Robust, Real-Time, Dense and Deformable 3D Organ Tracking in Laparoscopic Videos

Toby Collins(✉), Adrien Bartoli, Nicolas Bourdel, and Michel Canis

ALCoV-ISIT, UMR 6284 CNRS/Université d'Auvergne, Clermont-Ferrand, France
toby.collins@gmail.com

Abstract. An open problem in computer-assisted surgery is to robustly track soft-tissue 3D organ models in laparoscopic videos in real-time and over long durations. Previous real-time approaches use locally-tracked features such as SIFT or SURF to drive the process, usually with KLT tracking. However this is not robust and breaks down with occlusions, blur, specularities, rapid motion and poor texture. We have developed a fundamentally different framework that can deal with most of the above challenges and in real-time. This works by densely matching tissue texture at the pixel level, without requiring feature detection or matching. It naturally handles texture distortion caused by deformation and/or viewpoint change, does not cause drift, is robust to occlusions from tools and other structures, and handles blurred frames. It also integrates robust boundary contour matching, which provides tracking constraints at the organ's boundaries. We show that it can track over long durations and can handles challenging cases that were previously unsolvable.

1 Introduction and Background

There is much ongoing research to develop and apply Augmented Reality (AR) to improve laparoscopic surgery. One important goal is to visualise hidden subsurface structures such as tumors or major vessels by augmenting optical images from a laparoscope with 3D radiological data from *e.g.* MRI or CT. Solutions are currently being developed to assist various procedures including liver tumor resection such as [6], myomectomy [3] and partial nephrectomy [9]. To solve the problem one must register the data modalities. The general strategy is to build a deformable 3D organ model from the radiological data, then to determine the model's 3D transformation to the laparoscope's coordinate system at any given time. This is very challenging and a general, automatic, robust and real-time solution does not yet exist. The problem is especially hard with monocular laparoscopes because of the lack of depth information. A crucial missing component is a way to robustly compute dense matches between the organ's surface and the laparoscopic images. Currently, real-time results have only been achieved with sparse feature-based matches using KLT [5,10], however this is quite fragile, suffers from drift, and can quickly break down for a number of reasons including occlusions, sudden camera motion, motion blur and optical blur.

S. Ourselin et al. (Eds.): MICCAI 2016, Part I, LNCS 9900, pp. 404–412, 2016.
DOI: 10.1007/978-3-319-46720-7_47

To reliably solve the problem a much more advanced, integrated framework is required, which is the focus of this paper. Our framework is fundamentally a *template-driven* approach which works by matching each image directly to a deformable 3D template, which in our case is a textured 3D biomechanical model of the organ. The model's intrinsic, physical constraints are fully integrated which allows a high level of robustness. This differs from registration using KLT tracks, where tracks are made by independently tracking points frame-to-frame without being constrained by the model. This causes a lack of robustness and drift, where over time the tracked points no longer corresponds to the same physical point. We propose to solve this by densely and robustly matching the organ's texture at the pixel level, which is designed to overcome several fundamental limitations of feature-base matching. Specifically, feature-based matches exist only at sparse. discriminative, repeatable feature points (or *interest points*), and for tissues with weak and/or repetitive texture it can be difficult to detect and match enough features to recover the deformation. This is especially true with blurred frames, lens smears, significant illumination changes, and distortions caused by deformations or viewpoint change. By contrast we match the organ's texture densely, without requiring any feature detection or feature matching, and in a way that naturally handles texture distortions and illumination change.

2 Methodology

We now present the framework, which we refer to as Robust, Real-time, Dense and Deformable (R2D2) tracking. Figure 1 gives an overview of R2D2 tracking using an *in-vivo* porcine kidney experiment as an example.

Model Requirements. We require three main models. The first is a *geometric model* of the organ's outer surface, which we assume is represented by a closed surface mesh $\mathcal{S}$. We denote its interior by $\Omega \subset \mathbb{R}^3$. The second is a *deformation model*, which has a *transform function* $f(\mathbf{p}; \mathbf{x}_t) : \Omega \to \mathbb{R}^3$ that transforms a 3D

