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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 June 2014 (19.06.2014)

W I P O | P C T

(10) International Publication Number
WO 2014/093824 A1

(51) International Patent Classification:
A61B 19/00 (2006.01) *A61B 1/045* (2006.01)

(21) International Application Number:
PCT/US20 13/0750 14

(22) International Filing Date:
13 December 2013 (13. 12.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/737,172 14 December 2012 (14. 12.2012) US

(71) Applicant: **THE TRUSTEES OF COLUMBIA UNI¬VERSITY IN THE CITY OF NEW YORK** [US/US]; 412 Low Memorial Library, 535 West 116th Street, New York, NY 10027 (US).

(72) Inventors: **REITER, Austin;** 255 Great Neck Road, Apt. 310, Great Neck, NY 11021 (US). **ALLEN, Peter, K.;** 60 Broadway, Pleasantville, NY 10570 (US).

(74) Agent: **CHIARINI, Lisa, A.;** Hughes Hubbard & Reed LLP, One Battery Park Plaza, New York, NY 10004 (US).

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

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

Published:

— *with international search report (Art. 21(3))*

(54) **Title:** MARKERLESS TRACKING OF ROBOTIC SURGICAL TOOLS

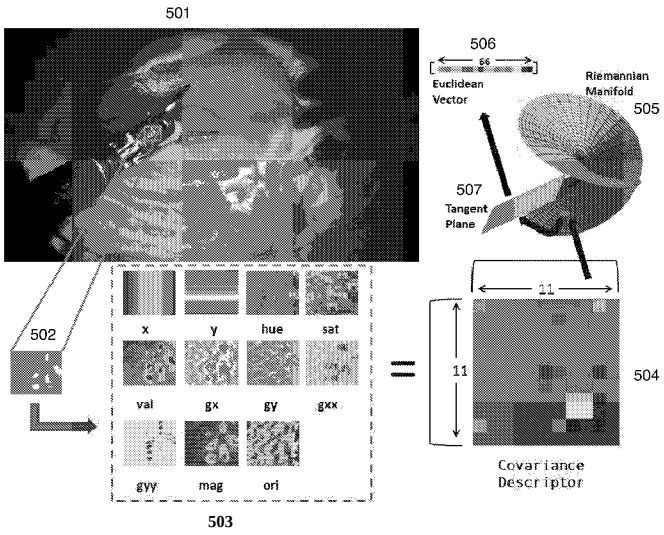

FIG. 5

(57) **Abstract:** Appearance learning systems, methods and computer products for three-dimensional markerless tracking of robotic surgical tools. An appearance learning approach is provided that is used to detect and track surgical robotic tools in laparoscopic sequences. By training a robust visual feature descriptor on low-level landmark features, a framework is built for fusing robot kinematics and 3D visual observations to track surgical tools over long periods of time across various types of environments. Three-dimensional tracking is enabled on multiple tools of multiple types with different overall appearances. The presently disclosed subject matter is applicable to surgical robot systems such as the da Vinci® surgical robot in both ex vivo and in vivo environments.

WO 2014/093824 A1

MARKERLESS TRACKING OF ROBOTIC SURGICAL TOOLS

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 61/737,172, filed December 14, 2012.

BACKGROUND OF THE DISCLOSED SUBJECT MATTER

Field of the Disclosed Subject Matter

[0002] Embodiments of the disclosed subject matter relate generally to three-dimensional markerless tracking of robotic medical tools. More particularly, embodiments of the subject matter relate to systems, methods, and computer products for the acquisition and tracking of robotic medical tools through image analysis and machine learning.

Description of Related Art

[0003] Technological breakthroughs in endoscopy, smart instrumentation, and enhanced video capabilities have allowed for advances in minimally invasive surgery. These achievements have made it possible to reduce the invasiveness of surgical procedures. Computer-aided surgical interventions have been shown to enhance the skills of the physician, and improve patient outcomes. Particularly, robotic hardware and intelligent algorithms have opened the doors to more complex procedures by enhancing the dexterity of the surgeon's movements as well as increasing safety through mechanisms like motion scaling and stereo imaging. To further enhance the surgeon's abilities, robotic surgery systems may include tool tracking functionality

to determine the locations of instruments within the surgical field whether within sight of the surgeon or not.

[0004] Knowledge of the location and orientation of medial tools in the endoscopic image can enable a wide spectrum of applications. For example, accurate tool localization can be used as a Virtual Ruler capable of measuring the distances between various points in the anatomical scene, such as the sizes of anatomical structures. Graphical overlays can indicate the status of a particular tool, for example, in the case of the firing status of an electro-cautery tool. These indicators can be placed at the tip of the tool in the visualizer which is close to the surgeon's visual center of attention, enhancing the overall safer}' of using such tools. It can also be useful in managing the tools that are off the screen increasing patient's safety, or for visual serving of motorized cameras.

[0005] Tool tracking techniques are generally divided into marker-based systems and marker-less systems. As an example of a markerless tool tracking system, the joints of a robotic surgical system can be equipped with encoders so that the pose of the instruments can be computed through forward kinematics. However, the kinematics chain between the camera and the tool tip can involve on the order of 18 joints over 2 meters. As a result, such approaches are inaccurate, resulting in absolute error on the order of inches.

[0006] Previous approaches to marker-based tool tracking have employed specialized fiducial markers to locate the tool *in vivo.* There are practical challenges to these approaches such as manufacturability and cost. In some approaches, either color or texture is used to mark the tool, and in cases where information about the tool is known *a priori,* a shape model can be used to confine the search space. One method is to design a custom marker for the surface of the

surgical tool, to assist in tool tracking. In one approach, a color marker is designed by analyzing the Hue-Saturation- Value color space to determine what color components aren't common in typical surgical imagery, and the marker is fabricated and placed on a tool to be tracked. A training step creates a kernel classifier which can then label pixels in the frame as either foreground (tool) or background. In some approaches, a marker may comprise three stripes that traverse the known diameter of the tool which allows the estimation of depth information of the tool's shaft from the camera. An alternative example of a marker is a barcode.

[0007] Another technique to aid in tracking is to affix assistive devices to the imaging instrument itself. For example, a laser-pointing instrument holder may be used to project laser spots into the laparoscopic imaging frames. This is useful for when the tools move out of the field-of-view of the camera. The laser pattern projected onto the organ surface provides information about the relative orientation of the instrument with respect to the organ. Optical markers are used on the tip of the surgical instruments, and these markers used in conjunction with the image of the projected laser pattern allow for measurements of the pointed organ and the instrument,

[0008] Prior approaches to visual feature detection and matching in the computer vision community have applied scale and affine invariant feature descriptors, which have been very successful in matching planar features. However, they work poorly for features on metal surfaces with lighting changes, as in the case of surgical tools with varying poses and light directions. Other prior approaches either result in low accuracy or require the addition of distracting or impractical additional indicia to the surfaces of tools. Thus, there remains a need for an accurate non-invasive tool tracking system that provides knowledge of the locations of tools and enables the use of precise graphical overlays.

SUMMARY OF THE DISCLOSED SUBJECT MATTER

[0009] In one aspect of the disclosed subject matter, a robotic surgical tool tracking method and computer program product is provided. A descriptor of a region of an input image is generated. A trained classifier is applied to the descriptor to generate an output indicative of whether a feature of a surgical tool is present in the region. The location of the feature of the surgical tool is determined based on the output of the trained classifier.

[0010] In some embodiments, the descriptor is a covariance descriptor, a scale invariant feature transform descriptor, a histogram-of-orientation gradients descriptor, or a binary robust independent elementary features descriptor. In some embodiments, the trained classifier is a randomized tree classifier, a support vector machine classifier, or an AdaBoost classifier. In some embodiments, the region is selected from within a predetermined area of the input image. In some embodiments, the region is selected from within a mask area indicative of the portion of the input image that corresponds to a tip portion of the surgical tool. In some embodiments wherein the input image contains a plurality of surgical tools, it is determined to which of the plurality of surgical tools the feature corresponds. In some embodiments, the mask area is generated by applying a Gaussian mixture model, image segmentation by color clustering, image segmentation by thresholding, or image segmentation by application of a graph cut algorithm.

[0011] In some embodiments where the descriptor is a covariance descriptor, the covariance descriptor comprises an x coordinate, a y coordinate, a hue, a saturation, a color value, a first order image gradient, a second order image gradient, a gradient magnitude, and a gradient orientation. In some embodiments where the classifier is a randomized tree classifier, the randomized tree classifier additionally comprises weights associated with each tree and applying

the classifier comprises applying the weights associated with each tree to the outputs of each tree.

[0012] It is to be understood that both the foregoing general description and the following detailed description are exemplary and are intended to provide further explanation of the disclosed subject matter claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] FIG. 1 is a schematic diagram showing the modules of an exemplary embodiment of a system according to the disclosed subject matter.

[0014] FIGS. 2A-P depicts sample input and output of the Scene Labeling Module according to embodiments of the disclosed subject matter.

[0015] FIGS. 3A-C depict robotic surgical tools according to embodiments of the present disclosure.

[0016] FIG. 4 depicts seven naturally-occurring landmarks on a robotic surgical tool in accordance with the system of the present disclosure.

[0017] FIG. 5 provides a schematic view of a feature descriptor in accordance with an embodiment of the present subject matter.

[0018] FIG. 6 depicts shaft boundary detection output in accordance with an embodiment of the present subject matter.

[0019] FIGS, **7A-J** depict kinematics output in accordance with an embodiment of the present subject matter.

[0020] FIGS, 8A-B depict an evaluation scheme in accordance with the disclosed subject matter.

[0021] FIGS, 9A-D depict an example of kinematic latency.

[0022] FIGS, 10A-B depict applications of tool tracking in accordance with the present disclosure.

[0023] FIG. 11 depicts example appearance changes in a robotic surgical tool typically encountered under different lighting and perspective effects in accordance with the present disclosure.

[0024] FIG. 12A shows seven features of a robotic surgical tool that are analyzed in accordance with the present subject matter.

[0025] FIG. 12B-H show sample likelihoods on the tip of the robotic surgery tool of FIG. 12A tool overlaid with extrema locations in accordance with the present subject matter.

[0026] FIG. 13 is a histogram depicting relative performance of several combinations of descriptors and classifiers according to embodiments of the present disclosure.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0027] Reference will now be made in detail to exemplary embodiments of the disclosed subject matter, examples of which are illustrated in the accompanying drawing. The method and corresponding steps of the disclosed subject matter will be described in conjunction with the detailed description of the system.

[0028] Generally, the subject matter described herein provides a system, method, and computer product for tracking robotic surgical tools *in vivo* or *ex vivo* via image analysis that provides a level of accuracy not available in existing systems.

[0029] In one aspect, a tracking system is provided that learns classes of natural landmarks on articulated tools off-line. The system learns the landmarks by training an efficient multi-class classifier on a discriminative feature descriptor from manually ground-truthed data. The classifier is run on a new image frame to detect all extrema representing the location of each feature type, where confidence values and geometric constraints help to reject false positives. Next, stereo matching is performed with respect to the corresponding camera to recover 3D point locations on the tool. By knowing *a priori* the locations of these landmarks on the tool part (from the tool's CAD model), the pose of the tool is recovered by applying a fusion algorithm of kinematics and these 3D locations over time and computing the most stable solution of the configuration. Multiple tools are handled simultaneously by applying a tool association algorithm and the system of the presently disclosed subject matter is able to detect features on different types of tools. The features detected are small-scaled (~2% of the image), vary in the amount of texture, and are observed under many different perspective views. The features are designed to be used within a marker-less pose estimation framework which fuses kinematics with vision, although this is out-of-the-scope of the current paper. The learning system of the presently disclosed subject matter is extends to multiple tool types and multiple tools tracked simultaneously as well as various types of surgical data.

[0030] The da Vinci ® surgical robot is a tele-operated, master-slave robotic system. The main surgical console is separated from the patient, whereby the surgeon sits in a stereo viewing console and controls the robotic tools with two Master Tool Manipulators (MTM) while viewing

stereoscopic high-definition video. The patient-side hardware contains three robotic manipulator arms along with an endoscopic robotic arm for the stereo laparoscope. A typical robotic arm has 7 total degrees-of- freedom (DOFs), and articulates at the wrist. The stereo camera system is calibrated for both intrinsics and stereo extrinsics using standard camera calibration techniques. Although the cameras have the ability to change focus during the procedure, a discrete number of fixed focus settings are possible, and camera calibration configurations for each setting are stored and available at all times, facilitating stereo vision approaches as described below.

