3-D Pose Estimation of Articulated Instruments in Robotic Minimally Invasive Surgery

M. Allan, S. Ourselin, D. J. Hawkes, J. D. Kelly, and D. Stoyanov

***Abstract*—Estimating the 3-D pose of instruments is an important part of robotic minimally invasive surgery for automation of basic procedures as well as providing safety features, such as virtual fixtures. Image-based methods of 3-D pose estimation provide a non-invasive low cost solution compared with methods that incorporate external tracking systems. In this paper, we extend our recent work in estimating rigid 3-D pose with silhouette and optical flow-based features to incorporate the articulated degrees-of-freedom (DOFs) of robotic instruments within a gradient-based optimization framework. Validation of the technique is provided with a calibrated *ex-vivo* study from the da Vinci Research Kit (DVRK) robotic system, where we perform quantitative analysis on the errors each DOF of our tracker. Additionally, we perform several detailed comparisons with recently published techniques that combine visual methods with kinematic data acquired from the joint encoders. Our experiments demonstrate that our method is competitively accurate while relying solely on image data.**

***Index Terms*—Surgical instrument detection, articulated pose estimation, robotic surgery.**

I. Introduction

MINIMALLY invasive surgery (MIS) has provided surgeons with a less invasive method of accessing the surgical site with a cost of having less control and information about the operation compared with open surgery. Laparoscopic instruments reduce the surgeon's dexterity and ability to sense force feedback from applied tissue pressure and the limited field of view of the surgical camera makes self-localization challenging and increases the cognitive workload on the surgeon. In addition to this, the learning curve for MIS is steep with surgeons taking significant periods of time to obtain mastery of the techniques [1]. In recent years, computer assisted surgery (CAS) and robotics have played a large role in reducing these complications through advanced instruments, control and visualization. Using the surgical console or laparoscope display, pre- and intra-operative imaging can be integrated to the surgical workflow improving planning and understanding during the operation. In robotic systems, master manipulators are used to control articulated instruments which provide the surgeon with precision and dexterity which rival open surgery. However, significant challenges remain with achieving full integration of computer assistance and robotics within MIS. An important aspect of this involves understanding the 3D position and orientation of the instruments the surgeon is working with during the operation. This can be used to provide direct benefits such as dynamic motion constraints [2] or to detect tool-tissue interactions [3] or alternatively the motion data from tracked instruments can be used to help quantify the training process for junior surgeons, giving specific feedback on areas of weakness or to provide metrics for surgical skill.

Early methods of instrument tracking involved attaching external electromagnetic or optical markers to the instruments and then estimating pose with a specialized tracking system [4], [5] and these methods remain popular today. However, the process of attaching markers to instruments as well as introducing tracking systems to the operating room (OR) complicates the surgical workflow and adds issues with sterilization and cost. In contrast, image based solutions based on computer vision provide an alternative that can be realised entirely in software with no modification to the surgical setup. This is hugely advantageous as methods can be easily translated to clinic use without an extensive process of distributing markers to hospitals and training medical staff how to attach them correctly [6].

Estimating the pose of instruments using the images from a surgical camera involves a process of extracting image features such as edges, points or regions and then solving alignment cost functions which measure the agreement between parameterized models of the target object and the extracted features. This has been achieved using pipelines of simple models [7] where manually specified thresholds are iteratively applied to estimate parameters. This has also been achieved from an information maximization perspective [8]. More recent methods achieve greater robustness and accuracy by building much more complex cost functions where parametrized models are iteratively fit to image data however optimization in the case of articulated instruments has proved challenging [9]. As an alternative to complex generative models, discriminative models have also shown strong performance, particularly when accompanied by larger training datasets. These usually take

Manuscript received October 29, 2017; revised January 11, 2018; accepted January 11, 2018. Date of publication February 19, 2018; date of current version May 1, 2018. The work of M. Allan was supported in part by EPSRC (EP/J500331/1) and in part by the Rabin Ezra Foundation. *(Corresponding author: M. Allan.)*

M. Allan was with University College London, London WC1E 6BT, U.K. He is now with Intuitive Surgical, Sunnyvale, CA 94086 USA (e-mail: max.allan@intusurg.com).

S. Ourselin, D. J. Hawkes, and D. Stoyanov are with the Centre for Medical Image Computing, University College London, London WC1E 6BT, U.K.

J. D. Kelly is with the Division of Surgery and Interventional Science, University College London Hospital, London WC1E 6AU, U.K.

Digital Object Identifier 10.1109/TMI.2018.2794439

the form of 2D sliding window detectors [10], [11] but dealing with in-plane rotation of laparoscopic instruments is challenging. This can be achieved with rotated features [12] however online updates to the window orientation requires an additional tracker. As more procedures are carried out with robotic instruments, interest in tracking these articulated joints has increased. Using deep neural networks to directly regress articulated joint locations has been demonstrated with excellent results [13]–[15]. However, for surgical instruments these methods are limited to 2D pose estimation and for 3D localization, mainstream computer vision methods [16], [17] have achieved success in learning pose distributions from vast datasets which are used to find plausible candidates. However, for robotic surgical instruments, training data in the quantities required to perform this type of modelling does not yet exist and in this case the most straightforward method of achieving 3D pose estimation is to use the kinematics of the robot, for which the several mm of absolute positioning error at the tip is corrected by 2D detections, for instance using learned texture features on the instrument head [18] and with rendered templates [19]. Although these methods achieve excellent accuracy, they are limited as they require real-time access to the robot API to read the joint data. Although this is feasible in controlled laboratory setups, in the operating room this access is uncommon. In addition to this, articulated laparoscopic instruments are unlikely to support joint access at any point reducing the scope of this type of method.

In our recent work [20], we demonstrated a region-based tracking method which solved for 3D pose by aligning a rigid CAD model with image features and optical flow. In this work we have made several significant improvements. Firstly, the original work was limited as it could not track the articulated DOF of robotic instruments as the optimization was only performed over a parameter set of a single rigid Euclidean transform. Here we incorporated the articulated DOFs which can be achieved naturally within the CAD model alignment system. This involves extending the jacobians to take into account the rotation of the wrist and claspers of the robotic instruments. To the best of our knowledge, this is the first method of gradient-based optimization which is capable of tracking articulated robotic instruments in 3D without the need for external markers or kinematic data from the robot. This is a significant advantage of our method as it is applicable to both articulated laparoscopic instruments and robotic systems that generally do not give access to public APIs to read joint encoder data. Additionally, our method enables the tracking of flexible [21] and hydraulic [22] surgical robots which typically provide very inaccurate encoder based tracking. Our method also allows retrospective analysis of the numerous available datasets where only video data has been captured. A further improvement of our method is that we introduce an online learning system to dynamically update the color models used to generate segmentations. This enables our method to handle more complex appearance and lighting changes. A final contribution of our current work is the extensive comparative evaluation against 2 currently published 3D robotic instrument tracking methods, this is a meaningful contribution as very few published works make direct comparison to other methodologies.

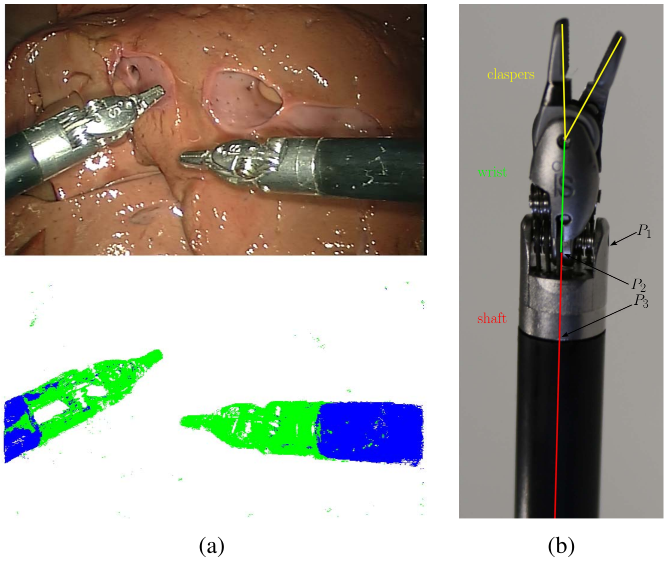


Fig. 1. (a) The feature distribution for each of the $K = 3$ classes with output classification. Region based pose estimation seeks to align projections of 3D CAD models with classifications images. (b) The typical shaft/head divide for many robotic surgical instruments. Together we refer to the the wrist and the claspers as the head. Points $P_{1,2,3}$ refer to the 3 reference points used in the experiments section.

