Real-time 3D Surface Tracking and Its Applications

Nicholas A. Ramey, Jason J. Corso, William W. Lau, Darius Burschka and Gregory D. Hager
Computational Interaction and Robotics Laboratory
The Johns Hopkins University
Baltimore, MD 21218
nramey@cs.jhu.edu

Abstract

We present a general technique for directly estimating and tracking surfaces from a stream of rectified stereo pairs in real-time. These techniques are based on the iterative updating of surface representations directly from image information and use no disparity search except during initialization. We perform the tracking through an iteratively re-weighted least squares minimization wherein a mask is incorporated to increase robustness to occlusion. The algorithms are formulated for a general family of linear in parameters surface models and discussed for the cases of planar surfaces and tensor product surfaces. These algorithms have been implemented on standard hardware and run at or near frame rate, with accuracy on the order of 1/20 of a pixel. We discuss applications of the technique including mobile robot localization, general deforming surface tracking, and biometry of biological surfaces.

1. Introduction

Computational stereo has the potential to provide dense, accurate range information to a set of visible surfaces. Indeed, over the last decade, the advent of cheap, fast stereo systems has led to a resurgence of interest in stereo vision. However, most real-time systems are currently based on traditional “brute force” search techniques using local match measures. Such methods are well-known to suffer in cases of occlusion and areas of low-texture, and provide depth information of limited (and sometimes questionable) accuracy [15].

We have developed effective multi-camera processing algorithms that can reconstruct and *track* the evolution of the set of rigid or deforming surfaces that comprise a scene. As described in Section 3, we have found the results to be fast, stable, and suggestive of accuracy on the order of 1/20 of a pixel in disparity resolution running at frame rate. Our formulation imposes no scale or structural constraints and implicitly verifies surfaces at every frame. The projection of patterned and polarized light allows accurate tracking of surfaces with specularities and low texture. We apply this method in multiple settings, including robot localization and biomedical surface tracking.

Our approach is motivated by previous work in image registration [12, 9, 17, 18] and template tracking [4] which poses the temporal correspondence problem as one of objective function minimization over a family of allowed image deformations. In our case, we consider the stereo disparity map on an image region to be a time-varying parametric function and optimize a set of parameters describing that map. We extend and generalize previous work on tracking and registration as follows. In [17, 18], uniform, bi-linear splines are used as a registration technique to compute optical flow. In the case of (calibrated) stereo we incorporate the epipolar constraint into the optimization process, therefore reducing the dimensionality of the problem. Furthermore, we formulate the problem in a computationally efficient, time-varying framework and, in that context, include methods to handle surface discontinuities and occlusions. In monocular tracking, one of the principle difficulties is the lack of 3D information. Indeed, almost all monocular tracking methods make some implicit or explicit assumption about the 3D structure of the tracked object [11, 5] and compute inter-frame or sequence motion based on it. In our case, we are directly inferring the 3D structure of the surface and do not explicitly track the motion of points on the surface [7, 16]. Finally, since we do not track motion, our algorithms can benefit from projected scene texture to improve local surface discrimination and accuracy [8]. In fact, we can even tailor the light to best improve the performance of local optimization.

The remainder of this paper is structured as follows. In the next section, we formulate the optimization problem and present a solution for parameter updates and mask computation. In Section 3, we describe two implementations of our algorithm and present results demonstrating its performance. In Section 4, we discuss some extensions of our algorithms and in Section 5 we conclude.

2. Mathematical Formulation

In this development, we assume a calibrated stereo system. Thus, incoming pairs of images can be rectified to form an equivalent non-verged stereo pair. Let $L(u, v, t)$ and $R(u, v, t)$ denote the left and right rectified image pair at time t, respectively.

In the non-verged case, the disparity map, D is a mapping from image coordinates to a scalar offset such that $L(u, v, t)$ and $R(u + D(u, v), v, t)$ are the projection of the same physical point in 3D space. As outlined above, our objective is to estimate a set of parameters $\mathbf{p} \in \Re^n$ that describe a parametric disparity map $D : \Re^n \times \Re^2 \rightarrow \Re^1$. This disparity map is defined on a given region A of pixel locations in the left image. For simplicity, we will consider A to be an enumeration of image locations and write $A = \{(u_i, v_i)'\}, 1 \leq i \leq N$.

In traditional region-based stereo, correspondences are computed by a search process that locates the maximum of a similarity measure defined on image regions. As we intend to perform a continuous optimization over $\mathbf{p}$, we are interested in analytical similarity measures. Candidate functions include sum of squared differences (SSD), zero-mean SSD (ZSSD), and normalized cross-correlation (NCC) to name a few. Robust objective functions [6] might also be considered. As we show below, we achieve similar effects using a reweighting loop in the optimization [3].

