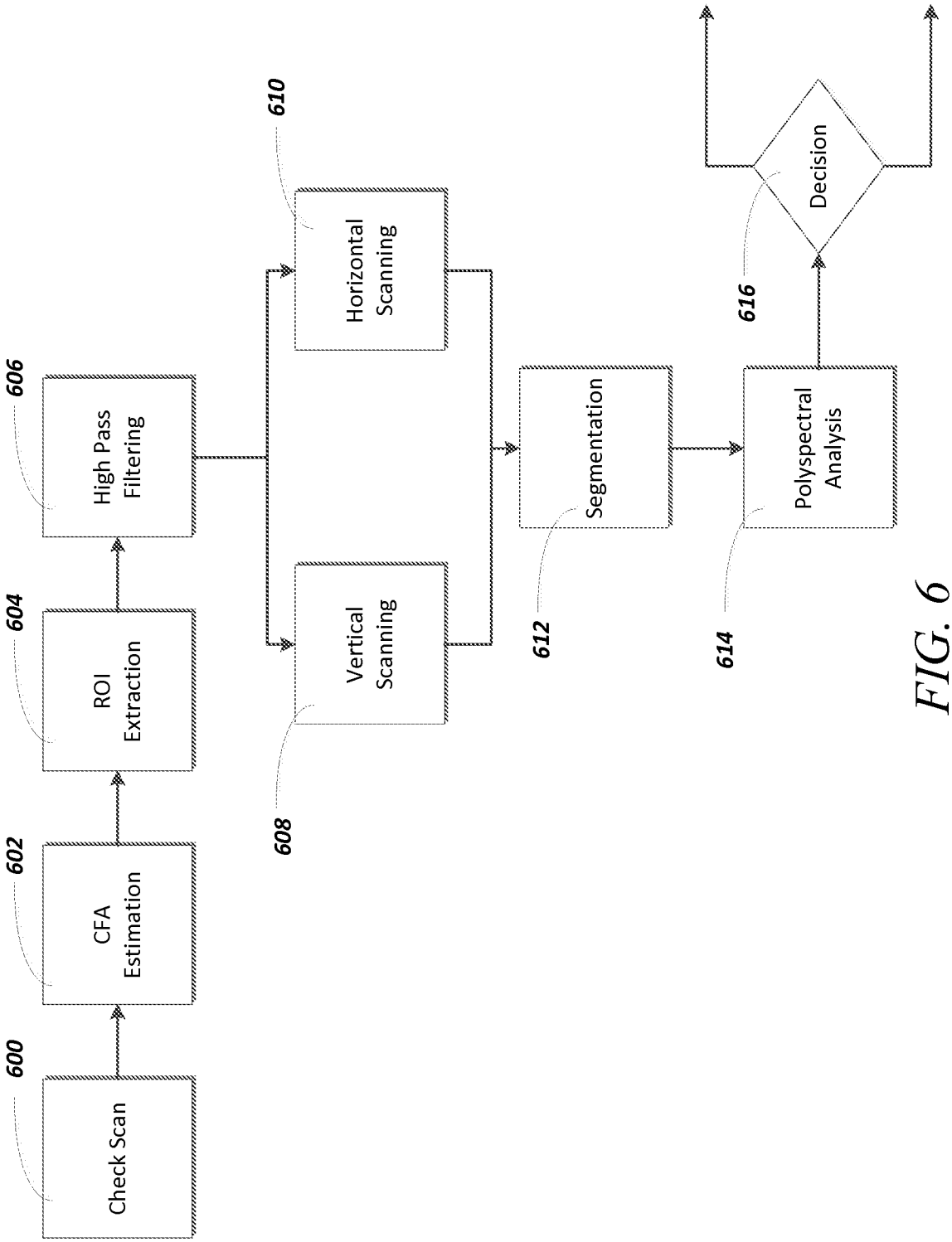
600
Check Scan
602
CFA Estimation
604
ROI Extraction
606
High Pass Filtering
608
Vertical Scanning
610
Horizontal Scanning
612
Segmentation
614
Polyspectral Analysis
616
Decision

