US 20170236258A1

(19) **United States**

(12) **Patent Application Publication**

Hsu et al.

(10) **Pub. No.: US 2017/0236258 A1**

(43) **Pub. Date: Aug. 17, 2017**

(54) **SYSTEM AND METHOD FOR FINDING LINES IN AN IMAGE WITH A VISION SYSTEM**

(71) Applicant: **Cognex Corporation**, Natick, MA (US)

(72) Inventors: **Yu Feng Hsu**, Chupei City (TW); **Lowell D. Jacobson**, Grafton, MA (US); **David Y. Li**, West Roxbury, MA (US)

(21) Appl. No.: **15/338,445**

(22) Filed: **Oct. 31, 2016**

Related U.S. Application Data

(60) Provisional application No. 62/249,918, filed on Nov. 2, 2015.

Publication Classification

(51) **Int. Cl.**

G06T 5/20 (2006.01)

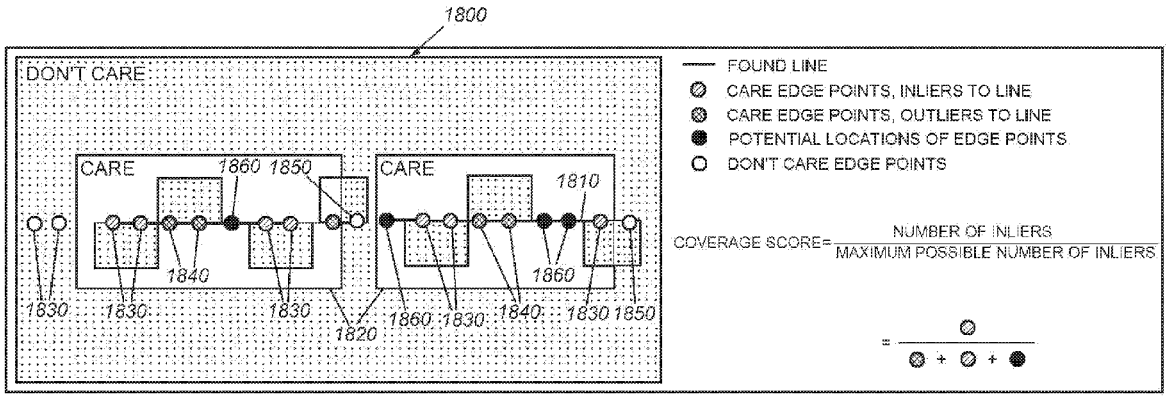
G06T 7/13 (2006.01)

(52) **U.S. Cl.**

CPC . ***G06T 5/20*** (2013.01); ***G06T 7/13*** (2017.01)

(57) **ABSTRACT**

This invention provides a system and method for finding multiple line features in an image. Two related steps are used to identify line features. First, the process computes x and y-components of the gradient field at each image location, projects the gradient field over a plurality subregions, and detects a plurality of gradient extrema, yielding a plurality of edge points with position and gradient. Next, the process iteratively chooses two edge points, fits a model line to them, and if edge point gradients are consistent with the model, computes the full set of inlier points whose position and gradient are consistent with that model. The candidate line with greatest inlier count is retained and the set of remaining outlier points is derived. The process then repeatedly applies the line fitting operation on this and subsequent outlier sets to find a plurality of line results. The process can be exhaustive RANSAC-based.



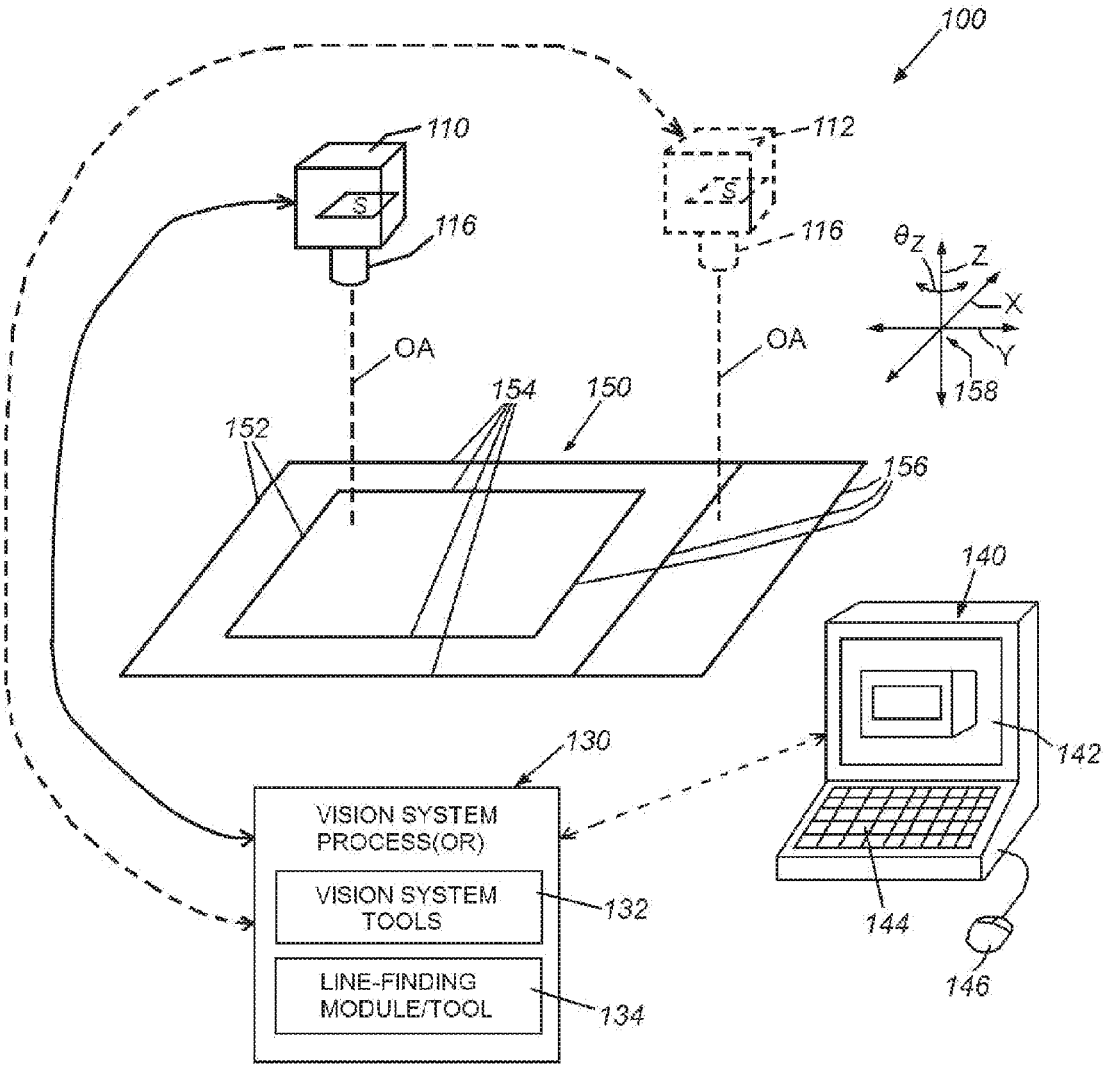

100
110
112
S
S
116
116
θZ
Z
X
Y
158
OA
OA
154
150
152
156
140
130
142
VISION SYSTEM PROCESS(OR)
VISION SYSTEM TOOLS
132
LINE-FINDING MODULE/TOOL
134
144
146

Fig. 1

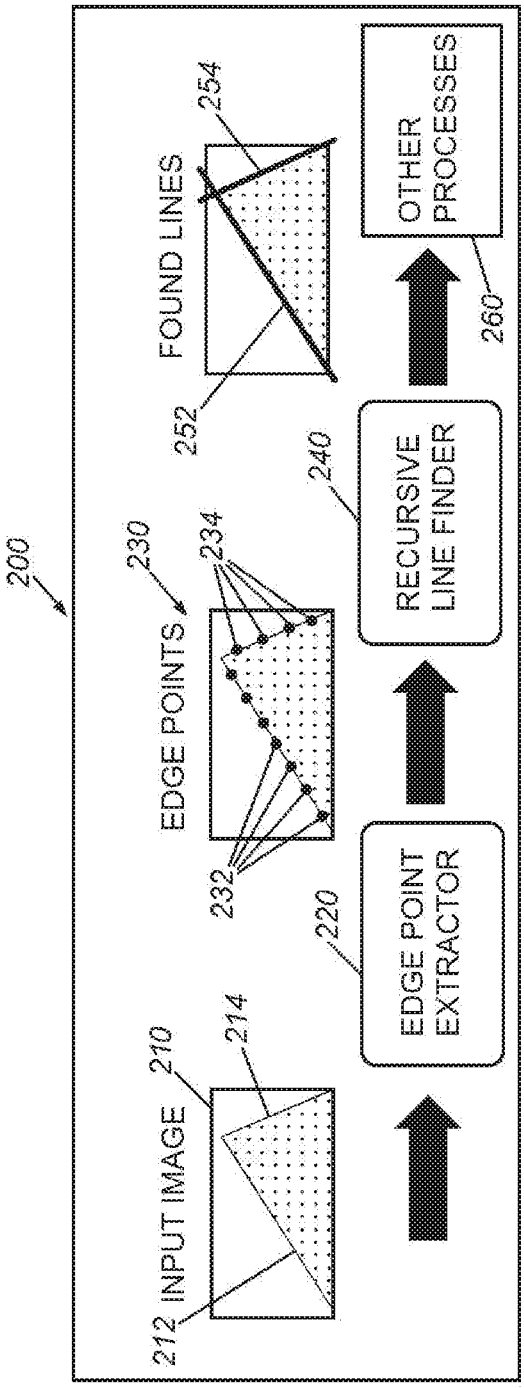

200
INPUT IMAGE
210
212
214
EDGE POINT EXTRACTOR
220
EDGE POINTS
230
232
234
RECURSIVE LINE FINDER
240
FOUND LINES
252
254
OTHER PROCESSES
260

