(12) STANDARD PATENT APPLICATION (11) Application No. **AU 2018256555 A1**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Image and feature quality, image enhancement and feature extraction for ocular-vascular and facial recognition, and fusing ocular-vascular with facial and/or sub-facial information for biometric systems

(51) International Patent Classification(s)
G06K 9/00 (2006.01) ***G06K 9/62*** (2006.01)
G06K 9/46 (2006.01)

(21) Application No: **2018256555** (22) Date of Filing: **2018.10.31**

(43) Publication Date: **2018.11.22**
(43) Publication Journal Date: **2018.11.22**

(62) Divisional of:
2016319775

(71) Applicant(s)
EyeVerify Inc.

(72) Inventor(s)
SARIPALLE, Sashi K.;GOTTEMUKKULA, Vikas;DERAKHSHANI, Reza R.

(74) Agent / Attorney
Spruson & Ferguson, GPO Box 3898, Sydney, NSW, 2001, AU

IMAGE AND FEATURE QUALITY, IMAGE ENHANCEMENT AND FEATURE EXTRACTION FOR OCULAR-VASCULAR AND FACIAL RECOGNITION, AND FUSING OCULAR-VASCULAR WITH FACIAL AND/OR SUB-FACIAL INFORMATION FOR BIOMETRIC SYSTEMS

ABSTRACT

Biometric enrollment and verification techniques for ocular-vascular, periocular, and facial regions are described. Periocular image regions can be defined based on the dimensions of an ocular region identified in an image of a facial region. Feature descriptors can be generated for interest points in the ocular and periocular regions using a combination of patterned histogram feature descriptors. Quality metrics for the regions can be determined based on region value scores calculated based on texture surrounding the interest points. A biometric matching process for calculating a match score based on the ocular and periocular regions can progressively include additional periocular regions to obtain a greater match confidence.

2018256555 31 Oct 2018

IMAGE AND FEATURE QUALITY, IMAGE ENHANCEMENT AND FEATURE EXTRACTION FOR OCULAR-VASCULAR AND FACIAL RECOGNITION, AND FUSING OCULAR-VASCULAR WITH FACIAL AND/OR SUB-FACIAL INFORMATION FOR BIOMETRIC SYSTEMS

Cross-Reference to Related Application

[0001] This application is a divisional application of Australian Patent Application No. 2016319775, filed 9 September, 2016, which claims priority to and the benefit of U.S. Provisional Patent Application 62/217,660, filed on 11 September, 2015, and entitled "Image Enhancement, Quality Metrics, Feature Extraction, Information Fusion, Pose Estimation and Compensation, and Template Updates for Biometric Systems," the entirety of which is incorporated by reference herein. The contents of Australian Patent Application No. 2016319775 and US Provisional Patent Application

Background

[0002] The present disclosure relates to biometric authentication and, more specifically, to systems and methods for image enhancement, quality metrics, feature extraction, and information fusion for multi-region (sub-ocular to face) or other image-based biometric systems.

[0003] Biometric systems can be used to authenticate the identity of an individual to either grant or deny access to a resource. For example, image scanners can be used by a biometric security system to identify an individual based on unique structures on the individual's face, for example, in the individual's eye and its surrounding regions. Biometric data captured from an individual, such as that captured during an enrollment process, can be stored as a template that is used to verify the identity of the individual at a later time. Biometric scanning technology would, however, benefit from improved techniques in using periocular features instead of or in addition to ocular features for enrollment and verification.

Brief Summary

[0004] It is an object of the present invention to meet this need or substantially overcome, or at least ameliorate, one or more disadvantages of existing arrangements.

[0005] Systems and methods relating to biometric techniques applicable to ocular-vascular, periocular, and facial regions are disclosed. In one aspect, a computer-implemented method comprises the steps of: receiving an image of a facial region of a user, the facial region including an eye and an area surrounding the eye; processing the image to define an ocular image region including at least a portion of the eye in the image of the facial region; defining a plurality of periocular image regions each including at least a portion of the area surrounding the eye in the image of the facial region, wherein the periocular regions are defined based on dimensions of the defined ocular region; calculating one or more biometric match scores based on the ocular image region and at least one of the periocular regions; and designating the image of the facial region as authentic or not authentic based on the one or more biometric match scores.

[0006] In one implementation, the plurality of periocular image regions comprises at least four periocular image regions. The at least four periocular image regions can comprise a periocular image region disposed below the ocular image region, a periocular image region disposed to the right of the ocular image region, a periocular image region disposed to the left of the ocular image region, and a periocular image region disposed above the ocular image region.

[0007] Defining the plurality of periocular image regions can comprise defining a lower periocular image region disposed below the ocular image region, the lower periocular image region having a width substantially equal to a width of the ocular image region width and a height in the range of 10% to 300% of a height of the ocular image region. Defining the plurality of periocular image regions can also comprise defining a right periocular image region disposed to the right of the ocular image region, the right periocular image region having a width in the range of 10% to 80% of a width of the ocular image region and a height of 120% to 550% of a height of the ocular image region. Defining the plurality of periocular image regions can further comprise defining a left periocular image region disposed to the left of the ocular image region, the left periocular image region having a width in the range of 10% to 50% of a width of the ocular image region and a height of 120% to 550% of a height of the ocular image region. Defining the plurality of periocular image regions can also comprise defining an upper periocular image region disposed above the ocular image region, the upper periocular image region having a width substantially equal to a width of the ocular image region width and a height in the range of 10% to 150% of a height of the ocular image region

[0008] In another implementation, calculating the one or more biometric match scores comprises: calculating a first biometric match score based on the ocular image region and an ocular enrollment template; and in response to determining that the first biometric match score does not meet a first match threshold, calculating a second biometric match score based on the ocular image region, a first one of the periocular image regions, the ocular enrollment template, and a periocular enrollment template. Calculating the one or more biometric match scores can further comprise, in response to determining that the second biometric match score does not meet a second match threshold, calculating one or more further biometric match scores by iteratively including additional ones of the periocular image regions in calculating the further biometric match scores until either a particular further biometric match score meets a corresponding match threshold or no further periocular image regions are available for inclusion. The plurality of periocular image regions can be ranked based at least on discriminative power and/or quality, and the additional periocular image regions can be iteratively included based on the respective rankings of the additional periocular image regions. One or more of the plurality of periocular image regions can be divided into sub-regions based on one or more facial features derived from the periocular image regions, and the additional periocular image regions can be iteratively included based on cluster importance or the respective rankings of the additional periocular image regions.

[0009] In a further implementation, calculating the one or more biometric match scores comprises: identifying a first set of pairs of matched points based on the ocular image region and an enrollment template; and identifying a second set of pairs of matched points based on at least one of the periocular image regions and the enrollment template. Calculating the one or more biometric match scores can further comprises: determining one or more inlier matched points by inputting a combination of the first and second sets of pairs of matched points into an outlier detection algorithm; determining that a number of the inlier matched points that correspond to the ocular image region meets a minimum ocular inlier count; and calculating a particular biometric match score based at least in part on the inlier matched points. The minimum ocular inlier count can be equal to 3.

[0010] In yet another implementation, calculating the one or more biometric match scores further comprises: determining one or more first inlier matched points by inputting the first set of pairs of matched points into an outlier detection algorithm; determining one or more second inlier matched points by inputting the second set of pairs of matched points into an outlier detection algorithm; and calculating a particular biometric match score based at least in part on an output of an outlier detection algorithm using a combination of the first and second inlier matched points as input. Calculating the one or more biometric match scores can further comprise determining that a number of inlier matched points, obtained from the output of the outlier detection algorithm, that correspond to the ocular image region meets a minimum ocular inlier count. The minimum ocular inlier count can be equal to 3.

[0011] In another aspect, a computer-implemented method comprises the steps of: receiving an image of a facial region of a user, the facial region including an eye and an area surrounding the eye; processing the image to: (i) define an ocular image region including at least a portion of the eye in the image of the facial region and (ii) define one or more periocular image regions each including at least a portion of the area surrounding the eye in the image of the facial region; identifying a plurality of points of interest in at least one of the ocular image region and the one or more periocular image regions; generating, for each point of interest, a feature descriptor based on a combination of a plurality of patterned histogram feature descriptors; and storing the generated feature descriptors in a biometric template.

[0012] In one implementation, processing the image comprises enhancing at least a portion of the image using local gradient Gabor pattern (LGGP). Enhancing at least a portion of the image using LGGP can comprise: calculating, at each of a plurality of angles, a Gabor phase image for the at least a portion of the image; aggregating the calculated Gabor phase images to form a combined Gabor phase image; calculating, at each of a plurality of angles, a local gradient of the combined Gabor phase image; and retaining a maximum value of each local gradient to form an enhanced image.

[0013] A particular patterned histogram feature descriptor can comprise a patterned histogram of extended multi-radii local binary patterns (PH-EMR-LBP), a patterned histogram of extended multi-radii center symmetric local binary patterns (PH-EMR-CSLBP), or a patterned histogram of extended multi-radii local ternary patterns (PH-EMR-LTP). Generating the feature descriptor for a particular point of interest can comprise generating a PH-EMR-LBP feature descriptor by: defining an image region comprising the particular point of interest; calculating a plurality of local binary pattern (LBP) codes for each pixel in the image region to form a multi-radii LBP (MR-LBP) image; dividing the MR-LBP image into a plurality of sub-regions; deriving a plurality of histograms comprising frequencies of each MR-LBP bit location within each sub-region; and combining the plurality of histograms to form the PH-EMR-LBP feature descriptor. Generating the feature descriptor for a particular point of interest can comprise generating a PH-EMR-CSLBP feature descriptor by: defining an image region comprising the particular point of interest; calculating a plurality of center symmetric local binary pattern (CSLBP) codes for each pixel in the image region to form a multi-radii CSLBP (MR-CSLBP) image; dividing the MR-LBP image into a plurality of sub-regions; deriving a plurality of histograms comprising frequencies of each MR-CSLBP bit location within each sub-region; and combining the plurality of histograms to form the PH-EMR-CSLBP feature descriptor. Generating the feature descriptor for a particular point of interest can comprise generating a PH-EMR-LTP feature descriptor by: defining an image region comprising the particular point of interest; calculating a plurality of local ternary pattern (LTP) codes for each pixel in the image region to form a multi-radii LTP (MR-LTP) image; dividing the MR-LBP image into a plurality of sub-regions; deriving a plurality of histograms comprising frequencies of each MR-LTP bit location within each sub-region; and combining the plurality of histograms to form the PH-EMR-LTP feature descriptor.

[0014] In another implementation, generating the feature descriptor for a particular point of interest comprises: calculating a PH-EMR-LBP feature descriptor, PH-EMR-CS-LBP feature descriptor, and PH-EMR-LTP feature descriptor for the particular point of interest; and combining the PH-EMR-LBP feature descriptor, PH-EMR-CS-LBP feature descriptor, and PH-EMR-LTP feature descriptor to form a combined feature descriptor. Generating the feature descriptor for a particular point of interest can further comprise applying a variance analysis to the combined feature descriptor to form a top feature descriptor comprising a subset of features from the combined feature descriptor. Generating the feature descriptor for a particular point of interest can further comprise performing dimensionality reduction on the top feature descriptor to form the feature descriptor for the particular point of interest.

[0015] In a further implementation, in generating a particular patterned histogram feature descriptor, local binary pattern or local ternary pattern feature descriptors are calculated using a neighborhood defined by a square with or without corners.

[0016] In another aspect, a computer-implemented method comprises the steps of: receiving an image of a facial region of a user, the facial region including an eye and an area surrounding the eye; defining an ocular image region including at least a portion of the eye in the image of the facial region; defining one or more periocular image regions each including at least a portion of the area surrounding the eye in the image of the facial region; identifying a plurality of points of interest in at least one of the ocular image region and the one or more periocular image regions; calculating, for each point of interest, a region value for texture surrounding the point of interest; and determining at least one quality metric for at least a portion of the image of the facial region based on the points of interest and the respective calculated region values.