II. Method

A. 3D Tracking With Level Sets

3D instrument tracking attempts to estimate the parameters of the transform ${}^c\mathbf{T}_m$ between the camera coordinate frame $\mathcal{F}_c$ and a model centric instrument coordinate system $\mathcal{F}_m$ (see Figure 2a). When the target object is fully rigid, this transform is composed of a 6 DOF Euclidean transform made up of a rigid rotation $\mathbf{R} \in \mathbb{SO}3$ and a translation $\mathbf{t} \in \mathbb{R}^3$. However, for complex articulated and deforming objects, ${}^c\mathbf{T}_m$ contains the standard rigid transformation but is augmented with a separate transform which articulates the model relative to its base coordinate frame ${}^m\mathbf{T}_{warp}$. The entire rigid transform is parameterized by a vector $\boldsymbol{\theta}$ however we generally omit this for brevity and refer to ${}^c\mathbf{T}_m\ {}^m\mathbf{T}_{warp}(\boldsymbol{\theta})$ as $\mathbf{T}$.

Region-based methods of estimating the parameters of $\mathbf{T}$ involve using an estimate of this transform to position the vertices of a CAD model of the instrument in $\mathcal{F}_c$ and generating one or more silhouette regions from the projection of these vertices onto the camera plane using the classic pinhole camera model (see Figure 2). Pose estimation is then formulated by finding the set of parameters such that the generated *model* silhouettes match *data* silhouettes obtained from a pixel-wise classification of the image pixels [23]–[27]. Many methods [27], [28] perform a 2 step estimation process whereby a full data silhouette is extracted from the image and backprojected to allow reverse engineering of the pose parameters in a separate step. However, [23], [29] proposed a direct method of which bypasses obtaining a full data silhouette and instead assesses the model silhouette using local information from around the projection. This formulation is

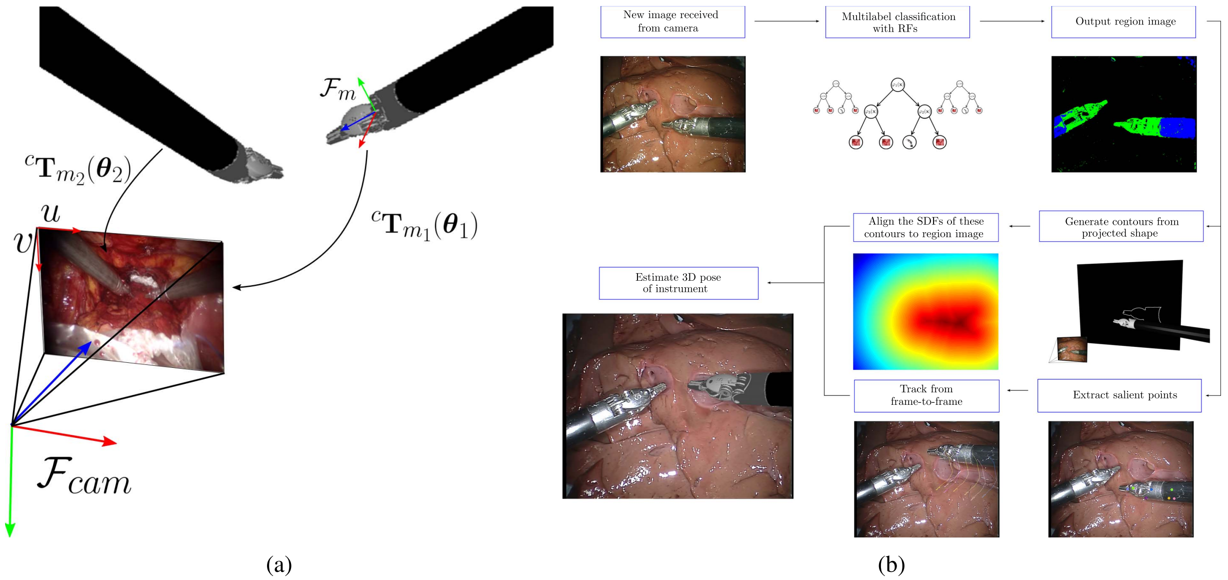

$\mathcal{F}_m$
${}^c\mathbf{T}_{m_2}(\boldsymbol{\theta}_2)$
u
v
${}^c\mathbf{T}_{m_1}(\boldsymbol{\theta}_1)$
$\mathcal{F}_{cam}$
(a)
New image received from camera
Multilabel classification with RFs
Output region image
Align the SDFs of these contours to region image
Generate contours from projected shape
Estimate 3D pose of instrument
Track from frame-to-frame
Extract salient points
(b)

Fig. 2. (a) Shows 2 instruments in front of a camera where a transform from model coordinates to camera coordinates is performed with T. (b) Shows the workflow of the entire algorithm, where the projection of the instrument is aligned with the RF output and 2D tracked points to estimate 3D pose.

greatly simplified over working with a 2 step process as it does not require complex regularizations to maintain a suitable shape when finding the data silhouette, instead relying on a strict shape prior provided by the CAD model projection. Bayesian approaches using learned shape spaces have also been used to this end [30]–[32].

In typical 3D tracking frameworks, a single contour is used to model the entire shape [24], [33], [34]. This allows the problem to be cast as contour matching using silhouettes. This simplification affords a great deal of invariance with respect to the chosen object and typically works well when the appearance model between foreground and background is strong, resulting in a clean contour. However, for manufactured robotic instruments, this simplification ignores strong internal homogeneous regions which can be useful in generating strong delineating contours (see Figure 1b) between the plastic shaft and the metallic clevis. A particular advantage of this additional contour is that it constructs a fully visible single contour, which is not the case for a binary silhouette as this contour intersects the edge of the image, and this can in principal provide information about foreshortening and additionally constrain the instrument when the clevis is occluded by tissue.

Estimating the optimal 3D pose using region-based methods involves defining an energy functional E_r (r denotes region) which measures the alignment of K data silhouettes obtained from statistical models over the image data with K model silhouettes generated from projections of a surgical instrument CAD model. This functional is composed of a sum over K binary alignments, where the form of each summed-cost mirrors a standard region-based segmentation [35]:

$$E_r(\boldsymbol{\theta}) = -\sum_{i}^{K} \int_{\Omega} \log(H(\phi^i(\mathbf{x}, \mathcal{C}_i(\boldsymbol{\theta})))f(I(\mathbf{x}), \chi_i) + (1 - H(\phi^i(\mathbf{x}, \mathcal{C}_i(\boldsymbol{\theta})))f(I(\mathbf{x}), \chi_{n(i)}))d\mathbf{x} \quad (1)$$

where the terms $f(I(\mathbf{x}), \chi_i)$ and $f(I(\mathbf{x}), \chi_{n(i)})$ are functions which return the probability that the pixel data $I(\mathbf{x})$ belongs to either the class i or the set of all other classes $n(i)$. Each statistical model is dependent on appearance parameters for the i^{th} region χ_i. The term $H(.)$ represents the smoothed Heaviside function, which is commonly used in mathematical models to filter other functions by discreet membership and in this case is used to indicate if a pixel $\mathbf{x}$ belongs to the silhouette i or the background. This silhouette is described by a closed contour $\mathcal{C}_i$ which is described as a level set by embedding it in a signed distance function ϕ. This is a beneficial representation over parametric competitors such as splines as it allows greater mathematical flexibility and does not suffer from numerical problems during optimization. This distance function is directly generated from the projection of the model and hence this function, and the contour, are parameterized by $\boldsymbol{\theta}$.

We use random forests (RFs) to provide the response $f(.)$ allowing data silhouettes to be extracted from a single background region. RFs are popular for solving many challenging problems including pose estimation [36], semantic image segmentation [37] and camera relocalization [38]. They have been shown to be fast, parallelizable and accurate while providing simplicity to the user and an ability to handle even high dimensional data [39]. An RF is an ensemble learner where a collection of randomized decision trees vote on a hypothesis for an input $\mathbf{x}$ which is aggregated into a single output using an averaging scheme. The decision trees are constructed as a sequence of linear classifiers $y = \mathbf{w}\mathbf{x}$ which direct input samples to one of two *child* nodes depending a thresholding of y. This parent to child splitting is applied recursively until $\mathbf{x}$ reaches a *leaf* node where a posterior distribution is assigned.