Fig. 2

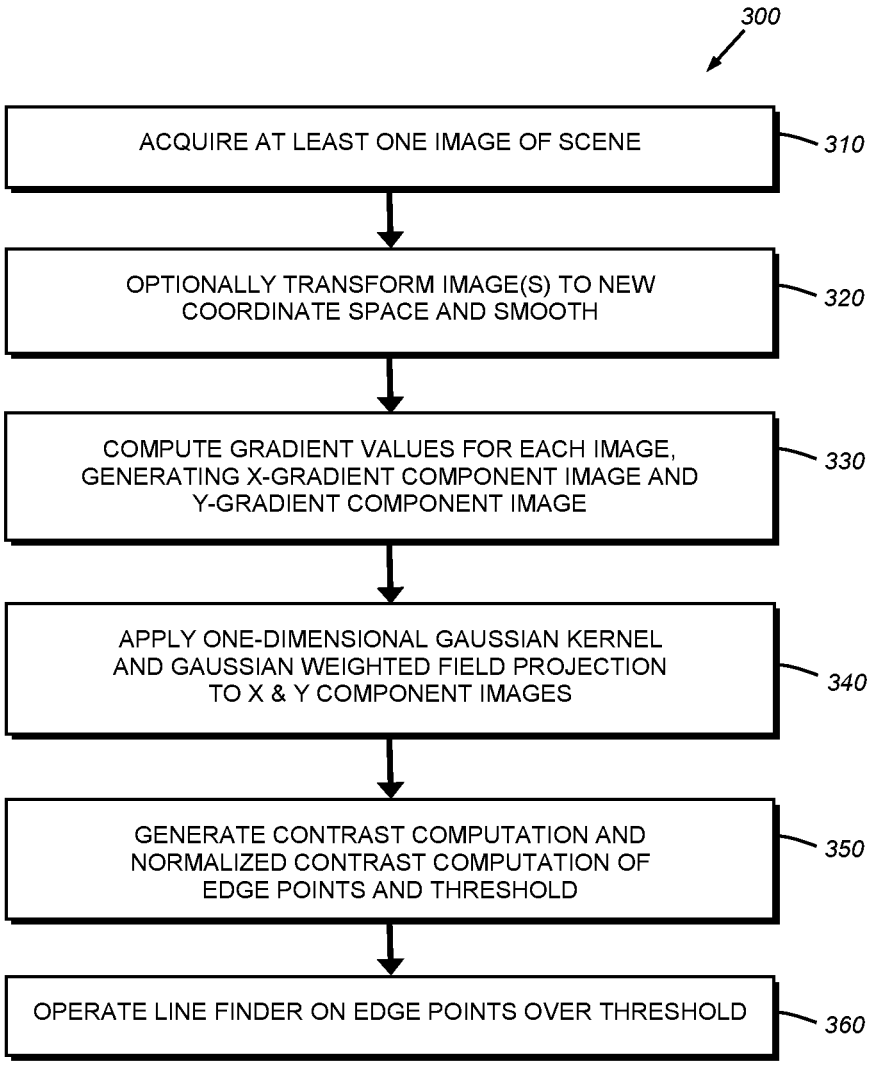

300
ACQUIRE AT LEAST ONE IMAGE OF SCENE
310
OPTIONALLY TRANSFORM IMAGE(S) TO NEW COORDINATE SPACE AND SMOOTH
320
COMPUTE GRADIENT VALUES FOR EACH IMAGE, GENERATING X-GRADIENT COMPONENT IMAGE AND Y-GRADIENT COMPONENT IMAGE
330
APPLY ONE-DIMENSIONAL GAUSSIAN KERNEL AND GAUSSIAN WEIGHTED FIELD PROJECTION TO X & Y COMPONENT IMAGES
340
GENERATE CONTRAST COMPUTATION AND NORMALIZED CONTRAST COMPUTATION OF EDGE POINTS AND THRESHOLD
350
OPERATE LINE FINDER ON EDGE POINTS OVER THRESHOLD
360

Fig. 3

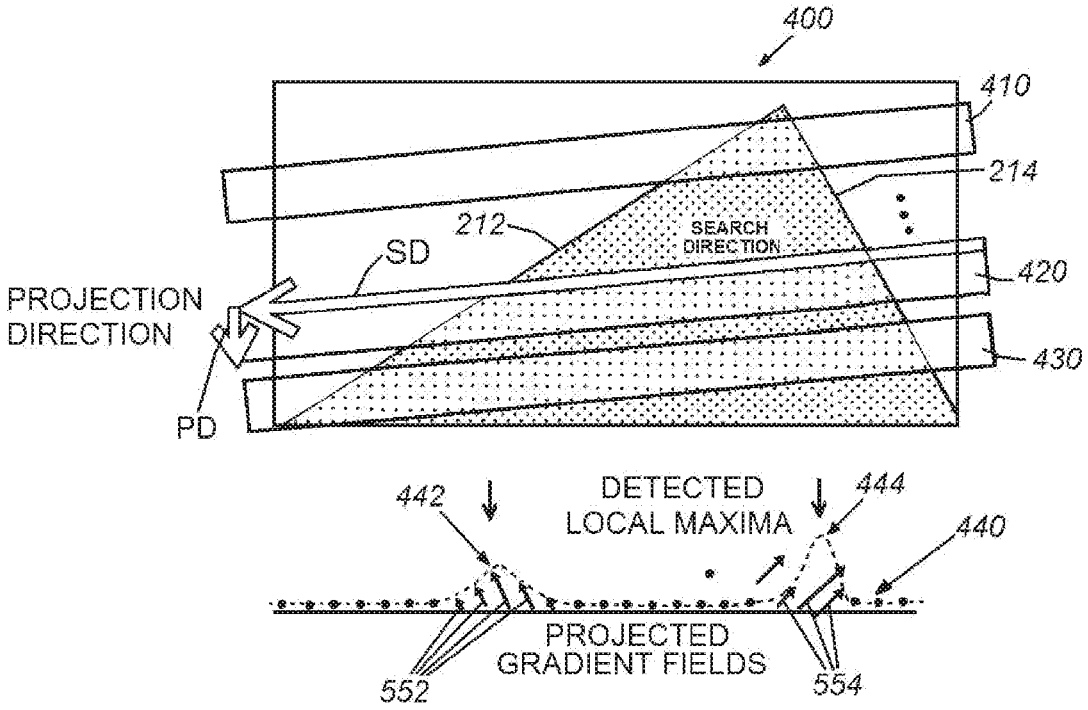

400
410
214
212
SEARCH DIRECTION
SD
420
PROJECTION DIRECTION
430
PD
442
DETECTED LOCAL MAXIMA
444
440
PROJECTED GRADIENT FIELDS
552
554