FIG. 6

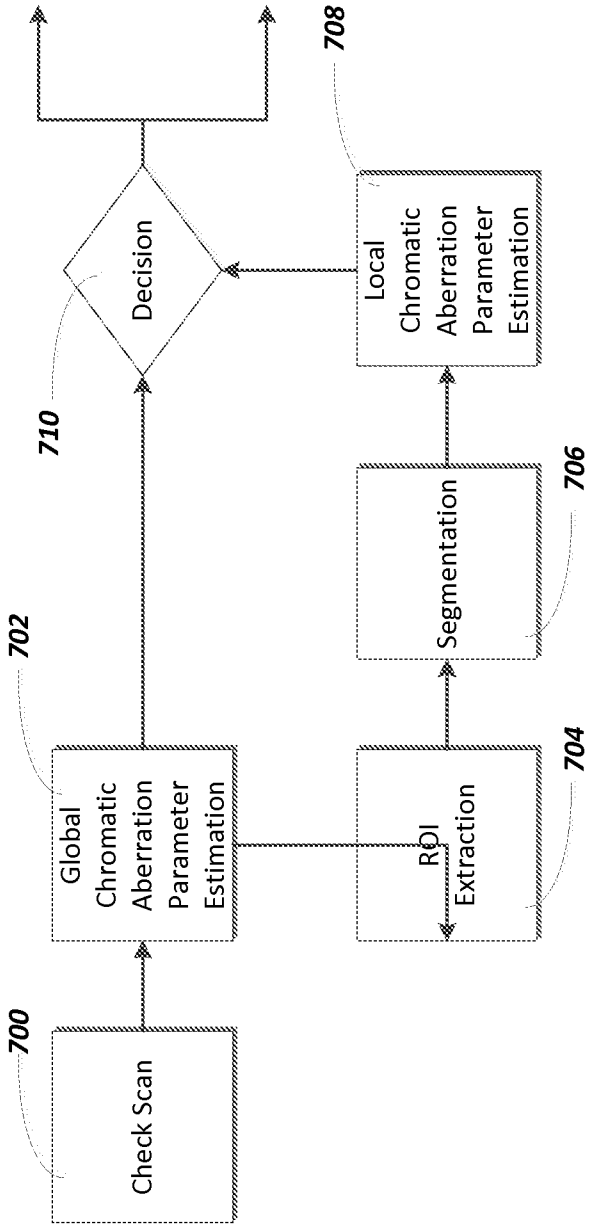
700
Check Scan
702
Global Chromatic Aberration Parameter Estimation
704
ROI Extraction
706
Segmentation
708
Local Chromatic Aberration Parameter Estimation
710
Decision

FIG. 7

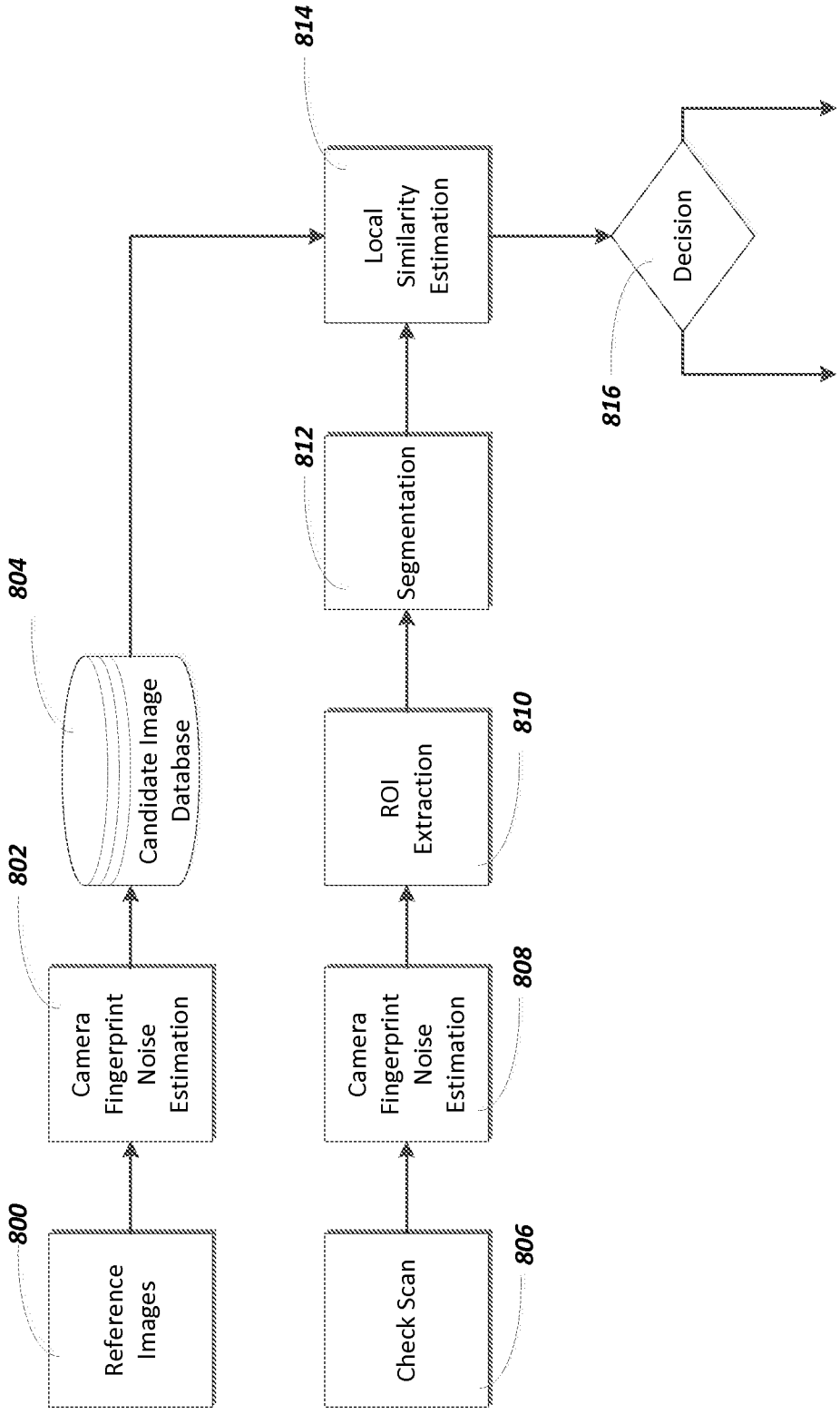
800
Reference Images
802
Camera Fingerprint Noise Estimation
804
Candidate Image Database
806
Check Scan
808
Camera Fingerprint Noise Estimation
810
ROI Extraction
812
Segmentation
814
Local Similarity Estimation
816
Decision