[0017] In one implementation, calculating the region value for a particular point of interest comprises: calculating at least one local binary pattern in a square-shaped neighborhood (BP) for the particular point of interest; and calculating at least one BP for one or more points offset from the particular point of interest. Calculating the region value for the particular point of interest can further comprise setting the region value to an average of region values calculated for the particular point of interest and a plurality of the offset points. Calculating at least one BP for the particular point of interest can comprise calculating a plurality of BPs, each having a different neighborhood, for the particular point of interest, and calculating at least one BP for the offset points can comprise calculating a plurality of BPs, each having a different neighborhood, for each offset point. Calculating the plurality of BPs for a particular point of interest or offset point can comprise: reducing the plurality of BPs to a Noisy Binary Pattern (NBP); and creating a general binary pattern (genBP) from the NBP. Calculating the plurality of BPs for a particular point of interest or offset point can further comprise: creating a weighted pattern H from the genBP; and calculating a region value for the particular point of interest or offset point as:

$$\frac{(L * \sum H)}{144}$$

where L comprises a maximum length of continuous zeros in the genBP. The offset points can comprise a plurality of pixel locations uniformly shifted in different directions from the particular point of interest.

[0018] In another implementation, determining the quality metric comprises: creating an ordered list of the points of interest based on respective region values of the points of interest; and calculating distances between consecutive points of interest in the ordered list. Determining the quality metric can further comprise calculating the quality metric as:

$$\sum_{n=1}^{p} s_n * sw_n * dw_n$$

where p comprises the number of points of interest, s_n comprises the region value calculated for point of interest n, sw_n comprises a weighted index for point of interest n, and dw_n comprises a weight for the distance corresponding to point n in the ordered list.

[0019] In a further implementation, the at least a portion of the image of the facial region comprises the ocular region or at least one of the periocular regions. The method can further comprise ranking the periocular image regions based on respective quality metrics separately calculated for each of the periocular image regions. The method can further comprise ranking the periocular image regions based on at least one of a respective discriminative power of each periocular image region. The method can further comprise using the periocular image regions in a progressive biometric matcher based at least in part on the respective quality metric and/or discriminative power rankings of the periocular image regions.

[0020] In yet another implementation, the method further comprises: determining that a difference in calculated quality metrics between a first one of the ocular or periocular image regions and a second one of the ocular or periocular image regions exceeds a threshold; and indicating a likely presence of a spoof based on the determined difference in calculated quality metrics.

[0021] In another aspect there is provided a computer-implemented method comprising: receiving an image of a facial region of a user, the facial region including an eye and an area surrounding the eye; processing the image to: (i) define an ocular image region including at least a portion of the eye in the image of the facial region and (ii) define one or more periocular image regions each including at least a portion of the area surrounding the eye in the image of the facial region, wherein processing the image comprises enhancing at least a portion of the image using local gradient Gabor pattern (LGGP) by: calculating, at each of a plurality of angles, a Gabor phase image for at least the portion of the image; aggregating the calculated Gabor phase images to form a combined Gabor phase image; calculating, at each of a plurality of angles, a local gradient of the combined Gabor phase image; and retaining a maximum value of each local gradient to form an enhanced image; identifying a plurality of points of interest in at least one of the ocular image region and the one or more periocular image regions; generating, for each point of interest, a feature descriptor based on a combination of a plurality of patterned histogram feature descriptors; and storing the generated feature descriptors in a biometric template.

[0022] In another aspect there is provided a system comprising: at least one memory for storing computer-executable instructions; and at least one processing unit for executing the instructions stored on the at least one memory, wherein execution of the instructions programs the at least one processing unit to perform operations comprising: receiving an image of a facial region of a user, the facial region including an eye and an area surrounding the eye; processing the image to: (i) define an ocular image region including at least a portion of the eye in the image of the facial region and (ii) define one or more periocular image regions each including at least a portion of the area surrounding the eye in the image of the facial region, wherein processing the image comprises enhancing at least a portion of the image using local gradient Gabor pattern (LGGP) by: calculating, at each of a plurality of angles, a Gabor phase image for at least the portion of the image; aggregating the calculated Gabor phase images to form a combined Gabor phase image; calculating, at each of a plurality of angles, a local gradient of the combined Gabor phase image; and retaining a maximum value of each local gradient to form an enhanced image; identifying a plurality of points of interest in at least one of the ocular image region and the one or more periocular image regions; generating, for each point of interest, a feature descriptor based on a combination of a plurality of patterned histogram feature descriptors; and storing the generated feature descriptors in a biometric template.

[0023] Other aspects of the above include corresponding systems and non-transitory computer-readable media. The details of one or more implementations of the subject matter described in the present specification are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

Brief Description of the Drawings

[0024] In the drawings, like reference characters generally refer to the same parts throughout the different views. Also, the drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the implementations. In the following description, various implementations are described with reference to the following drawings, in which:

[0025] FIG. 1 depicts an example method for creating an ocular and periocular enrollment template, according to an implementation.

[0026] FIG. 2A and 2B depict an example method for defining ocular and periocular regions, and corresponding facial images, according to an implementation.

[0027] FIG. 3 depicts an example modified local binary pattern (genBP) process, according to an implementation.

[0028] FIG. 4 depicts an example technique for biometric progressive matching with enrollment and verification phases, according to an implementation.

[0029] FIG. 5 depicts an example system for performing biometric scanning and analysis, according to an implementation.

Detailed Description

[0030] The white of the eye includes complex patterns (mostly due to vascular structures), which are not only readily visible and scannable, but are also unique to each individual. In some cases, other non-vascular formations may also be visible and can be taken into account by a biometric matching algorithm. Thus, these visible structures seen on the white of the eye, mostly due to vasculature of the conjunctiva and episclera, can be scanned and advantageously used as a biometric. This biometric can be used to authenticate a particular individual or identify an unknown individual from a larger set of candidates. Implementations of solutions for imaging and pattern matching the blood vessels in the white of the eye and for vascular point detection, feature extraction and matching are described, for example, in U.S. Patent No. 8,369,595, issued on February 5, 2013, and entitled "Texture Features for Biometric Authentication," and U.S. Patent No. 9,390,327, issued on July 12, 2016, and entitled "Feature Extraction and Matching for Biometric Authentication," the entireties of which are incorporated by reference herein. To add robustness to these and other biometric verification techniques, described here are methods for progressively including visible non-ocular-vascular structures outside the white of the eye, such as periocular or partial or even complete face or other visible identifiable patterns, into an ocular-vascular biometric matching scheme in the event, for example, that additional information is needed to form a more accurate analysis.

[0031] Ocular features are those arising from the patterns seen external to the corneal limbus and bounded by the eyelids, which are mostly due to vascular patterns seen on the white of the eyes. Henceforth, such features are interchangeably referred to herein as ocular or ocular-vascular. The white of the eye has a number of layers. The sclera is an opaque, fibrous, protective, layer of the eye containing collagen and elastic fiber. The sclera is covered by the episclera, which has a particularly large number of blood vessels and veins that run through and over it. The episclera is covered by the bulbar conjunctiva, which is a thin clear membrane that interfaces with the eyelid or the environment when the eyelid is opened, which also contains unique and rich vascular structures. Blood vessels run through all of these layers of the white of the eye and can be detected in images of the eye. The eye also includes eyelashes and sometimes drooping eyelids that can sometimes obscure portions of the white of the eye in an image.

[0032] Periocular refers to the immediate region of the face surrounding the eye (external to and, in some instances, abutting the eyelid edges), which is textured and can be used for biometric applications. The periocular region can include one or more regions that can be, but are not necessarily, contiguous, and that can take various forms with respect to shape and size. While the periocular region has no hard-defined boundaries, standardizing the region of interest can minimize registration issues, further improving the accuracy of biometric system. Such standardization can be achieved using certain measurable value from the user face, such as distance between eye corners, radius of iris, inter-ocular distance and/or width and height of the derived eye crop (while using eye detection algorithms). During user authentication, one or more digital images of a user’s eye and periocular region are captured, one or more verification templates are generated from the captured image or images, and the identity of the user can be verified by matching the corresponding ocular and periocular structures as expressed in the enrollment and verification templates.

[0033] In other implementations, additional regions of the face extending beyond the periocular region can be used in addition to or instead of regions immediately surrounding the eye. For example, when heavy facial expressions are observed, one or more regions on and/or around the nose or on/around the face can be considered as extended periocular regions. In one implementation, facial features can complement ocular features for an eye-based biometric matcher. In another implementation, the ocular features used by an eye-based biometric matcher are complemented by both periocular and extended facial features. In another implementation, an entire face can be used in addition to or instead of the regions immediately surround the eye. The face can complement ocular features for an eye-based biometric matcher.

Both periocular and face regions can complement the ocular features used by an eye-based biometric matcher.

[0034] In various other implementations, other information such as the skin tone of the periocular region can also be measured and used as a soft biometric. Skin tone can be estimated using histograms of the intensity values of the different color bands of image, such as normalized red, green, and blue (RGB) channels. During verification, the histogram information of the enrollment template can be used to stop a verification process when the distance between histograms of enrollment and verification images are higher than a threshold.

[0035] In one implementation, skin tone can be estimated by a clustering technique that identifies the most prominent colors. As one example, the RGB pixels of the periocular region are converted to Lab color space and clustered to N number of classes. The optimal N can be determined by AIC (Akaike Information Criterion) or BIC (Bayesian Information Criterion). Next, these clusters are sorted in ascending order based on their occurrence, and the top three clusters are grouped with the pre-defined skin tones. The closest skin tone can be determined with a distance measure and/or correlation measure. In one implementation, the distance measure is Euclidean distance, although other distance measures are contemplated. Next, the verification procedure can be stopped if the distance and/or correlation between the enrollment and verification image are higher than a threshold. The same process can be applied to other colored parts of the image, such as the iris.

[0036] FIG. 1 depicts an example method for creating an ocular and periocular enrollment template. In step 102, one or more images of a user’s facial region(s) are captured using an image sensor, e.g., a camera, that can be associated with a device to which access by the user is to be authenticated (e.g., a smartphone, a smart watch, smart glasses, a notebook computer, a tablet computer, etc.). By way of illustration, the camera can be a digital camera, a three-dimensional (3D) camera, a light field sensor and/or a near infrared sensor or other monochromatic and/or multispectral imaging sensor. The images can include one or more ocular regions (regions defining an eye area of the user) and/or one or more periocular regions, as well as other facial regions, in some instances. The images can be captured either in still mode or in video mode or a combination thereof. The images can be captured in various wavelengths. In some implementations, the user is prompted (by visual, or audible, or haptic feedback) to move the device closer/farther to trigger the optimal distance for finding eyes and/or face.

[0037] In step 104, an initial region of interest (ROI) is identified, e.g., one or both eyes may be located. Viola-Jones algorithms or similar learning algorithms trained on various different forward-looking eye images can be used for this purpose. In another implementation, Viola-Jones algorithms trained on various different glancing eye images can be used. Thereafter, a gaze tracking algorithm, such as one using Haar filters, can be used to quantize the amount of gaze to acquire one or more images, once a selected gaze direction is detected.

[0038] Once the initial ROI is localized on the face shown in the captured image(s), additional computation can be performed to obtain the final ROI on the image. Typically, the final ROIs are cropped to obtain RGB and/or near-infrared images of one or more eyes. Accordingly, as used herein, "image" or "captured image" can also refer to a cropped image of a final ROI.

[0039] With reference to FIG. 1, in the Ocular and Periocular Stages, one or more averaged or otherwise enhanced or naturally captured images, which can be enrollment images or verification images, are preprocessed in image enhancement sub-stage 106 and image quality sub-stage 108, and features from the preprocessed images are extracted in feature extraction sub-stage 110, to generate one or more ocular and periocular templates. The preprocessing sub-stages 106 and 108 can include image enhancement and image quality techniques, explained further below. The feature extraction sub-stage 110 can include interest point detection and local descriptor extraction, which are described in detail below. The operations performed during the sub-stages 106, 108, and 110 in the Ocular and Periocular Stages need not be the same. Rather, different operations that account of the particular features of ocular and periocular regions can be used in those respective stages. The enrollment ocular and/or periocular templates, prior to storing, can be encrypted or otherwise protected at step 112.

Periocular Boundaries

[0040] Referring now to FIGS. 2A and 2B, periocular regions can be extracted using various methods. On acquiring an image of a user's face (step 200), one or more eye regions are detected (step 202). Based on the size of eye crops identified in step 202, the eye region size can be recalculated (step 204). Proceeding to the ocular stage, in one implementation, an integro-differential algorithm, Hough circles or Hessian blob detector is used to detect the iris boundary (segmentation of iris region in step 208). Similarly, a Gabor filtering based algorithm can be used to detect the upper and lower eyelid, and further whites of the eye can be separated after removing the iris region (segmentation of scleral region in step 206). The periocular region can be derived after subtracting both white of the eye and iris regions from the captured images (Step 210). In some implementations, color-based methods are used to segment the above-mentioned regions.

