Video to CT Registration for Image Overlay on Solid Organs

Balazs Vagvolgyi, Li-Ming Su, Russell Taylor, and Gregory D. Hager

Center for Computer-Integrated Surgical Systems and Technology
Johns Hopkins University
Baltimore, MD 21218
{vagvoba, hager}@cs.jhu.edu

Abstract. This paper describes a general-purpose system for computing registered stereoscopic video overlays of pre-operative imagery during minimally invasive surgery. There are three key elements to our approach. The first element is a real-time computer vision system that operates on stereoscopic video acquired during minimally invasive surgery to extract geometric information. We present two variations on this system: a dense stereo algorithm and a sparse point-based method. The second element is an efficient deformable surface-to-surface ICP registration. The final element is a novel display system that has been customized to operate well with stereo vision. By combining these elements, we show that we are able to perform video to volume registration and display in real time. This in turn facilitates rendering of annotations and visualization of sub-surface information on structures within the surgical field. Experimental results are shown on video sequences recorded during animal and human surgeries.

1 Introduction

Minimally invasive surgery (MIS) is a technique whereby instruments are inserted into the body via small incisions (or in some cases natural orifices), and surgery is carried out under video guidance. While advantageous to the patient, MIS presents numerous challenges for the surgeon due to the restricted field of view presented by the endoscope, the tool motion constraints imposed by the insertion point, and the loss of haptic feedback.

One means of overcoming some of these limitations is to present the surgeon with addition visual information. This paper describes a system that provides the surgeon with a three-dimensional information overlay registered to pre-operative or intra-operative volumetric data. The novelty of the system lies in its use of real-time stereo video data, online deformable registration, and rendering without recourse to an external tracking system. We have implemented a version of the system for augmenting the surgical view during laparoscopic kidney procedures.

Previous work on image overlay using stereo has largely focused on rigid structures and non-real-time visualization. In particular, [1] presents a system that makes use of stereo vision to perform image overlay of MRI images of the head. More recently, [2] briefly describes an attempt to use stereo area-matching on da Vinci images, but they largely conclude that area-matching does not work well on these images. Kanbara et al.

[3] demonstrated a video overlay using traditional image navigation techniques (which rely on an external tracking system) and rigid anatomy. Stoyanov et al. [4] described a traditional single-frame region matching stereo system and validated it against CT, and in [5] a real-time motion estimation system for discrete points was presented.

2 Methods

Briefly, our implemented system provides three general functions: 1) extraction of 3D information from stereo video data; 2) registration of video data to preoperative images; and 3) rendering and information display. We have implemented two methods for computing depth information and performing registration: a dense stereo matching algorithm, and a local point-based tracking algorithm. In all that follows, we assume that the endoscope has been calibrated to determine the corresponding 2D projection parameters [6].

2.1 Video to CT Registration

Extracting Surfaces From Stereo Most real-time stereo algorithms make use of area-matching techniques [8]. However, given the challenges of endoscopic video imagery, these techniques are often ineffective due to image noise, specular highlights and lack of texture. In recent years, global optimization methods have been developed to improve the accuracy of stereo. Original work focused on scanline dynamic programming [9]; hierarchical methods [10], multiple smoothness constraints [11], and graph cuts [12]. Our approach was to develop a highly optimized dynamic programming stereo algorithm which provides a desirable tradeoff between reliability and speed.

Fig. 1. Upper left: Standard image from stereo literature [7]; Upper right: Disparity map from upper left image (DP stereo, with sub-pixel disparity estimation and left-right check); Lower left: Image of a kidney surface; Lower right: Depth map of kidney image (DP stereo).

In the following discussion, we assume fully rectified color stereo images $L(u,v)$ (the left image) and $R(u,v)$ (the right image). In the top-left cornered image coordinate systems, for any pixel location (u,v) in the left camera, we define the disparity value for a corresponding point (u,v') in the right image as $D(u,v) = v - v'$. Given a known camera calibration, it is well known how to convert a dense disparity representation into a set of 3D points [8].