We choose our objective to be ZSSD. In practice, zero-mean comparison measures greatly outperform their non-zero-mean counterparts [1] as they provide a measure of invariance over local brightness variations. If the average is computed using Gaussian weighting, then this difference can be viewed as an approximation to convolving with the Laplacian of a Gaussian. Indeed, such a convolution is often employed with the same goal of achieving local illumination invariance.

Let $\overline{L}(u, v, t) = L(u, v, t) - (L * M)(u, v, t)$ and $\overline{R}(u, v, t) = R(u, v, t) - (R * M)(u, v, t)$ where $*$ denotes convolution and M is an appropriate averaging filter kernel in the spatial-temporal domain. Define $d_i = D(\mathbf{p}; u_i, v_i)$. We can then write our chosen optimization criterion as

$$O(\mathbf{p}) = \sum_{(u_i,v_i)\in A} w_i(\overline{L}(u_i, v_i, t) - \overline{R}(u_i + d_i, v_i, t))^2 \tag{1}$$

where w_i is an optional weighting factor for location $(u_i, v_i)'$.

For compactness of notation, consider A to be fixed and write $\overline{L}(t)$ to denote the $N \times 1$ column vector $(\overline{L}(u_1, v_1, t), \overline{L}(u_2, v_2, t), \ldots \overline{L}(u_N, v_N, t))'$. Likewise, we define $\overline{R}(\mathbf{p}, t) = (\overline{R}(u_1 + d_1, v_1, t), \ldots \overline{R}(u_N + d_N, v_N, t))'$.

We now adopt the same method as in [12, 9, 4] and expand $\overline{R}(\mathbf{p}, t)$ in a Taylor series about a nominal value of $\mathbf{p}$. In this case, we have

$$\begin{aligned} O(\triangle\mathbf{p}) &= \|(\overline{L}(t) - \overline{R}(\mathbf{p} + \triangle\mathbf{p}, t))W^{1/2}\|^2 \\ &\approx \|(\overline{L}(t) - \overline{R}(\mathbf{p}, t) - J(\mathbf{p}, t)\triangle\mathbf{p})W^{1/2}\|^2 \\ &= \|(\mathbf{E}(\mathbf{p}, t) - J(\mathbf{p}, t)\triangle\mathbf{p})W^{1/2}\|^2 \end{aligned} \tag{2}$$

where $\mathbf{E}(\mathbf{p}, t) \equiv \overline{L}(t) - \overline{R}(\mathbf{p}, t)$, $J(\mathbf{p}, t) = \partial\overline{R}/\partial\mathbf{p}$ is the $N \times n$ Jacobian matrix of $\overline{R}$ considered as a function of $\mathbf{p}$, and $W = \operatorname{diag}(w_1, w_2, \ldots w_N)$. Furthermore, if we define $J_D(\mathbf{p}) = \partial D/\partial\mathbf{p}$, we have

$$J(\mathbf{p}, t) = \operatorname{diag}(\overline{L}_x(t))J_D(\mathbf{p}) \tag{3}$$

where $\overline{L}_x(t)$ is the vector of spatial derivatives of $\overline{L}(t)$ taken along the rows [1]

It immediately follows that the optimal $\triangle\mathbf{p}$ is the solution to the (overdetermined) linear system

$$\left[J(\mathbf{p}, t)^t W J(\mathbf{p}, t)\right] \triangle\mathbf{p} = J(\mathbf{p}, t)^t W \mathbf{E}(\mathbf{p}, t) \tag{4}$$

In the case that the disparity function is linear in parameters, J_D is a constant matrix and J varies only due to time variation of the gradients on the image surface.

At this point, the complete surface tracking algorithm can now be written as follows:

1. Acquire a pair of stereo images and rectify them.
2. Convolve both images with an averaging filter and subtract the result.
3. Compute spatial x derivatives in the zero-mean left image.
4. Warp the right image by a nominal disparity map (e.g. that computed in the previous step) and subtract from the zero mean left image.
5. Solve (4).

The final two steps may be iterated if desired to achieve higher precision. The entire procedure may also be repeated at multiple scales to improve convergence, if desired. In practice we have not found this to be necessary.

[1] Here, we should in fact use the spatial derivatives of the right image after warping or a linear combination of left and right image derivatives. However in practice using just left image derivatives works well and avoids the need to recompute image derivatives if iterative warping is used.