[0041] In one implementation, as shown by the periocular region stage 212 in FIGS. 2A and 2B, periocular regions around the final ocular ROI of size width W x height H (where W and H are the number of pixels in the horizontal and vertical direction, respectively) can be defined as described below. The ocular ROI is not necessarily a rectangular shape; rather, the ROI can take a form similar to the shape of the eye, as shown in FIG. 2B, and the W and H can be defined as the width and height, respectively, of a bounding box encompassing the ocular ROI. There can be various numbers of regions; however, the implementation described below includes four periocular regions (1 through 4) ranked by discriminative power, which essentially refers to how useful the region is in a biometric matching operation. More specifically, periocular region 1 represents a facial region with more distinctive, or discriminatory, characteristics than region 2, which has greater discriminative power compared to region 3, and so on. Various periocular region sizes are contemplated (here, defined by a bounding box of width W x height H). The ranges and values of the periocular regions disclosed herein were experimentally determined based on large scale testing of which periocular regions have sufficient discriminative power for the purposes of performing biometric matching operations. As used below, the term "substantially equal" means equal to a particular value or within +/- 10% of the value.

[0042] In one implementation, the four periocular regions are defined as follows, where W and H refer to the width and height, respectively, of the ocular region:

- Region 1:
  - ○ Width is substantially equal to W.
  - ○ Height is in the range of 0.1*H to 3*H below the eye crop of the ocular region, inclusive. In one implementation, height is substantially equal to 0.3*H.

- Region 2:
  - Width is in the range of 0.1*W to 0.8*W, inclusive. In one implementation, width is substantially equal to 0.2*W.
  - Height is in the range of 1.2*H (0.1*H above the eye crop of the ocular region and 0.1*H below the eye crop) to 5.5*H (1.5*H above the eye crop and 3*H below the eye crop), inclusive. In one implementation, height is substantially equal to 1.45*H (0.15*H above the eye crop and 0.3*H below the eye crop).
- Region 3:
  - Width is in the range of 0.1*W to 0.5*W, inclusive. In one implementation, width is substantially equal to 0.15*W.
  - Height is in the range of 1.2*H (0.1*H above the eye crop of the ocular region and 0.1*H below the eye crop) to 5.5*H (1.5*H above the eye crop and 3*H below the eye crop), inclusive. In one implementation, height is substantially equal to 1.45*H (0.15*H above the eye crop and 0.3*H below the eye crop).
- Region 4:
  - Width is substantially equal to W.
  - Height is in the range of 0.1*H to 1.5*H above the eye crop of the ocular region, inclusive. In one implementation, height is substantially equal to 0.15*H.

[0043] Other non-rectangular or even non-contiguous periocular (or other facial) sub-regions can be used based on the differentiability and reliability of those region(s) per individual, certain population, or all of the user(s). In some implementations, when drooping eyelids are often observed, an optimal periocular region around the tight crop of size WxH can be defined as follows:

- Widths of regions 2 and 3 are substantially equal to 0.2*W and 0.15*W, respectively.
- Heights of regions 2 and 3 are substantially equal to 1.45*H each (which starts from the top of the eye crop).
- Widths of regions 1 and 4 are substantially equal to W.
- Heights of regions 1 and 4 are substantially equal to 0.45*H and 0.1*H, respectively.

[0044] The range of values for heights and widths for various regions are same as explained above. Other periocular region sizes are contemplated. The values of W and H can vary based on imaging sensor resolution.

Image Enhancement

[0045] Image enhancement techniques, such as those performed in step 106 of FIG. 1, will now be explained. The images that are captured from an image sensor can have varying quality due to, for instance, exposure and motion blur artifacts. In step 106, which is optional, several images can be registered (i.e., spatially aligned) and averaged to reduce image noise. In some implementations, image correlation methods are used to measure the dissimilarity between the obtained images in order to align the images for averaging and discard those that are most different (e.g. due to motion blur or eye blink) and, hence, are not suitable for registration and averaging. For example, n consecutive frames with minimal motion can be averaged after registration. The number of consecutive frames (disregarding the occasional dropped frames explained above) that are averaged can depend on the frame rate, noise level of the image sensor at the given settings, environment conditions in which images are being captured.

[0046] In one example, if the ambient light is greater than 800 lumens during the images acquisition, two consecutive frames are registered and averaged. If the ambient light is in the range of 450-800 lumens, 3 consecutive frames are registered and averaged. If the ambient light is in the range of 0-450 lumens, 4 consecutive frames are registered and averaged. Other combinations of selecting the number of frames based on ambient light are possible for whites of the eye, periocular region and face, including non-averaging option if the region of interest is bright enough for the sensor noise to be negligible at the given settings.

[0047] In another implementation, the number of frames used to average can also be adapted to the motion artifacts across consecutive frames. For example, the motion may be caused due to changing expressions, eye and body movements and/or environmental aberrations. Motion can be measured using correlation across frames, mutual information across frames, etc. In one implementation, the number of frames used to average relies on both environmental lighting and observed motion artifacts.

[0048] In some implementations, image enhancements algorithms include Self Quotient Image, or SQI. A simple version of SQI image (Q) can be represented as follows:

$$Q = I/\hat{I} = I/(F*I)$$

where, Î is the smoothed version of I, and F is the smoothing kernel. SQI method is known for its lighting invariance property, which is useful for periocular regions with variable illumination due to shadows and other lighting conditions. Other image enhancements such as Weber Linear Descriptors (WLD), Coherence Diffuse Filters, Atrous Wavelets, Retinex (and its variants), single or multi frame bilateral filtering, deep learning methods and other sparse enhancement can be used.

[0049] In one implementation, local gradient Gabor pattern (LGGP) can be used as an image enhancement technique for a particular image or image region, as described in the following steps.

[0050] Step 1: Calculate the phase of Gabor for the green channel of the image region using even and odd Gabor responses as follows:

[0051] Step 1a: A 2D even Gabor kernel, similar to band-pass filters, is a cosine function modulated by 2D Gaussian envelope, and 2D odd Gabor is a sinusoidal function modulated by 2D Gaussian envelope. The even and odd Gabor kernels can be derived as follows:

$$G_{even}(x, y, f, \emptyset) = \ exp\left\{\frac{-1}{2}\left[\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right]\right\} \cos(2\pi f x')$$

$$G_{odd}(x, y, f, \emptyset) = \ exp\left\{\frac{-1}{2}\left[\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right]\right\} \sin(2\pi f x')$$

Any scale and orientation of Gabor kernel can be achieved through the following coordinate transformation:

$$x' = xcos(\emptyset) + ysin(\emptyset)$$

$$y' = -xsin(\emptyset) + ycos(\emptyset)$$

where σ_x and σ_y define spread of the Gaussian envelope along x and y-axes respectively, f is the frequency of the modulating sine or cosine, and Ø is orientation of the kernel. In one implementation, frequency is set to 6 and spread of Gaussian along x and y-axes is set to 2.5. In one implementation the choice of orientations is pruned to 6, ranging from 0 to 5 π/6, each π/6 radians apart.

[0052] Step 1b: The phase of Gabor image is calculated as follows:

$$I_{Phase}(\emptyset) = tan^{-1}\left(\frac{G_{odd}(\emptyset) \otimes Image}{G_{even}(\emptyset) \otimes Image}\right)$$

where $\otimes$ is the convolution operator. At any given orientation, as an example, $I_{Phase}(\emptyset = 0)$ is an image of size equal to the size of the green channel of the image region.

[0053] Step 2: The phase of Gabor responses (images) at various orientations are aggregated to form an output image. Three example methods for achieving this are: (1) retain the maximum value of the intensity of a pixel at a given location (x, y) across all the orientations; (2) set the intensity of a pixel at a given location (x, y) to the weighted average of responses across all the orientations at that particular location; and (3) multiply the maximum value of the intensity of a pixel at a given location (x, y) across all the orientations with the original image, followed by normalization.

[0054] Step 3: The local gradients in four directions (0, 45, 90, and 135 degrees) are then calculated for the aggregated responses and the maximum gradient value is retained to form an output image.

[0055] In one implementation, color normalization such as white balancing using a gray world algorithm can be used before the aforementioned enhancement techniques.

[0056] In one implementation, additional processing of the enhanced image includes image histogram and contrast adjustments such as Contrast Limited Adaptive Histogram Equalization (CLAHE). CLAHE generally operates in small regions of the image referred to as tiles. Typically, each tile's contrast is enhanced such that the histogram of the output approximately matches the histogram specified by a particular distribution (e.g., uniform, exponential, or Rayleigh distribution). The neighboring tiles are then combined using an interpolation (e.g. bilinear interpolation) to eliminate any artificially induced boundaries. In some implementations, selecting a linear or nonlinear combination of the red, green, or blue color components that have the best contrast between the ocular vessels or periocular features and the background can enhance the image region. For example, the green component can be preferred in a RGB image of the ocular-vascular, because it can provide a better contrast between vessels and the background.

Image and Feature Quality Metrics

[0057] Referring back to step 108 in FIG. 1, the quality of each enhanced image can be measured, and those that meet a certain quality threshold are retained for further processing. Such quality can be measured jointly or separately for the ocular and periocular region. An image quality metric can also act as a match-predictive quality metric (measure of the respective biometric sample's ability to match) and, in some scenarios, can be fused into a final match score to improve biometric system performance.

[0058] In one implementation, detected interest points in an image can determine the quality of the image. One point based quality metric, which is a no-reference image quality metric, will be referred to herein as EV_QM. EV_QM can be calculated using three steps: Interest Point Detection, Region Value Calculation, and Quality Metric Score Generation. Interest points can be detected using vascular point detection (VPD), explained further below. Region value refers to the amount of texture around each interest point. The texture around each interest point can be determined using Local Ternary Pattern (LTP). Quality metric score generation is measured using corner location and region value.

[0059] Referring to FIG. 3, in one implementation, region value can be calculated using the following modified local binary pattern (LBP) process. Three squares (with or without corners) with corresponding different half-lengths (neighborhoods) are calculated around an interest point, with their pixel loci computed (as depicted in pixel grid 304). For example, these half-lengths can be 2, 4, 6, or other number of pixels. The interest point can be further offset (shifted) in 8 different locations (as depicted in pixel grid 302), further creating 3 squares (minus their corners) around each offset point (similar to that depicted in pixel grid 304, with the offset point serving as the center point). LBP for each interest and offset points are calculated across each of the three squares, which are further referred to as Binary Patterns (BP). Thus, each interest point and corresponding offset points have three different BPs associated with them.

[0060] The three BPs can be further reduced to obtain a final region value (texture score) as follows:

[0061] Step 1: Each BP is evaluated to determine the uniformity score. If the BP is not uniform (e.g., more than 4 bits change or fewer than 2 bits change), then the BP is discarded and the score (final region value) across the corresponding offset or interest point is set to zero.

[0062] Step 2: If all of the BPs are uniform, a filtering process, as graphically depicted in step 306 of FIG. 3 is performed. Each element in BP1, BP2, and BP3 is added using the following formula:

$$Result_n = XOR(BP1_n, BP2_n) + XOR(BP2_n, BP3_n)$$

There are three possible values (1, 0, and 2) that can be seen in each element of $Result_n$. Each of these bits is further mapped according to the corresponding values (-1, 0, and 1, respectively) in *Value*, as shown in step 308. This result is further referred to as *Noisy Binary Pattern (NBP)*

[0063] Step 3: If there are more than four noisy pixels, the score for the corresponding interest point is set to zero. Otherwise, each noisy pixel in the *NBP* is replaced with its closest neighbor.

[0064] Step 4: The final result is a single binary pattern with length of BP for half-length 2. This final binary pattern is further referred to as *genBP*.

[0065] Step 5: *genBP* is further weighted based on formulae described in step 310:

$$H = \begin{cases} 0 & if\ \mathrm{XOR}(genBP_{n-1} + genBP_n) + \mathrm{XOR}(genBP_n + genBP_{n+1}) = 2 \\ 0.5 & if\ \mathrm{XOR}(genBP_{n-1} + genBP_n) + \mathrm{XOR}(genBP_n + genBP_{n+1}) = 1 \\ 1 & if\ \mathrm{XOR}(genBP_{n-1} + genBP_n) + \mathrm{XOR}(genBP_n + genBP_{n+1}) = 0 \end{cases}$$

The result is further referred to as weighed pattern, or *H*.

[0066] Step 6: Calculate the maximum length of continuous zeros in *genBP* and let it further be referred to as *L*.

[0067] Step 7: The final region value s_n can be calculated using the formula described in step 312:

$$s_n = \frac{\{L * \mathrm{sum}(H)\}}{144}$$

[0068] Step 8: If at least three of the points including the interest point and corresponding offset points compute a region value, the region values of the interest point and those offset corresponding points computing a region value are averaged and assigned to s_n. Otherwise, s_n is set to zero.