Our dynamic programming stereo algorithm optimizes the following objective function:

$$C_{tr}(u,v,d,d') = \frac{1}{2}\left(C(u-1,v,d') + C(u,v-1,d')\right) + r(d,d') + e(u,v,d) \qquad (1)$$

where d and d' are disparity values of the current pixel and the upper left neighbor, r is the cost of change in disparity between neighbors, e is an image match cost of two color pixels, and C is the cost of disparities which is initialized to 0 outside the image boundaries. The image matching function is the sum of absolute differences (SAD) of a pair of color pixels. Although the absolute difference between the two pixels proved to be adequate for most cases we observed that in certain challenging lighting conditions integrating the difference over a small region of pixels may improve matching performance. We impose a cost limit of 25 gray values which has been experimentally proven to improve matching performance [7]. The smoothness term is a linear function of the disparity values. We note that (1) is an approximation in the following sense. At a given location (u,v), it is possible that the left and upper neighbors could have differing disparities. Thus, in principle the regularization function should include separate terms for both neighbors, and the minimization in (1) should operate on two independent disparity values. However, this leads to a quadratic (in disparity) complexity. In practice, the depth resolution of stereo is far less than the lateral (pixel) resolution. As a result, in most cases there are large patches of consistent disparity. Thus, the approximation of constant local disparity is quite good, and is well worth the computational savings.

Once (1) is computed for the entire image, the disparity map satisfying both the horizontal and vertical smoothness criteria can be read out recursively from the memoization buffer M as

$$\begin{aligned} D(u_{max}, v_{max}) &= \min_d M(u,v,d) \\ D(u,v) &= \frac{1}{2}\Big[M\big(u+1,v,D(u+1,v)\big) + M\big(u,v+1,D(u,v+1)\big)\Big] \end{aligned} \tag{2}$$

In order to improve performance, we make two additional modifications. First, we choose a reduced scale (typically a factor of two to four) to perform computations. Each factor of two reduction improves performance by a factor of eight. Second, rather than searching over the complete disparity range $\mathscr{D}_s$ in every image, we only search over a small bracket of disparities about the previous stereo pair in the image sequence. With camera motion there are areas of the image (primarily at discontinuities) that occasionally violate this assumption of small change. In practice these areas converge to the correct answer within a small number of frames. As an optional feature, the algorithm is capable of computing sub-pixel disparity estimates by fitting a parabola on the costs associated to the neighbors of the winning discrete disparity. The location of the apex of resulting parabola determines the estimate of the sub-pixel disparity value.

To reduce the effects of illumination, the video data was preprocessed by first globally adjusting the brightness and color values of the left video channel to the values measured on the right channel, and then applying a a Laplacian high-boost filter in advance to increase the fine detail contrast.

Dense Registration Methods Given a 3D point cloud from stereo, a CT surface segmentation, and a good starting point (typically available based on prior knowledge of the procedure), our first goal is to compute a rigid registration (R_t, T_t) of images taken at time t to a and preoperative surface given a previous estimate (R_{t-1}, T_{t-1}).

For this purpose, we use a modified version of the classical ICP algorithm [13] applied to the depth map computed from the stereo endoscopic video stream as one

point cloud (P_{stereo}) and the 3D model of the anatomy placed in the FOV as the other point cloud (P_{model}). While P_{stereo} is a surface mesh that contains only the visible 3D details of the anatomy, P_{model} contains all the visible and occluded anatomical details. We assume that P_{stereo} is a small subset of the surface points of P_{model}, thus before finding the point correspondence the algorithm renders the z-buffer of the pre-operative model P_{model} using R_{t-1}, T_{t-1} and extracts those points that are visible by the virtual camera ($P_{modelsurface}$). The resulting $P_{modelsurface}$ point cloud is a surface mesh similar to P_{stereo}, thus finding the point correspondence with P_{stereo} is now possible. The implemented method for finding the point matches is accelerated by using a $k-d$ tree [14]. For finding the rigid transformation we used the closed-form solution technique employing SVD [15].

After finding the estimated rigid transformation using ICP, a deformable surface registration is computed. For these purposes, a set of points are defined below the surface in the CT volume, and a spring-mass system is defined as reported in [16]. For efficiency, the current implementation computes just the forces between the reconstructed surface and the CT surface. Given the point correspondence computed by the rigid transformation, we can easily compute the strain ($F(v)$) between the corresponding surface points as

$$F(v) = \gamma(P_{stereo}(CP(v)) - P_{modelsurface}(v)) \tag{3}$$

where the parameter $\gamma \in [0,1]$ determines the strength of deformation. In our results we used $\gamma = 1/3$.

In an ideal case, the strain vectors could be applied to deform the model directly. However ICP is a rigid registration algorithm thus the point correspondence between the model and the deformed surface will always be somewhat incorrect. In order to overcome this difficulty the algorithm filters the strain field (F_{filt}) before applying deformation. The filtering is done with a Gaussian kernel on the neighboring strain vectors. The neighborhood is defined in 2D on the visible surface mesh of the model. Finally the force field is applied on the model surface to yield the deformed surface ($P_{defsurface}$):

$$P_{defsurface}(v,t) = (1-\lambda)(P_{modelsurface}(v) + F_{filt}(v)) + \lambda P_{defsurface}(v,t-1) \tag{4}$$

where $\lambda \in [0,1]$ allows adjusting the temporal consistency of deformation.