Fig. 4

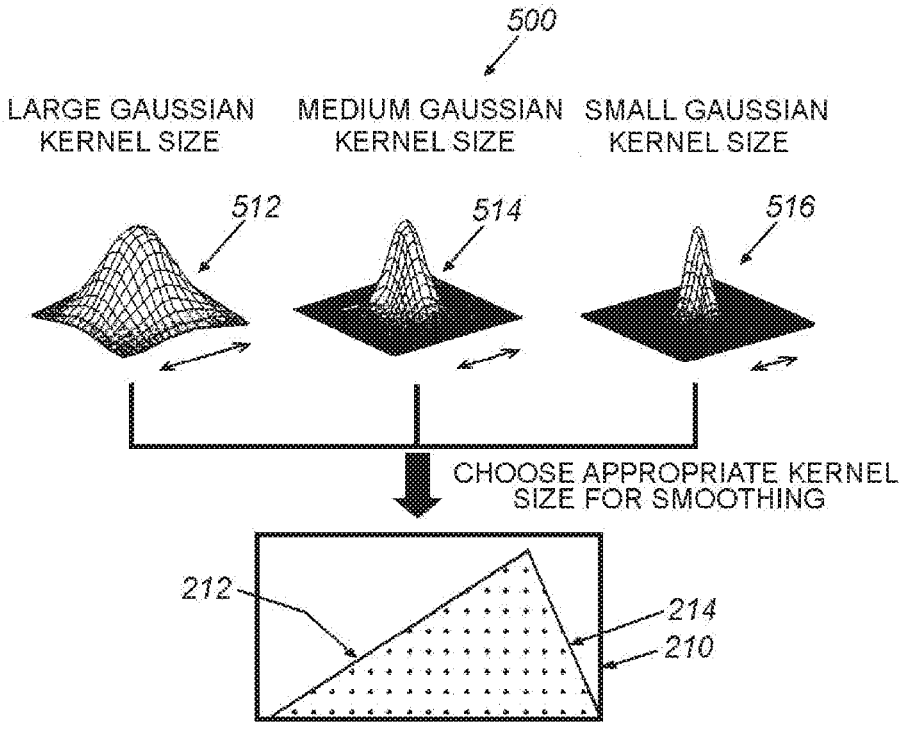

500
LARGE GAUSSIAN KERNEL SIZE
MEDIUM GAUSSIAN KERNEL SIZE
SMALL GAUSSIAN KERNEL SIZE
512
514
516
CHOOSE APPROPRIATE KERNEL SIZE FOR SMOOTHING
212
214
210

Fig. 5

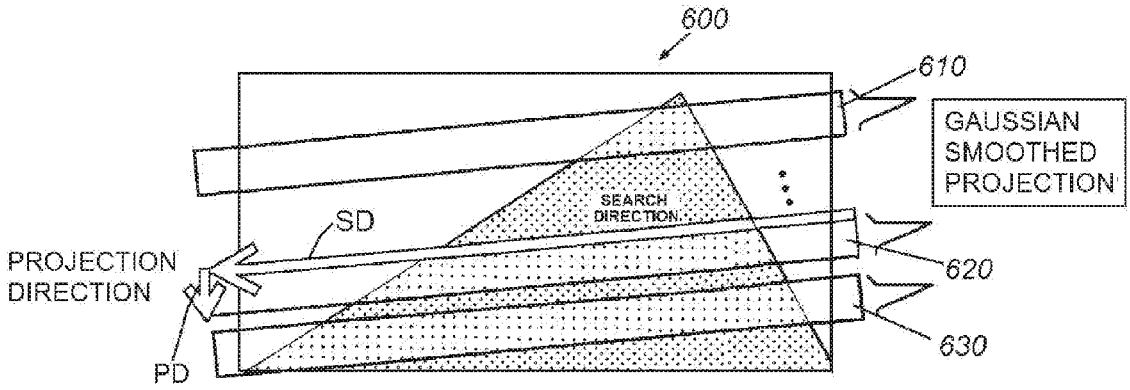

600
610
GAUSSIAN
SMOOTHED
PROJECTION
SEARCH
DIRECTION
SD
PROJECTION
DIRECTION
620
630
PD

Fig. 6

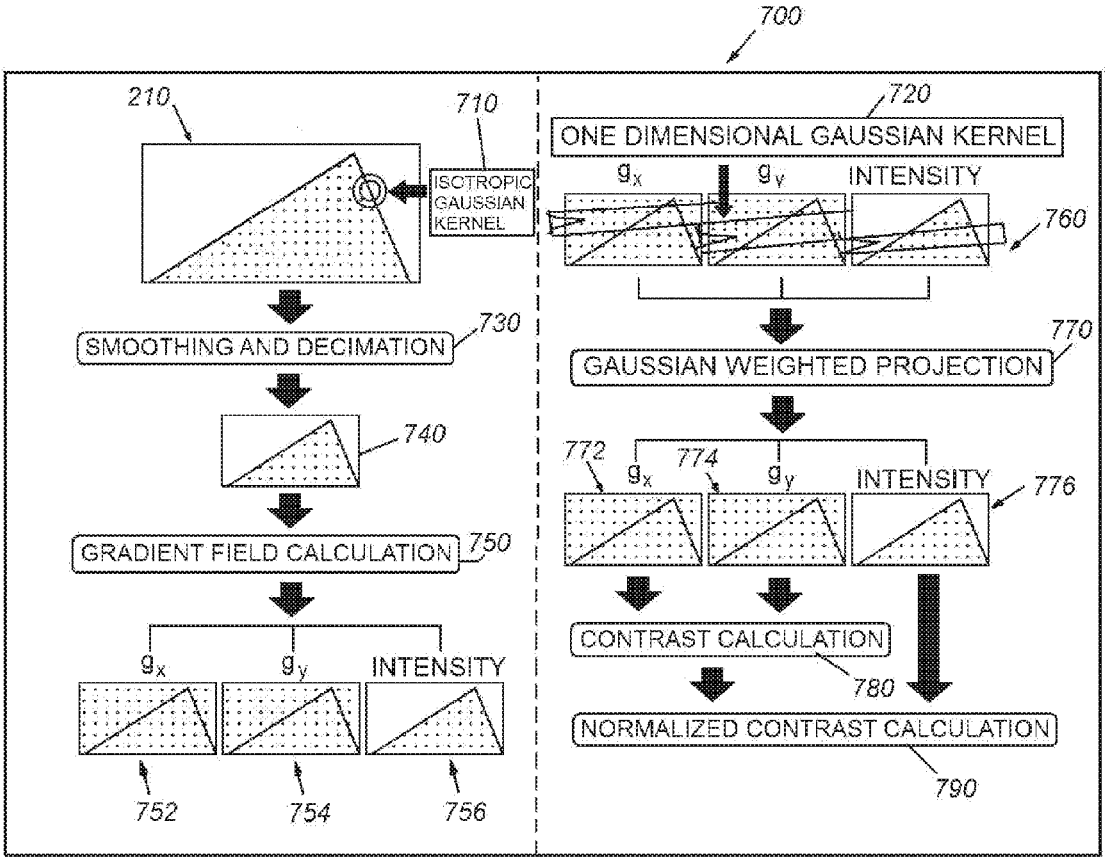

700
210
710
ISOTROPIC GAUSSIAN KERNEL
720
ONE DIMENSIONAL GAUSSIAN KERNEL
g_x
g_y
INTENSITY
760
SMOOTHING AND DECIMATION
730
770
GAUSSIAN WEIGHTED PROJECTION
740
772
774
776
GRADIENT FIELD CALCULATION
750
CONTRAST CALCULATION
780
NORMALIZED CONTRAST CALCULATION
790
752
754
756