Rather than using RGB pixel intensities directly, we instead transform our training data into the Opponent 1, Red, a from the CIE Lab color space and Gabor filter output. A small but important modification which we make to our training implementation compared with [20] is to use class balancing. In normal MIS images, background data are much more common than instrument data which, in the case of a 0-1 indicator loss function, leads to learning decision boundaries which favor selecting background labels over foreground labels in ambiguous cases. However, when working within our silhouette based framework, correctly labelling foreground examples so that a complete silhouette is observed is more important than eliminating isolated regions of noise (effectively false negatives are much more detrimental than false positives).

To improve the quality of the segmentation used to drive the region-based pose estimation, we can make improvements to the RF. Firstly, as we only wish to classify the background and foreground in regions near the model contour, it makes sense to learn a highly specific model for the appearance using only pixels which sit close to this boundary. As we have a full 3D model of the instrument, we can generate automatic ground truth segmentations from the signed distance function ϕ and select training data from a 30 pixel wide boundary, this value was chosen experimentally. After 5 frames, we retrain the forest. For further details, see Figure 3. Preliminary experiments showed that the most effective strategy was to learn a constant foreground model from the first frame and update the background model data online by sampling from the first frame. This prevents model drift from affecting the training data significantly by incorrectly placing background pixels into the foreground class and vice-versa. This works as we use a bag-of-pixels model which is resilient to movements of the tissue that occur in normal operating interaction. However, upon camera motion the background model would have to be relearned. We could in principal detect this motion with optical flow and reinitialize the model from the segmentation boundary once the camera motion ceases. This technique was discovered to be much more effective than using the current frame to update the background model as this leads to drift when tracking begins to fail.

B. Optical Flow Tracking

When using a silhouette to estimate the pose of any object, a significant challenge arises because of ambiguities in the mapping between pose and silhouette. A simple example being when a sphere is rotated to any angle, the silhouette does not change. A similar problem occurs with the near cylindrical shape of the instruments used in minimally invasive surgery which, when undergoing rotation around the roll axis, do not change their silhouette significantly.

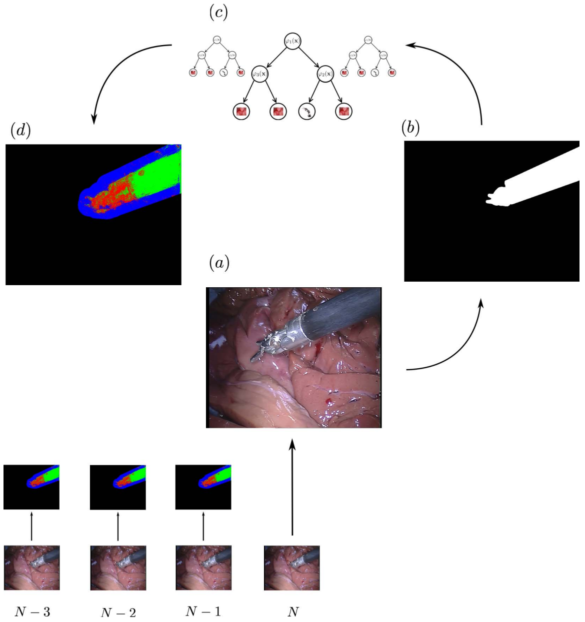


Fig. 3. The online forest algorithm. (a) For each new frame N, we check if the forest needs to be re-learned and generate a ground truth mask from the projection of the estimate of the pose at frame N (b) onto the frame 1. By only using pixels from a fixed size region around the contour, we are able to generate background samples to learn a new model (c) the output of which is shown in (d). The foreground samples are not refreshed from the first frame. The advantage of resampling from the first frame is that we obtain robustness to model drift which causes the projection at frame N to be inaccurate.

To solve this problem, we propose to combine the silhouette based features, which represent the surface appearance of the instrument as a bag of pixels, with multiple independent Lucas-Kanade optical flow features [40]. This retains enough surface spatial information to allow the ambiguous DOF to be estimated without the penalty of a highly non-convex cost function, which is common in full photo-consistency based object tracking. The idea of tracking 2D information on the instrument surface as an additional method of constraining the pose estimation is very simple and works on the principal that if we can match several 2D tracked image points to 3D points on the model surface, we can estimate the 3D transformation to the instrument by minimizing the reprojection error between the predicted 2D point locations $[x, y]^T$ and their correspondences $[\hat{x}, \hat{y}]^T$ in the image. This can be defined by with objective energy function E_p, where similarly to Equation 1, p denotes the use of a point-based cost:

$$E_p(\boldsymbol{\theta}) = \sum_{i \in W^{t+1}} ||\mathbf{K}\mathbf{T}\mathbf{X}_i^t - [\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]^T||_2^2 \qquad (2)$$

where $||.||_2^2$ denotes the squared L_2 norm, although other distance metrics are commonly used [41]. $[\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]^T$ denotes a corresponding point location in the frame at time $t+1$ which was matched with the point projected from the vertex location $\mathbf{X}_i^t$ at t. W^{t+1} is the set of matched points between frames at times t and $t+1$. $\mathbf{K}$ is the calibration matrix for the classic pinhole camera model.

C. Modelling Articulation With Kinematic Chains

In MIS, manufactured robotic manipulators such as surgical instruments have a known set of possible transformations which constrain the vertices of each joint to rotate or translate around or along a single axis (see Figure 5). Hence, this allows the warping transform ${}^{m}\mathbf{T}_{warp}$ to be represented as a composition of several single axis transforms ${}^{n-1}\mathbf{T}_n$ which are applied consecutively to different subsets of the model vertices.

A kinematic chain is the most common method of describing a robot manipulator by dividing it into an assembly of Γ links or rigid bodies each of which define a coordinate frame $\mathcal{F}$. These links are connected together at a shared axis known as a joint, where for an Γ link chain there are at most $\Gamma - 1$ joints. The coordinate frames of consecutive links are related with a single 4×4 transform ${}^{n-1}\mathbf{T}_n$ which is described with one or more DOFs, which specifies how many parameters are required to fully locate the geometry of the connected n^{th} link in the reference frame of the parent $n-1^{\text{th}}$ link [42]. The most common case for robotic manipulators is to use a single DOF joint where the transform is defined to rotate around 1 axis (rotary) or translate along 1 axis (prismatic) and in fact any K DOF joint can be modelled as a series of single DOF joints [42].

When combined together, the links and joints of a kinematic chain describe how a point $\mathbf{X}$ defined in the local coordinate system of the $j^{\text{th}} : j \leq \Gamma$ link $\mathcal{F}_j$ can be transformed into the coordinate system of the base frame of the robot as:

$$\mathbf{X}_{\mathcal{F}_0} = {}^0\mathbf{T}_1 \ {}^1\mathbf{T}_2 \ldots {}^{j-1}\mathbf{T}_j \mathbf{X}_{\mathcal{F}_j} \tag{3}$$

where ${}^0\mathbf{T}_1 \ {}^1\mathbf{T}_2 \ldots {}^{j-1}\mathbf{T}_j$ can be compactly represented as ${}^0\mathbf{T}_j$, $\mathbf{X}_{\mathcal{F}_j}$ is the representation of $\mathbf{X}$ in $\mathcal{F}_j$ and $\mathbf{X}_{\mathcal{F}_0}$ is the representation of $\mathbf{X}$ in $\mathcal{F}_0$.