2.1. Surface Formulations

In practice, we have found this formulation most effective for tracking disparity functions that are linear in their parameters (thus avoiding the problem of recomputing the Jacobian of the disparity function at runtime). A example is when the viewed surface is planar [2]. In this case, it is not hard to show that disparity is an affine function of image location, that is:

$$D(a, b, c; u, v) = au + bv + c \tag{5}$$

A more general example of a linear in parameters model is a B-spline. Consider a set of scan-line locations α and row locations β, such that $(\alpha, \beta) \in A$. With m parameters per scan-line and n parameters for row locations, a pth by qth degree tensor B-spline is a disparity function of the form

$$D(\mathbf{p}; \alpha, \beta) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,p}(\alpha)\ N_{j,q}(\beta)\ \mathbf{p}_{i,j} \tag{6}$$

To place this in the framework above, let κ denote an indexing linear enumeration of the mn evaluated basis functions, and define $B_{i,k} = N_{k,p}(\alpha_i) * N_{k,q}(\beta_i)$ for all $(\alpha_i, \beta_i) \in A$. It immediately follows that we can create the $N \times mn$ matrix $\mathcal{B}$

$$\mathcal{B} \equiv \left[\begin{array}{c} B_{1,1}, B_{1,2} .... B_{1,mn} \\ B_{2,1}, B_{2,2} .... B_{2,mn} \\ \vdots \\ B_{N,1}, B_{N,2} .... B_{N,mn} \end{array} \right]$$

and write

$$D(\mathbf{p}) = \mathcal{B}\mathbf{p} \tag{7}$$

It follows that the formulation of the previous section applies directly with $J_D = \mathcal{B}$.

2.2. Reweighting

One of the potential limitations with the system thus far is that it assumes all pixels in the region of interest fall on a continuous surface. In particular, an occluding surface introduces a $\mathcal{C}^0$ discontinuity into the problem. As we discuss in Section 4, it is possible to directly introduce $\mathcal{C}^0$ discontinuities into the spline formulation. However, for now we consider such "outliers" to be undesirable and to be avoided.

There are any number of methods for incorporating some type of robustness into an otherwise smooth $L2$ style optimization. Examples include Iteratively Re-Weighted Least Squares and Expectation-Maximization. Here, we adopt an approach that takes advantage of the spatial properties of the image. We define a weighting matrix at each new time step $W(t+1) = NCC(\overline{L}(t), \overline{R}(\mathbf{p}_t, t))$. That is, the weight for a pixel at each new iteration is the normalized cross correlation between the left and right images under the computed disparity function.

3. Applications

3.1. Implementation

The algorithms presented above have been implemented in Matlab/mex and in C. The Matlab version is used to gather data and verify results while the C version runs near framerate and is used as a demonstration system. The C version uses the OpenGL API to render the reconstructed surface with the video stream texture mapped onto the surface in real-time, and it also uses the XVision2 and Intel Integrated Performance Primitives Libraries for video and image processing. Unless otherwise noted, we run the real-time system on a Pentium IV running Linux with an IEEE 1394 stereo camera. The tracking system operates as fast as the stereo vision system, providing a rectified stream of images at a maximum of 26Hz. Biomedical tracking systems run at frame rate, although intra-operative sequences are processed post-operatively.

In all cases, processing is initiated with a standard correspondence-based stereo calculation. However, as the results indicate, the algorithm admits an approximating plane for the seed.

3.2. Mobile Robot Localization

The algorithm is applied to robot navigation. Many tasks on a mobile robot require knowledge about the incremental changes in position during the operation. We observe that when viewed in a non-verged stereo system, planes project to a linear function in the disparity (5). Thus, tracking the three parameters $\left[\begin{array}{ccc} a & b & c \end{array} \right]^T$ is sufficient to track the 3D plane. For further information and a complete discussion of detecting and segmenting planar regions from input images, refer to [2].

The relative localization between consecutive camera acquisitions is based on significant planes in the field of view of the camera. Each plane allows the estimation of three out of the six possible parameters of the pose. A set of two non-coplanar planes allows the estimation of the 2D position in the ground plane of the local area and all rotation angles of the robot. Therefore, relative localization is possible when at least two planes are tracked between frames.

For the experiments discussed in this section, we are using a stereo head with 5.18mm lenses, a 92mm baseline, and square pixels 0.12mm wide. The plane being observed, unless otherwise specified, is roughly orthogonal to the viewing axis and at a depth of one-meter.

Table 1: Parameter estimation accuracy for a plane at a distance of about one meter.