Sparse Registration Methods There are many cases where surface geometry alone is inadequate for a unique, stable registration. For such cases, we have also included a stronger point-based registration method. To use this method, we first assume that the model has been brought into registration with the video, either using a dense registration or by other manual means. Once a registration is known, a set of image feature locations, $p_1, p_2, \ldots p_n$ in one image are chosen. A disparity map is calculated as described above. With this, the corresponding points in the second image are known, and the 3D locations of those points in CT coordinates are given by the registration. Thus, a direct 3D to 3D point registration can be performed using [15]. To maintain the registration, a simple brute-force template tracking algorithm has been implemented to recompute the feature points in each image. In every frame of the video, the new feature locations are used to recompute the reconstructed 3D points, and the model is re-registered.

2.2 Rendering and Display

One of the major challenges is to perform the video processing, registration, and stereoscopic rendering of the 3D overlay in real time. In order to eliminate the redundancies in the displaying and registration pipelines, a special-purpose 3D rendering engine has been developed. The current system does not use any graphics hardware acceleration for 3D rendering in order to make shared memory buffers easily accessible by the registration algorithm. This rendering engine incorporates all of the functionality of a typical graphics pipeline, including a a full geometrical transformation engine with Z-buffering, several lighting models, and various forms of transparent display. The graphics pipeline supports fast stereo rendering with no redundancy in the lighting and transformation phases, and shared texture and model memories. By sharing the same memory layout for representing the 3D geometry and having direct access to the intermediate processing steps, we can easily extract the list of visible triangles and the Z-buffer from the 3D rendering pipeline and reuse them during the dense 3D to 3D registration. The gains in memory efficiency and computational complexity are significant. The final system can render 5 million stereo triangles per second with Texture + Lighting + Transparency on a Dual Pentium 4 3.2 GHz.

During development, we asked the help of a Urologic surgeon to design the visual appearance of the 3D models so that they are visible but not obtrusive. Moreover the surgeon helped us to build other 3D models that provide additional intra-operative visual guidance for dissecting the tumor. In particular, we also display the kidney collecting system to help the surgeon understand the underlying anatomy relative to the video view. Figure 4 shows the the final display used for partial nephrectomy.

3 Results

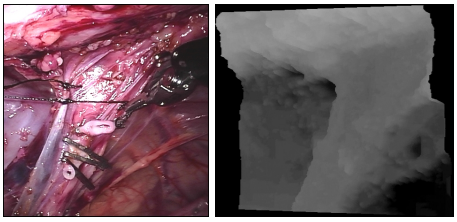

Fig. 2. Left: input image for stereo. Right: 3D mesh rendered with depth shading.

Here, we present details of the performance of the previously described stereo and registration methods. We have also included supplementary material demonstrating the system operating on representative video sequences. In terms of performance, the speed of the stereo and display engines is more than adequate for the models used in our experiments. In the case of the former, we can compute dense stereo to 1/4 pixel resolution on 240x320 images (one half VGA resolution) at over 10 frames/second. With respect to the latter, we typically have no more than 30,000 triangles on our 3D scenes which can be rendered in stereo by the engine over 100 frames per second. Both of these operate in parallel on separate CPU cores. Feature tracking is about as fast as dense stereo (10 frames/second) because large template size is required for the high accuracy feature tracking on endoscopic video data. All performance numbers include the required pre-processing computations.

3.1 Stereo and Registration on In-Vivo Data Animal Data

Dynamic Programming Stereo The dynamic programming method demonstrates very stable 3D reconstruction on an intra-operative sequence (Figure 2, left). The high depth

resolution and the fine details demonstrate that the algorithm had no difficulties dealing with the discontinuities of the anatomical surface (Figure 2 right). The only cases where significant discontinuities may occur are the areas of the surgical tools in the field of view.