There are several methods to define the transform between neighbouring links and for general transforms, 6 DOFs are required to fully specify the relative orientation. However, for single DOF joints, the Denavit Hartenberg (DH) representation [43] defines the n^{th} joint to be parallel to the $x = 0$ plane of $\mathcal{F}_{n-1}$, effectively cancelling out 2 degrees of freedom, 1 in rotation and 1 in translation reducing the number of parameters to 4, 2 distances and 2 angles [44]. 1 distance parameter is required to describe how far along the x axis of $\mathcal{F}_{n-1}$ the plane defined by joints $n-1$ and n lies and 1 angle parameter describes the rotation between the joints in this plane. These 2 parameters are denoted a_{n-1} and α_{n-1} respectively. Describing how $\mathcal{F}_n$ is attached to the z axis of $\mathcal{F}_n$ and orientated relative to $\mathcal{F}_{n-1}$ involves a further 2 parameters. Firstly, the distance along this common axis between where a_{n-1} from link $n-1$ intersects the common axis and where a_n from link n intersects the common axis is defined as d_n and describes the vertical shift between the two links. Additionally, the rotation around the z axis of $\mathcal{F}_n$ between the 2 links is defined as θ_n. When applied to a prismatic joint i a_i, α_i, θ_i are fixed and d_i is the DOF whereas for a revolute joint i, a_i, α_i, d_i are fixed and θ_i is the DOF. These 4 rotation and translation operations are applied consecutively to provide a single transform ${}^{n-1}\mathbf{T}_n$ as:

$${}^{n-1}\mathbf{T}_n = R_{x_{n-1}}(\alpha_{n-1}) \cdot T_{x_{n-1}}(a_{n-1}) \cdot R_{z_n}(\theta_n) \cdot T_{z_n}(d_n) \tag{4}$$

where $R_{x_{n-1}}$ refers to a 4×4 transform composing a rotation matrix around the x axis of frame F_{n-1} with a zero translation and R_{z_n} has the same meaning but the rotation component is defined around the z axis of frame $\mathcal{F}_n$. $T_{x_{n-1}}$ and T_{z_n} refer to same concept but the rotation part of the transform is the identity matrix and the translation part is a translation along the x and z axes of frames $\mathcal{F}_{n-1}$ and $\mathcal{F}_n$ respectively.

D. DH Parameters for da Vinci Robotic Instruments

In this work we focus solely on working with the instruments of the da Vinci robotic system, particularly the LND instrument which is commonly used in surgical procedures to control a suturing needle. However, the methods are easily applicable to any robotic instrument with the appropriate minor modifications. The LND, like any da Vinci instrument, has 3 DOFs on the wrist: firstly, the wrist pitch (WP) which articulates the entire wrist to mimic the motion of a human wrist enabling the mirroring of motions such as stitching to be captured more precisely. The second DOF is the wrist yaw (WY) which corresponds to a coordinated motion of two mechanical joints representing the claspers and enables the claspers to be oriented towards a target. The final DOF allows the clasper to open and close so that the instrument can grasp and hold objects. This results in the final parameterisation of our instrument being the 6 rigid DOFs of the model to camera rotation and translation and a further 3 DOFs which describe how the instrument wrist is oriented relative to the shaft of the instrument. For reference, the DH parameters for the LND are shown in Table I.

TABLE I
LARGE NEEDLE DRIVER DH PARAMETERS FOR THE ARTICULATED WRIST. THESE REFER TO THE LAST 3 JOINTS IN A 7 DOF DA VINCI ARM. THE MEANINGS OF THE TERMS CAN BE SEEN IN FIGURE 4 AND THE RELATIONSHIP OF EACH FRAME TO THE INSTRUMENT LINKS CAN BE SEEN IN FIGURE 5

	$\mathbf{a_{i-1}}$ (m)	$\boldsymbol{\alpha_{i-1}}$ (rads)	$\mathbf{d_i}$ (m)	$\boldsymbol{\theta_i}$ (rads)
Wrist Pitch $\mathcal{F}_1$	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
Wrist Yaw $\mathcal{F}_2$	0.009	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$
Grip $\mathcal{F}_3$	0	$-\frac{\pi}{2}$	0	0

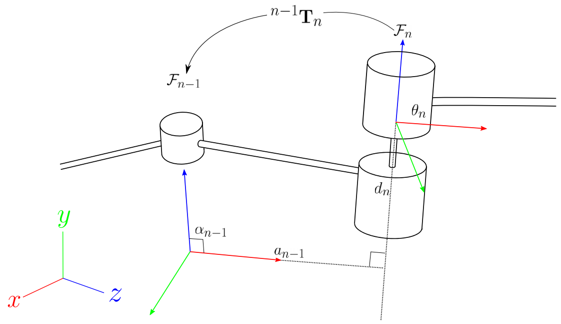


Fig. 4. The coordinate system transforms used in modified DH parameter setup. A point defined in the frame $\mathcal{F}_n$ can be transformed into the frame $\mathcal{F}_{n-1}$ with the transform ${}^{n-1}T_n$.

E. Optimization

We jointly optimize over the region based energy, referred to from here on as $E_r(\boldsymbol{\theta})$, and point based energy computed optical flow, $E_p(\boldsymbol{\theta})$ using gradient descent and a weighting factor λ to allow both terms to have more equitable influence. In our experiments we set λ so that the Jacobians from the point estimates have 0.8 of the magnitude of the Jacobians from the region-based energy:

$$E(\boldsymbol{\theta}) = E_r(\boldsymbol{\theta}) + \lambda E_p(\boldsymbol{\theta}) \tag{5}$$

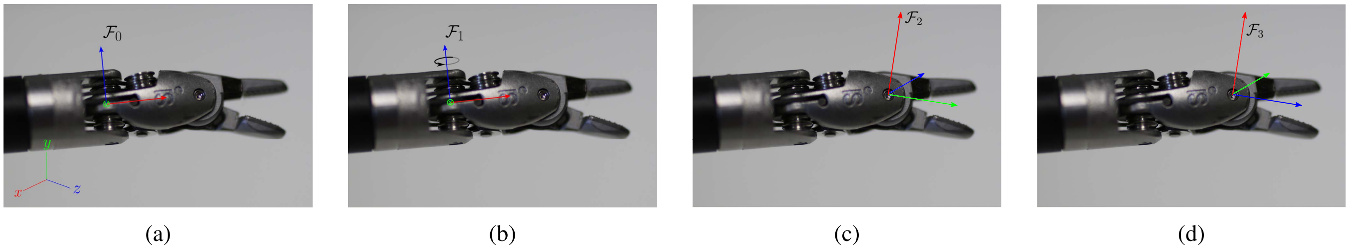


Fig. 5. (a) The base frame $\mathcal{F}_0$ for the robotic instrument which is oriented relative to the surgical camera with the rigid body transform ${}^{c}\mathrm{T}_{m}$. (b) The wrist frame $\mathcal{F}_1$ which enables the instrument head to rotate around the z axis of this frame. (c) The claspers rotate together around the z axis of $\mathcal{F}_1$ defining a new frame $\mathcal{F}_2$ which has its x axis pointing in the direction of the claspers. (d) The claspers rotate around the *z* axis of this frame in opposite directions allowing opening and closing.

where the derivative is computed as:

$$\frac{\partial E(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial E_r(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \lambda \frac{\partial E_p(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{6}$$

and the individual cost derivatives are:

$$\frac{\partial E_r(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\sum_{k \in K}\sum_{i \in \Omega} \frac{f(I(\mathbf{x}), \chi_k) - f(I(\mathbf{x}), \chi_{n(k)})}{W} \frac{\partial H}{\partial \boldsymbol{\theta}} \tag{7}$$

where

$$\begin{aligned} W = {} & H(\phi^k(\mathbf{x}, \boldsymbol{\theta})) f(I(\mathbf{x}), \chi_k) \\ & + (1 - H(\phi^k(\mathbf{x}\boldsymbol{\theta})) f(I(\mathbf{x}), \chi_{n(k)}) \end{aligned} \tag{8}$$

and

$$\frac{\partial H}{\partial \boldsymbol{\theta}} = \delta(\mathbf{x}) \left[\frac{\partial \phi^k(\mathbf{x}, \boldsymbol{\theta})}{\partial x}\frac{\partial x}{\partial \boldsymbol{\theta}}, \frac{\partial \phi^k(\mathbf{x}, \boldsymbol{\theta})}{\partial y}\frac{\partial y}{\partial \boldsymbol{\theta}} \right] \tag{9}$$

where $\partial \phi^k(\mathbf{x}, \boldsymbol{\theta})/\partial x, y$ can be computed using finite differences and $\delta(.)$ is the derivative of the smoothed Heaviside function and corresponds to a smoothed Dirac delta function which has the effect of weighting the derivative terms so that only the points around the contour contribute to the optimization.