	Z Mean	Z Std Dev	Normal Error Std Dev
1	1064.8mm	2.2359mm	0.2947°
2	1065.3mm	1.7368mm	0.2673°
3	1065.2mm	1.5958mm	0.2258°

3.2.1 Convergence Radius

For a controlled environment with a stationary plane and robot, we calculated an initial guess for the plane parameters and then varied this guess to test the robustness of the tracking algorithm to initialization error.

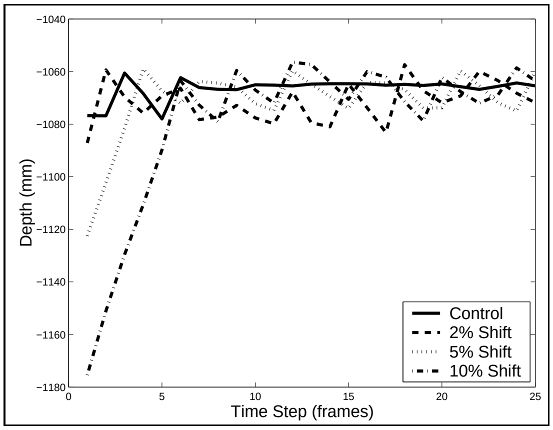

Figure 1: Graph for convergence while introducing error into the seed's depth by 2, 5 and 10 percent toward the camera.

In Figure 1, we show the time to convergence when we shift the seed's depth closer to the camera at varying levels. The convergence speed is directly proportional to the magnitude of the introduced error. We note that the convergence speed is only about 5 frames for an error of 10%.

3.2.2 Accuracy of Parameter Estimation

Assuming a suitably textured scene, the algorithm estimates a plane's parameters with sub-pixel accuracy (approximately 1 pixel per cm). However, this estimation accuracy varies with the depth of the plane being tracked because the depth-per-disparity increases as the distance to the plane increases. Table 1 shows the statistics for the plane.

For a non-stationary scene, we show the accuracy of our system against the robots internal odometry. Figure 2 shows the robot performing oscillatory rotations in front of a plane (700 mm distance). We see that the algorithm performs extremely well for the rotational motion. The estimated orientation lags minimally behind the odometric values; the length of the lag is proportional to the convergence speed.

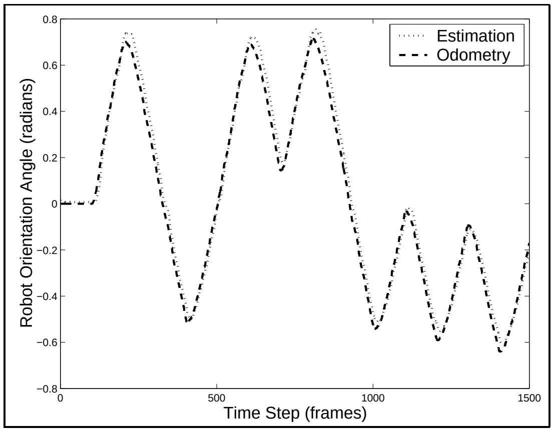

Figure 2: Accuracy of robot orientation.

3.3. Tracking Deforming Surfaces

The algorithm may also be applied to tracking deformable surfaces in a variety of applications. This technique offers a means to physically model the flow or irregular deformation of continuous surfaces without imposing constraints on scale or structure. The system implicitly verifies the surface at every step. These features ensure that it can reliably track the true motion of a flag in the wind, the gentle swell of an ocean wave, and even the otherwise indistinguishable irregularities of a beating heart.

Since the computationally limiting step of the system is dominated by the solution to the large linear system (4) which is dependent on the size of the region being observed and the resolution of the control points, it is ambiguous to give hard frame-rates. However, in the typical case, we track an image region about 20 percent of the image, and use bi-cubic surfaces approximated with 4 to 6 control points in each direction, running at frame rate. Further improvements will take advantage of the banded nature of the linear system and other simple algorithmic considerations.

3.3.1 Convergence vs. Parameter Density

As noted by other authors [15], it is difficult to measure the accuracy of stereo algorithms as there is usually no way to get ground truth. One measure of performance is the ability of the algorithm to correctly register the left and right images. To this end, we plot the mean image difference between the left and the warped right image on a representative sequence for three different control point resolutions (Figure 3). The graphs show the average pixel error per iteration. The noticeable peaks correspond to new images in the sequence; for a given frame of the sequence we continuously refine our surface approximation until the intra-iteration update is below a threshold. For our experiments, we use a convergence threshold of 10^{-3} pixels. As expected, for a low control point density, the average pixel error is slightly higher than for higher control point densities. However, the convergence speed is slower for higher control point densities. It should be noted that the real-time system is not left to converge on 10^{-3}. Instead, a nominal number (2-5) of iterations yields satisfactory results without jeopardizing accuracy (i.e. inter-frame image difference remains small in realtime recordings).