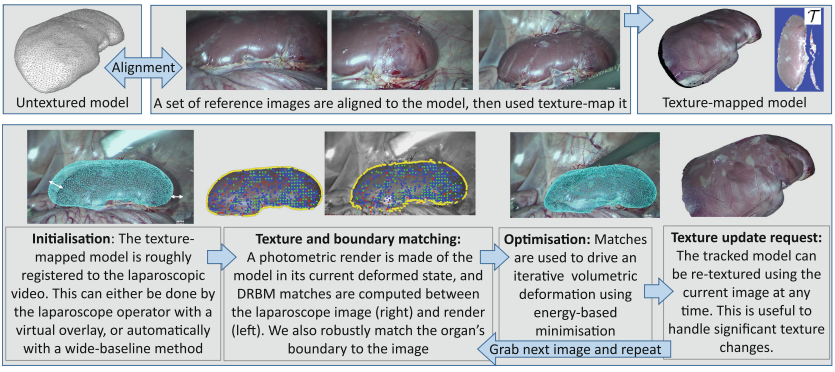

Fig. 1. Overview of R2D2 tracking with monocular laparoscopes. Top row: modelling the organ's texture by texture-mapping it from a set of reference laparoscopic images. Bottom row: real-time tracking of the textured model.

point $\mathbf{p} \in \Omega$ to the laparoscope's coordinates frame at time t. The vector $\mathbf{x}_t$ denotes the model's parameters at time t, and our task is to recover it. We also require the deformation model to have an *internal energy function*, which gives the associated energy for transforming the organ according to $\mathbf{x}_t$. We use E_t to regularise the tracking problem. In the presented experiments the deformation models used are tetrahedral finite element models, generated by a regular 3D vertex grid cropped to the organ's surface mesh (sometimes called a cage), and we compute f with trilinear interpolation. Thus $\mathbf{x}_t$ denotes the unknown 3D positions of the grid vertices in the laparoscope's coordinate frame. For $E_{internal}$ we have used the isotropic Saint Venant-Kirchoff (StVK) strain energy, which has been shown to work well for reconstructing deformations from 2D images [5]. Further modelling deails are given in the experimental section. The third model that we require is a *texture model*, which models the photometric appearance of $\mathcal{S}$. Unlike feature-based tracking, where the texture model is essentially a collection of 2D features, we will be densely tracking its texture, and so we require a dense texture model. We do this with a *texture-map*, which is a common model used in computer graphics. Specifically, our texture-map is a 2D colour image $\mathcal{T}(u, v) : \mathbb{R}^2 \rightarrow [0, 255]^3$ which models the surface appearance up to changes of illumination.

Texture-Map Construction. Before tracking begins we construct $\mathcal{T}$ through a process known as *image-based texture-mapping*. This requires taking laparoscopic images of the organ from several viewpoints (we call these *reference images*). The reference images are then used to generate $\mathcal{T}$ through an image mosaicing process. To do this we must align $\mathcal{S}$ to the reference images. Once done $\mathcal{T}$ can be constructed automatically using an existing method (we currently use Agisoft's Photoscan's method [1], using a default mosaic resolution of 4096×4096 pixels). The difficult part is computing the alignments. Note that this is done just once so it does not need to be real-time. We do this using an existing semi-automatic approach based on [3], which assumes the organ does not deform when the reference images are taken. This requires a minimum of two reference images, however more can be used to build a more complete texture model (in our experiments we use between 4 and 8 reference images), taking approximately three minutes to compute with non-optimised code.

Tracking Overview. Our solution builds on a new technique called *Deformable Render-based Block Matching* (DRBM) [2], which was originally proposed to track thin-shell objects such as cloth and plastic bottles, yet has great potential for our problem. It works by densely matching each image $\mathcal{I}_t$ to a time-varying 2D photometric render $\mathcal{R}_t$ of the deforming object. The render is generated from the camera's viewpoint and is continuously updated to reflect the current deformation. Matching is performed by dividing $\mathcal{R}_t$ into local pixel windows, then each window is matched to $\mathcal{I}_t$ with an illumination-invariant score function and a fast coarse-to-fine search process. At a final stage most incorrect matches, caused by *e.g.* occlusions or specularities are detected and eliminated using several consistency tests. The remaining matches are used as deformation constraints, which

are combined with the model's internal energy, then $\mathbf{x}_t$ is solved with energy minimisation. Once completed the new solution is used to update the render, the next image is acquired and the process repeats. Because this process tracks the model frame-to-frame a mechanism is needed for initialisation (to provide an initial extimate of $\mathbf{x}_t$ at the start) and re-initialisation (to provide and initial estimate if tracking fails). We discuss these mechanisms below.