Fig. 7

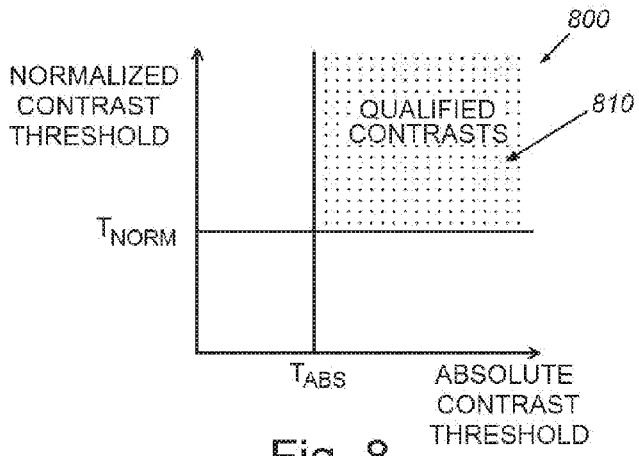

800
NORMALIZED CONTRAST THRESHOLD
QUALIFIED CONTRASTS
810
T_{NORM}
T_{ABS}
ABSOLUTE CONTRAST THRESHOLD

Fig. 8

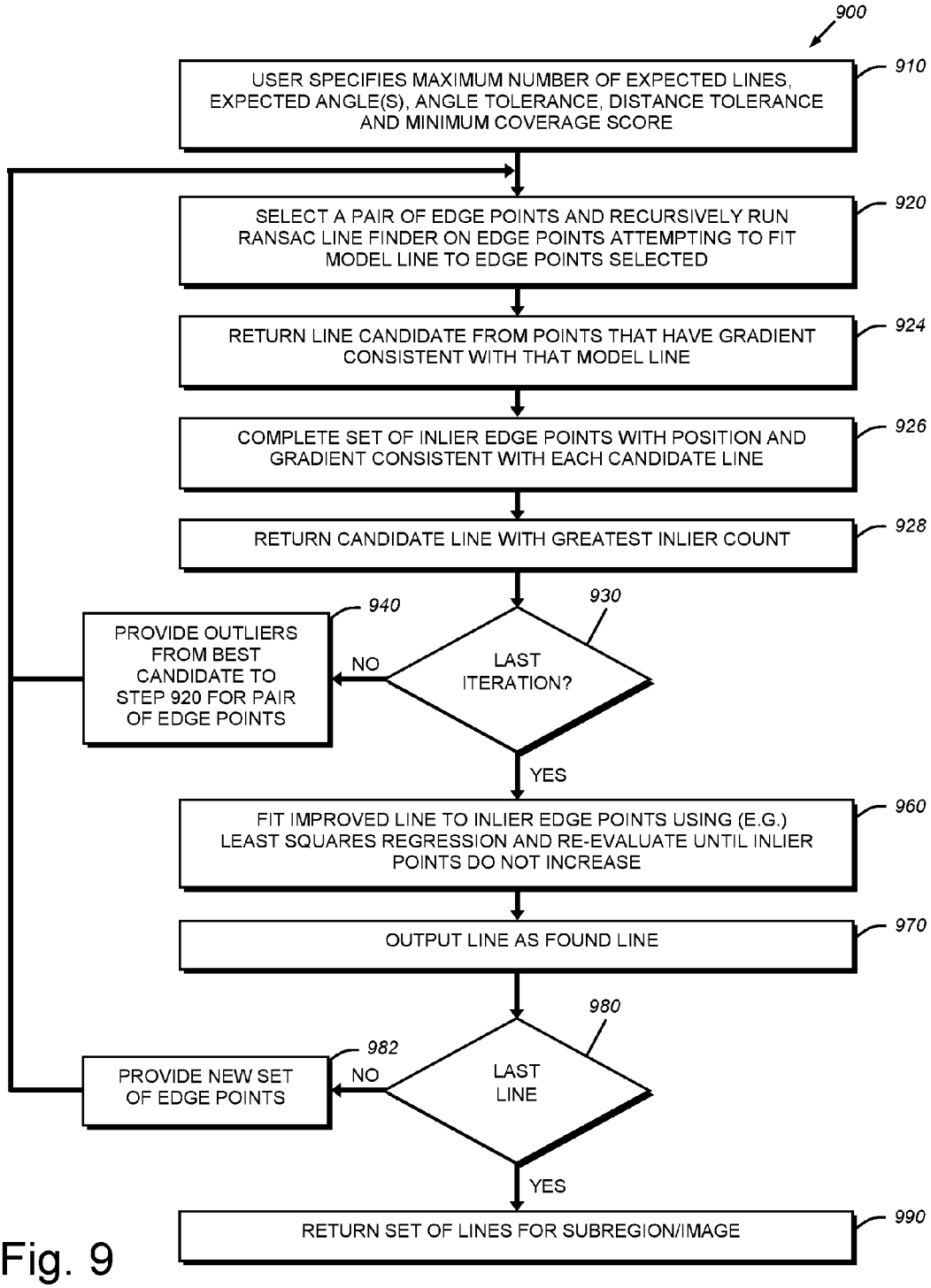

900
USER SPECIFIES MAXIMUM NUMBER OF EXPECTED LINES, EXPECTED ANGLE(S), ANGLE TOLERANCE, DISTANCE TOLERANCE AND MINIMUM COVERAGE SCORE
910
SELECT A PAIR OF EDGE POINTS AND RECURSIVELY RUN RANSAC LINE FINDER ON EDGE POINTS ATTEMPTING TO FIT MODEL LINE TO EDGE POINTS SELECTED
920
RETURN LINE CANDIDATE FROM POINTS THAT HAVE GRADIENT CONSISTENT WITH THAT MODEL LINE
924
COMPLETE SET OF INLIER EDGE POINTS WITH POSITION AND GRADIENT CONSISTENT WITH EACH CANDIDATE LINE
926
RETURN CANDIDATE LINE WITH GREATEST INLIER COUNT
928
930
LAST ITERATION?
NO
940
PROVIDE OUTLIERS FROM BEST CANDIDATE TO STEP 920 FOR PAIR OF EDGE POINTS
YES
FIT IMPROVED LINE TO INLIER EDGE POINTS USING (E.G.) LEAST SQUARES REGRESSION AND RE-EVALUATE UNTIL INLIER POINTS DO NOT INCREASE
960
OUTPUT LINE AS FOUND LINE
970
980
LAST LINE
NO
982
PROVIDE NEW SET OF EDGE POINTS
YES
RETURN SET OF LINES FOR SUBREGION/IMAGE
990

Fig. 9

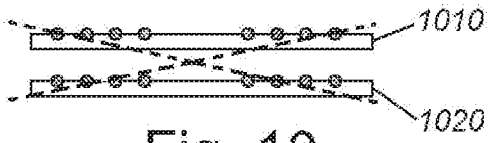


Fig. 10

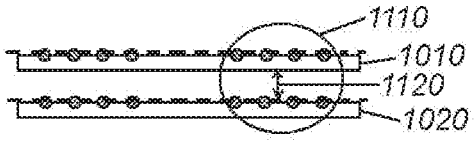


Fig. 11

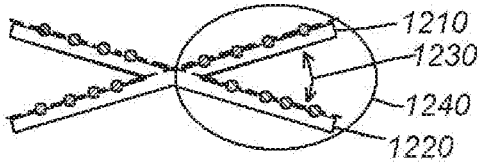


Fig. 12

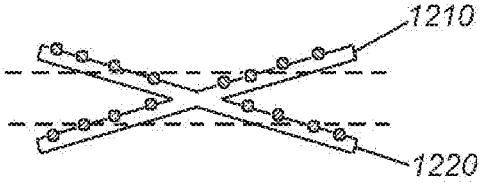
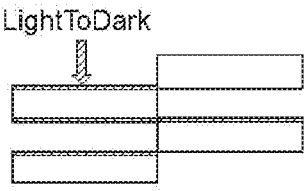
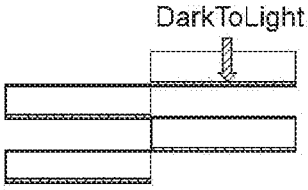