$$\begin{aligned} \frac{\partial E_p(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \sum_{i \in W_{t+1}} \frac{\partial}{\partial \boldsymbol{\theta}} ||\mathbf{KTX}_i^t - [\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]||_2^2 \\ &= \sum_{i \in W_{t+1}} 2[\mathbf{KTX}_i^t - [\hat{x}_i^{t+1}, \hat{y}_i^{t+1}]] \\ &= \left[x_i^t - \hat{x}_i^t, y_i^t - \hat{y}_i^t] \right]^T \cdot \left[\frac{\partial x}{\partial \boldsymbol{\theta}}, \frac{\partial y}{\partial \boldsymbol{\theta}} \right] \end{aligned} \tag{10}$$

Equations 9 and 10 requires derivatives of 2D pixel coordinates with respect to the transform $\mathbf{T}$.

$$\frac{\partial x}{\partial \boldsymbol{\theta}} = f_u \frac{1}{Z^2}\left(Z\frac{\partial X}{\partial \boldsymbol{\theta}} - X\frac{\partial Z}{\partial \boldsymbol{\theta}} \right) \tag{11}$$

$$\frac{\partial y}{\partial \boldsymbol{\theta}} = f_v \frac{1}{Z^2}\left(Z\frac{\partial Y}{\partial \boldsymbol{\theta}} - Y\frac{\partial Z}{\partial \boldsymbol{\theta}} \right) \tag{12}$$

where $[X, Y, Z]^T = {}^{c}\mathbf{T}_i \mathbf{X}_{\mathcal{F}_i}$ is the representation of the vertex which generated the pixel (x, y) transformed from the link frame $\mathcal{F}_i$ into camera coordinates. The derivatives of these terms with respect to the translation and rotation are well known [24] however the derivatives of the parameters of the articulated components merit further discussion. They are obtainable in closed form by differentiating the kinematic chain with respect to each articulated component parameter. The variables of Equation 11 and 12 which depends on these components is the projected 3D vertex position $\mathbf{x} = \mathbf{K}\,{}^{c}\mathbf{T}_i \mathbf{X}_{\mathcal{F}_i}$, where $\mathbf{X}_{\mathcal{F}_i}$ is defined in the local coordinate system of the link i on which $\mathbf{X}$ lies and ${}^{c}\mathbf{T}_i$ defines the transform from the camera frame to this frame. The Jacobian of the frame to camera transform part of this equation breaks down as:

$$\frac{\partial\, {}^{c}\mathbf{T}_i \mathbf{X}_{\mathcal{F}_i}}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}\, {}^{c}\mathbf{T}_0\, {}^{0}\mathbf{T}_{j-1}\, {}^{j-1}\mathbf{T}_j\, {}^{j}\mathbf{T}_i \mathbf{X}_{\mathcal{F}_i} \tag{13}$$

where ${}^{j-1}\mathbf{T}_j$ is the transform from the parent of frame $\mathcal{F}_j$ to $\mathcal{F}_j$. If we consider the parameter θ_j which is responsible for rotating the jth link around the z axis of its frame (see Section II-C), then the derivative becomes:

$$\begin{aligned} \frac{\partial^{c}\mathbf{T}_i(\boldsymbol{\theta})\mathbf{X}_i}{\partial \theta_j} &= {}^{c}\mathbf{T}_0\, {}^{0}\mathbf{T}_{j-1} \left(\frac{\partial}{\partial \theta_j}\, {}^{j-1}\mathbf{T}_j \right) {}^{j}\mathbf{T}_i \mathbf{X}_{\mathcal{F}_i} && (14) \\ &= {}^{c}\mathbf{T}_0\, {}^{0}\mathbf{T}_{j-1} \left(\mathbf{z} \times \mathbf{X}_{\mathcal{F}_j} \right) && (15) \end{aligned}$$

where the product rule is applied to each transform of the kinematic chain and, as each parameter influences directly only a single $\mathbf{T}$, all but a single term is zero. The vertex $\mathbf{X}_{\mathcal{F}_i}$ is effectively transformed into the coordinate frame $\mathcal{F}_j$ as this equation measures how motion of the frame j influences vertices in frames towards the distal end of the kinematic chain.

III. Experiments

To evaluate the accuracy of the articulated tracking we perform quantitative ex-vivo and qualitative in-vivo studies. However, as several recently published methods of articulated instrument tracking provide comparison datasets, we can also perform a quantitative comparison with these methods.

A. Implementation Details

Our implementation[1] makes use of OpenGL/GLSL and we describe our model as a tree of nodes in a parent-child relationship. For the example da Vinci LND model, this consists of a base frame containing the shaft which has a single child node containing the wrist model (see Fig. 5). This again has a single child node containing the clasper axis but no geometry which in turn has 2 child nodes containing

[1] https://github.com/surgical-vision/ttrack/

each clasper. At each successive pose iteration, the vertices of each node are projected to an *index image* which contains the numerical index of node which owns the geometry of the vertex. This is used to determine which vertices influence each term in the Jacobian computation. Currently our non-optimized method is not real-time, with processing time for a single 720×576 image taking ≈ 0.3 seconds per gradient descent step with between 10-20 steps required for convergence. However, the cost function gradients are evaluated as an independent sum-over-pixels and is therefore highly parallelizable, with similar implementations achieving real-time performance [24]. We solve our cost-function by reinitializing from the pose in the previous frame but do not incorporate any motion modelling to make forward predictions. Our method requires manual initialization in the first frame, which we achieve with a GUI based tool.[2] This is used to initialize the pose of the instrument model which in turn is used to generate the initial ground truth image segmentation to train the RF.

B. Ex-Vivo Experiments

We construct 2 ex-vivo experiments using the da Vinci LND instrument and several different animal tissue samples. The camera maintains a static position and observes 1000 frame sequences showing an instrument moving with articulation of the wrist and claspers. The DVRK platform is used to capture synchronised joint and video data and we use the GUI based manual initialization technique to correct errors in the joint configuration and obtain a more accurate ground truth. Plots showing the translation and rotation parameters of the instrument reference frames, the errors in the wrist and clasper position and errors in the relative position of 3 static points on the MR LK tracked model and the ground truth model (see Figure 1b) are shown in Figures 6 and 8. We evaluate parameter errors in 3D space directly, rather than measuring 2D projection error given that most applications of 3D tracking are impacted more heavily by errors in world space. Furthermore, using the error between corresponding points allows us to represent the accuracy of our algorithm without dependence on an arbitrarily chosen origin. We also show renderings of the instrument pose over the video frames are shown in Figures 7 and 9.

TABLE II

OVERLAP PRECISION, RECALL AND F1 SCORE FOR THE 4 FRAMES USED IN THE EVALUATION IN [9]. AS WE PERFORMED THIS EVALUATION OURSELVES USING HAND-CRAFTED MASKS THE RESULTS REPORTED IN THIS TABLE FOR THE METHOD OF [9] ARE SLIGHTLY DIFFERENT, ALBEIT BETTER THAN THE RESULTS IN THE ORIGINAL PAPER

	Precision - [9]	**Recall** - [9]	**F1** - [9]	**Precision** - Ours	**Recall** - Ours	**F1** - Ours
Frame 25	**0.96**	0.70	0.81	0.84	**0.96**	**0.90**
Frame 75	**0.96**	0.85	0.90	0.83	**0.99**	**0.91**
Frame 125	0.84	0.60	0.70	**0.93**	**0.92**	**0.92**
Frame 175	**0.94**	0.80	0.87	0.90	**0.85**	0.87
Average	**0.93**	0.74	0.82	0.87	**0.93**	**0.90**

C. Quantitative Comparison Results

Recent articulated robotic tracking methods [9], [19], [45] allow us to provide a quantitative comparison method between our fully visual technique and methods that combine visual tracking with robotic kinematic information. Our first comparison is between our method and that of [9] which provided a method of tracking general 3D articulated object and contained a validation section on robotic surgical instruments. This method used a similar region overlap type metric to our technique incorporating multiple instrument regions to provide added robustness. However, this was formulated within a gradient-free optimization as the simple overlap metric did not allow for analytical Jacobians to be computed. This lead to slow and often inaccurate solutions for robotic instruments although the method worked well for retinal instruments and human hands. We show results using the 4 frame evaluation used in the original paper where the 25th, 75th, 125th and 175th frames are manually segmented. We use classification metrics of precision, recall and the F1 score to compare the overlap between the manual segmentation and the rendering of the instrument in that frame. Precision (P), Recall (R) and F1 score (F1) are computed as

$$\begin{aligned} P &= \frac{TP}{TP + FP} \\ R &= \frac{TP}{TP + FN} \\ F1 &= 2(P \times R)/(P + R) \end{aligned} \tag{16}$$

where the F1 score is the harmonic mean of the precision and recall and is often used as a weighted average of the two measures. Quantitative results of these scores are shown in Table II and corresponding qualitative results in Figure 10. The original work of [9] tends to underlap the ground truth slightly, whereas our method tends to overlap slightly which is reflected in the higher precision value for [9] and the higher recall value for our work. However, when taken together, the F1 score shows much higher performance in our method. In this dataset, we make one modification to our method, as the first frame of video does not show a good view of the instrument clasper meaning the color distribution for this class was badly learned from the first frame. To counter this, we chose a later frame to learn our RF, however this is similar to the original authors who chose frames from across the video to learn their color model.