We use DRBM as a basis and extend it to our problem. Firstly, DRBM requires at least some texture variation to be present, however tissue can be quite textureless in some regions. To deal with this additional constraints are needed. One that has rarely been exploited before are organ boundary constraints. Specifically, if the organ's boundary is visible (either partially or fully) it can be used as a tracking constraint. Organ boundaries have been used previously to semi-automatically register pre-operative models [3], but not for automatic real-time tracking. This is non-trivial because one does not know which points correspond to the organ's boundary *a priori*. Secondly, we extend it to volumetric biomechanical deformable models, and thirdly we introduce semi-automatic texture map updating, which allows strong changes of the organ's appearance to be handled, due to *e.g.* coagulation.

Overview and Energy-Based Formulation. To ease readability we now drop the time index. During tracking texture matches are found using DRBM, which outputs a quasi-dense set of texture matches $C_{texture} \stackrel{\text{def}}{=} \{(\mathbf{p}_1, \mathbf{q}_1), \ldots, (\mathbf{p}_N, \mathbf{q}_N)\}$ between 3D points $\mathbf{p}_i \in \mathbb{R}^3$ on the surface mesh $\mathcal{S}$ and points $\mathbf{q}_i \in \mathbb{R}^2$ in the image. We also compute a dense set of boundary matches $C_{bound} \stackrel{\text{def}}{=} \{(\tilde{\mathbf{p}}_1, \tilde{\mathbf{q}}_1), \ldots, (\tilde{\mathbf{p}}_M, \tilde{\mathbf{q}}_M)\}$ along the model's boundary, as described below. Note that this set can be empty if none of its boundaries are visible. The boundary matches work in an Iterative Closest Point (ICP) sense, where over time the boundary correspondences slide over the surface as it deforms.

Our energy function $E(\mathbf{x}) \in \mathbb{R}^+$ encodes tracking cues from the image ($C_{texture}$, C_{bound}) and the model's internal deformation energy, and has the following form:

$$E(\mathbf{x}) = E_{match}(\mathbf{x}; C_{texture}) + \lambda_{bound} E_{match}(\mathbf{x}; C_{bound}) + \lambda_{internal} E_{internal}(\mathbf{x}) \tag{1}$$

The term E_{match} is a *point-match energy*, which generates the energy for both texture and boundary matches. This is defined as follows:

$$E_{match}(\mathbf{x}; C) \stackrel{\text{def}}{=} \sum_{(\mathbf{p}_i, \mathbf{q}_i) \in C} \rho\left(\|\pi(f(\mathbf{p}_i; \mathbf{x})) - \mathbf{q}_i\|_2\right) \tag{2}$$

where $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the camera's projection function. We assume the laparoscope is intrinsically calibrated, which means π is known. The function $\rho : \mathbb{R} \rightarrow \mathbb{R}^+$ is an *M-estimator* and is crucial to achieve robust tracking. It ts purpose is to align the model point $\mathbf{p}_i$ with the image point $\mathbf{q}_i$, but to do so robustly to account for erroneous matches, which are practically unavoidable.

When a match is erroneous the model should not align the match, and the M-estimator provides this by reducing the influence of an erroneous match on E. We have tested various M-estimators and found good results are obtained with pseudo-L1 $\rho(x) \stackrel{\text{def}}{=} \sqrt{x^2 + \epsilon}$ with $\epsilon = 10^{-3}$ being a small constant to make E_{match} differentiable everywhere.

The terms λ_{bound} and $\lambda_{internal}$ are influence weights, and discuss how they have been set in the experimental section. We follow the same procedure to minimise E as described in [2]. This is done by linearising E about the current estimate (which is the solution from the previous frame), then we form the associated linear system and solve its normal equations using a coarse-to-fine multi-grid Gauss-Newton optimisation with backtracking line-search.

Computing Boundary Matches. We illustrate this process in Fig. 2(k). First we take $\mathcal{R}$ and extract all pixels $\mathcal{P}$ on the render's boundary. For each pixel $\mathbf{p}_i \in \mathcal{P}$ we denote its 3D position on the model by $\tilde{\mathbf{p}}_i$, which is determined from the render's depthmap. We then conduct a 1D search in $\mathcal{I}$ for a putative match $\tilde{\mathbf{q}}_i$. The search is centred at $\mathbf{p}_i$ in the direction orthogonal to the render's boundary, which we denote by the unit vector $\mathbf{v}_i$. We search within a range $[-l, +l]$ in one pixel increments where l is a free parameter, and measure the likelihood $b(\mathbf{p}) \in \mathbb{R}$ that a sample $\mathbf{p}$ corresponds to the organ's boundary. We currently compute b with a hand-crafted detector, based on the fact that organ boundaries tend to occur at low-frequency intensity gradients, which correspond to a change of predominant tissue albedo. We give the precise algorithm for computing b in the supplementary material. We take $\tilde{\mathbf{q}}_i$ as the sample with the maximal b beyond a detection threshold b_τ. If no such sample exists then we do not have a boundary match. An important stage is then to eliminate false positives because there may be other nearby boundary structures that could cause confusion. For this we adopt a conservative strategy and reject the match if there exists another local minimum of b along the search line that also exceeds b_τ.