Fig. 13



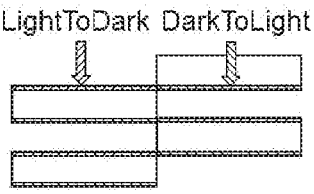
LightToDark POLARITY

Fig. 14



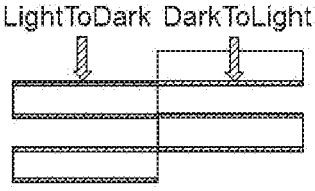
DarkToLight POLARITY

Fig. 15



EITHER POLARITY

Fig. 16



MIXED-POLARITY

Fig. 17

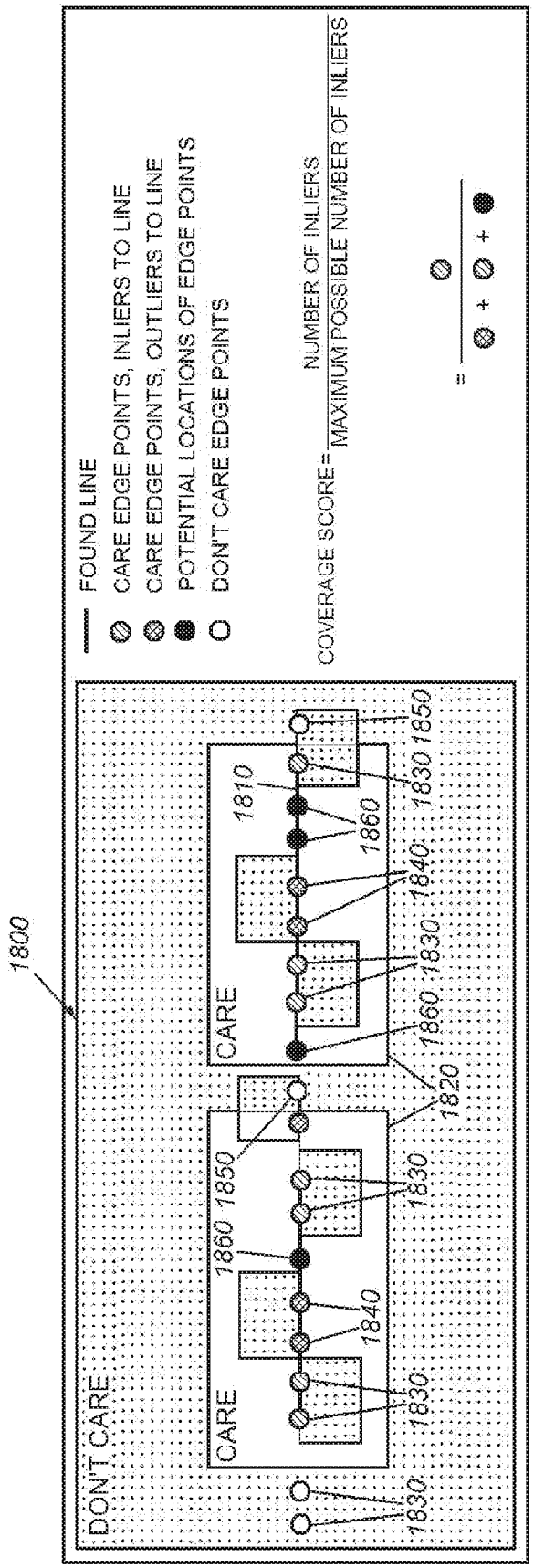

1800
DON'T CARE
CARE
CARE
1810
1820
1830
1840
1850
1860
FOUND LINE
CARE EDGE POINTS, INLIERS TO LINE
CARE EDGE POINTS, OUTLIERS TO LINE
POTENTIAL LOCATIONS OF EDGE POINTS
DON'T CARE EDGE POINTS
COVERAGE SCORE = NUMBER OF INLIERS / MAXIMUM POSSIBLE NUMBER OF INLIERS
= ⊘ / (⊘ + ⊘ + ●)

Fig. 18

SYSTEM AND METHOD FOR FINDING LINES IN AN IMAGE WITH A VISION SYSTEM

RELATED APPLICATION

[0001] This application claims the benefit of co-pending U.S. Patent Provisional Application Ser. No. 62/249,918, entitled SYSTEM AND METHOD FOR FINDING LINES IN AN IMAGE WITH A VISION SYSTEM, filed Nov. 2, 2015, the teachings of which are incorporated herein by reference.

FIELD OF THE INVENTION

[0002] This invention relates to machine vision systems, and more particularly to vision system tools that find line features in acquired images

BACKGROUND OF THE INVENTION

[0003] Machine vision systems (also termed herein, simply "vision systems") are used for a variety of tasks in manufacturing, logistics, and industry. Such tasks can include surface and part inspection, alignment of objects during assembly, reading of patterns and ID codes, and any other operation in which visual data is acquired and interpreted for use in further processes. Vision systems typically employ one or more cameras that acquire images of a scene containing an object or subject of interest. The object/subject can be stationary or in relative motion. Motion can also be controlled by information derived by the vision system, as in the case of manipulation of parts by a robot.

[0004] A common task for a vision system is finding and characterizing line features in an image. A variety of tools are used to identify and analyze such line features. Typically, these tools rely upon a sharp contrast difference that occurs in a portion of the image. This contrast difference is analyzed using e.g. a caliper tool to determine if the individual points in the image with contrast difference can be assembled into a line-like feature. If so, then a line is identified in the image. Notably, the tools that find edge points and those that attempt to fit a line to the points act independently of each other. This increases processing overhead and decreases reliability. Where an image contains multiple lines, such tools may be limited in ability to accurately identify them. Furthermore, traditional, line-finding tools that are designed to find a single line in an image can be problematic to use when the image contains multiple closely spaced lines with similar orientation and polarity.

SUMMARY OF THE INVENTION

[0005] This invention overcomes disadvantages of the prior art by providing a system and method for finding line features in an image that allows multiple lines to be efficiently and accurately identified and characterized. First, the process computes x and y-components of the gradient field at each location of the image, projects the gradient field over a plurality of image subregions, and detects within each subregion a plurality of gradient extrema, yielding a plurality of edge points with associated position and gradient. Next, the process iteratively chooses two edge points, fits a model line to those edge points, and if the gradients of those edge points are consistent with the model line, computes the full set of inlier points whose position and gradient are consistent with that model line. The candidate line with greatest inlier count is retained as a line result and the set of remaining outlier points is derived. The process then repeatedly applies the line fitting operation on this and subsequent outlier sets to find a plurality of line results. The line-fitting process can be exhaustive, or based on a random sample consensus (RANSAC) technique.

[0006] In an illustrative embodiment, a system for finding line features in an acquired image is provided. A vision system processor receives image data of a scene containing line features. An edge point extractor generates intensity gradient images from the image data and finds edge points based upon the intensity gradient images. A line-finder then fits the edge points to one or more lines based upon the intensity gradient in the edge points. Illustratively, the line finder operates a RANSAC-based process to fit inlier edge points to new lines including iteratively defining lines from outlier edge points with respect to previously defined lines. The edge point extractor performs a gradient field projection of line-feature-containing regions of the intensity gradient images. Illustratively the gradient field projection is oriented along a direction set in response to an expected orientation of one or more or the line features and the gradient field projection can define a granularity based on a Gaussian kernel. Illustratively, the image data can comprise data from a plurality of images acquired from a plurality of cameras and transformed into a common coordinate space. The image data can also be smoothed using a smoothing (weighting) kernel, which can comprise a 1D Gaussian kernel or another weighting function. The edge points can be selected based upon a threshold defined by an absolute contrast and a contrast that is normalized based on average intensity of the image data. Illustratively, the line finder is constructed and arranged to exchange edge points representing portions of parallel lines or crossing lines to correct erroneous orientations, and/or to identify lines with polarity variation, including mixed polarities in line features based on gradient values in the edge points. Also, illustratively, the edge point extractor is arranged to find a plurality of gradient magnitude maxima in each of the gradient projection sub-regions. These gradient magnitude maxima can be respectively identified as some of the plurality edge points, and can be described by a position vector and a gradient vector. Additionally, the line finder can be arranged to determine consistency between at least one edge point of the extracted plurality of edge points and at least one candidate line of the found plurality of lines by computing a metric. This metric can be based upon a distance of the at least one edge point from the candidate line and an angle difference between a gradient direction of the edge point and a normal direction of the candidate line.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The invention description below refers to the accompanying drawings, of which:

[0008] FIG. **1** is a diagram of an exemplary vision system arrangement acquiring images of an object that includes multiple edge features and a vision system processor including an edge-finding tool/module in accordance with an illustrative embodiment;

[0009] FIG. **2** is a diagram showing an overview of the system and method for edge-point extraction and line finding from an acquired image according to an illustrative embodiment;

[0010] FIG. **3** is a flow diagram of an edge point extraction procedure according to the system and method of FIG. **2**;

[0011] FIG. **4** is a diagram of a field projection on a region of an image containing edge features, which is part of the edge point extraction procedure of FIG. **3**;

[0012] FIG. **5** is a diagram showing application of a Gaussian kernel to the image to smooth the image, for use in the edge point extraction procedure of FIG. **3**;

[0013] FIG. **6** is a diagram of a field projection, including application of a Gaussian kernel for smoothing of the projection, for use in the edge point extraction procedure of FIG. **3**;

[0014] FIG. **7** is a diagram showing a graphical overview of the edge point extraction procedure of FIG. **3** including application of Gaussian kernels and calculation of absolute and normalized contrast thresholds for edge points;

[0015] FIG. **8** is graph showing a region of qualified contrasts for edge points having a sufficient absolute contrast threshold and normalized contrast threshold;

[0016] FIG. **9** is a flow diagram of a line-finding procedure based upon found edge points in FIG. **3**, using an exemplary RANSAC process according to an illustrative embodiment;

[0017] FIGS. **10** and **11** are diagrams showing erroneous and correct alignment of edge points relative to closely spaced parallel line features, respectively;

[0018] FIGS. **12** and **13** are diagrams showing correct and erroneous alignment of edge points relative to crossing line features, respectively, which can be resolved according to the line-finder of the illustrative system and method;

[0019] FIGS. **14-17** are diagrams showing examples of lines exhibiting, respectively, light-to-dark polarity, dark-to-light-polarity, either light-to-dark or dark-to-light polarity, or mixed polarity, which can be resolved according to the line-finder of the illustrative system and method; and

[0020] FIG. **18** is a diagram showing modification of the coverage score for a found line in view of a user-defined mask.