[0031] FIG. 1 provides an overview of the modules and algorithm of a detection and tracking system in accordance with an embodiment of the disclosed subject matter. Generally, the system includes a Scene Labeling Module 101 which applies a multi-feature training algorithm to label all pixels in an image of an anatomical scene with medical tool(s), a Feature Classification Module 102 which uses a classifier on feature descriptors to localize known landmarks on the tool tips, and a Shaft Extraction Module 103 that uses a shaft mask from the Scene Labeling Module 101 to fit cylinders to the shaft pixels in the image for all visible tools, whenever possible. A Patient-Side Manipulator (PSM) Association Module 104 uses class-labeled feature detections output from the Feature Classification Module 102 to determine which feature is associated with which tool in the image and a Fusion and Tracking Module 105 takes outputs from both the Shaft Extraction Module 103 and the Patient-Side Manipulator Association Module 104 to fuse visual observations with raw kinematics and track the articulated tools over time. In the paragraphs that follow, each of these modules is explained further.

Scene Labeling Module

[0032] Scene Labeling Module 101 labels every pixel in an input image. Referring to **FIG.** 2A, the input image is the scene image 201, which typically includes the anatomical scene 202 and

medical tool(s) 203 and 204. The scene is labeled with one of three classes: Metal, Shaft, or Background. A Gaussian Mixture Model (GMM) of several color and texture features is learned off-line for each of these three classes. Subsequently, a class-conditional probability is assigned for each of the classes to every pixel and a label is assigned.

[0033] FIG. 2 shows an example result of the pixel labeling routine described with reference to FIG. 1. FIG. 2A shows the original image 201 from an *in-vivo* porcine sequence of first and second robotic tools 203 and 204 performing a suturing procedure using the da Vinci ® Surgical System. FIG. 2B shows the metal likelihood *(e.g.,* tool tip, clevis), with mask regions 205 and 206 corresponding to the highest probability locations of metal. FIG. 2C shows the shaft likelihood, with mask regions 207 and 208 corresponding to the highest probability locations of shaft. FIG. 2D shows the background likelihood, with mask region 209 corresponding to the highest probability location of the background. The metal class represents all pixels located at the distal tip of the tool, from the clevis to the grippers. All of the features to be detected by the Feature Classification Module 102 are located in this region. Additionally, it is described below how the shaft class is used to fit a cylinder to the tool's shaft, whenever possible.

[0034] Typically, surgeries performed with the da Vinci © are quite zoomed in, and so the shaft is not usually visible enough to fit a cylinder (the typical approach to many tool tracking algorithms)). However, at times the camera is zoomed out and so this scene pixel labeling routine allows the algorithm to estimate the 6-DOF pose of the shaft as additional information. By estimating the approximate distance of the tool from the camera using stereo matching of sparse corner features on the tool's tip, it can be determined if the shaft is visible enough to attempt to fit a cylinder. When the camera is zoomed out, although the shaft is visible the features on the tool tip are not so easily detected. Therefore, recognition can be based on a

combination of shaft features, tool-tip features, and a hybrid in between depending on the distance of the tool to the camera. These pixel labelings help to assist in both feature detection and shaft detection, as described further in the following text.

Feature Classification Module

[0035] Feature Classification Module 102 analyzes only the pixels which were labeled as Metal by Scene Labeling Module 101 (mask regions 205 and 206 of FIG. **2B).** This reduces both the false positive rate as well as the computation time, helping to avoid analyzing pixels which are not likely to be one of the features of interest (because they are known beforehand to be located on the tool tip). A multi-class classifier is trained using a discriminative feature descriptor. Class-labeled features are then localized in the image. Next, these candidate feature detections are stereo matched and triangulated to localize as 3D coordinates. These feature detection candidates are analyzed further using known geometric constraints to remove outliers and then are fed into the fusion and tracking stage of the algorithm.

[0036] According to one aspect of the present subject matter, data is collected for the purposes of training the classifier. In one embodiment, nine different video sequences are used that span various in-vivo experiments, to best cover a range of appearance and lighting scenarios. For training, a Large Needle Driver (LND) tool can be used (shown in **FIG. 3A).** However, as discussed below, this will extend well to other types of tools, such as the Maryland Bipolar Forceps (MBF) (shown in FIG. **3B)** and Round Tip Scissors (RTS) (shown in **FIG.** 3C). With training only on the Large Needle Driver, the system of the present disclosure is able to track on the Large Needle Driver, Maryland Bipolar Forceps and Round Tip Scissors. Seven naturally-occurring landmarks are manually selected as shown in **FIG.** 4 overlain on an image of the LND. The features chosen are of the pins that hold the distal clevis together 401, 402 and 403, the IS

logo in the center 404, the wheel 405, wheel pin 406, and the iDot 407. From time-to-time this combination of landmarks is referred to as a marker pattern, Mi. The features chosen may also include known, invariant locations on the mid-line of the shaft axis to this marker partem to be used in the fusion module.

[0037] For each frame in the ground truth procedure, the best encompassing bounding-box is manually dragged around each feature of interest, to avoid contamination from pixels which don't belong to the tool. To obtain as large a dataset as possible with reasonable effort, some embodiments of the presently disclosed subject matter coast through small temporal spaces using Lucas-Kanade optical flow (KLT) to predict ground truth locations between user clicks as follows; the user drags a bounding-box around a feature of interest; the software uses KLT optical flow to track this feature f om frame-to-frame (keeping the same dimensions of the box); as the user inspects each frame, if either the track gets lost or the size changes, the user drags a new bounding-box and starts again until the video sequence ends.

[0038] This allows for faster ground truth data collection while still manually-inspecting for accurate data. Overall, a training set can comprise ~20,000 total training samples across the seven feature classes.

[0039] A feature descriptor capable of discriminating these feature landmarks from each other robustly is disclosed. A discriminative and robust region descriptor to describe the feature classes is required because each feature is fairly small (*e.g.*, 17-**25** pixels wide, or ~ 2% of the image). In one embodiment of the present subject matter, a Region Covariance Descriptor is used, where the symmetric square covariance matrix of $d$features in a small image region serves as the feature descriptor (depicted in **FIG. 5).** Given an image I of size $[W \times H]$, d-ll features

are extracted, resulting in a $[W \times H \times d]$ feature image, as shown in equation (1), where x, y are the pixel locations; *Hue, Sat, Val* are the hue, saturation, and luminance values from the HSV color transformation at pixel location (x, y); I_x, I_y are the first-order spatial derivatives; I_{xx}, I_{yy} are the second-order spatial derivatives; and the latter two features are the gradient magnitude and orientation, respectively. The first two pixel location features are useful because their correlation with the other features are present in the off-diagonal entries in the covariance matrix. The $[d \times d]$ covariance matrix C_R of an arbitrary rectangular region R within F then becomes our feature descriptor.

$$F = [x\ y\ Hue\ Sat\ Val\ I_x\ I_y\ I_{xx}\ I_{yy}\ \sqrt{I_x^2 + I_y^2}\ \arctan\left(\frac{I_y}{I_x}\right)] \qquad (1)$$

[0040] According to some embodiments of the present subject matter and as shown in FIG. 5, several independent features are combined compactly into a single feature descriptor. Eleven features are used overall (shown in the dotted box 503), specifically the (x, y) locations, hue/saturation/value color measurements, first and second order image gradients, and gradient magnitude and orientation. A rectangular region 502 (inset box shown zoomed from the original image 501) of the image 501 is described by using the covariance matrix of these 11 features within that region, yielding an 11x11 symmetric matrix 504. In order to use this matrix as a descriptor with typical linear mathematical operations, it must be mapped from it's natural Riemannian space 505 to a vector space using Lie Algebra techniques, yielding a 66-dimensional vector space descriptor 506, described in further detail below.

[0041] Each C_R can be computed efficiently using integral images. The sum of each feature dimension as well as the sum of the multiplication of every two feature dimensions is computed. Given these first and second order integral image tensors, the covariance matrix 504 of any

rectangular region 502 can be extracted in $O(d^2)$ time. Using the ground truth data collected in accordance with the method given above, covariance descriptors of each training feature are extracted and the associated feature label is stored for training a classifier. However, the *d*-dimensional noiisingular covariance matrix descriptors 504 cannot be used as is to perform classification tasks directly because they do not lie on a vector space, but rather on a connected Riemamiian manifold 505, and so the descriptors must be post-processed to map the $[d \times d]$ dimensional matrices C_R 540 to vectors $G_J \in \mathbb{R}^{d(d+1)/2}$ 506.

[0042] Methods for post-processing the covariance descriptors to a vector space are known in the art. Symmetric positive definite matrices, of which the noiisingular covariance matrices above belong, can be formulated as a connected Riemamiian manifold 505. A manifold is locally similar to a Euclidean space, and so every point on the manifold has a neighborhood in which a homeomoiphism can be defined to map to a tangent vector space. According to one embodiment of the present subject matter, the $[d \times d]$ dimensional matrices above 504 are mapped to a tangent space 507 at some point on the manifold 505, which will transform the descriptors to a Euclidean multi-dimensional vector-space for use within the classifier according to the following method. Given a matrix X, the manifold-specific exponential mapping at the point Y is defined according to equation (2), and logarithmic mapping according to equation (3).

$$exp_X(Y) = X^{\frac{1}{2}}\exp(X^{-\frac{1}{2}}YX^{-\frac{1}{2}})X^{\frac{1}{2}} \quad (2)$$

$$log_X(Y) = X^{\frac{1}{2}}\log(X^{-\frac{1}{2}}YX^{-\frac{1}{2}})X^{\frac{1}{2}} \quad (3)$$

[0043] In these formulations, exp and log are the ordinary matrix exponential and logarithmic operations. An orthogonal coordinate system is defined at a tangent space with the vector operation. To obtain the vector-space coordinates at X for manifold point Y, the operation of

equation (4) is perfumed, where ***upper*** extracts the vector form of the upper triangular part of the matrix. In the end, the result is a vector space with dimensionality $\boldsymbol{q} = \frac{d(d+1)}{2}$.

$$\boldsymbol{vec}_{\boldsymbol{x}}(\boldsymbol{Y}) = \boldsymbol{upper}(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{Y}\boldsymbol{X}^{-\frac{1}{2}}) \qquad (4)$$

[0044] The manifold point at which a Euclidean tangent space is constructed is the mean covariance matrix of the training data. To compute the mean matrix $\mu_{\text{¾}}$ in the Riemannian space, the sum of squared distances is minimized according to equation (5). This can he computed using the update rule of equation (6) in a gradient descent procedure. The logarithmic mapping of Y at μ_{CR} is used to obtain the final vectors. The training covariance matrix descriptors are mapped to this Euclidean space and are used to train the multi-class classifier, described below.

$$\mu_{C_R} = \underset{\boldsymbol{Y} \in \boldsymbol{M}}{argmin} \sum_{i=1}^{N} \mathbf{d}^2(\boldsymbol{X}_i, \boldsymbol{Y}) \qquad (5)$$

$$\mu_{C_R}^{t+1} = exp_{\mu_{C_R}^{t}}\left[\frac{1}{N}\sum_{i=1}^{N} log_{\mu^t}(\boldsymbol{X}_i)\right] \qquad (6)$$

[0045] Various multi-class classifiers known in the art may suit this problem. However, runtime is an important factor in the choice of a learning algorithm to be used in accordance with the present subject matter. Consequently, in one embodiment of the present disclosure, multi-class classification is performed using a modified Randomized Tree (RT) approach. In addition to providing feature labels, the approach of the present disclosure, allows retrieval of confidence values for the classification task which will be used to construct class-conditional likelihood images for each class. Various feature descriptors, such as Scale-Invariant Feature Transforms (SIFT), Histograms-of-Oriented Gradients (FioG), and the Covariance Descriptors previously discussed may be paired with various classification algorithms such as Support Vector Machines

(SVM) or the two variants on RTs, described below. This results in a total of nine possible descriptor/classifier combinations: SIFT/SVM, SIFT/RT, SIFT/BWRT, HoG/SVM, HoG/RT, HoG/BWRT, Covar/SVM, Covar/RT, and Covar/BWRT. In one embodiment of the present disclosure, the Covariance Descriptor is paired with the adapted RTs to achieve a sufficient level of accuracy and speed.