Initialisation, Re-localisation and Texture Model Updating. There are various approaches one can use for initialisation and re-localisation. One is with an automatic wide-baseline pose estimation method such as [7]. An alternative is to have the laparoscope operator provide them, by roughly aligning the live video with a overlaid render of the organ from some canonical viewpoint (Figs. 1 and 2(a)), and then tracking is activated. The alignment does not need to be particularly precise due to the robustness of our match terms, which makes it a practical option. For the default viewpoint we use the model's pose in one of the reference images from the texture-map construction stage. The exact choice is not too important so we simply use the one where the model centroid is closest to the image centre. During tracking, we have the option to update the texture model by re-texturing its front-facing surface regions with the current image. This is useful where the texture changes substantially during surgery. Currently this is semi-automatic to ensure the organ is not being occluded by tools or other organs in the current image, and is activated by a user notification. In future work aim to make this automatic, but this is non-trivial.

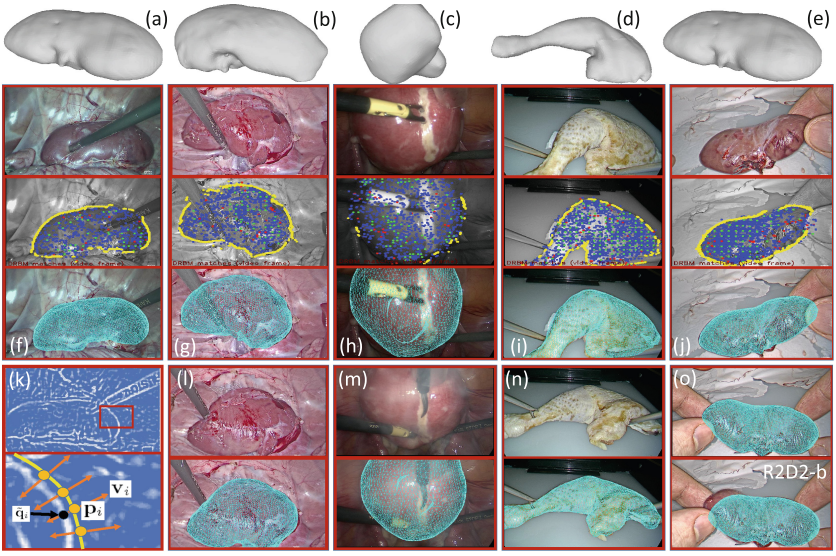

Fig. 2. Visualisations of the five test cases and tracking results. Best viewed in colour.

3 Experimental Results

We evaluate performance with five test cases which are visualised in Fig. 2 as five columns. These are two *in-vivo* porcine kidneys (a,b), an *in-vivo* human uterus (c), an *ex-vivo* chicken thigh used for laparoscopy training (d) and an *ex-vivo* porcine kidney (e). We used the same kidney in cases (a) and (e). The models were constructed from CT (a,b,d,e) and T2 weighted MRI (c), and segmented interactively with MITK. For each case we recorded a monocular laparoscopic video (10 mm Karl Storz 1080p, 25fps with CLARA image enhancement) of the object being moved and deformed with surgical tools (a,b,c,d) or with human hands (e). The video durations ranged from 1424 to 2166 frames (57 to 82 s). The objects never moved completely out-of-frame in the videos, so we used them to test tracking performance without re-localisation. The main challenges present are low light and high noise (c), strong motion blur (b,c), significant texture change caused by intervention (a,c), tool occlusions (a,b,c,d), specularities (a,b,c,d,e), dehydration (b), smoke (c), and partial occlusion where the organ disappears behind the peritoneum (b,c). We constructed deformable models with a 6 mm grid spacing with the number of respective tetrahedral elements for (a–e) being 1591, 1757, 8618, 10028 and 1591. Homogeneous StVK elements were used for (a,b,c,e) using rough generic Poison's ratio ν values from the literature. These were $\nu = 0.43$ for (a,b,e) [4] and $\nu = 0.45$ for (c). Note that when we use homogeneous elements, the Young's modulus E is not actually a useful parameter for us. This because if we double E and halve $\lambda_{internal}$ we end up with the same internal energy. We therefore arbitrarily set $E = 1$ for (a,b,c,e). For (d) we

used two coarse element classes corresponding to bone and all other tissue, and we set their Young's moduli using relative values of 200 and 1 respectively.