FIG. 8

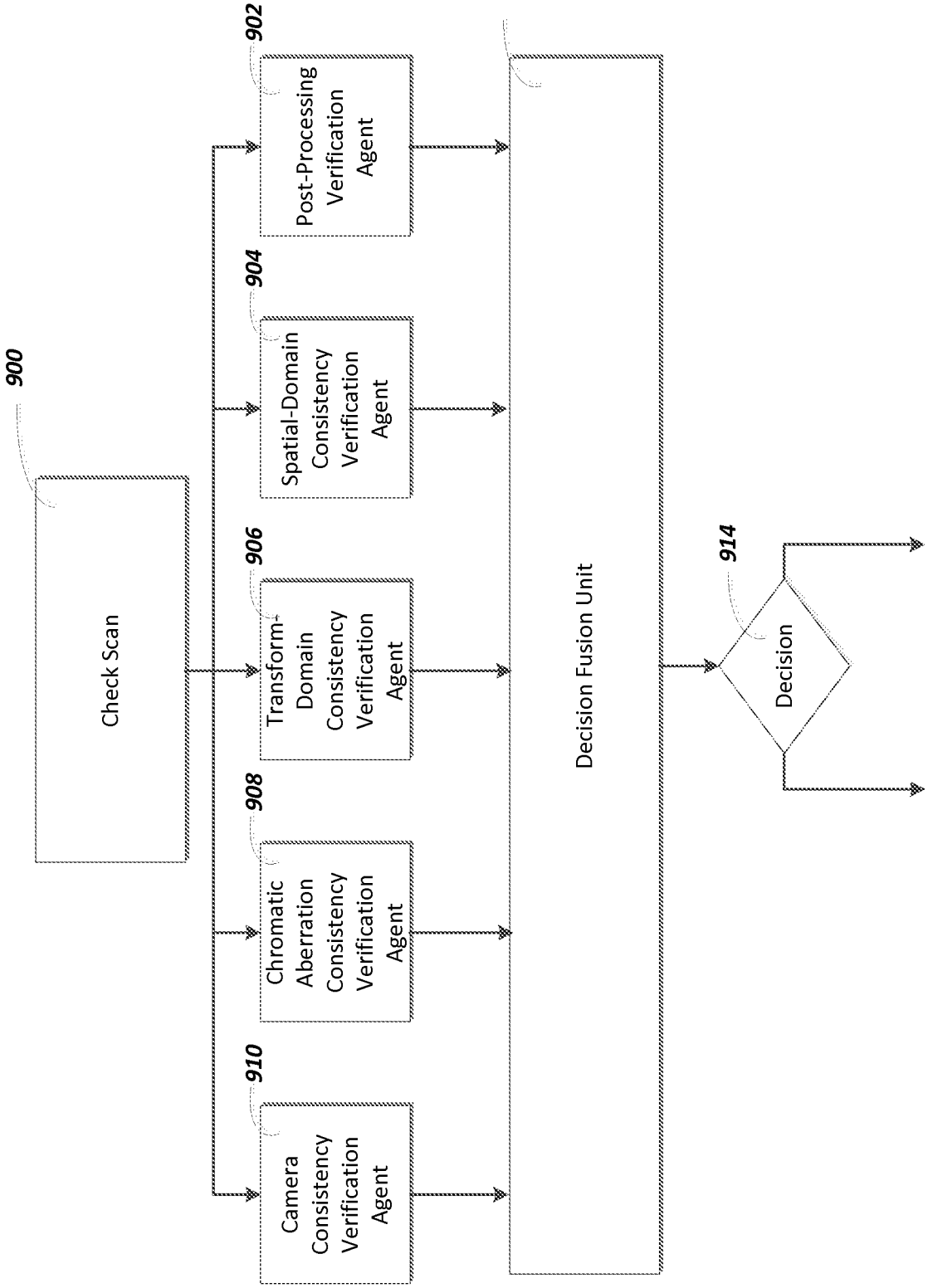
900
Check Scan
902
Post-Processing Verification Agent
904
Spatial-Domain Consistency Verification Agent
906
Transform-Domain Consistency Verification Agent
908
Chromatic Aberration Consistency Verification Agent
910
Camera Consistency Verification Agent
Decision Fusion Unit
914
Decision

FIG. 9