DETAILED DESCRIPTION

[0021] An exemplary vision system arrangement **100** that can be employed according to an illustrative embodiment is shown in FIG. **1**. The system **100** includes at least one vision system camera **110**, and can include one or more additional, optional cameras **112** (shown in phantom). The illustrative camera(s) **110**, **112** include(s) an image sensor (or imager) S and associated electronics for acquiring and transmitting image frames to a vision system process(or) **130** that can be instantiated in a standalone processor and/or a computing device **140**. The camera **110** (and **112**) includes an appropriate lens/optics **116** focused upon a scene that contains an object **150** under inspection. The camera **110** (and **112**) can include internal and/or external illuminators (not shown) that operate in accordance with the image acquisition process. The computing device **140** can be any acceptable processor-based system capable of storing and manipulating image data in accordance with the illustrative embodiment. For example, the computing device **140** can comprise a PC (as shown), server, laptop, tablet, smartphone or other similar device. The computing device **140** can include appropriate peripherals, such as a bus-based image capture card that interconnects to the camera. In alternate embodiments, the vision processor can be partially or fully contained within the camera body itself and can be networked with other PCs, servers and/or camera-based processors that share and process image data. The computing device **140** optionally includes an appropriate display **142**, which can support an appropriate graphical user interface (GUI) that can operate in accordance with vision system tools and processors **132** provided in the vision system process(or) **130**. Note that a display can be omitted in various embodiments and/or provided only for setup and service functions. The vision system tools can be part of any acceptable software and/or hardware package that is acceptable for use in the inspection of objects, such as those commercially available from Cognex Corporation of Natick, Mass. The computing device can also include associated user interface (UI) components, including, for example, a keyboard **144** and mouse **146**, as well as a touchscreen within the display **142**.

[0022] The camera(s) **110** (and **112**) image some or all of an object **150** located within the scene. Each camera defines an optical axis OA, around which a field of view is established based upon the optics **116**, focal distance, etc. The object **150** includes a plurality of edges **152**, **154** and **156** that are respectively arranged in different directions. For example, the object edges can comprise those of a cover glass mounted within a smartphone body. Illustratively, the camera(s) can image the entire object, or specific locations (e.g. corners where the glass meets the body). A (common) coordinate space can be established with respect to the object, one of the cameras or another reference point (for example a moving stage upon which the object **150** is supported). As shown, the coordinate space is represented by axes **158**. These axes illustratively define orthogonal x, y and z axes and rotation $\ominus_z$ about the z axis in the x-y plane.

[0023] According to an illustrative embodiment, the vision system process **130** interoperates with one or more applications/processes (running on the computing device **140**) that collectively comprise a set of vision system tools/processes **132**. These tools can include a variety of conventional and specialized applications that are used to resolve image data—for example a variety of calibration tools and affine transform tools can be used to transform acquired image data to a predetermined (e.g. common) coordinate system. Tools that convert image grayscale intensity data to a binary image based upon a predetermined threshold can also be included. Likewise, tools that analyze the gradient of intensity (contrast) between adjacent image pixels (and subpixels) can be provided.

[0024] The vision system process(or) **130** includes a line-finding process, tool or module **134** that locates multiple lines in an acquired image according to an illustrative embodiment. Reference is, thus, made to FIG. **2**, which graphically depicts an overview of a line-finding procedure **200** according to the illustrative embodiment. The procedure **200** consists of two primary parts. An input image **210** is provided to the processor. As shown, the image includes a pair of intersecting edges **212** and **214**. These can represent a corner region of the above-described object **150**. An edge point extractor **220** processes the input image **210** to obtain a set **230** of candidate edge points, including edge points **232** and **234** that respectively reside along edges **212** and **214**. The edge points **232**, **234**, and their associated data (e.g. intensity gradient information described below), are provided to a recursive line finder **240**, which performs a series of iterative processes on selected edge points. The goal of the iterative processes is an attempt to fit other found edge points to candidate line features. The line-finding process

240 results in found lines **252** and **254** as shown. These results can be provided to other downstream processes **260** that use the information—e.g. alignment processes, robot manipulation, inspection, ID reading, part/surface inspection, etc.

[0025] Reference is made to FIG. **3**, which describes a procedure for extracting edge points according to an embodiment. One or more images are acquired of the scene, which contains an object or surface having edge features to be found (step **310**). The image(s) can be extracted by a single camera or by a plurality of cameras. In either case, the image pixels can be (optionally) transformed by appropriate calibration parameters to a new and/or common coordinate space in step **320**. This step can also include smoothing of the image as described below. In certain embodiments, where a plurality of cameras image discontinuous regions of the scene—for example focusing on corner regions of a larger object—the common coordinate space can account for the empty region between camera fields of view. As described below, lines that extend between such fields of view (e.g. the object edge that connects two found corner regions) can be extrapolated by the system and method of the illustrative embodiment. The edge points required for finding lines are extracted from the image(s) in the appropriate coordinate space by the edge point extractor using gradient field projection in step **330**. Gradient values are first computed for each pixel, generating two images for x and y gradient components. The image(s) are further processed by projecting the gradient field over many caliper-like regions. Unlike a conventional caliper tool which projects the intensity value, by projecting the gradient field in accordance with the embodiment, gradient orientation can be preserved, which facilitates the subsequent line-finding process as described below.

[0026] In step **340**, and also referring to the diagram of FIG. **4**, a portion (a caliper-like region) **400** of the image containing the candidate edge features is subjected to a gradient field projection (represented by a plurality of projections **410**, **420**, **430**, searched across the (approximately) expected orientation of the edges in a search direction (arrow SD), with the projections repeated across the region **400** in an orthogonal projection direction (arrow PD). For each projection (e.g. projection **420**) edges appear as local maxima in a gradient field **440** associated with the projection. In general, a series of edge points within the projection that are associated with an edge will exhibit an intensity gradient (vectors **552**, **554**) orthogonal to the direction of extension of the edge. As described below, the user can define the projection direction based on expected line orientation. Alternatively, this can be provided by default or by another mechanism—e.g. analysis of the features in the image.

[0027] Two granularity parameters are involved in the above-described gradient projection step. Prior to gradient field calculation, the user can choose to smooth the image using an isotropic Gaussian kernel. A first granularity determines the size of this Gaussian smoothing kernel. As shown in the diagram **500** of FIG. **5**, an application of an appropriately sized Gaussian kernel (e.g. large **512**, medium **514**, small **516**) is made to smooth the image **210**. The first granularity parameter, hence, determines the size of the isotropic Gaussian smoothing kernel prior to field calculation.

[0028] After gradient field calculation, a Gaussian-weighted projection is thereby performed by the process, rather than uniform weighting in conventional caliper tools. Thus, a second granularity parameter determines the size of the one-dimensional (**1**D) Gaussian kernel used during field projection as shown in FIG. **6**, in which the region **600** is subjected to a Gaussian-smoothed kernel **610**, **620**, **630**. During a typical operation, the user verifies (using the GUI) all extracted edges overlaid on the image, and then adjusts the granularities and contrast thresholds until the number of extracted edges along the lines to be found appears satisfactory, while avoiding an excessive number of edges due to background noise in the image. In other words, this step allows the signal-to-noise ratio to be optimized for the image characteristic. This adjustment can also be performed automatically by the system, using a default value in various embodiments. Note that the use of a Gaussian weighting function is one of a variety of approaches for weighting the projection, including (e.g.) a uniform weighting.

[0029] The overall flow of gradient field extraction and projection is illustrated graphically in the diagram **700** of FIG. **7**. The two granularity parameters, the isotropic Gaussian kernel **710** and the **1**D Gaussian kernel **720**, are each shown in each half of the overall diagram **700**. As shown, each acquired image **210** is subjected to smoothing and decimation **730**. The resulting image **740** is then subjected to gradient field calculation **750**, as described above, to generate the two gradient images **752** and **754**. These gradient images are also represented as g_x and g_y, representing two orthogonal axes in the common coordinate space. Note that in addition to two gradient images, the intensity image **756** is also typically subjected to the smoothing, decimation and projection process **760** (using a Gaussian-weighted projection **770** based upon the 1D Gaussian kernel **720**) since the processed intensity information is also employed for calculating normalized contrasts in accordance with an embodiment—described below. The result is the projection profiles of gradient images **772**(g_x), **774** (g_y), and intensity image **776**.

[0030] Referring also to step **350** of the procedure **300** (FIG. **3**), qualified edge points are then extracted by combining the 1D projection profiles of both x & y gradient images. This is accomplished using a raw contrast calculation **780** and a normalized contrast calculation **790** based on the Intensity image. More particularly, any local peaks with both raw projected gradient magnitudes and normalized projected gradient magnitudes exceeding respective thresholds are considered a candidate edge point for subsequent line finding according to the following illustrative equations:

$$(g_x^2+g_y^2)^{1/2}>T_{ABS}$$

$$(g_x^2+g_y^2)^{1/2}/I>T_{NORM}$$

where g_x and g_y are the values of the x-gradient and y-gradient projections at a pixel location, respectively, I the intensity, T_{ABS} an absolute contrast threshold for raw projected gradient magnitudes and T_{NORM} is a normalized contrast threshold for intensity-normalized projected gradient magnitudes.