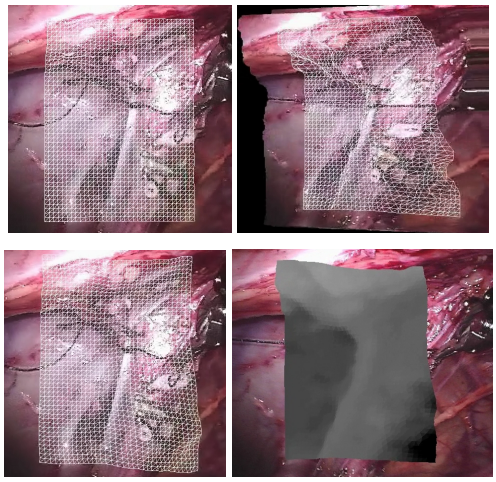

Fig. 3. Top row, rigid registration of the anatomical surface model. Bottom row, deformable registration of the anatomical surface model. On the left, the deformed wire-frame model and on the right, the deformed surface model rendered with depth shading.

Registration Results Since we did not have pre-operative 3D model of the anatomy corresponding to this surgery recording, we selected a video frame where the anatomy was not covered by any surgical tools and created a 3D surface model from the reconstructed surface mesh. We then used this model for rigid and deformable registration. As expected, the rigid registration gave perfect match for the video frame from which the model was created. For the rest of the video, frames the rigid registration provided a good approximation of the motion of the corresponding anatomical feature. ICP was configured to stop at 2 mm accuracy and process no more than 5 iterations. Enabling deformable registration improved the surface matching significantly. The deformed surface behaves like a latex surface: stretching, shrinking and sticking to the reconstructed surface (see Figure 3). For rigid registration the average error measured by ICP was below 2 mm per vertex in the video segment where the surgical tool was out of the work area (successful registration in 1 iteration). The deformable registration reduced the average registration error below 0.5 mm for most of the same video segment.

Evaluation on In-Vivo Patient Data We have applied our methods to two different interventions. Both were post-process after the surgery itself. In the first case, video data was recorded during a laparoscopic partial nephrectomy carried out using a surgical grade stereoscopic endoscope (Scholly America, West Boylston, MA). A segment of the video was chosen where the kidney surface had been exposed prior to surgical excision of the tumor. The corresponding CT image for this patient was segmented manually by a surgeon producing 3D models for the kidney surface, the tumor, and the collecting system in VTK file format.

Figure 4 shows the final display used for partial nephrectomy. The ring model that represents the cutting margins on the kidney surface around the tumor as well as the colors and the transparency levels were verified by the surgeon who performed the

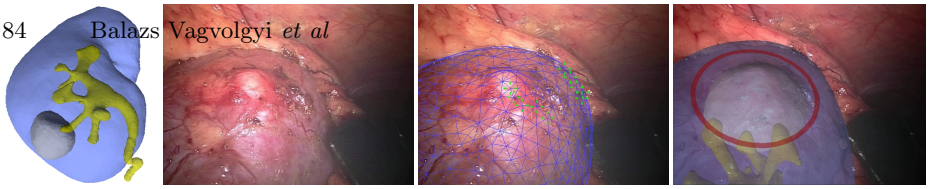

Fig. 4. Laparoscopic partial nephrectomy of a tumor (sequence 1, left to right): segmented CT model; source image (left channel); after manual registration and feature point selection; automatic registration and augmented reality overlay of the safety margin of dissection (red ring).

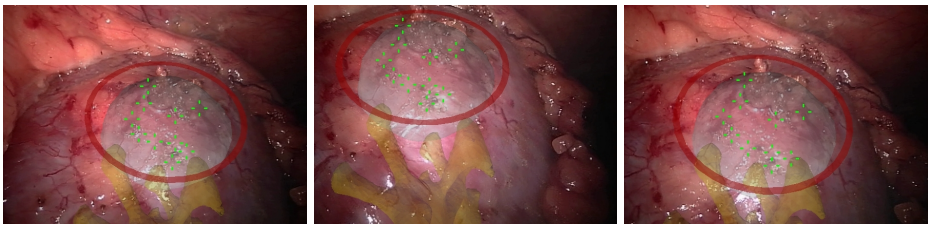

Fig. 5. Laparoscopic partial nephrectomy of a tumor (Sequence 2): automatic registration with augmented reality overlay of the safety margin of dissection (frames 154, 252, and 430).

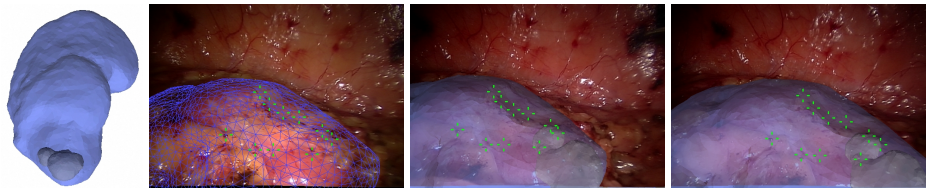

Fig. 6. Robot-assisted laparoscopic partial nephrectomy of large lower pole kidney stones (from left to right): segmented CT model; after manual registration and feature point selection; automatic registration on (frame 109); automatic registration on (frame 407).

surgery. We experimented with automated full-surface registration and manual registration followed by feature point selection and tracking. Due to the limited amount of kidney surface appearing in the video, we found that manual registration followed by "pinning" with surface feature points had superior stability as well as providing better overall performance. Figure 5 show several examples from a second sequence taken from the same case.