[0046] SIFT has been used as a descriptor for feature point recognition/matching and is often used as a benchmark against which other feature descriptors are compared. It has been shown that SIFT can be well approximated using integral images for more efficient extraction. In one embodiment of the present disclosure, ideas based on this method may be used for classifying densely at many pixels in an image.

[0047] HoG descriptors describe shape or texture by a histogram of edge orientations quantized into discrete bins (in one embodiment of the present disclosure, 45 are used) and weighted on gradient magnitude, so as to allow higher-contrast locations more contribution than lower-contrast pixels. These can also be efficiently extracted using integral histograms.

[0048] An SVM constructs a set of hyperplanes which seek to maximize the distance to the nearest training point of any class. The vectors which define the hyperplanes can be chosen as linear combinations of the feature vectors, called Support Vectors, which has the effect that more training data may produce a better overall result, but at the cost of higher computations. In an alternative embodiment of the present disclosure using SVM, Radial Basis Functions are used as the kernel during learning.

[0049] RTs naturally handle multi-class problems very efficiently while retaining an easy training procedure. The RT classifier Λ is made up of a series of L randomly-generated trees

$\Lambda = [\gamma_1, \ldots, \gamma_L]$, each of depth m. Each tree ¾ for i 6 1, . . . , L, is a fully-balanced binary tree made up of internal nodes, each of which contains a simple, randomly-generated test that splits the space of data to be classified, and leaf nodes which contain estimates of the posterior distributions of the feature classes.

[0050] To train the tree, the training features are dropped down the tree, performing binary tests at each internal node until a leaf node is reached. Each leaf node contains a histogram of length equal to the number of feature classes IK which in one embodiment of the present disclosure is seven (for each of the manually chosen landmarks shown in FIG. 4). The histogram at each leaf counts the number of times a feature with each class label reaches that node. At the end of the training session, the histogram counts are turned into probabilities by normalizing the counts at a particular node by the total number of hits at that node. A feature is then classified by dropping it down the trained tree, again until a leaf node is reached. At this point, the feature is assigned the probabilities of belonging to a feature class depending on the posterior distribution stored at the leaf from training.

[0051] Because it is computationally infeasible to perform all possible tests of the feature, L and m are chosen so as to cover the search space sufficiently and to best avoid random behavior. In one embodiment, L = 60 trees are used, each of depth m = 11. Although this approach is suitable for matching image patches, traditionally the internal node tests are performed on a small patch of the luminance image by randomly selecting 2 pixel locations and performing a binary operation (less than, greater than) to determine which path to take to a child. In one embodiment, feature descriptor vectors are used rather than image patches, and so the node tests are adapted to suit this specialized problem.

[0052] In one embodiment of the disclosed subject matter, for each internal tree node a random linear classifier h_i to feature vector x is constructed to split the data as shown in equation (7), where $\mathbf{n}$ is a randomly generated vector of the same length as feature x with random values in the range [-1, 1] and $z \in [-1, 1]$ is also randomly generated. This test allows for robust splitting of the data and is efficiently utilized as it is only a dot product, an addition, and a binary comparison per tree node. In this way, the tree is trained with vectorized versions of the covariance descriptors and build up probability distributions at the leaf nodes. The resulting RT classifier Λ is the final multi-class classifier. The results from each tree γ_i are averaged across all L trees. However, even with relatively small values for L and m for computation purposes, the search space is still quite large given the appreciable amount of choices for randomly-created linear dot products at the internal tree nodes, and this leaves the training approach susceptible to randomness. To alleviate this, the approach is further modified from a conventional RT.

$$h_i = \begin{cases} \mathbf{n}^T \mathbf{x} + z \leq 0 & go\ to\ right\ child \\ otherwise & go\ to\ left\ child \end{cases} \qquad (7)$$

[0053] In one aspect of the present subject matter, an improved RT approach is disclosed, which is referred to as Best Weighted Randomized Trees (BWRT). Each tree γ_i is essentially a weak classifier, but some may work better than others, and can be weighted according to how well they behave on the training data. Because of the inherent randomness of the algorithm and the large search space to be considered, an improvement is shown by initially creating a randomized tree bag Ω of size $E \gg L$. This allows us initial consideration of a larger space of trees, but after evaluation of each tree in Ω on the training data, the best L trees are selected for inclusion in the final classifier according to an error metric.

[0054] The latter point allows consideration of more of the parameter space when constructing the trees while retaining the computational efficiency of RTs by only selecting the best performers. In order to evaluate a particular tree on the training data, the posterior probability distributions at the leaf nodes is considered. First, the training data is split into training and validation sets (*e.g.*, ---70% is used to train and the rest to validate). Next, all trees from the training set in Ω are trained as usual. Given a candidate trained tree $\tilde{\gamma}_i \in \Omega$, each training sample is dropped from the validation set through f_i until a leaf node is reached. Given training feature X_j and feature classes 1, . . . , *b*, the posterior distribution at the leaf node contains *b* conditional probabilities $p_{\gamma_i}(y|X_j)$ where y € 1, . . . , b. To evaluate the goodness of tree f_i on X_j, $p_{\gamma_i}(y_j|X_j)$ is compared to the desired probability 1 of label y_j, and accumulate the root-mean squared (RMS) error of all training features Xj across all validation trees in Ω. The top *L* trees (according to the lowest RMS errors) are selected for the final classifier Λ. In some embodiments, the initial bag size is *E* = 125,000 candidate tree classifiers, cut down to *L* = 60 trained trees for the final classifier.

[0055] In one aspect of the disclosed subject matter, in addition to selecting the best trees in the bag, the error terms are used as weights on the trees. Rather than allowing each tree to contribute equally to the final averaged result, each tree is weighted as one-over-RMS so that trees that label the validation training data better have a larger say in the final result than those which label the validation data worse. As such, for each $\gamma_i \in \Lambda$ an associated weight w_i is computed such that $w_i = 1/rms_i$ where rms_i is the accumulated RMS error of tree γ_i on the validation data. At the end, all weights w_i for $i \in 1, \ldots, L$ are normalized to sum to 1 and the final classifier result is a weighted average using these weights.

[0056] Given the trained classifier Λ, features for eacli class label are detected on a test image by computing dense covariance descriptors **CR** *(e.g. ,* at many locations in the image) using the integral image approach for efficient extraction. Each **CR** is mapped to a vector space using the mean covariance μ_{C_R} of the training data as previously described, producing a Euclidean feature C_j. Each c_j is dropped through the trees γ_l and the probabilities are averaged at the obtained leaf nodes to get a final probability distribution $p_{l,}$, representing the probability of c_j belonging to each of the L feature classes. This results in L class-probability images. The pixel locations are obtained by non-maximal suppression in each class-probability image.

[0057] The probabilities are used instead of the classification labels because a classification of label l arises when its confidence is greater than all other $h - 1$ classes in the classifier. However, a confidence of 95% for one pixel location means more than a confidence of 51% for that same labeling at a different location. In this case, the pixel with the higher probability would be chosen (even given they both have the same label), and for this reason detect is performed in probability space rather than in labeling space.

[0058] After candidate pixel locations are determined for each feature class, the feature detections are stereo matched in the corresponding stereo camera using normalized cross-correlation checks along the epipolar line, the features are triangulated to retrieve 3D locations. Using integral images of summations and squared-summations correlation windows along these epipoles are efficiently computed.

Patient-Side **Manipulator** (PSM) Association **Module**

[0059] Referring back to FIG, 1, after deriving the 3D point locations (in the camera's coordinate system) and associated feature labels, Patient-Side Manipulator (PSM) Association

Module 104, determines with which tool each feature is associated. With multiple tools in the scene, it is unclear after determination of class-labeled 3D feature locations which feature is associated with which tool. Typically, the da Vinci © has three Patient-Side Manipulators (PSMs), only two of which are visible in the camera frame at any time. The manipulators are referred to as PSMo, PSM_1, and PSM_2. For example and not limitation, a case in which two tools (PSMo and PSM_1) simultaneously appear is discussed below. In this case, Patient-Side Manipulator (PSM) Association Module 104 associates feature detections with PSMs.

[0060] Each PSM has a marker pattern, M_0 and M_1, respectively, each in their zero-coordinate frame *(e.g.,* the coordinate system before any kinematics are applied to the marker). Using the forward kinematics estimate from each PSM, the marker patterns are rotated to achieve the estimated orientations of each PSM. The full rigid-body transform from the **forward** kinematics is not applied because most of the **error** is in the position, and although the rotation isn't fully **correct,** it's typically close enough to provide the geometric constraints require. This leaves equations (9) and (10), where *Roto* and *Roti* are the 3x3 rotation matrices from the full rigid-body transformations representing the forward kinematics for PSMo and PSM_1, respectively. Given $\tilde{M}_0$ *and* $\tilde{M}_1$ 3D unit vectors are computed between each of the rotated point locations within each **marker.** This yields 7x7 3D **unit** vectors in **a** 7x7x3 matrix **for** each **rotated** marker **pattern.** Additionally, a 7x7 distance matrix D_m is calculated between each marker location in its zero-coordinate frame.

$$\tilde{M}_Q = R\,\mathrm{ot}_0(M_0) \qquad (9)$$

$$\tilde{M}_1 = R\,\mathrm{oti}(M_1) \qquad (10)$$

[0061] Next, given *N* detected feature observations using the classification method described above, both an *NxN* distance matrix between each 3D feature observation and an *NxNx3* matrix of unit vectors are computed, similar to those computed for the marker patterns using the kinematics estimates from the robot. Finally, any feature observations which do not adhere to one of the pre-processed marker distance and rotation configurations according to the PSMs are rejected. Using empirically determined distance *(e.g.,* ~3-5 mm) and orientation *(e.g.,* ~10°-20°) thresholds, the PSM associated with each feature is determined, allowing only one assignment per feature class to each PSM.

Shaft Extraction Module

[0062] Referring back to FIG. 1, Shaft Extraction Module 103 determines the location of the shaft in an input image. As noted above, it is not guaranteed that there are enough shaft pixels visible to compute valid cylinder estimates, and so in one embodiment of the present disclosure, stereo vision is used to estimate the distance of the tool tip to the camera. If the algorithm determines that the tools are situated far enough away from the camera so that the shaft is sufficiently visible, the shaft likelihood mask as provided by the Scene Labeling Module 101 is used to collect pixels in the image (potentially) belonging to one of the two tools' shafts. Assuming that each tool shaft is represented as a large, rectangular blob, using connected components and 2D statistical measures *(e.g.,* aspect ratios, total pixel areas) those areas of the image which are not likely to be one of the tool shafts are eliminated.

[0063] As shown in FIG. 6, 2D boundary lines 601, 602, 603, and 604 are fitted to each candidate shaft blob. By extracting the boundary lines of the shaft (outer pairs of lines 601-602 and 603-604), the mid-line axis (inner lines 605 and 606), and then the intersection location between the tool's shaft and the clevis (dots 607 and 608 on inner lines 605 and 606), shaft

observations are provided to the Fusion and Tracking Module 105 along with the feature observations. Using projective geometry a 3D cylinder is fit to each pair of 2D lines, representing a single tool's shaft. Then, the intersection point in the 2D image where the tool shaft meets the proximal clevis is located by moving along the cylinder axis mid-line from the edge of the image and locating the largest jump in gray-scale luminance values, representing where the black shaft meets the metal clevis (dots 607 and 608 on inner lines 605 and 606). A 3D ray is projected through this 2D shaft/clevis pixel to intersect with the 3D cylinder and localize on the surface of the tool's shaft. Finally, this 3D surface location is projected onto the axis mid-line of the shaft, representing a rotationally-mvariant 3D feature on the shaft. This shaft feature is associated with its known marker location and is added to the fusion stage 105 along with the feature classification detections.