[0069] In another implementation, region value can be calculated using entropy of the region surrounding the interest points and the corresponding offset points around each interest point. The entropy of the region around interest point is calculated as follows:

[0070] Step 1: Initialize a threshold to quantize the texture in the region of interest around the interest point and corresponding offset points.

[0071] Step 2: Determine the number of levels (level_map) in the region of interest by subtracting the center pixel intensity with all the other intensities in the region of interest.

[0072] Step 3: level_map is further quantized by grouping the pixel intensities using the threshold determined in Step 1.

[0073] Step 4: The number of unique elements in each group of level_map is calculated.

[0074] Step 5: The value of the region based on entropy around an interest point is calculated as follows:

$$\text{Entropy(Region Value}, s_n) = \sum_{\mathrm{i}} \mathrm{N_i} * \log(1/\mathrm{N_i})$$

where N is the length of elements in group ‘i’.

[0075] Step 6: Average s_n across the interest point and corresponding offset points, and assign the averaged value to s_n.

[0076] In another implementation, region value can be calculated using the entropy of gradient information around the interest points in ‘D’ directions. In one implementation, D includes 0, 45, 90, and 135 degrees. The gradient information can be calculated in multiple scales around an interest point.

[0077] In one implementation, corner locations and corresponding region values can be used to determine the quality metric score generation. Quality metric score generation can be implemented as follows:

[0078] Step 1: Sort the interest points based on their region values in descending order and note the corresponding interest point locations.

[0079] Step 2: Let the distances between corresponding interest points after rearranging them based on Step 1 be $d = \{d_1, d_2, d_3, \ldots, d_{n-2}, d_{n-1}, d_n\}$, where d1 is the distance between the first and second interest point. d_n will be zero.

[0080] Step 3: Let the weights of the distances be computed as the logarithm of distances: $dw_n = \log(d)$.

[0081] Step 4: Compute the weighted index as:
$\mathrm{sw_n} = \exp^{(1/[1,2,3,\ldots,n-2,n-1,n])}$, where n is the number of interest points

[0082] The final quality metric score can be computed using the following formula:

$$\text{EV\_QM} = \sum_{n=1}^{p} s_n * sw_n * dw_n, \text{ where } p \text{ is the number of interest points}$$

[0083] Other interest point detection algorithms can be used, such as can be calculated using Accelerated Segment Test (FAST) or Speeded Up Robust Features (SURF) algorithms.

[0084] EV_QM can be calculated separately for ocular and periocular regions, and further, each ROI can be ranked based on calculated EV_QM. The ROIs can be matched in the order they were ranked if a progressive matcher (described below) is being used. Of note, the quality metrics described herein can be used in spoof detection techniques. Reproductions of facial features of an eye or face, such as a physical photographs or printouts, or digital images or videos of the genuine user played back on a screen for malicious intent, are often of reduced quality (e.g., lower resolution, fuzzy, discolored, noisy, blurred, etc.) compared to a real physical presence at the time of scan. After filtering out low quality points of interest during the verification process, a low quality reproduction generally will not have a sufficient number of recognizable points of interest and, therefore, will fail verification.

[0085] Similar techniques can be used to detect partial spoofs as well. For example, a person may hold a printout of a valid eye over his or her own eye in an attempt to pass the eye verification process. In performing verification, the determined quality of the eye printout (e.g., the recognizable points of interest) can be compared to the determined quality of periocular or other facial regions. If the difference in quality between the eye and one or more other regions exceeds a threshold, this may indicate the presence of a partial spoof, and the verification can fail. Other spoof detection techniques using the disclosed quality metrics are contemplated.

Interest Point Detection

[0086] Various interest point detection algorithms can be used within a captured image region of ocular-vascular, periocular, and face, separately. For example, the Speeded Up Robust Features (SURF) algorithm is a "blob" type feature detection algorithm that can be used to identify areas within an image region centered on an interest point. The Features from Accelerated Segment Test (FAST) algorithm is a corner detection algorithm that can also be used to identify interest points within an image region. The vascular point detector (VPD) can also be used to identify the points that latch on the vasculature within an image region. Further, VPD can also be used to detect points on the periocular region and face. In some implementations, candidate points can also be identified at multiple image scales. For example, if the original image size is 100x100 (Scale 0), points can be identified from the 100x100 original image and also when the original image is resized to 50x50 (Scale 1) and 25x25 (Scale 2). Other region specific configurations of point detectors can be considered.

Point Suppression

[0087] The number of candidate points that are produced using an interest point detection algorithm can vary based on the amount of texture and quality of the image. Moreover, such algorithms may latch onto noisy or irrelevant (with respect to the given application) information, especially with non-VPD algorithms. Such noisy or irrelevant candidate points can be removed by a candidate point suppression algorithms.

[0088] In one implementation, a non-vascular point suppression (NVS) algorithm is used to determine the candidate point quality. The algorithm is explained in the following steps:

[0089] Step 1: Extract a Region (R) around an interest point. Let the size of R be MxM.

[0090] Step 2: Extract Local Patches (LP) within R of size NxN, where N<M. Assign the LP centered in R as Center Patch (CP).

[0091] Step 3: Local patches inside the region R are sparsely populated.

[0092] Step 4: Calculate the histogram distributions of all LPs enclosed in region R, and subtract the histogram of CP from each calculated histogram for LPs.

[0093] Step 5: For each subtraction, calculate the spread score as the number of bins being occupied divided by the total available bins.

[0094] Step 6: Derive a combination of bandpass filters based on the distribution of pixels in original image (*Band Pass Filter Computation*). These bandpass filters are used to measure the amount of spurious pixels such as glare against the vascular information pixels.

[0095] Step 7: Subtracted patches from Step 4 are filtered based on the derived band pass filter, and a corner response is calculated.

[0096] Step 8: Use spread score and corner response in cascade framework. Spread score is a binary classifier: it either rejects or accepts the point. Corner response provides a normalized scored between zero and one. Zero indicates non-vascularity while one indicates vascularity.

[0097] In another implementation, a *Bandpass Filter Computation* is used to determine the candidate point strength. The bandpass filter is dynamically generated based on the segmented whites of the eye region statistics. The bandpass filter can be generated as follows:

[0098] Step 1: Extract the green layer of the scleral region from an RGB image.

[0099] Step 2: Derive a histogram of the region. For example, use ‘N’ bins to extract a histogram from an uint8 image.

[0100] Step 3: Calculate the exponential of inverse normalized histogram derived in Step 2.

[0101] Step 4: Suppress the exponential by a factor k. Factor k generally varies from 0.1 to 0.3 and can be tuned based on the application or dataset.

[0102] Step 5: Calculate the response of inverse exponential with integer values ranging from 1 to N.

[0103] Step 6: Concatenate the first 5 elements from Step 4 and remaining from Step 5. This provides *Bandpass Filter 1.*

[0104] Step 7: For *Bandpass Filter 2*, construct a Gaussian with normal distribution of scleral pixels (green channel).

[0105] Step 8: Use *Bandpass Filter 1* and *Bandpass Filter 2* in parallel to establish the strength of the vascular point.

[0106] This filter construction process is adaptive to image pixels. If the image is dark, most of the scleral pixels fall near the lower tail of the histogram. So, *Bandpass Filter 1* would have a higher response over *Bandpass Filter 2*. This reduces the score of the point. Similarly, a glare-saturated image would have all its pixels in the upper tail and *Bandpass Filter 2* will have a higher response, eliminating the point with a low score.

[0107] In one implementation, the uniformity of the Local Binary Pattern (LBP) can be generated to determine the quality of a candidate point. Using the 8-bit LBP code, the number of bitwise transitions from 0 to 1 or vice-versa determines uniformity of the LBP code. A LBP code is considered uniform, if it has less than or equal to ‘n’ transitions. The range of transitions is 0 to 8. In one implementation, n is equal to 3. As an example, the LBP codes 00000000 with 0 transitions, 01111100 with 2 transitions, 01000001 with 3 transitions, are uniform. Similarly, 01010000 with 4 transitions and 01010010 with 6 transitions are non-uniform. The candidate points that are uniform are retained in the template.

[0108] The above mentioned quality algorithms for candidate points can be used in combination or separately for a given enrollment or verification process.

Local Feature Descriptors

[0109] A feature vector around each interest point can be generated using an image patch around the interest point for the ocular-vascular, periocular, and face regions, separately. The descriptors for an interest point can be generated using a single or multiple feature descriptor algorithms. The Fast Retina Keypoint (FREAK) algorithm is one example feature descriptor algorithm that can be used, for example, to generate descriptors for candidate points identified by the FAST algorithms. A FREAK descriptor can be, for example, a string of binary numbers that defines visual features surrounding a candidate point. Local Binary Patterns (LBP) and their variants such as Center Symmetric Local Binary Patterns (CSLBP) around the candidate points are examples of feature descriptors that can be used to describe image patches in a vicinity of a candidate point. Histograms of oriented Gradients (HoG), Histograms of LBP (HLBP), Histograms of CSLBP (HCSLBP), Patterned Histograms of Extended Multi-Radii LBP (PH-EMR-LBP), Patterned Histograms of Extended Multi-Radii CSLBP (PH-EMR-CSLBP), Patterned Histograms of Extended Multi-Radii Local Ternary Patterns (PH-EMR-LTP), and Patterned Histograms of Binary Patterns after Dimensionality Reduction (PHBP-DR) are other examples of feature descriptors that can be used to describe image neighborhoods around candidate points. Other feature descriptor algorithms or combinations of algorithms can be used to generate local image descriptors for candidate points of an image region.

[0110] In one implementation, the following steps are involved in generating PH-EMR-LBP descriptors:

[0111] Step 1: Calculate both LBP codes for 3x3 pixel and 5x5 pixel concentric squares around each pixel to form a multi-radii LBP (MR-LBP) image (note that, in contrast to regular LBP, the aforesaid technique uses a square or rectangular neighborhood, which can include or exclude corners) rather than a circular loci of pixels around the center pixel for LBP code derivation). In a 3x3 pixel region, comparing the center pixel with its immediate eight neighboring pixels generates a 3x3 LBP code. The result is an 8-bit code, whose bit values are either 0 or 1 (1 if the intensity value of the neighbor pixel is greater than the intensity of the center pixel, else 0). Similarly, in a 5x5 pixel region, comparing the center pixel with its next to immediate eight neighboring pixels (that is, sixteen pixels) generates a 5x5 LBP code (results is a 16-bit code). Hence, a MR-LBP has a 24-bit code (8 from 3x3, and 16 from 5x5 LBP code) for a given pixel in an image.

[0112] Step 2: An MxM patch (output of step 1) around each interest point is segmented into NxN sub-regions that can have K overlapped pixels.

[0113] Step 3: The histograms for each 24-bit MR-LBP within each sub-region are derived separately, and concatenated to deliver PH-EMR-LBP (note that in contrast to regular LBP, this histogram is calculated based on the frequency of LBP bit locations in the patch, rather than that of the decimal equivalent of the binary code).

[0114] The parameter values for M, N and K can be tuned based on the images' spatial frequency, resolution, and noise.

[0115] In one implementation, the following steps are involved in generating PH-EMR-CSLBP descriptors:

[0116] Step 1: Calculate both 3x3 pixel and 5x5 pixel center symmetric local binary patterns (CSLBP) codes at each pixel to form a multi-radii CS-LBP (MR-CSLBP) image. In a 3x3 region, comparing the intensity values of the diagonal pixels using the 8 boundary pixels generates a 4-bit 3x3 CS-LBP code (starting from the top-left pixel, consider the boundary pixels as numbered from 1-8 in a clock-wise direction, the 4-bit CS-LBP code is generated by comparing pixel 1, 2, 3, and 4 with 5, 6, 7, and 8 respectively). The values are 0 or 1 (1 if the intensity value of 1 is greater than 5, else 0—similarly, for other combinations). Likewise, in a 5x5 region, the 16 diagonal pixels of the outer ring are compared to generate an 8-bit 5x5 CS-LBP code. Hence, a MR-CS-LBP has a 12-bit code (4 from 3x3, and 8 from 5x5 CS-LBP code) for a given pixel in an image.

[0117] Step 2: An MxM patch (output of step 1) around each interest point is segmented into NxN sub-regions that may have K overlapped pixels.

[0118] Step 3:: The histograms for each 12-bit MR-CS-LBP within each sub-region are derived separately, and concatenated to deliver PH-EMR-CS-LBP.