Our second clinical case was the surgical removal of a large kidney stone. The data was again recorded with a surgical grade stereoscopic endoscope, this time in the context of a robotic surgery carried out with the da Vinci system (Intuitive Surgical, Sunnyvale, CA). In this case, we processed approximately 1 minute of video. The CT segmentation employed did not contain the collecting system, but did contain both the stone and the kidney surface. This segmentation was also performed manually. Figure 6 shows the resulting display at three points through the video (at the video frame which provided the baseline for manual registration, and two other video frames). The associated material for this paper contains the entire video sequence. As before, we tested both the pure surface-based registration and the registration using feature points, and found the latter to be much more stable.

4 Conclusion

We have presented a system for performing real-time deformable registration and display on solid organ surfaces observed with a stereo video endoscope. The system produces good results even under the challenging conditions found in intra-operative video. We have presented results of the stereo processing and registration system on real video data, and we have evaluated the displays on two human cases.

Although a promising start, it is clear that there are several immediate avenues for further improvement of the system. First, we intend to combine the surface and local feature tracking registration, and to automate the selection of points for the latter. Second, we are working to parallelize the algorithm to improve the speed of both stereo processing and registration. Finally, we are planning to perform a formal system validation in an animal model within the next few months.

Acknowledgements This material is based upon work supported by TATRC under grant W81XWH-06-1-0195 and the National Science Foundation under Grant No. EEC-9731478. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would like to thank Carol Reiley for the video recordings.

References

[1] Betting, F., Feldmar, J., Ayache, N., Devernay, F.: A new framework for fusing stereo images with volumetric medical images. In: CVRMed. (1995) 30–39
[2] Mourgues, F., Devernay, F., Coste-Maniere, E.: 3d reconstruction of the operating field for image overlay in 3d-endoscopic surgery. isar **00** (2001) 191
[3] M. Kanbara et al.: A stereoscopic video see-through augmented reality system based on real-time vision-based registration. In: Proc. Virt. Reality. (2000) 255–262
[4] Stoyanov, D., Darzi, A., Yang, G.Z.: Dense 3d depth recovery for soft tissue deformation during robotically assisted laparoscopic surgery. In: MICCAI (2). (2004) 41–48
[5] Stoyanov, D., Mylonas, G.P., Deligianni, F., Darzi, A., Yang, G.Z.: Soft-tissue motion tracking and structure estimation for robotic assisted mis procedures. In: MICCAI (2). (2005) 139–146
[6] Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(11) (2000) 1330–1334
[7] Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision **47**(1) (2002) 7–42
[8] Brown, M.Z., Burschka, D., Hager, G.D.: Advances in Computational Stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(8) (2003) 993–1008
[9] Cox, I.J., Hingorani, S.L., Rao, S., Maggs, B.M.: A maximum likelihood stereo algorithm. Computer Vision and Image Understanding **63**(3) (1996) 542–567
[10] van Meerbergen, G., Vergauwen, M., Pollefeys, M., Van Gool, L.: A hierarchical symmetric stereo algorithm using dynamic programming. IJCV **47**(1-3) (2002) 275–285
[11] Hirschmüller, H.: Stereo vision in structured environments by consistent semi-global matching. In: Proc. CVPR. (2006) 2386–2393
[12] Kolmogorov, V., Zabih, R.: Graph cut algorithms for binocular stereo with occlusions. In: Mathematical Models in Computer Vision: The Handbook. Springer-Verlag (2005)

[13] Besl, P., McKay, N.: A method for registration of 3D shapes. PAMI **14**(2) (1992) 239–256
[14] Williams, J.P., Taylor, R.H., Wolff, L.B.: Augmented k-d techniques for accelerated registration and distance measurement of surfaces. In: Computer Aided Surgery: Computer-Integrated Surgery of the Head and Spine, Linz, Austria (1997) P01–21
[15] Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. IEEE Trans. Pat. Anal. Machine Intell. **9** (1987) 698–700
[16] K. Montgomery et. al.: Spring: A general framework for collaborative real-time surgical simulation. In: Proc. MMVR. (2002)