Fusion and Tracking Module

[0064] Because features detected are not guaranteed to always be visible in any given frame, the robot kinematics are combined with the vision estimates in Fusion and Tracking Module 105 to provide the final articulated pose across time. The kinematics joint angles are typically available at a very high update rate, although they may not be very accurate due to the error accumulation at each joint.

[0065] For surgical robots like the da Vinci ®, it is important to keep the instrument insertion point (also termed remote center) stationary. This means that one part of the robotic arm holding the instrument does not move during the surgery (e.g., it is passive). The error of the end effector pose comes from both the error in zero calibration of the potentiometers at the joints and the error in the kinematics chain due to the link lengths. These are mostly static because the errors from the passive setup joints have more influence on the overall error as they are further

up in the kinematic chain and have longer link lengths than the active joints. Therefore, by solving for this constant error bias, it can be applied to the raw kinematics of the active joints to determine fairly accurate overall joint angle estimates. This bias essentially amounts to a rigid body pose adjustment at the stationary remote center. Although there is also error for the robotic arm holding the camera, when it does not move it is not necessary to include this in the error contributions.

[0066] To perform these adjustments on-line, an Extended Kalman Filter (EKF) is used. The state variables for the EKF contain entries for the offset of the remote center, which is assumed to be either fixed or slowly changing and so can be modeled as a constant process. The observation model comes from our 3D point locations of our feature classes. At least 3 non-colinear points are required for the system to be fully observable. The measurement vector is given in equation (11). The observation function which transforms state variables to observations is not linear, and so the Jacobians of equations (12) and (13) are required, where $\boldsymbol{p}^K$ is a 3D point location in the kinematics remote center coordinate frame KCS, $\boldsymbol{q}_i^K$ is a unit quaternion rotation between the true instrument joint coordinate system ICS and the KCS, and $\boldsymbol{cf}$ is the remote center location in the KCS.

$$>'_3 = [x\text{i}, y\text{i}, z_1, .., x_n, y_n, ¾]^T \quad (11)$$

$$J_1 = \frac{\partial p^K}{\partial q_i^K} \quad (12)$$

$$J_2 = \frac{\partial p^K}{\partial c_i^K} \quad (13)$$

[0067] It is unlikely that any realistic solution to a computer vision problem does not contain outliers. The image analysis is of principal concern as it is input to the fusion and tracking module 105. To address this issue, in some embodiments of the present disclosure, an initial

RANSAC phase is added to gather a sufficient number of observations and perform a parametric fitting of the rigid transformation for the pose offset of the remote center. This is used to initialize the EKF and updates online as more temporal information is accumulated. In some embodiments, a minimum of —30 total inliers are required for a sufficient solution to begin the filtering procedure. The rigid body transformation offset is computed using the 3D correspondences between the class-labeled feature observations, done separately for each PSM after the PSM association stage described above, and the corresponding marker patterns after applying the forward kinematics estimates to the zero-coordinate frame locations for each tool. Because the remote center should not change over time, this pose offset will remain constant across the frames, and so by accumulating these point correspondences temporally, a stable solution is achieved.

[0068] In some embodiments of the present disclosure, not all of the modules of Fig. 1 are present. For example, in one embodiment, Scene Labeling Module 101 and Shaft Extraction Module 103 are omitted and the input image is provided as input directly to the Feature Classification Module 102. In another embodiment, kinematics data is not used and so the Fusion and Tracking Module 105 is omitted and the pose of the Patient Side Manipulator is determined based on the output of the feature classification module. In another embodiment, there is only one Patient Side Manipulator, and the Patient Side Manipulator Association Module 104 is omitted. Other combinations of the modules of Figure 1 that do not depart from the spirit or scope of the disclosed subject matter will be apparent to those of skill in the art.

Experimental Results

[0069] The system of the present disclosure has been demonstrated to work on two types of datasets, both collected previously on a da Vinci ® surgical robot; (I) porcine data (in-vivo), and

(2) pork data (ex-vivo). The data which was used to test was specifically not included in the training collection procedure described above. After collecting and training the seven feature classes using —20,000 training samples with the Best Weighted Randomized Trees approach described above, the PSM association and geometric constraints method described above was applied, and finally the fusion and tracking stage was performed.

[0070] Overall, testing included 6 different video sequences, totaling 6876 frames (458 seconds worth of video). Each video frame had two tools visible at all times. Across these video sequences, three different types of da Vinci ® tools, were analysed, the Large Needle Driver (shown in FIG. 3A), Maryland Bipolar Firceps (shown in FIG. 3B) and Round Tip Scissors (shown in FIG. 3C). The system was trained only on the Large Needle Driver (LND) (shown in FIG. 3A), and tested on that same LND tool in addition to the Maryland Bipolar Forceps (MBF) (shown in FIG. 3B) and Round Tip Scissors (RTS) (shown in FIG. 3C). The method works on these other tools because there are many shared parts across the tools, including the pins used to hold the clevis together and the IS logo in the center of the clevis. Even though the overall appearance of each tool is quite different, results show that the method extends very well to different tools given that the lower-level features are consistent. However, if newer tools are introduced which don't share these parts in common, more training data and feature classes must be considered and included in training the classifier discussed above.

[0071] Ten sample results are shown in FIGS. 7A-J from various test sequences. FIGS. 7A-H show ex-vivo pork results with different combinations of the LND, MBF, and RTS tools. FIGS. 7I-J show a porcine in-vivo sequence with an MBF on the left and an LND on the right. In FIG. 7H, one tool is completely occluding the other tool's tip, however the EKF from the Fusion stage assists in predicting the correct configuration. For each, superposed lines 701-710 portray the

raw kinematics estimates as given by the robot, projected into the image frames. The lines 711-720 superposed on the tools show the fixed kinematics after running application of the detection and tracking system of the present disclosure. FIGS. 7A-B show the MBF (left) and LND (right). FIGS. **7C-D** show the RTS (left) and MBF (right). FIGS. **7E-F** show the LND (left) and RTS (right). FIGS. **7G-H** show the MBF (left) and MBF (right). FIGS. **7I-J** show the MBF (left) and LND (right). The significant errors are apparent, where in some images the estimates are not visible at all, motivating the need for the system and methods of the present disclosure. A visual inspection yields a fairly accurate correction of the kinematics overlaid on the tools.

[0072] Because joint-level ground truth for articulated tools is very difficult to collect accurately and on a large dataset, the accuracy of the tracking system of the present disclosure is evaluated in the 2D image space. **FIG. 8A** depicts the evaluation scheme for the kinematics estimates. The dotted lines 801, 802 define an acceptable boundary for the camera-projection of the kinematics, where the solid line 803 is a perfect result. FIG. 8B shows an example of an incorrect track 804 on the right-most tool. Using this scheme, each frame of the test sequences was manually inspected, and resulted in a 97.81% accuracy rate over the entire dataset.

[0073] TABLE **1** shows a more detailed breakdown of the evaluation. Overall, the system of the present disclosure was tested against 6 sequences, including both ex-vivo and in-vivo environments, all with two tools in the scene. TABLE 1 shows the test sequence name in the first (leftmost) column, the number of tracks labeled as correct in the second column, the total possible number of detections in that sequence in the third column, and the final percent correct in the last (rightmost) column. Note that in any given frame, there may be 1 or 2 tools visible, and this is how the numbers in the third column for the total potential number of tracks in that

sequence are computed. Finally, the last row shows the total number of correct tracks detected as 13315 out of a total possible of 13613, yielding the final accuracy of 97.81% correct. Also note that the accuracy was very similar across the sequences, showing the consistency of the system and methods of the present disclosure. Although the accuracy was evaluated in the 2D image space, this may not completely represent the overall 3D accuracy as errors in depth may not be reflected in the perspective image projections.

Sequence	# Correct	Potential	% Correct
Seq. 1	1890	1946	97.12%
Seq. 2	2114	2182	96.88%
Seq. 3	1447	1476	98.04%
Seq. 4	1611	1648	97.75%
Seq. 5	4376	4431	98.76%
Seq. 6	1877	1930	97.25%
TOTAL	**13315**	**13613**	**97.81%**

TABLE 1

[0074] The full tracking system of the present disclosure runs *at* approximately 1.0-1.5 sees/frame using full-sized stereo images (960x540 pixels). The stereo matching, PSM association, and fusion/EKF updates are negligible compared to the feature classification and detection, which takes up most of the processing time. This is dependent on the following factors: number of trees in A, depth of each tree γ_i, number of features used in the Region Covariance descriptor CR (in one embodiment of the present disclosure, 11 are used, but less could be used), and the quality of the initial segmentation providing the mask prior. However, by half-sizing the images a faster frame-rate can be achieved (0.6-0.8 sees/frame, an example of which is shown in Seq. 5) while achieving similar accuracy. Also, because a solution is found for a remote center bias offset which remains constant over time, the frames can be processed at a slower-rate without affectmg the overall accuracy of the tracking system. Finally, many stages

of the classification are parallelizable, and both the Covariance Descriptor and Randomized Trees can be implemented on a GPU processor. Test results on the covariance processing show a reduction in the processing time of the feature tensors (equation (1)) from ~70()ms to —100ms, and this can be reduced further.

[0075] There many variations are possible in the implementation of a tracking system in accordance with the present disclosure. One such variation is found in the size of the window to use when extracting covariance descriptors for classification throughout the image. The reason is that, during training, the best encompassing bounding box around each feature is used, and the descriptors are well tuned to representing the entire feature. When applying the classifier, if the window is too small or too large, the descriptors won't capture the features well. To alleviate this, prior knowledge of the 3D sizes of the features is used to guide computation of the optimal window size. Using the stereo vision approach which determines if the shaft is visible enough to extract (as discussed above) and estimating that the features are $\sim 2x3mm$ in size, the optimal window size in the image can be automatically determined dynamically on each frame. To further reduce errors, at every pixel location that is evaluated, a bounding box is extracted that is both full and half-sized according to this automatically determined window size to account for the smaller features *(e.g.,* the pins). This improves the overall feature detection system.

[0076] Upon further inspection of the errors encountered during evaluation on the test sequences, it is found that most of the incorrect fixed/tracked kinematic configurations are due to a latency in the raw kinematics which causes the video and raw kinematics to be out-of-sync from time-to-time. This situation is shown more precisely in FIGS. 9A-D, which shows an example of kinematic latency in the right tool. Often the kinematics and video get out-of-sync with each other. Most of our errors are due to this fact, manifesting in the situation shown in

FIGS. 9A-P. The four frames of FIGS. **9A-D** are consecutive to each other in order. In FIG. 9A (representing time *t*), both tools are tracked well (as shown by lines 901). Then, in FIG. **9B** (representing time *t+1)* and FIG. 9C (representing time *t+2),* the kinematics and video become out-of-sync and the right tool becomes inaccurately tracked (as shown by lines 902 and 903). However, in FIG. 9D (representing time *t+3*), the tools are tracked successfully again (as shown by lines 904). Looking at the overlay configuration 903 in FIG. 9C, which is essentially the same as the correct one immediately following in FIG. 9D (904), suggests this latency is the source of our errors. For the individual frames which had incorrect projections (according the scheme described above), the result would jump immediately to a correct configuration instead of getting lost completely, and the previous incorrect projection was in the location and configuration that the upcoming projection would eventually reach. Therefore, by logging the test data more precisely so that the video and kinematics are more in sync with each other, the accuracy would be expected to increase even further. However, in practice on a live system this kinematic latency does not exist.

[0077] The majority of tool tracking approaches in the literature work by estimating the cylinder of the shaft which is visible in the scene). However, as previously discussed, surgeons tend to work quite zoomed in, making this cylinder-fitting procedure very difficult, if not impossible, due to the limited number of visible shaft pixels. The remaining minority approaches work by analyzing the tip of the tool using features, however these will fail when the tool tip is too far away to be seen well by the camera. The approach described above is advantageous in that it dynamically decides which of these two approaches is optimal at any given time, and often uses both simultaneously to best track the tool over longer periods of time. Also, by using the pixel labeling method described above, the system of the present disclosure is able to tell more

accurately when parts of the tool are occluded. For example, if the metal tool tip is occluded then the pixel labeling won't label the incorrect pixels from the occluder as metal, and false positives will be avoided. Occlusion errors will similar!}' be avoided for the shaft.