Our tracking framework has several tunable parameters, which are *(i)* the energy weights, *(ii)* the boundary search length l, *(iii)* the boundary detector parameters and *(iv)* the DRBM parameters. To make them independent of the image resolution, we pre-scale the images to a canonical width of 640 pixels. For all five cases we used the same values of *(iii)* and *(iv)* (their respective defaults), and the same value for *(iii)* of $l = 15$ pixels. For *(i)*, we used the same value of $\lambda_{bound} = 0.7$ in all cases. For $\lambda_{internal}$ we used category-specific values, which were $\lambda_{internal} = 0.2$ for the uterus, $\lambda_{internal} = 0.09$ for kidneys and $\lambda_{internal} = 0.2$ for the chicken thigh. In the interest of space, the results presented here do not use texture model updating. This is to evaluate tracking robustness despite significant appearance change. We refer the reader to the associated videos to see texture model updating in action. We benchmarked processing speed on a mid-range Intel i7-5960X desktop PC with a single NVidia GTX 980Ti GPU. With our current multi-threaded C++/CUDA implementation the average processing speeds were 35, 27, 22, 17 and 31fps for cases (a-e) respectively. We also ran our framework without the boundary constraints ($\lambda_{bound} = 0$). This was to analyse its influence on tracking accuracy, and we call this version R2D2-b. We show snapshot results from the videos in Fig. 2. In Fig. 2(f–j) we show five columns corresponding to each case. The top image is an example input image, the middle image shows DRBM matches (with coarse-scale matches in green, fine-scale matches in blue, gross outliers in red) and the boundary matches in yellow. The third image shows an overlay of the tracked surface mesh. We show three other images with corresponding overlays in Fig. 2(l–n). The light path on the uterus in Fig. 2(h) is a coagulation path used for interventional incision planning, and it significantly changed the appearance. The haze in Fig. 2(m) is a smoke plume. In Fig. 2(o) we show the overlay with and without boundary constraints (top and bottom respectively). This is an example where the boundary constraints have clearly improved tracking.

We tested how well KLT-based tracking worked by measuring how long it could sustain tracks from the first video frames. Due to the challenges of the conditions, KLT tracks dropped off quickly in most cases. mostly due to blur or tool occlusions. Only in case (b) did some KLT tracks persist to the end, however they were limited to a small surface region which congregated around specularities (and therefore were drifting). By contrast our framework sustained tracking through all videos. It is difficult to quantitatively evaluate tracking accuracy in 3D without interventional radiological images, which were not available. We therefore measured accuracy using 2D proxies. These were *(i)* Correspondence Prediction Error (CPE) and *(ii)* Boundary Prediction Error (BPE). CPE tells us how well the tracker aligns the model with respect to a set of manually located point correspondences. We found approximately 20 per case, and located them in 30 representative video frames. We then measured the distance (in pixels) to their tracked positions. BPE tells us how well the tracker aligns the model's boundaries to the image. This was done by manually marking any contours in

Table 1. Summary statistics of the quantitative performance evaluation (in pixels). Errors are computed using a default image width of 640 pixels.