[0119] Similar to the previous descriptor, the parameter values for M, N and K can be tuned based on the images' spatial frequency, resolution, and noise.

[0120] In one implementation, the following steps are involved in generating PH-EMR-LTP descriptors:

[0121] Step 1: Calculate both 3x3 pixel and 5x5 pixel local ternary patterns codes at each pixel to form a multi-radii LTP (MR-LTP) image with a Step Size (SS). In a 3x3 region, comparing the center pixel with its immediate eight neighboring pixels generates a 3x3 LTP code. The result is a 16-bit code, whose values are 0 or 1 (1 if the intensity value of each neighbor pixel is greater than the intensity of the center pixel plus SS, else 0; and 1 if the intensity value of each neighbor pixel is less than the intensity of the center pixel minus SS, else 0). Similarly, in a 5x5 region, comparing the center pixel with its next to immediate eight neighboring pixels (that is 16 pixels) generates a 5x5 LTP code (result is a 32-bit code). Hence, a MR-LTP has a 48-bit code (16 from 3x3, and 32 from 5x5 LBP code) for a given pixel in an image.

[0122] Step 2: An MxM patch (output of step 1) around each interest point is segmented into NxN sub-regions that may have K overlapped pixels.

[0123] Step 3: The histograms for each 48-bit MR-LTP within each sub-region are derived separately, and concatenated to deliver PH-EMR-LTP.

[0124] Similar to the previous descriptor, the parameter values for M, N and K can be tuned based on the images' spatial frequency, resolution, and noise.

[0125] In one implementation, a PHBP-DR can be derived using the following steps:

[0126] Step 1: PH-EMR-LBP for a given interest point is derived by setting the M, N and K values to 9, 3 and 1 respectively. These configurations deliver a feature vector of length 384 (24x16; histograms of each of the 24-bit codes across 16 sub-regions. Note that 16 3x3 sub-regions are possible in a 9x9 patch with 1 pixel overlap) with values ranging from 0 to 9 (as we have 9 pixels in 3x3 sub-region).

[0127] Step 2: PH-EMR-CS-LBP for a given interest point is derived by setting the M, N and K values to 7, 3 and 1 respectively. These configurations deliver a feature vector of length 108 (12x9; histograms of each of the 12-bit codes across 9 sub-regions. Note that 9 3x3 sub-regions are possible in a 7x7 patch with 1 pixel overlap) with values ranging from 0 to 9 (as we have 9 pixels in 3x3 sub-region).

[0128] Step 3: PH-EMR-LTP PH-EMR-LBP for a given interest point is derived by setting the M, N and K values to 9, 3 and 1 respectively. These configurations deliver a feature vector of length 768 (48x16; histograms of each of the 48-bit codes across 16 sub-regions. Note that 16 3x3 sub-regions are possible in a 9x9 patch with 1 pixel overlap) with values ranging from 0 to 9 (as we have 9 pixels in 3x3 sub-region). SS is set to 5 after the intensity values of the image are normalized to 0-255.

[0129] Step 4: Feature vectors 1, 2, and 3 are concatenated to form a feature vector of length 1260.

[0130] Step 5: Using a variance analysis, only the top 720 features are retained. In some instances, as these highly variant features are pre-calculated, only these 720 features are generated during enrollment and verification process in the interest of reducing computational complexity.

[0131] Step 6: Finally, a dimensionality reduction method such as principal component analysis (PCA) is used to extract PHBP-DR. A separate PCA analysis can be performed for ocular and periocular regions to produce 103 and 98 long feature vectors, respectively. For face, either periocular features can be used as is, or a separate PCA analysis can be performed. Other lengths of feature vectors for ocular, periocular and face are possible.

[0132] At the end of the biometric enrollment process, an enrollment template can include a set of candidate points and the descriptors for ocular-vascular ROI, and a set of candidate points and the descriptors for a periocular region. In some implementations, an enrollment template can also include a set of candidate points and the descriptors for a face, and/or a set of candidate points and the descriptors for extended periocular regions. Note that the ocular-vascular and periocular ROI and template for left and right eyes are treated separately. Multiple descriptors for a set of interest points are contemplated. Original and processed images that are used to produce templates for ocular-vascular face, and periocular region can be discarded for security and privacy reasons.

Progressive Matcher

[0133] In one implementation of a biometric matching process, a final match score is generated using a progressive matcher. As an initial step, local image descriptors are matched to find Matched-Point-Pairs between enrollment and verification templates for both ocular and periocular regions using a distance measure. As an example, a Euclidean distance between descriptors of interest points can be calculated between the enrollment and verification descriptor vectors of ocular and periocular regions separately, and the pairs below a certain distance threshold can be retained as Matched-Point-Pairs.

[0134] In presence of noise or other aberrations, the Matched-Point-Pairs may have several outliers or otherwise erroneous extra matches. By assuming a plausible homography between the locations of Matched-Point-Pairs enrollment and verification images, outliers (spatially non-overlapping matched points after alignment under the assumed homography) can be removed from Matched-Point-Pairs.

[0135] In some implementations, a random sample consensus (RANSAC) or other outlier detection method can be used to determine the transformation needed to align candidate points in a verification image with points in an enrollment image, while rejecting outliers that do not fit a hypothesized transformation between genuine matches, in terms of geometries of ocular regions of interest encoded in enrollment and verification templates. In some implementations, the RANSACs of different ROIs (such as ocular-vascular and periocular ROIs) can be performed separately, and the collection of the surviving inlier points can be delivered to a final RANSAC for final score calculations and other related computations. In some implementations, a minimum number of inlier points from certain or all sub-regions of interest (e.g., vascular patches and points seen on top of the scleral ROI) can be required before proceeding to final matching. Eventually, the score is generated by combining the number of inliers that are found after RANSAC (N), recovered scale from the transformation matrix (RS), and recovered angle from the transformation matrix (RA), using the homography fitted to the locations of Matched-Point-Pairs by RANSAC or equivalent.

[0136] In one implementation, the match score is calculated using the following formula:

$$MS = \{(Cx + Cy)/2 * \log(N)\} \quad / \quad \{(1 + |\log2(RS+0.001)|) * (1+ (RA/0.2)^2)\}$$

where Cx and Cy are correlations between vectors of x and y coordinates of inlier matched points between enrollment and verification templates, respectively, N is the number of these aligned points, RA is the recovered angle which represents the change in angle resulting from the transformation of the locations of inlier matched verification points to the enrollment points for registration, and RS is the recovered scale which represents the change in scale resulting from the aforesaid transformation. RA and RS are derived from the similarity or alike geometric transformation matrix resulting from RANSAC or similar operation. Other measures such as the distance of the transformation/registration matrix from identity are contemplated, especially if the regions of interest (such as the ocular-vascular and periocular) are spatially pre-normalized.

[0137] In some implementations, M-SAC, Group-SAC and/or Optimal-RANSAC can replace RANSAC.

[0138] The match scores can be generated for both whites of the eye (SV) and the periocular region (SP) separately, combined (SVP), or in a sequential manner. In the progressive approach, different ROIs are progressively included into the matcher if there is insufficient information or quality for a robust decision in the starting region of interest. For instance, if certain source quality or a definite decision cannot be achieved using the vascular information in the whites of the eye, the matcher can progressively add more information from locations pertaining to periocular (and potentially beyond, like nose and face) as needed to achieve the desired certainty in asserting a biometric match.

[0139] In one implementation, the Matched-Point-Pairs from ocular and periocular regions are calculated separately and then combined for RANSAC to generate the final match score.

[0140] In another implementation, the Matched-Point-Pairs from ocular and periocular regions are calculated separately and then combined for RANSAC to generate the final match score. However, the final inliers generation is constrained to have at-least N points from the ocular region.

[0141] In a further implementation, the Matched-Point-Pairs from ocular and periocular regions are calculated separately, their respective inliers are discovered by ROI-specific RANSACs separately, and then the collection of region-specific RANSAC-filtered template elements are combined for a final RANSAC to generate the final match score. However, the final inliers generation is constrained to have at-least N points from the RANSAC of the ocular-vascular region. In some implementations, a typical minimum value for N is 3.

[0142] In some implementations, a progressive matcher is implemented to make the final decision based on comparison of match scores against thresholds. By way of example:

Step 1: If SV > 3.2, the user is authenticated, and steps 2 and 3 are skipped.
Step 2: If SVP > 3.4, the user is authenticated, and step 3 is skipped.
Step 3: If SP > 3.2, the user is authenticated.
End process.

[0143] In the event that the above progressive process does not authenticate the user, another enrollment template from the enrollment bank can be recalled, or a new verification image can be acquired, until a certain exit condition (such as a time limit) is reached. Other combinations of the SV, SVP and SP thresholds are contemplated. If one or more of the regions of interests is not available for scanning due to aberrations resulting from, for example, eyeglasses or glare artifacts, the systems can use other available regions.

[0144] FIG. 5 depicts one implementation of a method for biometric enrollment and authentication using progressive matching, including an enrollment phase in which enrollment templates are created based on ocular-vascular and periocular regions, and a verification phase in which progressive matching can be performed based on captured images and the enrollment templates. The steps illustrated in FIG. 5 can be performed in accordance with the various techniques described herein.

[0145] In some cases, a single biometric trait can be divided into multiple ROIs, which are then matched progressively. For example, significant regions of a periocular region can be divided into n parts that are then matched progressively.

[0146] In one implementation, the quality of some or all periocular ROIs can be measured and ranked, and the matching process can progressively add each ROI based on its respective rank as needed to achieve the desired certainty in asserting a biometric match.

[0147] In another implementation, an image-based distance metric or similar detects user poses, lighting conditions, or facial gestures that might distort some biometric regions of interest such as periocular. These variations can be induced during enrollment or added to a rolling template bank based on a template update policy. At the time of verification, the matcher can try to retrieve the most relevant templates from the bank based on the aforementioned image similarity metric.

[0148] In another implementation, if the matching process encounters a new pose or facial expression, the closest expression is used and a special case of progressive matcher is applied. For example, a special case can divide the periocular region into several segments using a clustering process, and within each segment the score and transformation matrix are generated. The final score can be determined by a weighted fusion of all the individual segment scores. These weights are determined by the amount of deformation observed across all the transformation matrices. Other score fusions are possible.

[0149] FIG. 5 illustrates one implementation of a localized system for generating secure biometric templates and performing user verification according to the techniques described herein. A user device 500 can include an image sensor 530, processor 540, memory 550, biometric hardware and/or software 560, and a system bus that couples various system components, including the memory 550 to the processor 540. User device 500 can include, but is not limited to, a smart phone, smart watch, smart glasses, tablet computer, portable computer, television, gaming device, music player, mobile telephone, laptop, palmtop, smart or dumb terminal, network computer, personal digital assistant, wireless device, information appliance, workstation, minicomputer, mainframe computer, or other computing device that is operated as a general purpose computer or a special purpose hardware device that can execute the functionality described herein.

[0150] Biometric hardware and/or software 560 includes an image processing module 562 for performing operations on images captured by image sensor 530. For example, image processing module 562 can perform segmentation and enhancement on images of the eye and surrounding facial area of a user 510 to assist in isolating vascular structures and other features of interest. Template module 564 creates biometric templates based on the vasculature imagery and can performs various obfuscating and scrambling operations on the templates. Verification module 566 validates the identity of a user 510 by performing matching operations between a biometric verification template formed upon capturing a biometric reading and a previously stored enrollment template. In some implementations, certain functionality can be performed on devices other than user device 500. For example, a user device can instead include only a biometric sensor, such as a camera, and image processing and verification functions can be performed on a remote server accessible to user device 500 over a network, such as the internet.

[0151] More generally, the systems and techniques described here can be implemented in a computing system that includes a back end component (e.g., as a data server), or that includes a middleware component (e.g., an application server), or that includes a front end component (e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the systems and techniques described here), or any combination of such back end, middleware, or front end components. The components of the system can be interconnected by any form or medium of digital data communication (e.g., a communication network). Examples of communication networks include a local area network ("LAN"), a wide area network ("WAN"), and the Internet.

[0152] The computing system can include clients and servers. A client and server are generally remote from each other and can interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention.

[0153] Implementations of the subject matter and the operations described in this specification can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Embodiments of the subject matter described in this specification can be implemented as one or more computer programs, i.e., one or more modules of computer program instructions, encoded on computer storage medium for execution by, or to control the operation of, data processing apparatus. Alternatively or in addition, the program instructions can be encoded on an artificially-generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. A computer storage medium can be, or be included in, a computer-readable storage device, a computer-readable storage substrate, a random or serial access memory array or device, or a combination of one or more of them. Moreover, while a computer storage medium is not a propagated signal, a computer storage medium can be a source or destination of computer program instructions encoded in an artificially-generated propagated signal. The computer storage medium can also be, or be included in, one or more separate physical components or media (e.g., multiple CDs, disks, or other storage devices).