[0078] The present disclosure provides a tool detection and tracking framework which is capable of tracking multiple types of tools and multiple tools simultaneously. The algorithm has been demonstrated on the da Vinci ® surgical robot, and can be used with other types of surgical robots. High accuracy and long tracking times across different kinds of environments (ex-vivo and in-vivo) are shown. By learning low-level features using a multi-class classifier, the system of the present disclosure overcomes different degrees of visibility for each feature. The hybrid approach of the present disclosure, using both the shaft and features on the tool tip, is advantageous over either of these methods alone. Using knowledge of the distance of the tool the system of the present disclosure can dynamically adapt to different levels of information into a common fusion framework. Finally, by fusing vision and kinematics, the system of the present disclosure can account for missed observations over time.

[0079] Example applications of tool tracking in accordance with the present disclosure are shown in FIGS. **10A-B.** In FIG, **1ΘA,** a picture of a measurement tool measuring the circumference 1001 and area 1002 of a mitral valve is shown. In FIGS, **10B,** an example scenario of a lost tool *(e.g.,* outside the camera's field-of-view) is shown, whereby the endoscopic image (top) shows only two tools, and with fixed kinematics and a graphical display (bottom), the surgeon can accurately be shown where the third tool 1003 (out of the left-bottom comer) is located and posed so they can safely manipulate the tool back into the field-of-view.

[0080] FIG. 11 depicts example appearance changes typically encountered of the IS Logo feature through different lighting and perspective effects, to motivate the need for a robust descriptor.

[0081] Although in one embodiment of the present disclosure, the Covariance Descriptor is paired with Best Weighted Randomized Trees to achieve a sufficient level of accuracy and speed, alternative combinations of descriptors and classifiers can be used. One method of evaluating available parings using the likelihood-space works as follows: given a test image, the multi-class classifier is run through the entire image, resulting in *h* probabilities at each pixel for each feature class. This yields *b* different likelihood images. In each likelihood, non-maximal suppression is performed to obtain the 3 best peaks in the likelihood. Then, a feature classification is marked correct if any of the 3 peaks in the likelihood is within a distance threshold (for example, 1% of the image size) of the ground truth for that feature type. This method is suitable because it is often the ease that there is a local peak at the correct location for a feature, but it is not always the global peak. Therefore, in a full tracking system a temporal coherence filter can eliminate these outliers. FIGS. 12A-H show sample likelihoods on the tip of the LND tool overlain with extrema locations. FIG. 12A depicts the individual features with circles (from top to bottom, iDot 1201, IS Logo 1202, Pin3 1203, Pinl 1204, Wheel 1205, Wheel Pin 1206, Pin 4 1207). Six of the seven features are correctly detected as peaks in the class-conditional likelihoods (FIG. *Mil* - iDot, **FIG.** 12C - IS Logo, FIG. 12D - Pin1, FIG. 12F - Pin4, FIG. 12G - Wheel, FIG. 12H - Wheel Pin), where the Pin3 (FIG. 12E) feature is incorrectly detected. This was produced using the Covar/RT approach.

[0082] To evaluate accuracy, testing was performed on video which was specifically not used in the training stage. Testing was performed on 1500 frames from in-vivo sequences, which

resulted in -4500 possible feature defections which were ground-truthed. The accuracy against the ground truth is shown in **FIG.** 13 for each individual feature type, separately. It is clear that different features are more reliably detected than others, which may be attributed to differences in size, texture, and uniqueness. However, it is obvious from this graph that the Region Covariance out-performs both the SIFT and HoG descriptors, regardless of the learning algorithm.

[0083] A more detailed analysis reveals that the SVM evaluates best overall, although both RT and BWRT are certainly comparable as different features perform differently. For example, Covar/SVM classifies the Wheel feature with 81% accuracy, whereas Covar/RT classifies that same feature at 84% and Covar/BWRT at 86%. Contrastly, Covar/SVM classifies the IS Logo feature at 80% against a classification rate of 59% for Covar/RT and 63% for Covar/BWRT.

[0084] The maximum achieved accuracy using the SIFT descriptor was 44% using SIFT/SVM on the Pinl feature. Using the HoG descriptor, the best achieved accuracy was 37% using HoG/SVM on the IS Logo feature.

[0085] In addition to accuracy, the per-frame processing time of each algorithm is considered. As mentioned previously, SVMs tend to become more complex and time consuming as more support vectors are added, which arises due to more training data. Conversely, the tree approaches are designed to be efficient as the node tests are low-cost and only *m* tests-per-tree across all *L* trees are needed to classify a feature (in this example, **w=10** and 1=90). In the case of the BWRTs, an initial tree bag of 1000 is used and the best 90 are selected fxom this bag.

[0086] During testing, for a given 640x480 image every other pixel is classified using the descriptor/classifier combinations. This amounts to 76,800 descriptor extractions and

classifications per-frame for each algorithm. A constant-size window is used for each descriptor (21 pixel diameter, empirically determined) for all descriptor types. The average run-time per-frame is analyzed and the results are shown in the third column of TABLE 2 in msecs/frame. The higher-dimensional feature vectors required more time, especial!}' in the case of the SVMs. Therefore, SIFT (d=128) had the largest run-time and HoG (d=45) had the smallest. The run-time for the RT and BWRT (d=66) cases should be very close as they are equivalent in terms of behavior, only differing in the values for the weights.

Descriptor	Classifier	Unmasked	Masked
Covar	SVM	60185.4	4431.18
	RT	8672.4	1171.01
	BWRT	8685.57	1086.8
SIFT	SVM	204696	13704.8
	RT	11914.3	915.163
	BWRT	12325.9	990.732
HoG	SVM	55634.6	4216.58
	RT	5231.53	551.321
	BWRT	5388.07	557.324

TABLE 2

[0087] The fastest algorithm was HoG/RT and HoG/BWRT, with the smallest complexity. An increase in speed can be applied to all cases if an initial mask prior were present, which would limit which pixels to analyze in the image (as applied above). The classifications can be confined to pixels only on the metal tip of the tool (as discussed above). The runtime results (including the time to compute the masks) are shown in the fourth column of TABLE 2, which shows a significant reduction in processing. This gets closer to a real-time solution, where, for example, the Covar/BWRT approach is reduced to a little over 1 sec/frame. Finally, the percent decrease in run-time from the SVM case to the RT/BWRT cases is analyzed for each descriptor. With a slight reduction in accuracy performance, this showed a reduction of up to 80% using

Covar, and 90% and 94% using the HoG and SIFT descriptors, respectively. These are not trivial speed-ups, and should be considered in the choice of the feature detection algorithm.

[0088] Although some feature types may not always be detected, a minimum of 3 are required on a given frame to recover the articulated pose (because the algorithm described above fuses kinematics with vision), and so across the 7 chosen landmarks, the percent correct achieved is sufficient for longterm tracking. Features with low probabilities are rejected based on the confidence. When considering tracking 2 tools simultaneously, kinematics can be used as a prior on geometric constraints to assign features to the most likely tool pairing.

[0089] It is understood that the subject matter described herein is not limited to particular embodiments described, as such ma}', of course, vary. For example, various image segmentation methods can be used in accordance with the present subject matter including thresholding, clustering, graph cut algorithms, edge detection, Gaussian mixture models, and other suitable image segmentation methods known in the art. Various descriptors can also be used in accordance with the present subject matter including covariance descriptors, Scale Invariant Feature Transform (SIFT) descriptors, Flistogram-of-Grientation Gradients (HoG) descriptors, Binary Robust Independent Elementary Features (BRIEF) descriptors, and other suitable descriptors known in the art. Various classifiers can also be used in accordance with the present subject matter including randomized tree classifiers, Support Vector Machines (SVM), AdaBoost, and other suitable classifiers known in the art. Accordingly, nothing contained in the Abstract or the Summary should be understood as limiting the scope of the disclosure. It is also understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to be limiting. Where a range of values is provided, it is understood that each intervening value between the upper and lower limit of that range and any

other stated or intervening value in that stated range, is encompassed within the disclosed subject matter.

[0090] Unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this disclosed subject matter belongs. Although any methods and materials similar or equivalent to those described herein can also be used in the practice or testing of the present disclosed subject matter, this disclosure may specifically mention certain exemplary methods and materials.

[0091] As used herein and in the appended claims, the singular forms "a," "an," and "the" include plural referents unless the context clearly dictates otherwise.

[0092] As will be apparent to those of skill in the art upon reading this disclosure, each of the individual embodiments described and illustrated herein has discrete components and features which may be readily separated from or combined with the features of any of the other several embodiments without departing from the scope or spirit of the present disclosed subject matter. Various modifications can be made in the method and system of the disclosed subject matter without departing from the spirit or scope of the disclosed subject matter. Thus, it is intended that the disclosed subject matter include modifications and variations that are within the scope of the appended claims and their equivalents.

WHAT IS CLAIMED IS:

1. A robotic surgical tool tracking method, comprising:

generating a descriptor of a region of an input image;

applying a trained classifier to the descriptor to generate an output indicative of whether a feature of a surgical tool is present in the region;

determining the location of the feature of the surgical tool based on the output of the trained classifier.

2. The method of claim 1, wherein the descriptor is selected from the group consisting of a covariaiice descriptor, a scale invariant feature transform descriptor, a histogram-of-orientation gradients descriptor, and a binary robust independent elementary features descriptor.

3. The method of claim 1, wherein the trained classifier is selected from the group consisting of a randomized tree classifier, a support vector machine classifier, and an AdaBoost classifier.

4. The method of claim 1, wherein the region is selected from within a predetermined area of the input image.

5. The method of claim 1, wherein the region is selected from within a mask area indicative of the portion of the input image that corresponds to a tip portion of the surgical tool.

6. The method of claim 1, wherein the input image contains a plurality of surgical tools, the method further comprising:

determining to which of the plurality of surgical tools the feature corresponds.

7. The method of claim 5, further comprising:

generating the mask area by applying a Gaussian mixture model

8. The method of claim 5, further comprising:

generating the mask area by image segmentation by color clustering.

9. The method of claim 5, further comprising:

generating the mask area by image segmentation by thresholding.

10. The method of claim 5, further comprising:

generating the mask area by image segmentation by application of a graph cut algorithm.

11. The method of claim 2, wherem the descriptor is a covariance descriptor.

12. The method of claim 11, wherein the covariance descriptor comprises an x coordinate, a y coordinate, a hue, a saturation, a color value, a first order image gradient, a second order image gradient, a gradient magnitude, and a gradient orientation.

13. The method of claim 1, wherem the classifier is a randomized tree classifier.

14. The method of claim 13, wherein the randomized tree classifier additionally comprises weights associated with each tree and applying the classifier comprises applying the weights associated with each tree to the outputs of each tree.

15. A non-transient computer readable medium for use with a robotic surgical tool tracking system, comprising:

instructions for generating a descriptor of a region of an input image;

instructions for applying a trained classifier to the descriptor to generate an output indicative of whether a feature of a surgical tool is present in the region;

instructions for determining the location of the feature of the surgical tool based on the output of the trained classifier.

16. The non-transient computer readable medium of claim 15, wherein the descriptor is selected from the group consisting of a covariance descriptor, a scale invariant feature transform descriptor, a histogram-of-orientation gradients descriptor, and a binary robust independent elementary features descriptor.

17. The non-transient computer readable medium of claim 15, wherein the trained classifier is selected from the group consisting of a randomized tree classifier, a support vector machine classifier, and an AdaBoost classifier.