	(a) In-vivo kidney 1			(b) In-vivo kidney 2			(c) In-vivo uterus			(d) Chicken Thigh			(e) Ex-vivo kidney		
BPE	R2D2	R2D2-d	R-HMA	R2D2	R2D2-d	R-HMA	R2D2	R2D2-d	R-HMA	R2D2	R2D2-d	R-HMA	R2D2	R2D2-d	R-HMA
Med.	1.37	**1.33**	2.67	**1.67**	4.96	3.80	**4.71**	5.19	28.99	**2.69**	3.16	5.19	**1.00**	2.03	4.18
IQR	**1.67**	1.69	6.34	**2.60**	6.44	10.53	6.77	**6.73**	55.13	**2.92**	4.04	8.50	**1.37**	2.43	14.28
Mean	**1.62**	1.63	16.34	**2.69**	6.29	16.60	**6.96**	7.87	45.71	**3.26**	4.34	8.87	**1.20**	3.35	16.18
S.D.	**1.18**	1.37	40.28	**3.30**	5.73	38.62	**8.35**	10.00	43.18	**2.59**	3.98	10.67	**0.97**	4.41	26.78
Max.	9.71	**8.98**	261.74	**30.20**	36.49	264.76	**61.12**	77.67	210.32	**21.44**	33.61	94.58	**8.49**	31.75	159.43
CPE	(a) In-vivo kidney 1			(b) In-vivo kidney 2			(c) In-vivo uterus			(d) Chicken Thigh			(e) Ex-vivo kidney		
Med.	**2.41**	2.51	3.61	**2.03**	2.22	3.01	**4.64**	4.69	14.56	**1.55**	1.59	5.21	**1.14**	2.89	3.73
IQR	**2.23**	2.40	4.27	**2.06**	2.16	4.03	**4.29**	4.42	144.67	**1.46**	1.52	8.63	**1.06**	3.24	16.99
Mean	**2.94**	3.09	11.98	**2.49**	2.52	9.08	7.21	**6.85**	288.83	**1.81**	1.96	9.50	**1.39**	3.48	15.67
S.D.	**2.36**	2.48	27.16	2.26	**1.67**	22.81	8.56	**8.45**	1100.44	**1.22**	1.64	12.84	**1.05**	2.76	31.01
Max.	**16.89**	18.75	169.57	22.27	**9.61**	203.52	42.73	**42.71**	9779.96	**7.82**	16.61	86.29	**6.48**	21.44	373.69

the representative images that corresponded to the object's boundary. We then measured the distance (in pixels) between each contour point and the model's boundary. The results are shown in Table 1, where we give summary statistics (median, inter-quartile range, median, standard deviation and maximum). The table also includes results from R2D2-b. To show the benefits of tracking with a deformable model, we also compare with a fast feature-based baseline method using a rigid transform model. For this we used SIFT matching with HMA outlier detection [8] (using the author's implementation) and rigid pose estimation using OpenCV's PnP implementation. We denote this by R-HMA. Its performance is certainly worse, which is because it cannot model deformation, and also because HMA was sometimes unable to find any correct feature clusters, most notably in (c) due to poor texture, blur and appearance changes.

4 Conclusion

We have presented a new, integrated, robust and real-time solution for dense tracking of deformable 3D soft-tissue organ models in laparoscopic videos. There are a number of possible future directions. The main three are to investigate automatic texture map updating, to investigate its performance using stereo laparoscopic images, and to automatically detect when tracking fails.

References

1. Agisoft Photoscan. http://www.agisoft.com. Accessed 30 May 2016
2. Collins, T., Bartoli, A.: Realtime shape-from-template: system and applications. In: ISMAR (2015)
3. Collins, T., Pizarro, D., Bartoli, A., Canis, M., Bourdel, N.: Computer-assisted laparoscopic myomectomy by augmenting the uterus with pre-operative MRI data. In: ISMAR (2014)

4. Egorov, V., Tsyuryupa, S., Kanilo, S., Kogit, M., Sarvazyan, A.: Soft tissue elastometer. Med. Eng. Phys. **30**(2), 206–212 (2008)
5. Haouchine, N., Dequidt, J., Berger, M., Cotin, S.: Monocular 3D reconstruction and augmentation of elastic surfaces with self-occlusion handling. IEEE Trans. Vis. Comput. Graph. **21**(12), 1363–1376 (2015)
6. Haouchine, N., Dequidt, J., Peterlik, I., Kerrien, E., Berger, M.-O., Cotin, S.: Image-guided simulation of heterogeneous tissue deformation for augmented reality during hepatic surgery. In: ISMAR (2013)
7. Puerto-Souza, G., Cadeddu, J.A., Mariottini, G.: Toward long-term and accurate augmented-reality for monocular endoscopic videos. Bio. Eng. **61**(10), 2609–2620 (2014)
8. Puerto-Souza, G., Mariottini, G.: A fast and accurate feature-matching algorithm for minimally-invasive endoscopic images. TMI **32**(7), 1201–1214 (2013)
9. Su, L.-M., Vagvolgyi, B.P., Agarwal, R., Reiley, C.E., Taylor, R.H., Hager, G.D.: Augmented reality during robot-assisted laparoscopic partial nephrectomy: toward real-time 3D-CT to stereoscopic video registration. Urology **73**, 896–900 (2009)
10. Tomasi, C., Kanade, T.: Detection and tracking of point features. Technical report CMU-CS-91-132 (1991)