[0154] The operations described in this specification can be implemented as operations performed by a data processing apparatus on data stored on one or more computer-readable storage devices or received from other sources.

[0155] The term "data processing apparatus" encompasses all kinds of apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, a system on a chip, or multiple ones, or combinations, of the foregoing The apparatus can include special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). The apparatus can also include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, a cross-platform runtime environment, a virtual machine, or a combination of one or more of them. The apparatus and execution environment can realize various different computing model infrastructures, such as web services, distributed computing and grid computing infrastructures.

[0156] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, declarative or procedural languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, object, or other unit suitable for use in a computing environment. A computer program may, but need not, correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language resource), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub-programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[0157] Embodiments of the subject matter described in this specification can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), an inter-network (e.g., the Internet), and peer-to-peer networks (e.g., ad hoc peer-to-peer networks).

[0158] The computing system can include clients and servers. A client and server are generally remote from each other and can interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. In some embodiments, a server transmits data (e.g., an HTML page) to a client device (e.g., for purposes of displaying data to and receiving user input from a user interacting with the client device). Data generated at the client device (e.g., a result of the user interaction) can be received from the client device at the server.

[0159] A system of one or more computers can be configured to perform particular operations or actions by virtue of having software, firmware, hardware, or a combination of them installed on the system that in operation causes or cause the system to perform the actions. One or more computer programs can be configured to perform particular operations or actions by virtue of including instructions that, when executed by data processing apparatus, cause the apparatus to perform the actions.

[0160] While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any inventions or of what may be claimed, but rather as descriptions of features specific to particular implementations of particular inventions. Certain features that are described in this specification in the context of separate implementations can also be implemented in combination in a single implementations. Conversely, various features that are described in the context of a single implementations can also be implemented in multiple implementations separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0161] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the implementations described above should not be understood as requiring such separation in all implementations, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

[0162] Thus, particular implementations of the subject matter have been described. Other implementations are within the scope of the following claims. In some cases, the actions recited in the claims can be performed in a different order and still achieve desirable results. In addition, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking and parallel processing may be advantageous.

CLAIMS:

1. A computer-implemented method comprising:

receiving an image of a facial region of a user, the facial region including an eye and an area surrounding the eye;

processing the image to:

(i) define an ocular image region including at least a portion of the eye in the image of the facial region and (ii) define one or more periocular image regions each including at least a portion of the area surrounding the eye in the image of the facial region, wherein processing the image comprises enhancing at least a portion of the image using local gradient Gabor pattern (LGGP) by:

calculating, at each of a plurality of angles, a Gabor phase image for at least the portion of the image;

aggregating the calculated Gabor phase images to form a combined Gabor phase image;

calculating, at each of a plurality of angles, a local gradient of the combined Gabor phase image; and

retaining a maximum value of each local gradient to form an enhanced image;

identifying a plurality of points of interest in at least one of the ocular image region and the one or more periocular image regions;

generating, for each point of interest, a feature descriptor based on a combination of a plurality of patterned histogram feature descriptors; and

storing the generated feature descriptors in a biometric template.

2. The method of claim 1, wherein generating the feature descriptor for a particular point of interest comprises generating a patterned histogram of extended multi-radii local binary patterns (PH-EMR-LBP), a patterned histogram of extended multi-radii center symmetric local binary patterns (PH-EMR- CSLBP), or a patterned histogram of extended multi-radii local ternary pattcrns (PH-EMR-LTP).

3. The method of claim 2, wherein generating the feature descriptor for a particular point of interest comprises generating a PH-EMR-LBP feature descriptor by:

defining an image region comprising the particular point of interest;

calculating a plurality of local binary pattern (LBP) codes for each pixel in the image region to form a multi-radii LBP (MR-LBP) image;

dividing the MR-LBP image into a plurality of sub-regions;

deriving a plurality of histograms comprising frequencies of each MR-LBP bit location within each sub-region; and

combining the plurality of histograms to form the PH-EMR-LBP feature descriptor.

4. The method of claim 2, wherein generating the feature descriptor for a particular point of interest comprises generating a PH-EMR-CSLBP feature descriptor by:

defining an image region comprising the particular point of interest;

calculating a plurality of center symmetric local binary pattern (CSLBP) codes for each pixel in the image region to form a multi-radii CSLBP (MR-CSLBP) image;

dividing the MR-LBP image into a plurality of sub-regions;

deriving a plurality of histograms comprising frequencies of each MR-CSLBP bit location within each sub-region; and

combining the plurality of histograms to form the PH-EMR-CSLBP feature descriptor.

5. The method of claim 2, wherein generating the feature descriptor for a particular point of interest comprises generating a PH-EMR-LTP feature descriptor by:

defining an image region comprising the particular point of interest;

calculating a plurality of local ternary pattern (LTP) codes for each pixel in the image region to form a multi-radii LTP (MR-LTP) image;

dividing the MR-LBP image into a plurality of sub-regions;

deriving a plurality of histograms comprising frequencies of each MR-LTP bit location within each sub-region; and

combining the plurality of histograms to form the PH-EMR-LTP feature descriptor.

6. The method of claim 2, wherein generating the feature descriptor for a particular point of interest comprises:

calculating a PH-EMR-LBP feature descriptor, PH-EMR-CS-LBP feature descriptor, and PH-EMR-LTP feature descriptor for the particular point of interest; and

combining the PH-EMR-LBP feature descriptor, PH-EMR-CS-LBP feature descriptor, and PH-EMR-LTP feature descriptor to form a combined feature descriptor.

7. The method of claim 6, wherein generating the feature descriptor for a particular point of interest further comprises applying a variance analysis to the combined feature descriptor to form a top feature descriptor comprising a subset of features from the combined feature descriptor.

8. The method of claim 7, wherein generating the feature descriptor for a particular point of interest further comprises performing dimensionality reduction on the top feature descriptor to form the feature descriptor for the particular point of interest.

9. The method of claim 1, wherein, in generating a particular patterned histogram feature descriptor, local binary pattern or local ternary pattern feature descriptors are calculated using a neighborhood defined by a square with or without corners.

10. A system comprising:

at least one memory for storing computer-executable instructions; and

at least one processing unit for executing the instructions stored on the at least one memory, wherein execution of the instructions programs the at least one processing unit to perform operations comprising:

receiving an image of a facial region of a user, the facial region including an eye and an area surrounding the eye;

processing the image to:

(i) define an ocular image region including at least a portion of the eye in the image of the facial region and (ii) define one or more periocular image regions each including at least a portion of the area surrounding the eye in the image of the facial region, wherein processing the image comprises enhancing at least a portion of the image using local gradient Gabor pattern (LGGP) by:

calculating, at each of a plurality of angles, a Gabor phase image for at least the portion of the image;

aggregating the calculated Gabor phase images to form a combined Gabor phase image;

calculating, at each of a plurality of angles, a local gradient of the combined Gabor phase image; and

retaining a maximum value of each local gradient to form an enhanced image;

identifying a plurality of points of interest in at least one of the ocular image region and the one or more periocular image regions;

generating, for each point of interest, a feature descriptor based on a combination of a plurality of patterned histogram feature descriptors; and

storing the generated feature descriptors in a biometric template.

11. The system of claim 10, wherein generating the feature descriptor for a particular point of interest comprises generating a patterned histogram of extended multi-radii local binary patterns (PH-EMR-LBP), a patterned histogram of extended multi-radii center symmetric local binary patterns (PH-EMR- CSLBP), or a patterned histogram of extended multi-radii local ternary patterns (PH-EMR-LTP).

12. The system of claim 11, wherein generating the feature descriptor for a particular point of interest comprises generating a PH-EMR-LBP feature descriptor by:

defining an image region comprising the particular point of interest;

calculating a plurality of local binary pattern (LBP) codes for each pixel in the image region to form a multi-radii LBP (MR-LBP) image;

dividing the MR-LBP image into a plurality of sub-regions;

deriving a plurality of histograms comprising frequencies of each MR-LBP bit location within each sub-region; and

combining the plurality of histograms to form the PH-EMR-LBP feature descriptor.

13. The system of claim 11, wherein generating the feature descriptor for a particular point of interest comprises generating a PH-EMR-CSLBP feature descriptor by:

defining an image region comprising the particular point of interest;

calculating a plurality of center symmetric local binary pattern (CSLBP) codes for each pixel in the image region to form a multi-radii CSLBP (MR-CSLBP) image;

dividing the MR-LBP image into a plurality of sub-regions;

deriving a plurality of histograms comprising frequencies of each MR-CSLBP bit location within each sub-region; and

combining the plurality of histograms to form the PH-EMR-CSLBP feature descriptor.

14. The system of claim 11, wherein generating the feature descriptor for a particular point of interest comprises generating a PH-EMR-LTP feature descriptor by:

defining an image region comprising the particular point of interest;

calculating a plurality of local ternary pattern (LTP) codes for each pixel in the image region to form a multi-radii LTP (MR-LTP) image;

dividing the MR-LBP image into a plurality of sub-regions;

deriving a plurality of histograms comprising frequencies of each MR-LTP bit location within each sub-region; and

combining the plurality of histograms to form the PH-EMR-LTP feature descriptor.

15. The system of claim 11, wherein generating the feature descriptor for a particular point of interest comprises:

calculating a PH-EMR-LBP feature descriptor, PH-EMR-CS-LBP feature descriptor, and PH-EMR-LTP feature descriptor for the particular point of interest; and

combining the PH-EMR-LBP feature descriptor, PH-EMR-CS-LBP feature descriptor, and PH-EMR-LTP feature descriptor to form a combined feature descriptor.

16. The system of claim 15, wherein generating the feature descriptor for a particular point of interest further comprises applying a variance analysis to the combined feature descriptor to form a top feature descriptor comprising a subset of features from the combined feature descriptor.

17. The system of claim 16, wherein generating the feature descriptor for a particular point of interest further comprises performing dimensionality reduction on the top feature descriptor to form the feature descriptor for the particular point of interest.

18. The system of claim 10, wherein, in generating a particular patterned histogram feature descriptor, local binary pattern or local ternary pattern feature descriptors are calculated using a neighborhood defined by a square with or without corners.

EyeVerify Inc.