[0031] Notably, a point is only considered a candidate edge point when its absolute and normalized contrasts both exceed their respective thresholds. This is shown by the upper right quadrant **810** in the exemplary graph **800** of normalized contrast threshold T_{NORM} versus absolute contrast threshold T_{ABS}. The use of dual (absolute and normal-

ized) thresholds differs generally from existing approaches that typically employ an absolute contrast threshold. The benefits of dual contrast thresholds are clear, by way of example, when an image includes both dark and bright intensity regions that both include edges of interest. In order to detect edges in dark regions of the image, it is desirable to set a low contrast threshold. However, such a low contrast setting can result in the detection of false edges in the bright portions of the image. Conversely, in order to avoid the detection of false edges in the bright regions of the image, it is desirable to set a high contrast threshold. However, with a high contrast setting, the system may fail to adequately detect edges in dark regions of the image. By using a second normalized contrast threshold, in addition to the traditional absolute contrast threshold, the system can appropriately detect edges both in dark and bright regions, and avoid detecting false edges in bright regions of the image. Hence, by enabling the detection of relevant edges while avoiding spurious edges, the use of dual contrast thresholds serves to maximize the speed and robustness of the subsequent line-finding stage of the overall process.

[0032] Referring further to procedure step **350** (FIG. **3**), once all edge points are extracted, they are represented and stored in a data structure that is convenient for subsequent line-finders to operate upon. Note, for example, the following tuple:

$$p=(x,y,gx,gy,gm,go,I,gm/I,m,n)$$

where (x,y) is the location of the edge point, (g_x,g_y) are the values of its respective x-gradient and y-gradient projections, (g_m,g_o) is the gradient magnitude and orientation computed from (g_x,g_y), I is the intensity at the edge point location, g_m/I is the intensity-normalized contrast obtained by dividing the gradient magnitude g_m by the intensity I, m is the image index and n is the projection region index. The location of the edge point, as in the standard caliper tool, can be interpolated for improved accuracy.

[0033] Note that the edge-point extraction process generally operates to run field projections in a single direction that substantially matches the expected line angle. The tool is, therefore, most sensitive to edges at this angle, and its sensitivity falls off gradually for edges at other angles, where the rate of fall-off depend on the granularity settings that indirectly determine the field projection length. As a result, the process is limited to finding lines whose angle is "near" the expected line angle, subject to the angle range specified by the user. While the process is adapted to find lines that are not orthogonal, it is contemplated that it could be generalized in various embodiments to find lines of any angle over 360 degrees by performing projections in multiple directions (omnidirectional line-finding), including orthogonal directions.

[0034] With reference now to step **360** of the procedure **300** (FIG. **3**), thresholded edge point candidates are provided to the line-finder in accordance with an illustrative embodiment. By way of example, the line-finder operates recursively and employs (e.g.) RANdom SAmple Concensus (RANSAC)-based techniques. Reference is also made to the line-finding procedure **900** in FIG. **9**. In step **910**, the user specifies the maximum number of expected lines in an image, along with an expected angle, angle tolerance, distance tolerance, and (illustratively) a minimum coverage score (defined generally below) via (e.g.) the GUI. These parameters are used by the line-finder to operate the following processes. The lines are found for each subregion of the image by recursively running a RANSAC line finder, the edge point outliers from one stage becoming the input points to the next stage. Thus, in step **920**, the procedure **900** selects a pair of edge points that are part of the group of edge points identified as extrema in the edge-finding process. The procedure **900** attempts to fit a model line to the selected edge points based on matching gradient values (within the selected range of tolerance) that are consistent with a model line. In step **924**, one or more line candidate(s) from step **922** is/are returned. Each line-finding stage returns a candidate line, its inliers and outliers. The returned line(s) is/are subjected to a computation of inlier edge points that have a position and gradient consistent with the line candidate (step **926**). In step **928**, the candidate line with the greatest inlier count is identified. The above-described line-finding stage (steps **920-928**) terminates when it reaches the maximum number of RANSAC iterations allowed (decision step **930**). The maximum number of iterations inside each line-finding stage are computed automatically using an internally computed worst case proportion of outliers and an assurance level specified by the user. Each line finding stage returns the line with the maximum number of captured edge points out of all its iterations—subject to a user-specified fit tolerance, geometric constraints and polarity. Each edge point can only be assigned to the inlier list of a single line and each line is only allowed to contain at most one edge point from each projection region. The gradient orientation of an edge point, along with its position, is used to determine whether it should be included in the inlier list of a candidate line. In particular, edge points should have gradient orientation that is consistent with the angle of a candidate line.

[0035] If the decision step **930** determines that more iterations are permitted, the outliers from the best inlier candidate are returned (step **940**) to the RANSAC process (step **920**) for use in finding a line candidate.

[0036] During each RANSAC iteration, two edge points belonging to different projection regions are randomly selected and a line will be fit to those two points. The resulting candidate line receives further consideration only if its angle is consistent with the gradient angles of both edges in the point pair and if the angle of the line is consistent with the uncertainty range specified by the user. In general, the gradient direction of an edge point is nominally orthogonal, but is allowed to differ by a user-configured angle tolerance. If a candidate line passes these initial tests, then the number of inlier edge points will be evaluated, otherwise a new RANSAC iteration is initiated. An edge point will be regarded as an inlier of a candidate line only if its gradient direction and position are consistent with the line—based on gradient angle and distance tolerances specified by the user.

[0037] When the RANSAC iterations reach the maximum (decision step **930**), the inliers of the best found line candidate are subjected to an improved line fit, using (for example) a least squares regression or another acceptable approximation technique, and the set of inlier edge points will be reevaluated, repeating these steps a maximum of N (e.g. three or more) times until the number of inliers ceases to further increase or decrease (step **960**). This is the line that is indicated as found in step **970**.

[0038] The decision step **980** determines whether more lines are to be found (based (e.g.) on searching further sub regions or another criteria), and if so, the process loops back to step **920** to operate on a new set of edge points (step **982**).

When the points have been exhausted or a maximum iteration count is reached, the procedure **900** returns a set of (i.e. multiple) found lines in the image in step **990**.

[0039] The multi-line finder is adapted to perform a final adjustment of existing results in cases where two line results intersect one another within the inspection region. As illustrated generally in FIGS. **10** and **11**, for closely spaced parallel lines **1010** and **1020**, erroneous line results (i.e. FIG. **10**) can sometimes be obtained due to the statistical nature of the RANSAC procedure. However, when such errors occur, an exchange of inlier point groups (arrow **1120** in groups **1110** in FIG. **11**) can sometimes locate the correct lines with increased coverage scores and reduced-fit residuals. Point exchanges can be most effective when an image contains closely spaced parallel lines as shown. Conversely, when the image contains lines **1210** and **1220** that actually do cross each other as shown in FIGS. **12** and **13**, then coverage scores are reduced after a point exchange (arrow **1230** in group **1240** in FIG. **12**), so the original results obtained before the exchange are retained by the process to successfully detect crossing lines.

[0040] Note that the RANSAC procedure is one of a variety of techniques by which the line-finder can fit points to a line. In alternate embodiments, the candidate points can be selected according to a set displacement therebetween or the image can be processed using (e.g.) an exhaustive search technique. Thus, as used herein the reference to the RANSAC technique should be taken broadly to include a variety of similar point-fitting techniques.

[0041] Additional functionalities of this system and method can be provided. These include support for mixed-polarity, automatically computing the projection region width, support multi-view line finding, and allowing the input image be free of pre-warpage to remove optical distortion. These functionalities are described further below.

[0042] With further reference to the examples of FIGS. **14-16**, the line-finding system and method of the illustrative embodiment generally supports standard LightToDark, DarkToLight and Either polarity settings (respectively) for the contrast between edges that are found. In addition, the system and method can also support a mixed-polarity setting (FIG. **17**) in which both a Light-To-Dark and Dark-to-Light characteristic appears in the same line. Line-finding results of all four settings are illustrated in the following figure. In an illustrative embodiment, the system and method can include a mixed-polarity setting that allows finding of a single line that contains edge points of opposite polarities. This differs from a conventional setting of "Either" polarity in which all edge points of a single line are either polarity—but only one polarity. The mixed-polarity setting can be advantageous when used to analyze the light and dark checkerboards of (e.g.) a calibration plate, among other applications.

[0043] The user can select improved shift invariance of line-finding. In such case, the edge point extractor employs substantially overlapped projection regions to improve result stability. When the regions are non-overlapping, pixels under consideration can potentially move out of the projection regions when the image is shifted, resulting in poor shift invariance in line-finding results. Overlapped projection regions ensure that the pixels under consideration are continuously covered by projection regions. If overlapped projection regions are used, then incremental computation can be performed to maintain speed, along with possible low-level optimization.

[0044] The user can provide masks that omit certain portions of the acquired image and/or imaged surface from analysis for line features. This can be desirable where the surface includes known line features that are not of interest (e.g. barcodes that are analyzed by other mechanisms, text, and any other structures that are not germane to the task for which lines are to be found. Thus, the edge point extractor can support image masking where "don't care" regions in an image can be masked out, and "care" regions are masked in. Where such masking occurs, the coverage scores of the found lines is illustratively reweighted according to the number of edge points falling within the mask.