FIG. 1

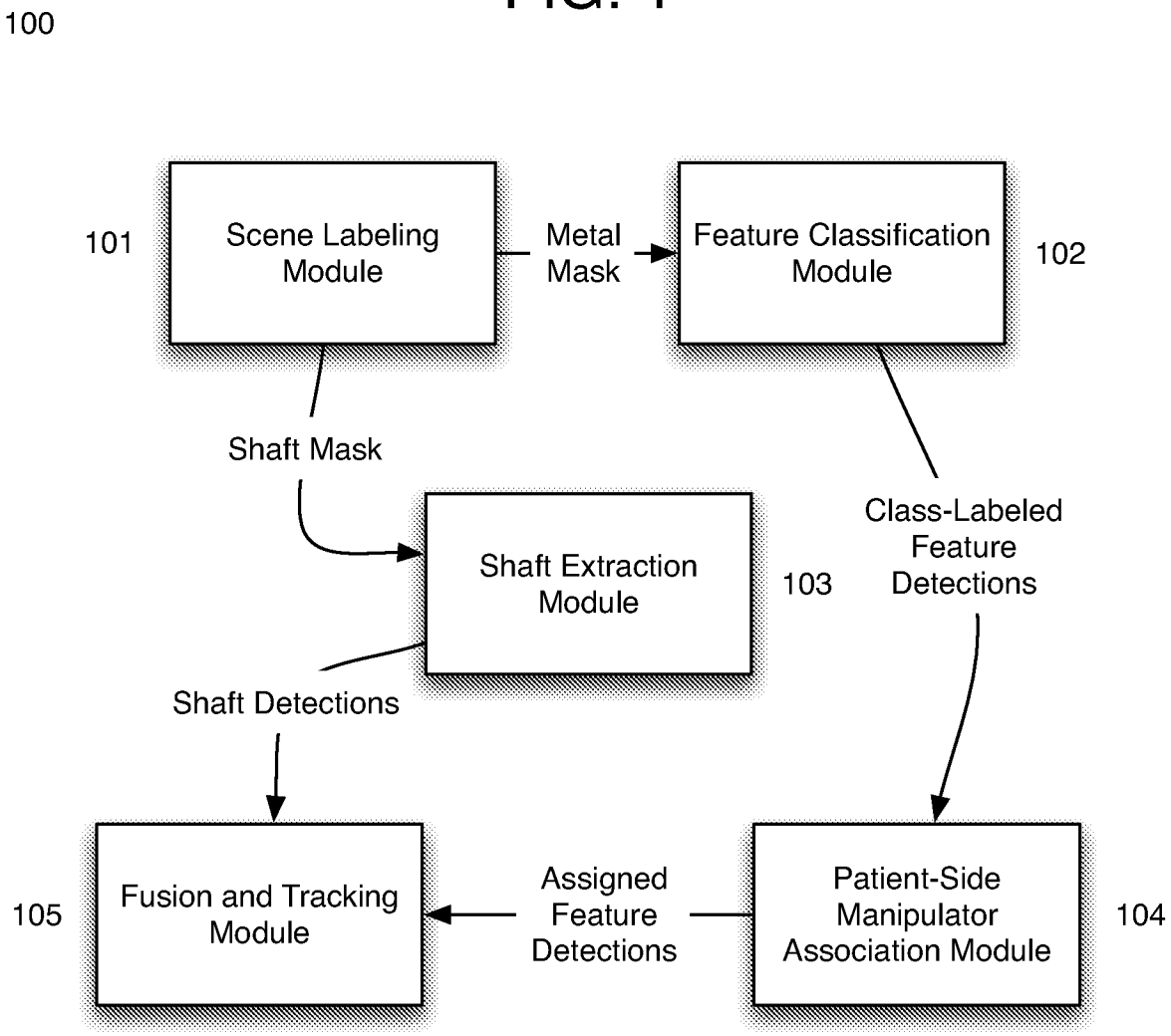
100
101
Scene Labeling Module
Metal Mask
Feature Classification Module
102
Shaft Mask
Class-Labeled Feature Detections
Shaft Extraction Module
103
Shaft Detections
105
Fusion and Tracking Module
Assigned Feature Detections
Patient-Side Manipulator Association Module
104

FIG. 2A FIG. 2B

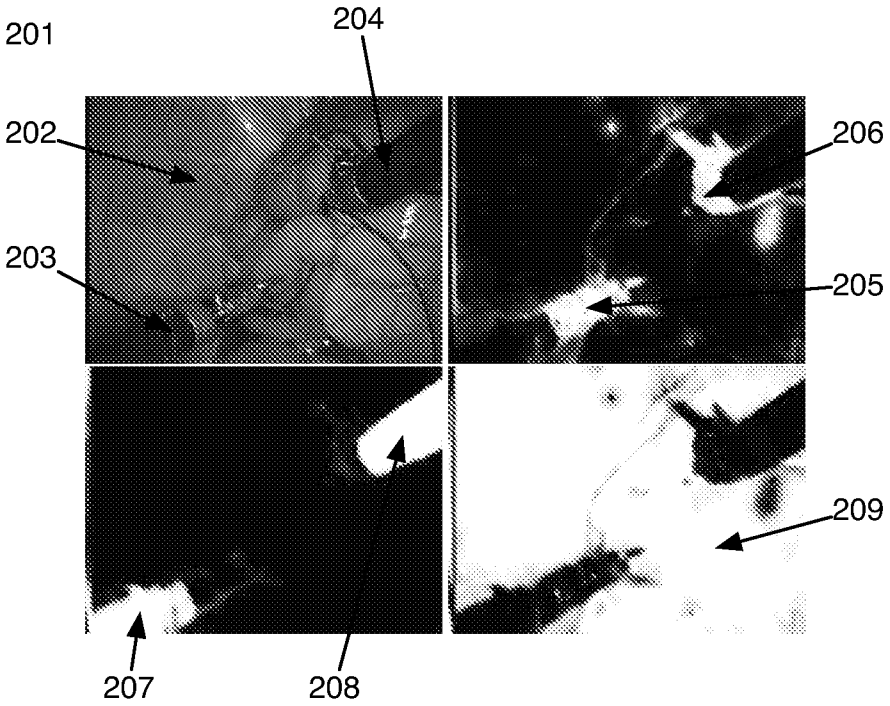

FIG. 2C FIG. 2D

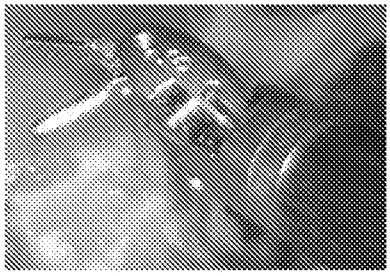

FIG. 3A

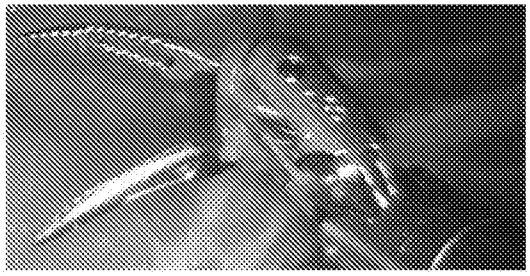

FIG. 3B

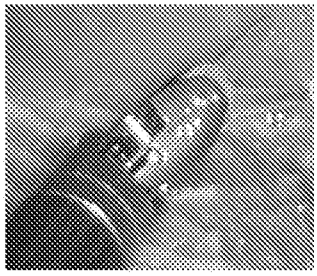

FIG. 3C

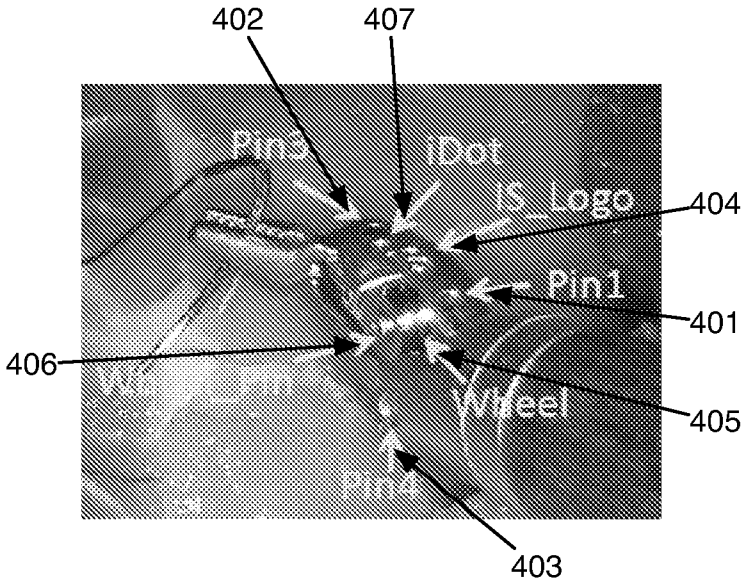

FIG. 4

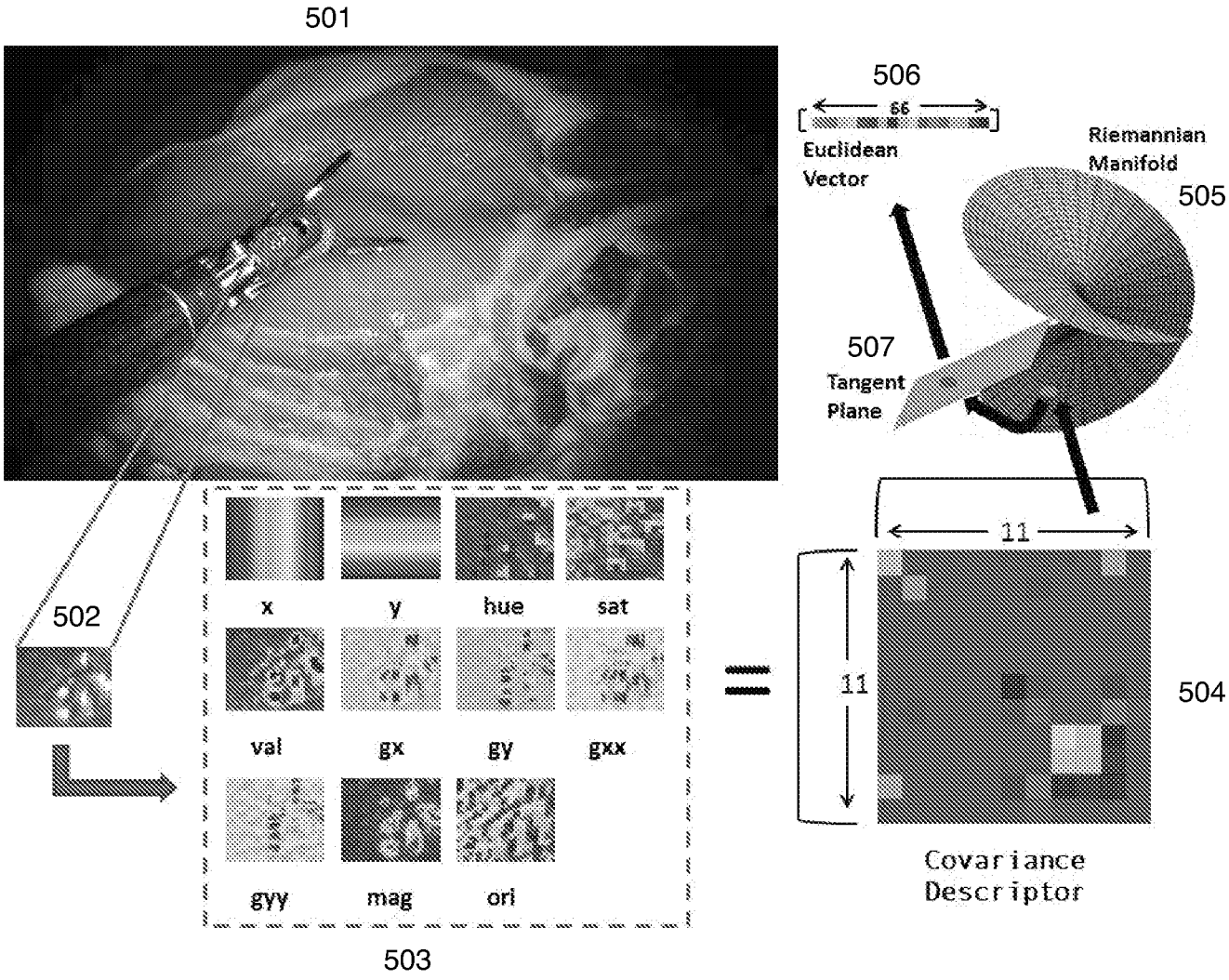
501
506
66
Euclidean
Vector
Riemannian
Manifold
505
507
Tangent
Plane
502
x
y
hue
sat
val
gx
gy
gxx
gyy
mag
ori
503
11
11
504
Covariance
Descriptor