Patent Attorneys for the Applicant/Nominated Person

SPRUSON & FERGUSON

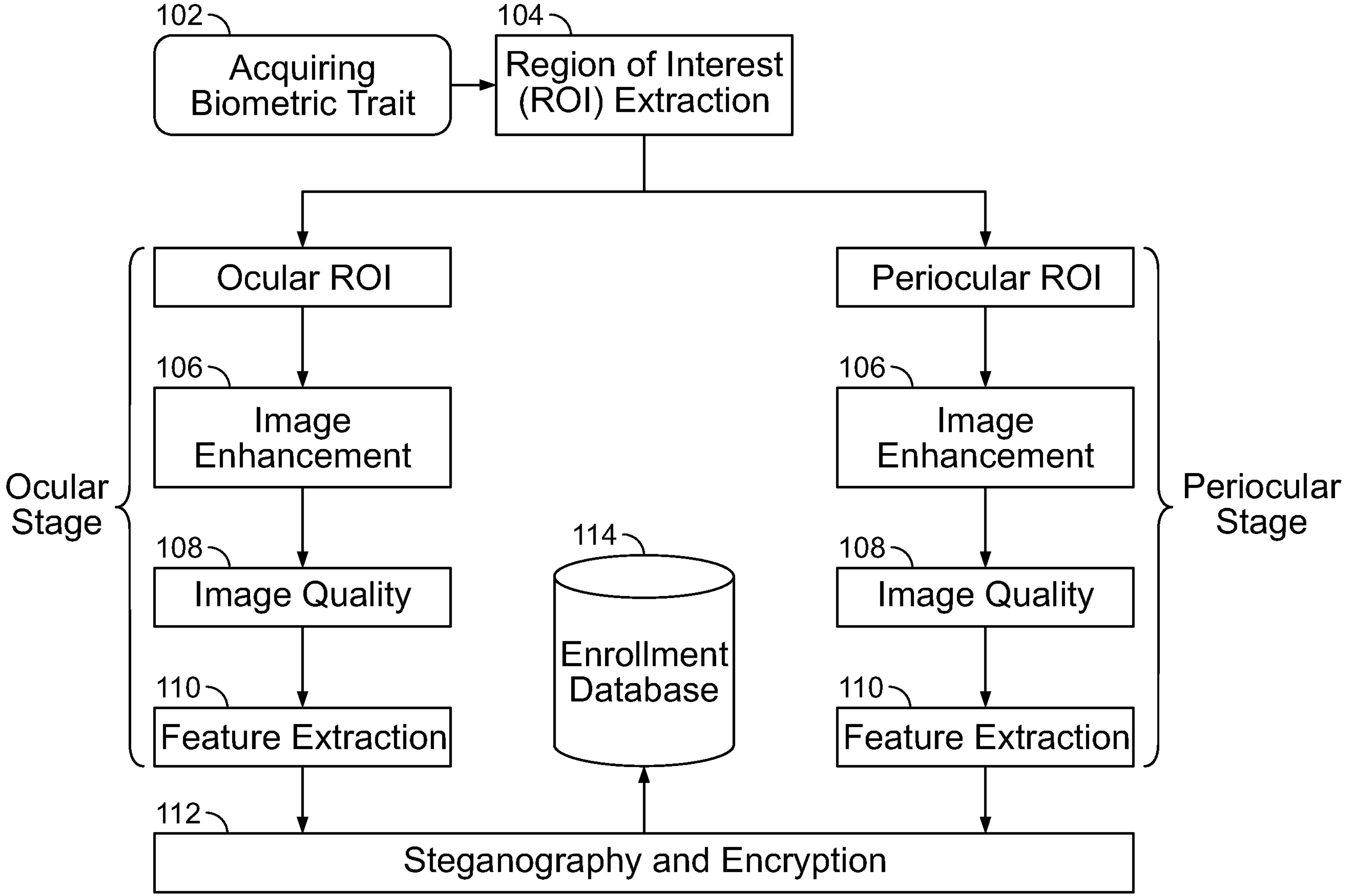

102
Acquiring Biometric Trait
104
Region of Interest (ROI) Extraction
Ocular Stage
Ocular ROI
106
Image Enhancement
108
Image Quality
110
Feature Extraction
Periocular ROI
106
Image Enhancement
108
Image Quality
110
Feature Extraction
Periocular Stage
114
Enrollment Database
112
Steganography and Encryption

FIG. 1

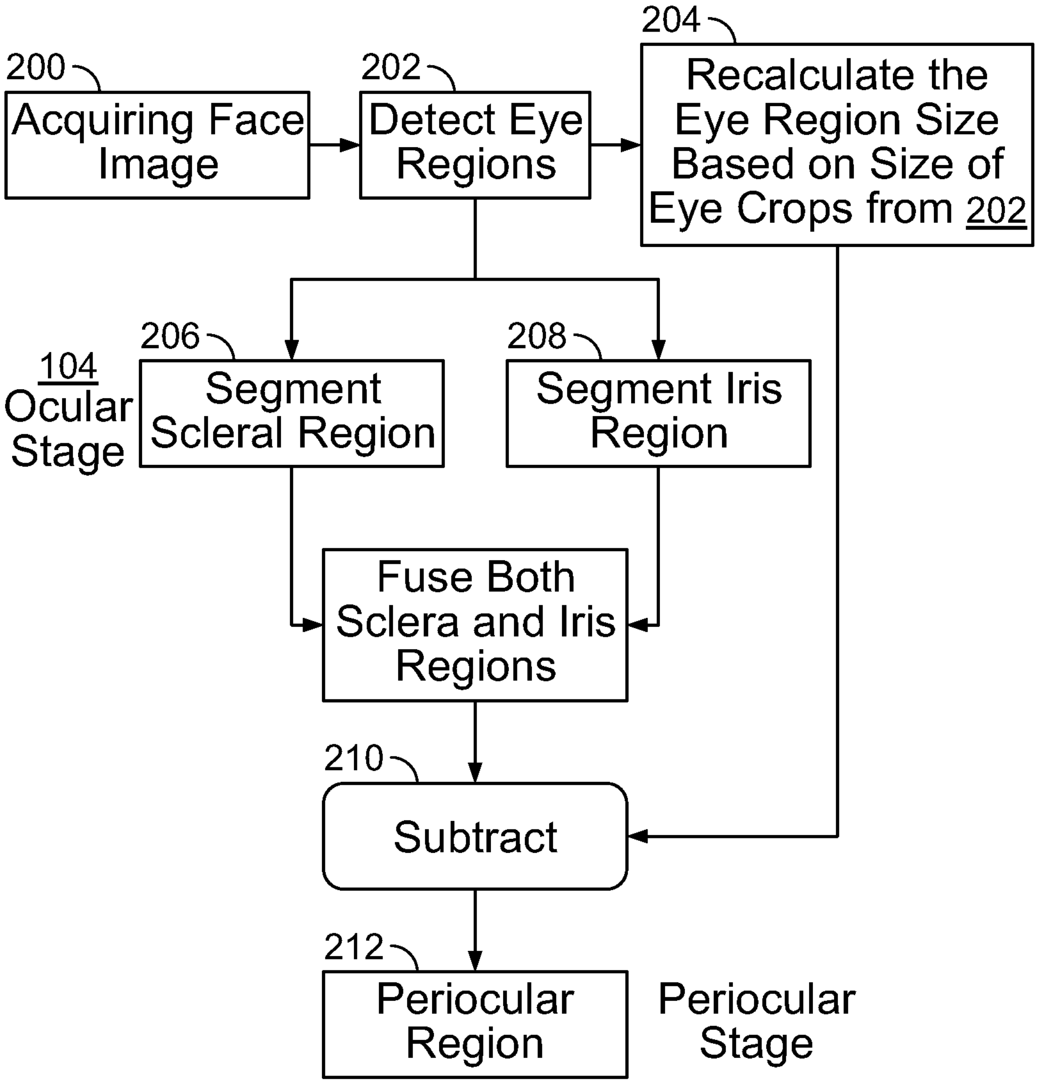

200
Acquiring Face Image
202
Detect Eye Regions
204
Recalculate the Eye Region Size Based on Size of Eye Crops from 202
104
Ocular Stage
206
Segment Scleral Region
208
Segment Iris Region
Fuse Both Sclera and Iris Regions
210
Subtract
212
Periocular Region
Periocular Stage

FIG. 2A

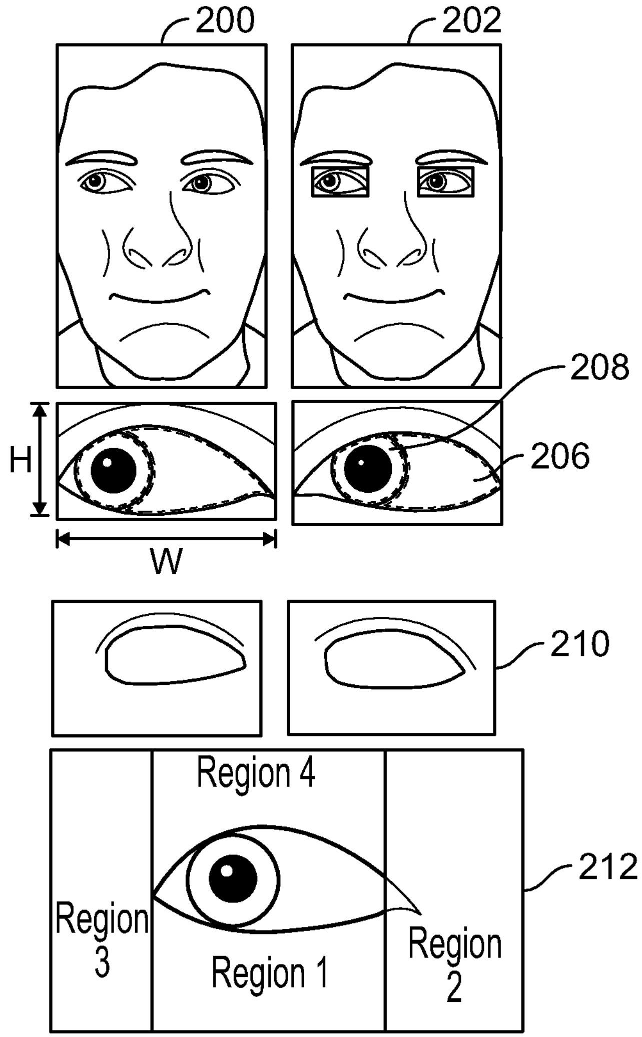
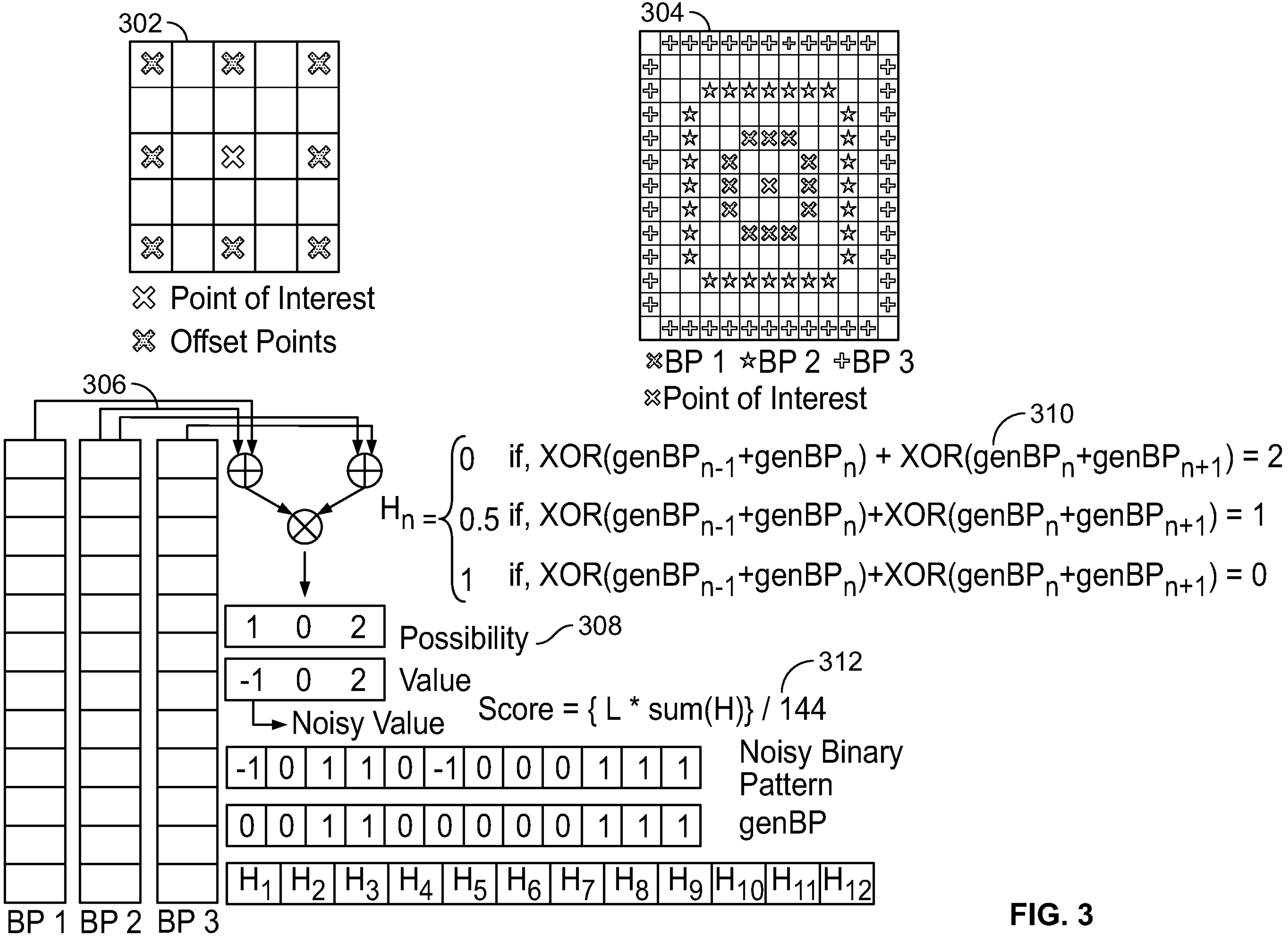