[0045] Reference is made to the exemplary image region **1800** of FIG. **18**, which shows coverage scores when image masks are present and the effect of image masking on such coverage scores. The edge point extractor supports image masking where "don't care regions" in an image can be masked out. As shown, the found line **1810** is characterized (based upon the "care" mask regions **1820**) by care edge points. Such care edge points consist of care edge point inliers **1830** to the line **1810** and care edge point outliers **1840** to the line **1810**. Don't care edge points **1850** on the line **1810** reside between care regions **1820** of the mask, as shown in this example, and are not included in the coverage score computation, even if they reside on the line as inliers. Potential locations **1860** for edge points along the line **1810** are also determined, as shown. These potential locations are positioned between known points at predictable spacing based on the spacing of found points. Illustratively, the coverage scores of the found lines are reweighted according to the number of edge points falling within the mask. The coverage score is, thus, modified as follows:

[0046] coverage score=number of care edge point inliers to line/(number of care edge point inliers to line+care edge point outliers to line+number of care potential locations of edge points).

[0047] After running the line-finding process according to the system and method herein, the found lines can be sorted in various ways based on sort criteria specified by the user (via (e.g.) the GUI). The user can choose from intrinsic sort measures such as inlier coverage score, intensity or contrast. The user can also choose from extrinsic sort measures such as signed distance or relative angle. When using extrinsic sort measures, the user can specify a reference line segment against which the extrinsic measures of the found lines are to be computed.

[0048] As described generally above, this system and method can include a Multi-Field-of-View (MFOV) overload, where a vector of images from different fields of view can be passed into the process. The images should all be in a common client coordinate space based upon a calibration. As noted above, this functionality can be extremely helpful in application scenarios where multiple cameras are used to capture partial areas of a single part. Because the edge points retain gradient information, line features that are projected between gaps in the field of view can still be resolved (when the gradients in both FOVs match for a given line orientation and alignment in each FOV.

[0049] Notably, the system and method does not require (allows the image to be free-of) removal of warpage (i.e. does not require the image to be unwarped) to remove nonlinear distortion, assuming the distortion is non-severe.

Where the image is not unwarped, the system and method can still detect candidate edge points, and map the point positions and gradient vectors through a nonlinear transform.

[0050] It should be clear that the line-finder provided according to the system, and method and various alternate embodiments/improvements is an effective and robust tool for determining multiple line features under a variety of conditions. In general, when used to find line features, the system and method has no particular limit on the maximum number of lines to be found in an image. Only memory and compute time will place practical limits on the number of lines that can be found.

[0051] The foregoing has been a detailed description of illustrative embodiments of the invention. Various modifications and additions can be made without departing from the spirit and scope of this invention. Features of each of the various embodiments described above may be combined with features of other described embodiments as appropriate in order to provide a multiplicity of feature combinations in associated new embodiments. Furthermore, while the foregoing describes a number of separate embodiments of the apparatus and method of the present invention, what has been described herein is merely illustrative of the application of the principles of the present invention. For example, as used herein the terms "process" and/or "processor" should be taken broadly to include a variety of electronic hardware and/or software based functions and components (and can alternatively be termed functional "modules" or "elements"). Moreover, a depicted process or processor can be combined with other processes and/or processors or divided into various sub-processes or processors. Such sub-processes and/or sub-processors can be variously combined according to embodiments herein. Likewise, it is expressly contemplated that any function, process and/or processor herein can be implemented using electronic hardware, software consisting of a non-transitory computer-readable medium of program instructions, or a combination of hardware and software. Additionally, as used herein various directional and dispositional terms such as "vertical", "horizontal", "up", "down", "bottom", "top", "side", "front", "rear", "left", "right", and the like, are used only as relative conventions and not as absolute directions/dispositions with respect to a fixed coordinate space, such as the acting direction of gravity. Additionally, where the term "substantially" or "approximately" is employed with respect to a given measurement, value or characteristic, it refers to a quantity that is within a normal operating range to achieve desired results, but that includes some variability due to inherent inaccuracy and error within the allowed tolerances of the system (e.g. 1-5 percent). Accordingly, this description is meant to be taken only by way of example, and not to otherwise limit the scope of this invention.

What is claimed is:

1. A system for finding line features in an acquired image comprising:

a vision system processor receiving image data of a scene containing line features, having an edge point extractor that

(a) computes a gradient vector field from the image data,

(b) projects the gradient vector field over a plurality of gradient projection sub-regions, and

(c) finds a plurality of edge points in respective of the gradient projection sub-regions based on the projected gradient data; and

a line-finder that generates a plurality of lines that are consistent with the edge points extracted from the image.

2. The system as set forth in claim **1** wherein the line finder operates a RANSAC-based process to fit inlier edge points to new lines including iteratively defining lines from outlier edge points with respect to previously defined lines.

3. The system as set forth in claim **1** wherein the gradient field projection is oriented along a direction set in response to an expected orientation of one or more or the line features.

4. The system as set forth in claim lwherein the gradient field projection defines a granularity based on a Gaussian kernel.

5. The system as set forth in claim **1** wherein the edge point extractor is arranged to find a plurality of gradient magnitude maxima in each of the gradient projection sub-regions, wherein the gradient magnitude maxima are respectively identified as some of the plurality edge points, being described by a position vector and a gradient vector.

6. The system as set forth in claim **1** wherein the line finder is arranged to determine consistency between at least one edge point of the extracted plurality of edge points and at least one candidate line of the found plurality of lines by computing a metric that is based upon a distance of the at least one edge point from the at least one candidate line and an angle difference between a gradient direction of the at least one edge point and a normal direction of the at least one candidate line.

7. The system as set forth in claim **1** wherein the image data includes data from a plurality of images acquired from a plurality of cameras, the images being transformed into a common coordinate space.

8. The system as set forth in claim **1** further comprising a smoothing kernel that smooths the image data.

9. The system as set forth in claim **8** wherein the smoothing kernel comprises a Gaussian kernel.

10. The system as set forth in claim **1** wherein the edge points are selected based upon a threshold defined by an absolute contrast and a contrast normalized based on average intensity of the image data.

11. The system as set forth in claim **1** wherein the line finder is arranged to exchange at least some edge points from the extracted plurality of edge points that represent portions of parallel lines or crossing lines from the found plurality of lines to correct erroneous orientations.

12. The system as set forth in claim **1** wherein the line finder is arranged to identify lines from the found plurality of lines having polarity variation

13. The system as set forth in claim **12** wherein the identified lines are lines defined by mixed polarities in the line features of the found plurality of lines based on gradient values in the extracted plurality of edge points.

14. A method for finding line features in an acquired image comprising the steps of:

receiving image data of a scene with a vision system processor, the data containing line features;

computing a gradient vector field from the image data;

projecting the gradient vector field over a plurality of gradient projection sub-regions;

finding a plurality of edge points in respective of the gradient projection sub-regions based on the projected gradient data; and

generating a plurality of lines that are consistent with the edge points extracted from the image.

15. The method as set forth in claim **14** further comprising applying a RANSAC-based process to fit inlier edge points to new lines, including iteratively defining lines from outlier edge points with respect to previously defined lines.

16. The method as set forth in claim **14** wherein the gradient field projection is oriented along a direction set in response to an expected orientation of one or more or the line features.

17. The method as set forth in claim **14** wherein the gradient field projection defines a granularity based on a 1D Gaussian kernel.

18. The method as set forth in claim **14** further comprising, finding a plurality of gradient magnitude maxima in each of the gradient projection sub-regions, wherein the gradient magnitude maxima are respectively identified as some of the plurality edge points, being described by a position vector and a gradient vector.

19. The method as set forth in claim **14** further comprising, determining consistency between at least one edge point of the extracted plurality of edge points and at least one candidate line of the found plurality of lines by computing a metric that is based upon a distance of the at least one edge point from the at least one candidate line and an angle difference between a gradient direction of the at least one edge point and a normal direction of the at least one candidate line

20. The method as set forth in claim **14** wherein the image data includes data from a plurality of images acquired from a plurality of cameras and transformed into a common coordinate space.

21. The method as set forth in claim **14** further comprising smoothing the image data using a smoothing kernel.

22. The method as set forth in claim **14** further comprising selecting the edge points based upon a threshold defined by an absolute contrast and a contrast normalized based on average intensity of the image data.

23. The method as set forth in claim **14** further comprising at least one of (a) exchanging edge points representing portions of parallel lines or crossing lines to correct erroneous orientations and (b) identifying lines with polarity variation in the line features, including mixed polarities in lines based on gradient values in the edge points.

* * * * *