FIG. 5

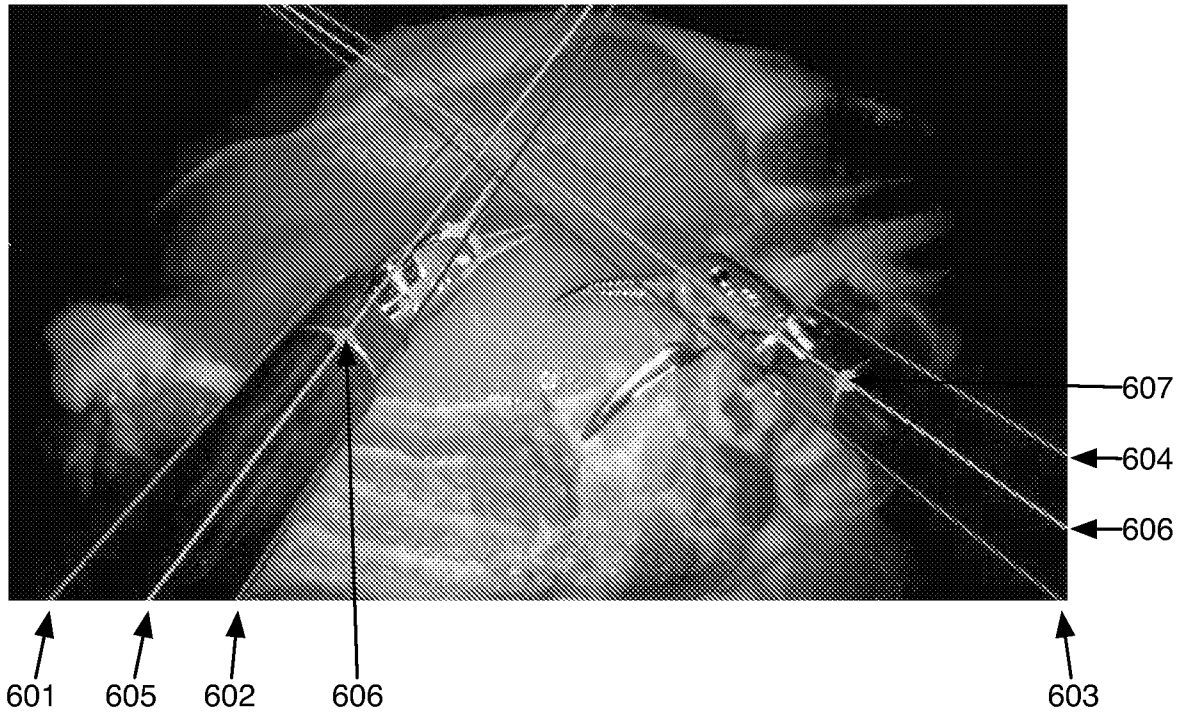

FIG. 6

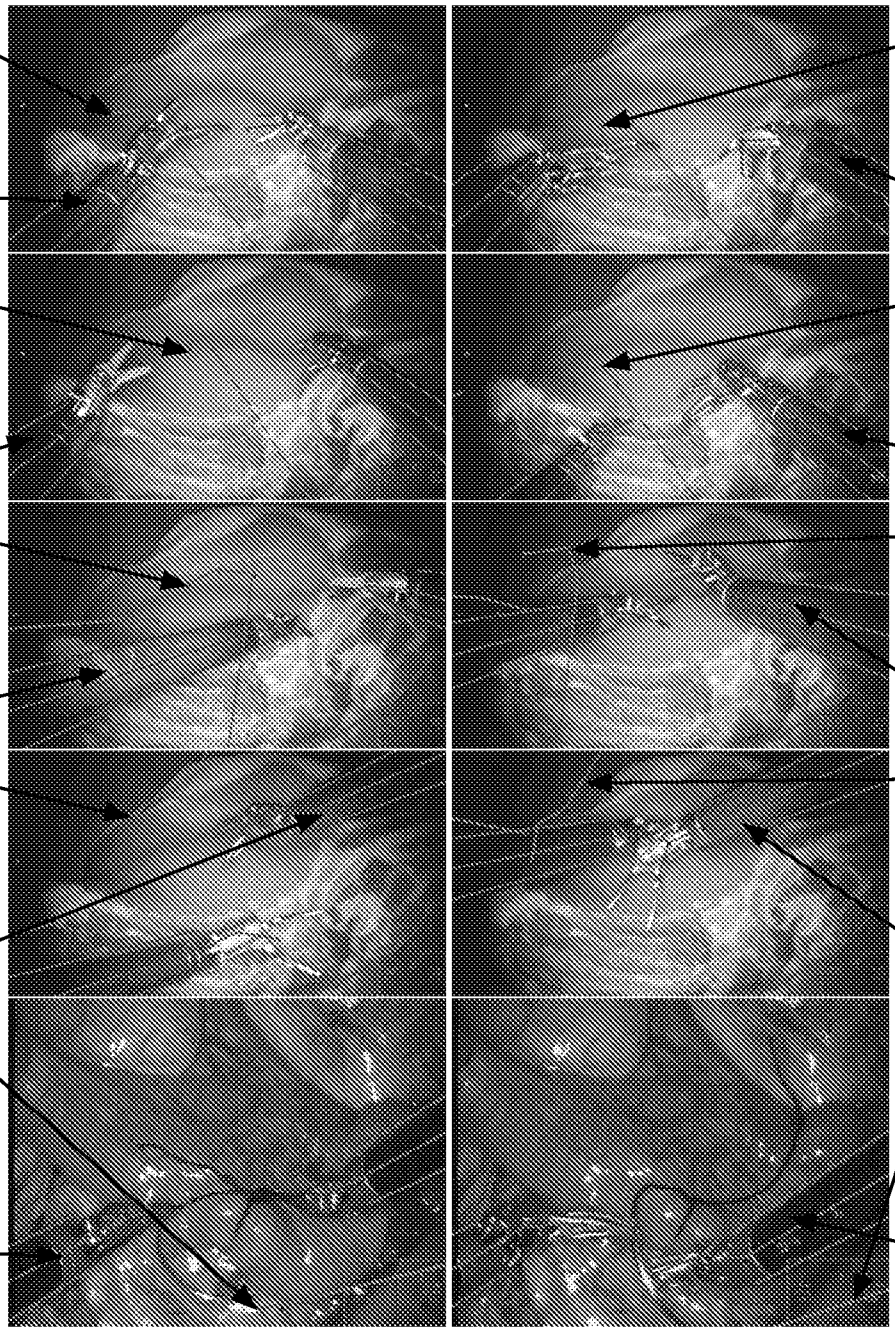

FIG. 7A FIG. 7B

FIG. 7C FIG. 7D

FIG. 7E FIG. 7F

FIG. 7G FIG. 7H

FIG. 7I FIG. 7J

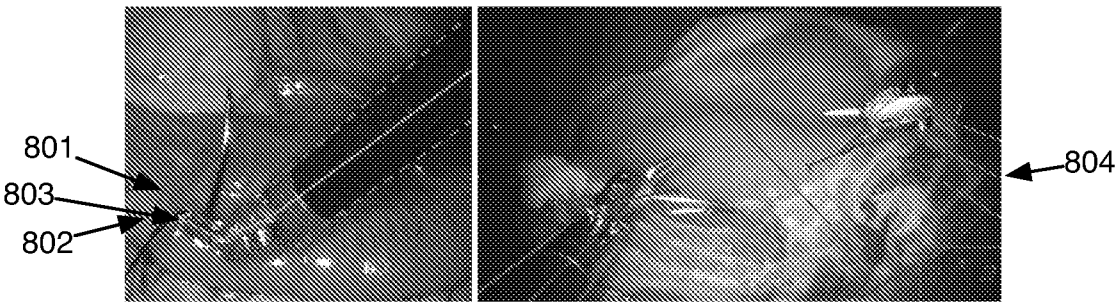

FIG. 8A FIG. 8B

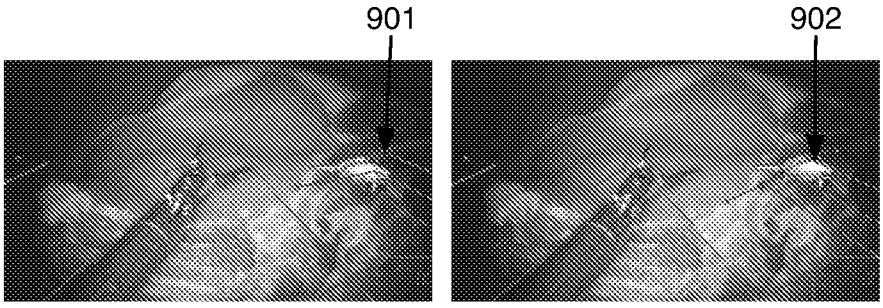

FIG. 9A FIG. 9B

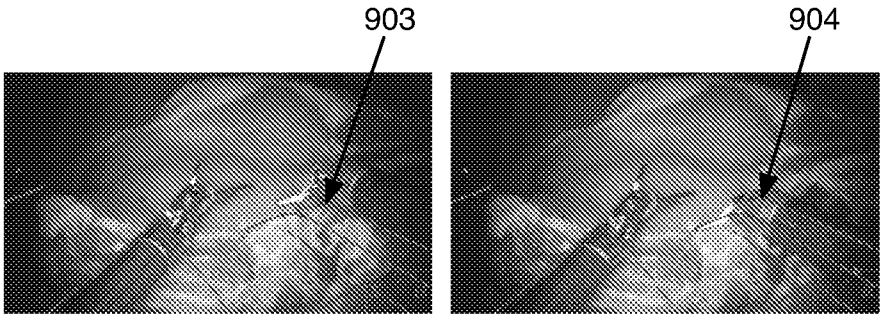

FIG. 9C FIG. 9D

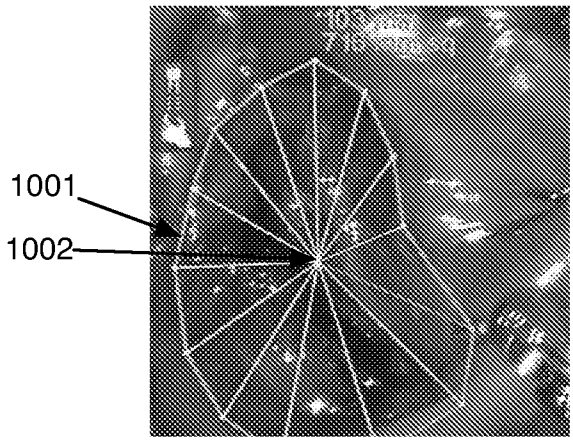

FIG. 10A

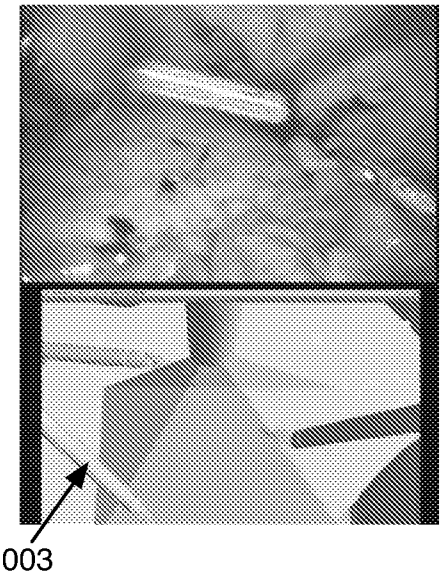

FIG. 10B

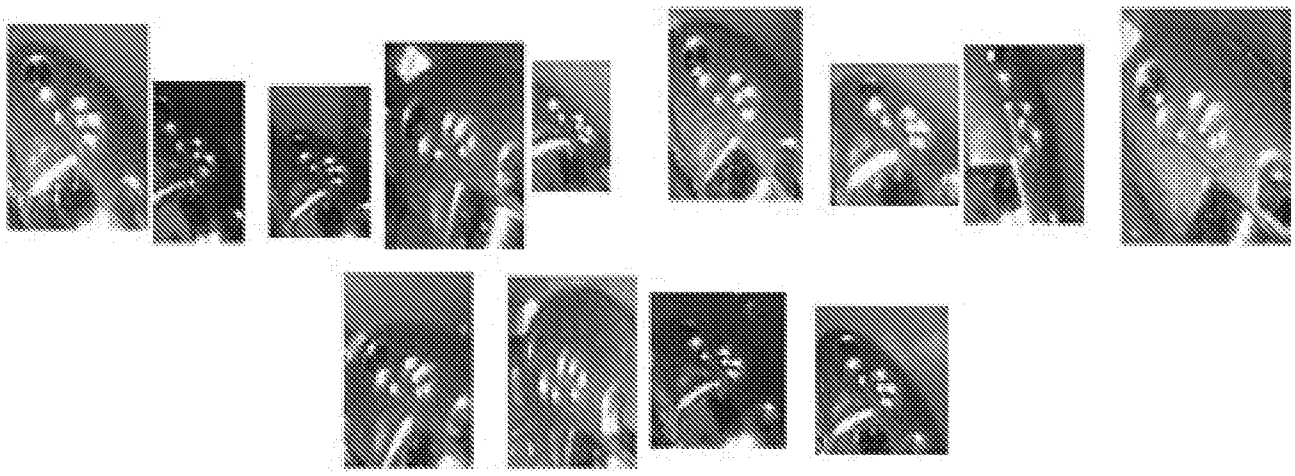

FIG. 11

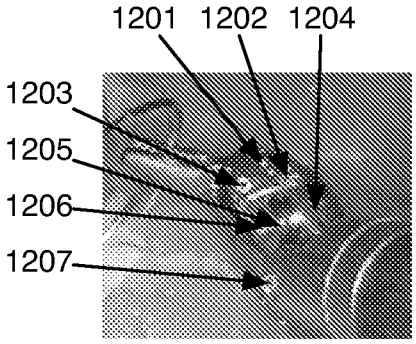

FIG. 12A

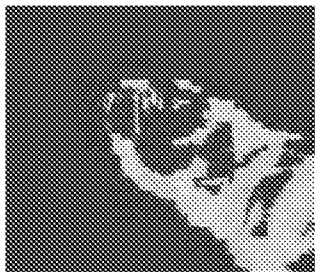

FIG. 12B

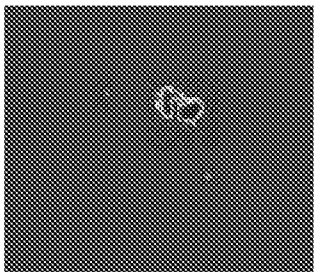

FIG. 12C

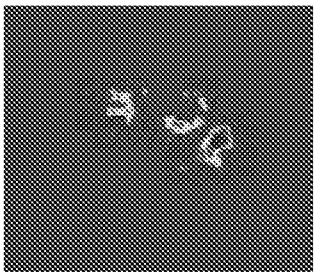

FIG. 12D

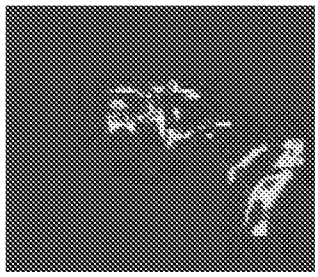

FIG. 12E

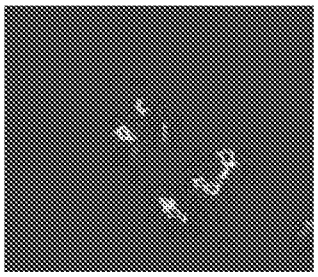

FIG. 12F

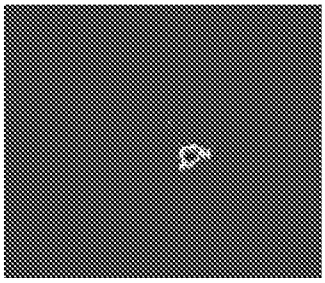

FIG. 12G

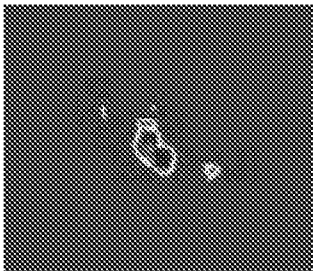

FIG. 12H

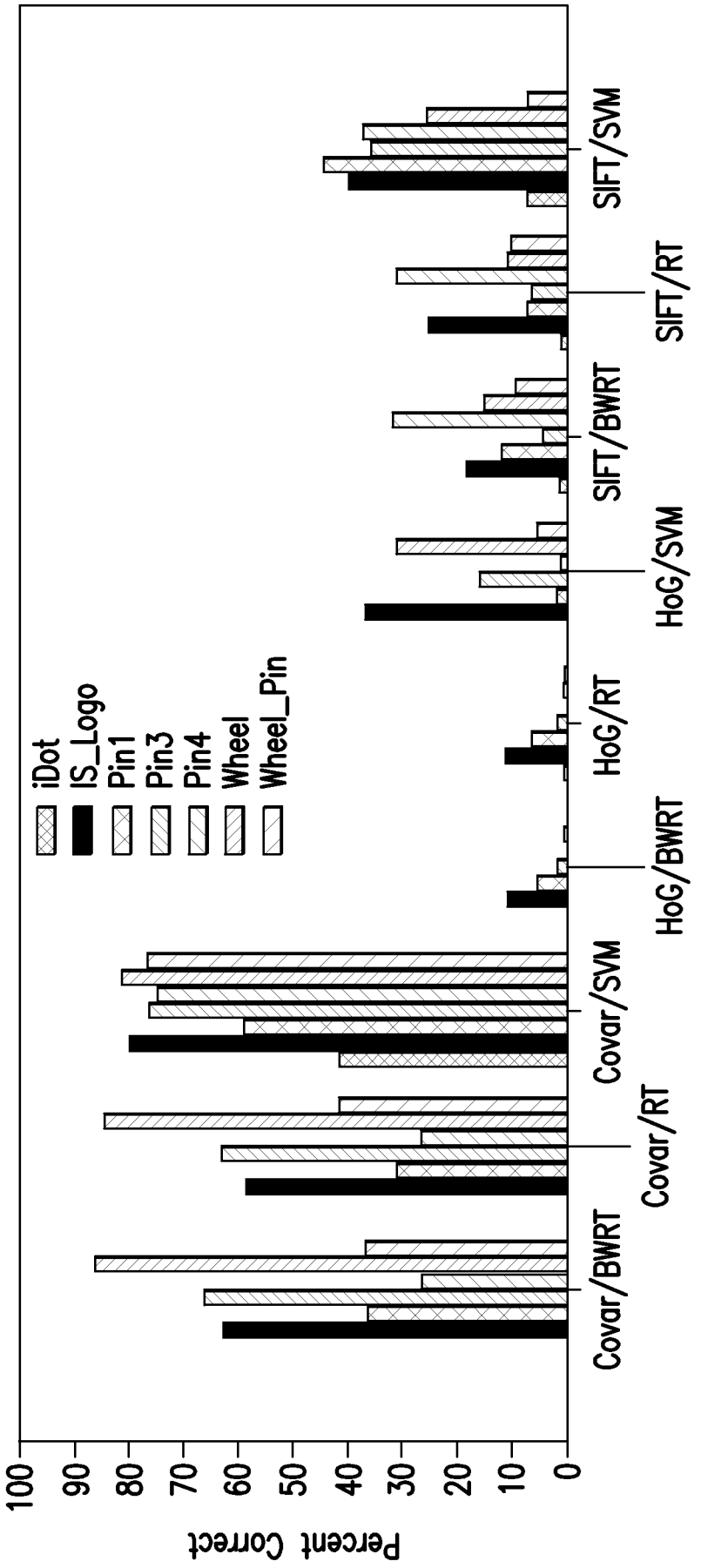
Percent Correct
100
90
80
70
60
50
40
30
20
10
0
iDot
IS_Logo
Pin1
Pin3
Pin4
Wheel
Wheel_Pin
Covar/BWRT
Covar/RT
Covar/SVM
HoG/BWRT
HoG/RT
HoG/SVM
SIFT/BWRT
SIFT/RT
SIFT/SVM

FIG. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US13/75014

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - A61B 19/00, 1/045 (2014.01)
USPC - 600/104, 111, 117
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8): A61B 19/00, 1/045 (2014.01)
USPC: 600/104, 111, 117

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
MicroPatent (US-G, US-A, EP-A, EP-B, WO, JP-bib, DE-C.B, DE-A, DE-T, DE-U, GB-A, FR-A); ProQuest; Google/Google Scholar; IP.com; KEYWORDS: Robot*, AutoMmat*, Machine*, Computer*, Mechan*, Electronic*, Surgical*, surger*, surgeon*, tool*, device*, implement*, equip*, uten*, Scalpel*, LaproMscop*, Forcep*, Glamp*, Track*, Local*, Identi*, Follow*, Find*, Descript*, Bound*, Limit*

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X -- Y	REITER, A, et al. Feature Classification For Tracking Articulated Surgical Tools. Medical Image Computing and Computer-Assisted Intervention-MICCAI. October 1-5, 2012; pages 592-600; abstract; section 2.1, paragraphs 1-2; section 2.2, paragraphs 1-2; section 2.3, paragraphs 1, 5; section 2.4, paragraphs 1-3; section 3, paragraphs 1-3, 6; figures 2(a)-(h), 3	1-7, 11-17 ---------- 8-10