The recent method and data of [19] allows us to compare with the state-of-the-art for 3D articulated instrument tracking which combines robot kinematics with a point based detector to provide accurate real-time tracking. We evaluate on 2 phantom sequences with LND instruments which contain complex articulations which make visual tracking extremely challenging. The results are evaluated quantitative in Table III where the authors manually labelled the centre locations of several tool parts that were used in their point-based detection system to obtain a ground truth. The authors then computed the relative pose between the predicted instrument location and the manually labelled instrument location for all frames in the video. Qualitative evaluation is show in Figure 11. In our analysis of dataset 2, we encountered 1 tracking failure for our method at frame 1200 when the left instrument obtained an inaccurate pose due to a challenging period of articulation.

[2] https://github.com/surgical-vision/viz/

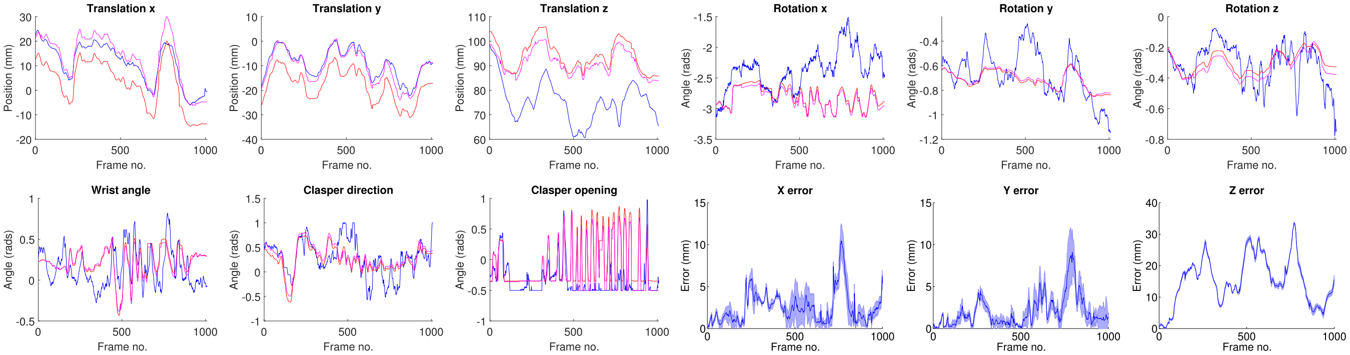
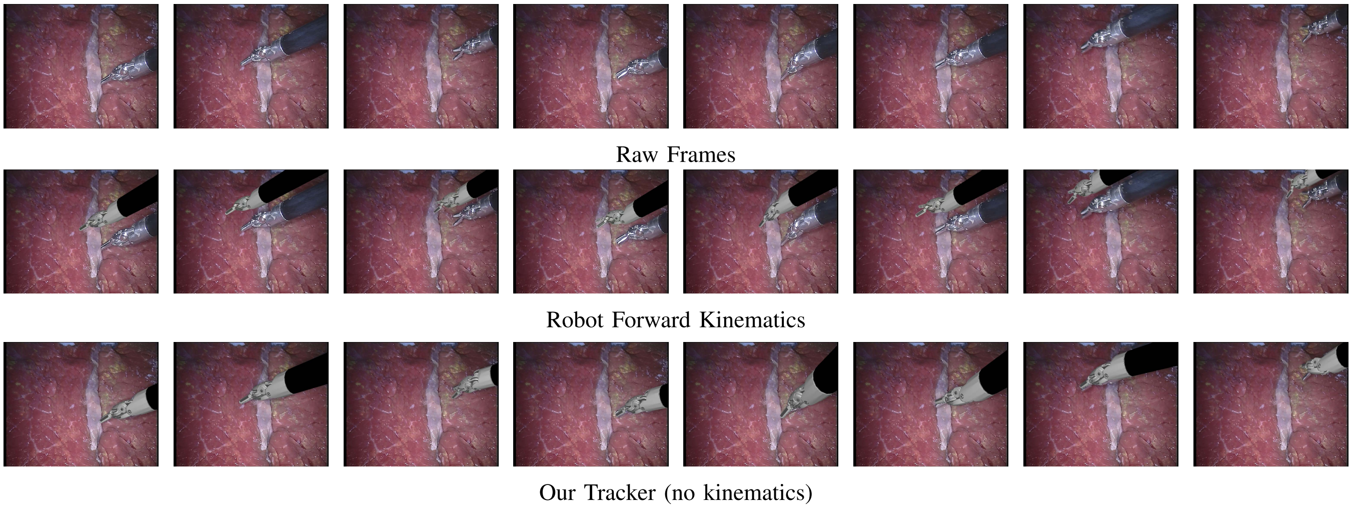


Fig. 6. Robot Kinematics, MR LK Tracker, Ground Truth. The top row shows the trajectories for our tracker and the kinematics compared with the hand corrected ground truth of dataset 1. There are some large rotation errors using the MR LK tracker and around 1.5 cm of t_z error. The t_y error increases and decreases over the sequence which occurs as the instrument converges to the correct pose and then loses tracking. Row 2 shows the trajectories for each of the 3 articulated degrees of freedom at the wrist and also the error distributions for corresponding points, where the blue line shows the mean error and the standard deviation is shown in light blue. Although the error in t_z is large the qualitative results in 7 show that the visual quality of the alignment is still good.


Raw Frames

Robot Forward Kinematics

Our Tracker (no kinematics)

Fig. 7. Qualitative results from dataset 1 showing frames 100, 200, 350, 400, 500, 600, 700 and 1000. The top row shows the original frames, the middle row shows the output from the uncorrected kinematics and the bottom row shows the MR LK tracker.

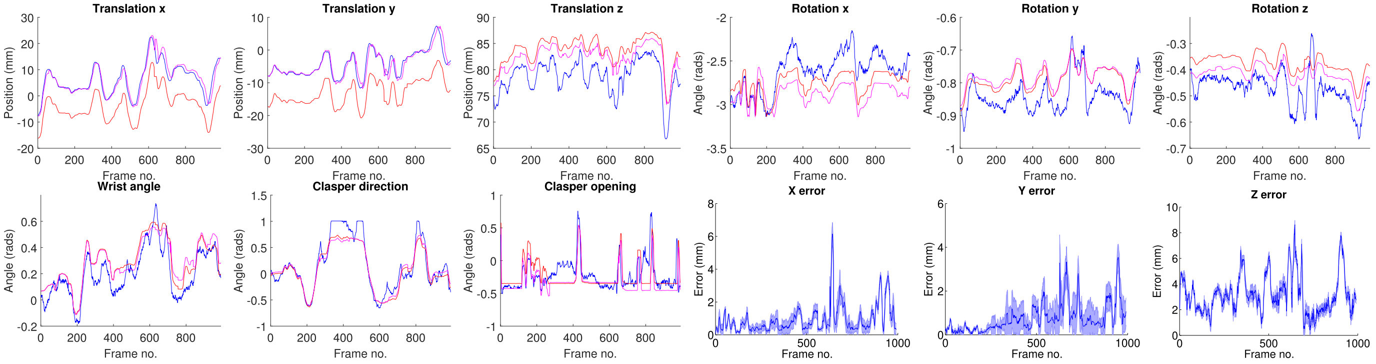


Fig. 8. Robot Kinematics, MR LK Tracker, Ground Truth. The top row shows the trajectories for our tracker and the kinematics compared with the hand corrected ground truth of dataset 2. The MR LK tracker is very accurate over this sequence, due to the excellent color classification against the clean background. Row 2 has the same meaning as in Figure 6.