200
202
208
206
H
W
210
212
Region 4
Region 3
Region 1
Region 2

FIG. 2B



$$H_n = \begin{cases} 0 & \text{if, } XOR(genBP_{n-1}+genBP_n) + XOR(genBP_n+genBP_{n+1}) = 2 \\ 0.5 & \text{if, } XOR(genBP_{n-1}+genBP_n)+XOR(genBP_n+genBP_{n+1}) = 1 \\ 1 & \text{if, } XOR(genBP_{n-1}+genBP_n)+XOR(genBP_n+genBP_{n+1}) = 0 \end{cases}$$

$$Score = \{ L * sum(H)\} / 144$$

FIG. 3

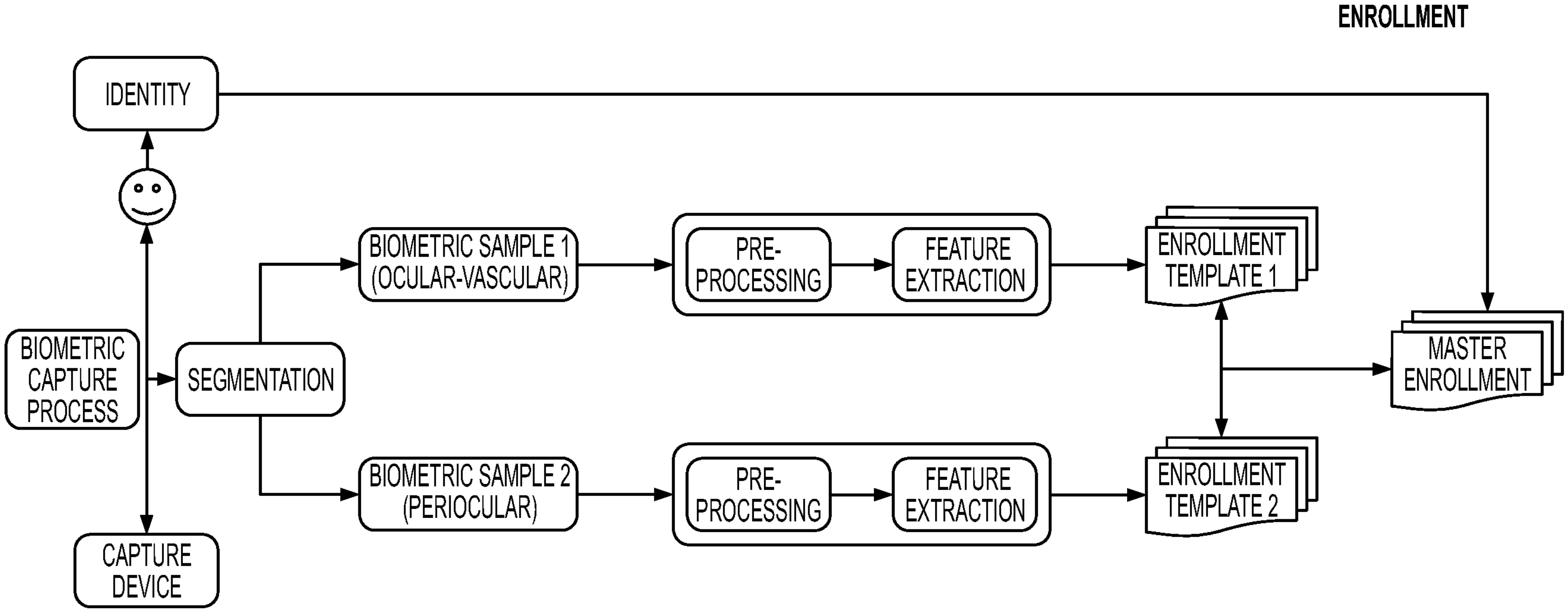

ENROLLMENT
IDENTITY
BIOMETRIC CAPTURE PROCESS
CAPTURE DEVICE
SEGMENTATION
BIOMETRIC SAMPLE 1 (OCULAR-VASCULAR)
BIOMETRIC SAMPLE 2 (PERIOCULAR)
PRE-PROCESSING
FEATURE EXTRACTION
PRE-PROCESSING
FEATURE EXTRACTION
ENROLLMENT TEMPLATE 1
ENROLLMENT TEMPLATE 2
MASTER ENROLLMENT

FIG. 4

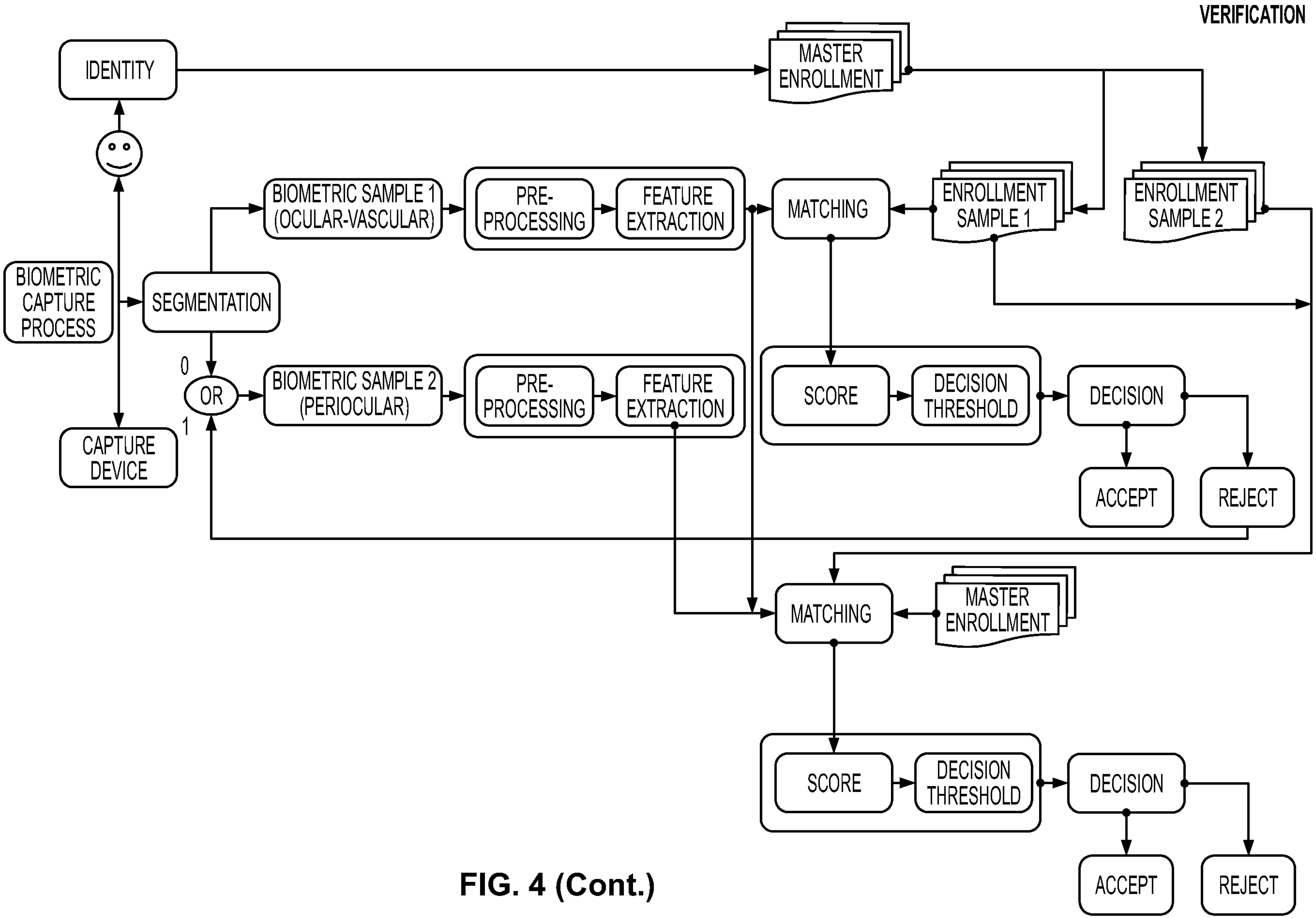

VERIFICATION
IDENTITY
MASTER ENROLLMENT
ENROLLMENT SAMPLE 1
ENROLLMENT SAMPLE 2
BIOMETRIC CAPTURE PROCESS
SEGMENTATION
CAPTURE DEVICE
0
OR
1
BIOMETRIC SAMPLE 1 (OCULAR-VASCULAR)
BIOMETRIC SAMPLE 2 (PERIOCULAR)
PRE-PROCESSING
FEATURE EXTRACTION
PRE-PROCESSING
FEATURE EXTRACTION
MATCHING
SCORE
DECISION THRESHOLD
DECISION
ACCEPT
REJECT
MATCHING
MASTER ENROLLMENT
SCORE
DECISION THRESHOLD
DECISION
ACCEPT
REJECT

FIG. 4 (Cont.)

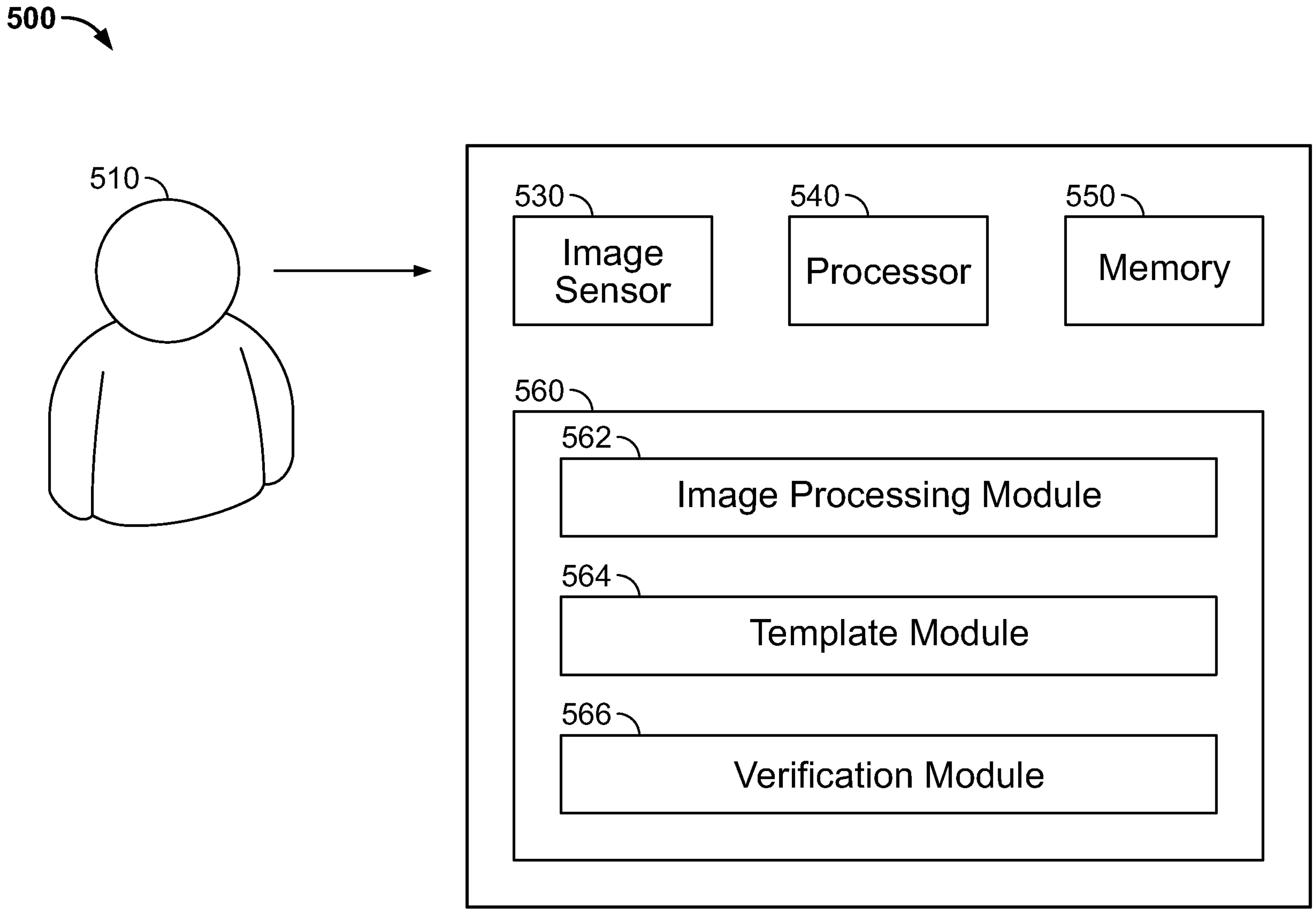

500
510
530
Image Sensor
540
Processor
550
Memory
560
562
Image Processing Module
564
Template Module
566
Verification Module

FIG. 5