Y	US 5820545 A (ARBTER, K et al.) October 13, 1998; column5, lines 1-4; column 6, lines 49-61	8
Y	US 8090177 B2 (VENKATARAMAN, S et al.) January 3, 2012; column 10, lines 10-22	9
Y	US 8073217 B2 (SUN, H et al.) December 6, 2011; column 4, lines 32-56	10
A	US 2012/0190981 A1 (HARRIS, RJ et al.) July 26, 2012; entire document	1-17
A	WO 2009/045827 A2 (ZHAO, W et al.) April 9, 2009; entire document	1-17
A	REITER, A et al. Marker-Less Articulated Surgical Tool Detection. Computer Assisted Radiology and Surgery (CARS). June 27-30, 2012; entire document	1-17
A	REITER, A et al. Learning Features On Robotic Surgical Tools. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). June 16-21, 2012; pages 38-43; entire document	1-17
A	REITER, A et al. A Learning Algorithm For Visual Pose Estimation Of Continuum Robots. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). September 25-30, 2011; pages 2390-2396; entire document	1-17

☒ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
05 FebruaTy 2014-(05:02:201 4)	24 FEB 2014

Name and mailing address of the ISA/US	Authorized officer:
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Shane Thomas PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	PEZZEMENTI, Z et al. Articulated Object Tracking By Rendering Consistent Appearance Parts. IEEE International Conference on Robotics and Automation. 2009; pages 3940-3947; entire document	1-17

专利名称(译)	无标记跟踪机器人手术工具		
公开(公告)号	EP2931161A4	公开(公告)日	2016-11-30
申请号	EP2013862359	申请日	2013-12-13
[标]申请(专利权)人(译)	纽约市哥伦比亚大学理事会		
申请(专利权)人(译)	哥伦比亚大学纽约市受托人		
当前申请(专利权)人(译)	哥伦比亚大学纽约市受托人		
[标]发明人	REITER AUSTIN ALLEN PETER K		
发明人	REITER, AUSTIN ALLEN, PETER, K.		
IPC分类号	A61B19/00 A61B1/045 A61B1/00 A61B5/00 A61B34/20 A61B34/30		
CPC分类号	A61B5/7267 A61B1/00149 A61B34/20 A61B34/30 A61B2034/2059 A61B2034/2065		
优先权	61/737172 2012-12-14 US		
其他公开文献	EP2931161A1		
外部链接	Espacenet		

摘要(译)

用于机器人手术工具的三维无标记跟踪的外观学习系统，方法和计算机产品。提供了一种外观学习方法，其用于在腹腔镜序列中检测和跟踪手术机器人工具。通过训练低级地标特征上的鲁棒视觉特征描述符，构建用于融合机器人运动学和3D视觉观察以在各种类型的环境中长时间跟踪外科手术工具的框架。在具有不同总体外观的多种类型的多个工具上启用三维跟踪。本公开的主题可应用于在离体和体内环境中的手术机器人系统，例如手术机器人。