Although both instruments go through periods of the video when they exhibit inaccurate tracking, this particular sequence was followed by a period when the instruments crossed over one another. This caused large drift in the left instrument which was deemed unrecoverable and a manual initialization was required.

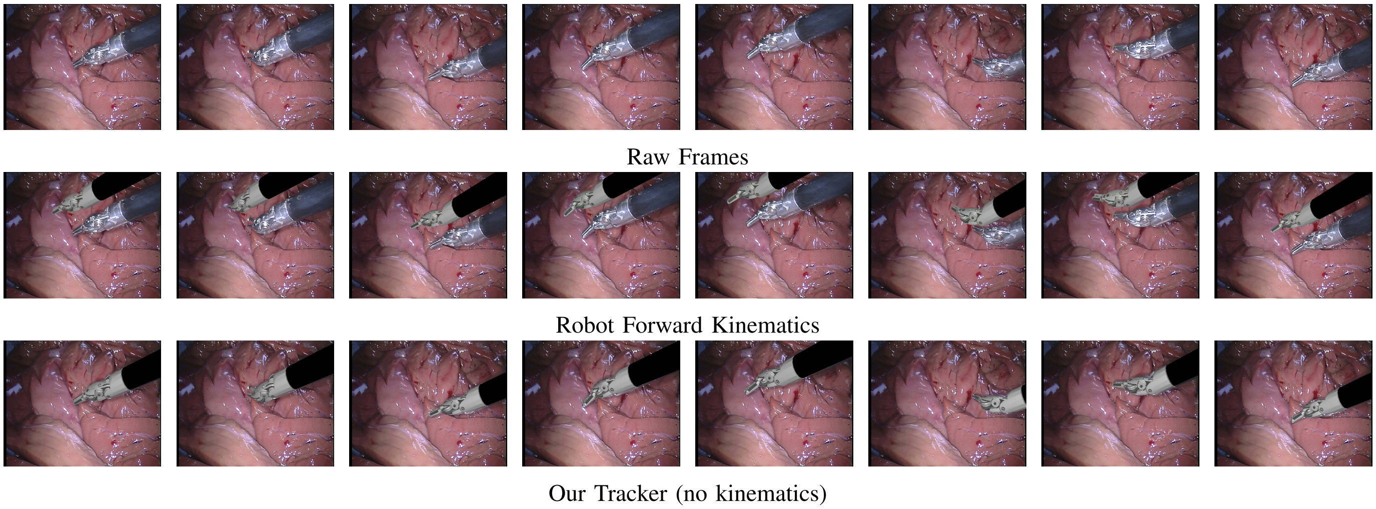


Fig. 9. Qualitative results from ex-vivo dataset 2 showing frames 100, 200, 300, 400, 500, 600, 700 and 800. The top row shows the original frames, the middle row shows the output from the raw, uncorrected kinematics and the bottom row shows the MR LK tracker. In frame 200, the instrument head rotates in and out of view and the MR LK method correctly tracks this.

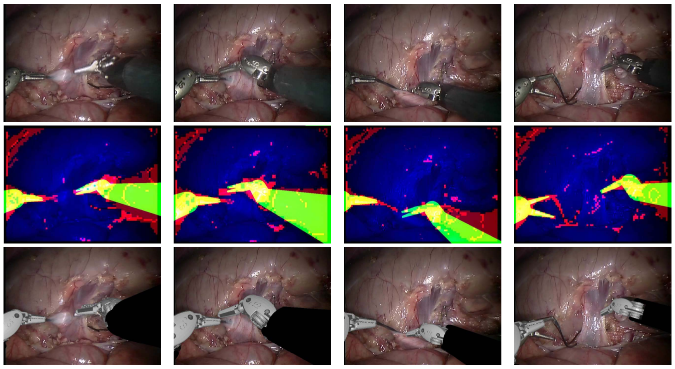


Fig. 10. Visual comparison for the dataset of [9]. This dataset shows a challenging in-vivo sequence with 2 da Vinci LND instruments. The top row shows the raw video frames 25, 75, 125 and 175, the corresponding frames from the method of [9] are in row 2 and the frames from our method are in row 3. Although the data is challenging, both methods show good alignment. Typically our method has better alignment but the right instrument fails to track the clasper opening in frame 175, which is correctly tracked by [9].

TABLE III

The Numerical Accuracy of Our Method Compared With [19]. The Rotation and Translation Error Is Computed for Each Frame From the Manually Labelled Ground Truth Part Locations. Although Our Results Are Not as Accurate as the Method of [19], we Are Still Able to Obtain Good Tracking Over the Majority of the Sequence and Critically Are Not Relying on Kinematics to Perform Our Estimation

Dataset	T error (mm) - Ours	R error (rads) - Ours	T error (mm) - [19]	R error (rads) - [19]
Dataset 1	5.07 ± 2.08	0.43 ± 0.26	**1.50 ± 1.12**	**0.12 ± 0.07**
Dataset 2	3.85 ± 3.64	0.58 ± 0.31	**3.14 ± 1.96**	**0.12 ± 0.08**

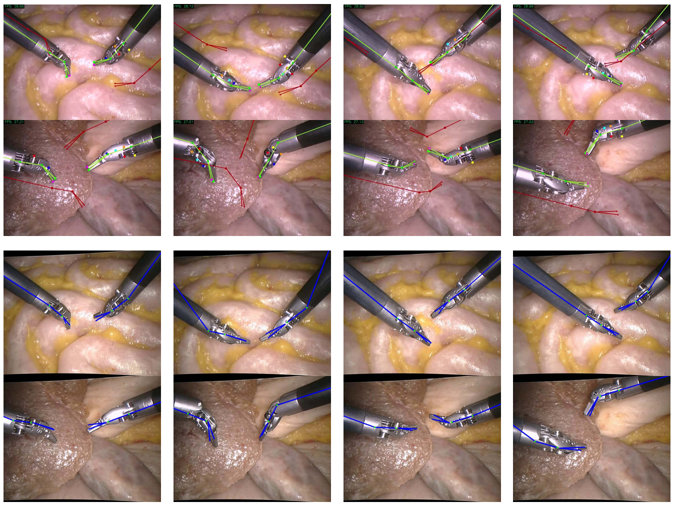


Fig. 11. Visual comparison for the dataset 1 and 2 of [19] where the first 4 images of the top row shows the results of [19] in frames 200, 400, 750 and 950 of dataset 1 and the last 4 images of the top row show frames 350, 450, 900 and 1200 of dataset 2. The bottom row shows our results where we overlay a skeleton of our pose estimation.

IV. Conclusion and Discussion

In this work, we present a novel system of tracking the articulated DOFs of surgical robotic instruments in 3D using a fully vision-based region and point based solution. Our system trivially extends to different instrument models and color schemes which greatly increases the range of robotic systems it can be tested on. Our extensive comparative evaluation draws together data from a wide varies of sources and demonstrates the superior performance of our method against the only other published 3D articulated instrument tracking method that does not make use of robot joint encoders demonstrating the advantage of using gradient based searches for pose estimation. We also obtain competitive results when compared with state-of-the-art methods which unlike our method rely heavily on the data from the robot joint encoders which is a well documented drawback [20]. The method however shows errors in the roll rotation DOF due to visual symmetry as this this DOF is explorer which prevents the region based tracker from locking onto reliable shape information. In principal this is best solved by incorporating more reliable detection information on the instrument surface, for instance making use of recent robust feature detection methods [13]. Additionally depth estimation is a challenge, particularly due to the small baseline of robotic surgical cameras. The main limitation of our method is its requirement for a manual initialization, however this can potentially be provided with user interaction, for instance using the GUI tool we have developed, and additionally we noticed in our experiments that the model suffers from drift, which is a common problem in model based tracking which incorporate temporal information. Future work will look mainly at the integration of prior information to restrain the rigid pose space from a 6 DOF transform to a restricted space and in principal these priors can be learned from kinematic data offline.

References

[1] J. H. Palep, "Robotic assisted minimally invasive surgery," *J. Minimal Access Surg.*, vol. 5, no. 1, pp. 1–7, 2009.

[2] H. Azimian, R. V. Patel, and M. D. Naish, "On constrained manipulation in robotics-assisted minimally invasive surgery," in *Proc. BioRob*, 2010, pp. 650–655.

[3] E. P. W. van der Putten, R. H. M. Goossens, J. J. Jakimowicz, and J. Dankelman, "Haptics in minimally invasive surgery—A review," *Minimally Invasive Therapy Allied Technol.*, vol. 17, no. 1, pp. 3–16, Jan. 2008.

[4] M. K. Chmarra, C. A. Grimbergen, and J. Dankelman, "Systems for tracking minimally invasive surgical instruments," *Minimally Invasive Therapy Allied Technol.*, vol. 16, no. 6, pp. 328–340, 2007.

[5] S. Speidel *et al.*, "Recognition of risk situations based on endoscopic instrument tracking and knowledge based situation modeling," *Proc. SPIE*, vol. 6918, p. 69180X, Mar. 2007.

[6] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin, "Vision-based and marker-less surgical tool detection and tracking: A review of the literature," *Med. Image Anal.*, vol. 35, pp. 633–654, Jan. 2017.

[7] S. Speidel, M. Delles, C. Gutt, and R. Dillmann, "Tracking of instruments in minimally invasive surgery for surgical skill analysis," in *Medical Imaging and Augmented Reality*. Berlin, Germany: Springer, 2006, pp. 148–155.

[8] R. Sznitman *et al.*, "Unified detection and tracking in retinal microsurgery," in *Medical Image Computing and Computer-Assisted Intervention*, vol 6891. Berlin, Germany: Springer, 2011, pp. 1–8.

[9] Z. Pezzementi, S. Voros, and G. D. Hager, "Articulated object tracking by rendering consistent appearance parts," in *Proc. ICRA*, May 2009, pp. 3940–3947.

[10] R. Sznitman, C. Becker, and P. Fua, "Fast part-based classification for instrument detection in minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Cham: Springer, 2014, pp. 692–699.

[11] D. Bouget, R. Benenson, M. Omran, L. Riffaud, B. Schiele, and P. Jannin, "Detecting surgical tools by modelling local appearance and global shape," *IEEE Trans. Med. Imag.*, vol. 34, no. 12, pp. 2603–2617, Dec. 2015.

[12] N. Rieke *et al.*, "Surgical tool tracking and pose estimation in retinal microsurgery," in *Proc. MICCAI*, 2015, pp. 266–273.

[13] I. Laina *et al.*, "Concurrent segmentation and localization for tracking of surgical instruments," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Cham: Springer, 2017, pp. 664–672.

[14] T. Kurmann *et al.*, "Simultaneous recognition and pose estimation of instruments in minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Cham: Springer, 2017, pp. 505–513.

[15] X. Du *et al.*, "Articulated multi-instrument 2D pose estimation using fully convolutional networks," *IEEE Trans. Med. Imag.*, to be published.

[16] G. Rogez and C. Schmid, "MoCap-guided data augmentation for 3D pose estimation in the wild," in *Proc. Annu. Conf. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 3108–3116. [Online]. Available: http://papers.nips.cc/paper/6563-mocap-guided-data-augmentation-for-3d-pose-estimation-in-the-wild

[17] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3D human pose estimation from monocular video," in *Proc. CVPR*, Jun. 2016, pp. 4966–4975.

[18] A. Reiter, P. K. Allen, and T. Zhao, "Articulated surgical tool detection using virtually-rendered templates," in *Proc. IJCARS*, 2012, pp. 1–8.

[19] M. Ye, L. Zhang, S. Giannarou, and G. Z. Yang, "Real-time 3D tracking of articulated tools for robotic surgery," in *Proc. MICCAI*, 2016, pp. 386–394.

[20] M. Allan *et al.*, "Image based surgical instrument pose estimation with multi-class labelling and optical flow," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Cham: Springer, 2015, pp. 331–338.

[21] J. Rosen *et al.*, "Roboscope: A flexible and bendable surgical robot for single portal minimally invasive surgery," in *Proc. ICRA*, May 2017, pp. 2364–2370.

[22] D. R. Berg, P. Y. Li, and A. G. Erdman, "Achieving dexterous manipulation for minimally invasive surgical robots through the use of hydraulics," in *Proc. ASME*, 2012, pp. 429–438.

[23] S. Dambreville, R. Sandhu, A. Yezzi, and A. Tannenbaum, "Robust 3D pose estimation and efficient 2D region-based segmentation from a 3D shape prior," in *Proc. ECCV*, 2008, pp. 169–182.

[24] V. Prisacariu and I. Reid, "PWP3D: Real-time segmentation and tracking of 3D objects," *Int. J. Comput. Vis.*, vol. 98, no. 3, pp. 335–354, 2012.

[25] J. Gall, B. Rosenhahn, and H.-P. Seidel, "Robust pose estimation with 3D textured models," in *Advances in Image and Video Technology*. Berlin, Germany: Springer, 2006, pp. 84–95.

[26] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers, "Combined region and motion-based 3D tracking of rigid and articulated objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 402–415, Mar. 2010.

[27] B. Rosenhahn, T. Brox, and J. Weickert, "Three-dimensional shape knowledge for joint image segmentation and pose estimation," in *Pattern Recognition*. Berlin, Germany: Springer, 2005, pp. 109–116.

[28] T. Cashman and A. Fitzgibbon, "What shape are dolphins? Building 3D morphable models from 2D images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 232–244, Jan. 2013.

[29] C. Schmaltz, B. Rosenhahn, T. Brox, D. Cremers, L. Wietzke, and G. Sommer, "Region-based pose tracking," in *Proc. IbPRIA*, 2007, pp. 56–63.

[30] D. Cremers, "Dynamical statistical shape priors for level set-based tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1262–1273, Aug. 2006.

[31] T. Chan and W. Zhu, "Level set based shape prior segmentation," in *Proc. CVPR*, vol. 2. Jun. 2005, pp. 1164–1170.

[32] V. A. Prisacariu and I. Reid, "Nonlinear shape manifolds as shape priors in level set segmentation and tracking," in *Proc. CVPR*, Jun. 2011, pp. 2185–2192.

[33] M. Allan, S. Ourselin, S. Thompson, D. J. Hawkes, J. Kelly, and D. Stoyanov, "Toward detection and localization of instruments in minimally invasive surgery," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 4, pp. 1050–1058, Apr. 2013.

[34] M. Allan *et al.*, "2D-3D pose tracking of rigid instruments in minimally invasive surgery," in *Information Processing in Computer-Assisted Interventions*, vol 8498. Cham, Switzerland: Springer, 2014, pp. 1–10.

[35] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *Int. J. Comput. Vis.*, vol. 72, no. 2, pp. 195–215, 2007.

[36] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. CVPR*, Jun. 2011, pp. 1297–1304.

[37] P. Kontschieder, S. R. Bulo, H. Bischof, and M. Pelillo, "Structured class-labels in random forests for semantic image labelling," in *Proc. ICCV*, Nov. 2011, pp. 2190–2197.

[38] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. CVPR*, Jun. 2013, pp. 2930–2937.

[39] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Found. Trends Comput. Graph. Vis.*, vol. 7, nos. 2–3, pp. 81–227, Feb. 2012.

[40] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI*, 1981, pp. 674–679.

[41] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. ISMAR*, Nov. 2007, pp. 225–234.

[42] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. Hoboken, NJ, USA: Wiley, 2008.

[43] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *J. Appl. Mech.*, vol. 22, no. 2, pp. 215–221, Jun. 1955.

[44] A. Bartoli and P. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment," *Comput. Vis. Image Understand.*, vol. 100, no. 3, pp. 416–441, 2005.

[45] A. Reiter, P. K. Allen, and T. Zhao, "Appearance learning for 3D tracking of robotic surgical tools," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 342–356, 2013.