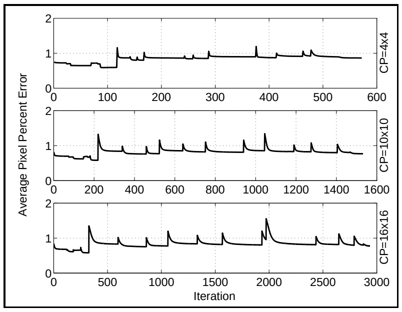

Figure 3: Tracking Convergence.

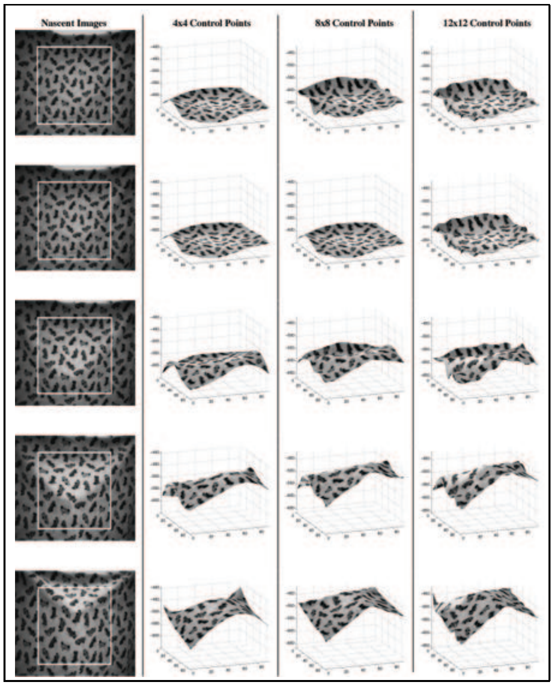

Figure 4: Tracking Cloth with Varying Control Point Densities.

3.3.2 Tracking Performance

To evaluate the system's performance, it is run on a sequence of a deforming surface (Figure 4). For this sequence, the model was bi-cubic with varying control point density (4x4, 8x8, 12x12); the recreated surface and nascent left intensity image are provided for selected key frames, along with average pixel error at each key frame (Figure 5). The increase in pixel error arises from an elevated inter-frame difference for later frames (recording at 0.5Hz, with increased speed of deformation starting at approximately frame 10). Given the above data on time to convergence (number of iterations), it is possible to intuit the appropriate density of parameters based on the desired speed and accuracy (determined by evaluation metrics including residual and parameter variances). Although an automated method for deciding this density has not been fully implemented, the underlying principles have been formulated and are under investigation. These principles are built on minimum description length formulations [14, 19]. It is important to note that the average pixel error is relatively low in both scenes, compared to the accepted noise level of 2 pixel values for these cameras.

3.3.3 Structured Light

This experiment tests the performance of the tracking system (bi-cubic with 4x4 control point density) in a scene with projected patterned light, or structured light. Structured light can provide the surface texture necessary to fuel the optimization process in regions of inherently low surface texture. Certain patterns of light provide better textures than others. For example, a pattern with vertically-repeting horizontal lines may cause the system to enter a local minimum where the updated disparity mask is actually ”matching” too distant or too near correspondences in the subtraction process. In the next sequence (Figure 6), note that as the frequency of the pattern increases to a point, the pixel error also rises. This may be attributed to the increase in overall texture of the higher frequency images. This figure demonstrates that as the frequency increases beyond this point, the pixel error decreases again. As the frequency of the pattern increases to an extreme, the projected light becomes homogeneous, and essentially acts as a flood light. Thus, we arrive at the first condition of no projected light, where pixel error is low due to lack of texture.

3.3.4 Occlusion Robustness

This experiment is designed to test the efficacy of the framework in the face of occlusions. Assuming occlusions can be represented as $\mathcal{C}^0$ discontinuities, the tracker effectively masks out occlusions from the optimization process (2), prohibiting the occlusion from erroneously altering the understanding of the surface. Key-frames of a sequence are run through the tracker, recording the mask and reconstruc-

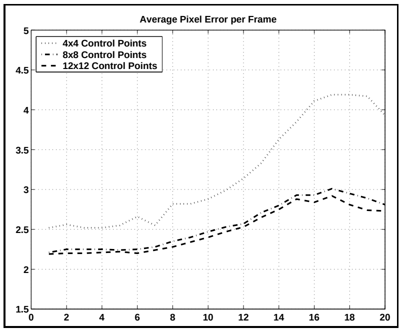

Figure 5: Average Pixel Error per Frame.

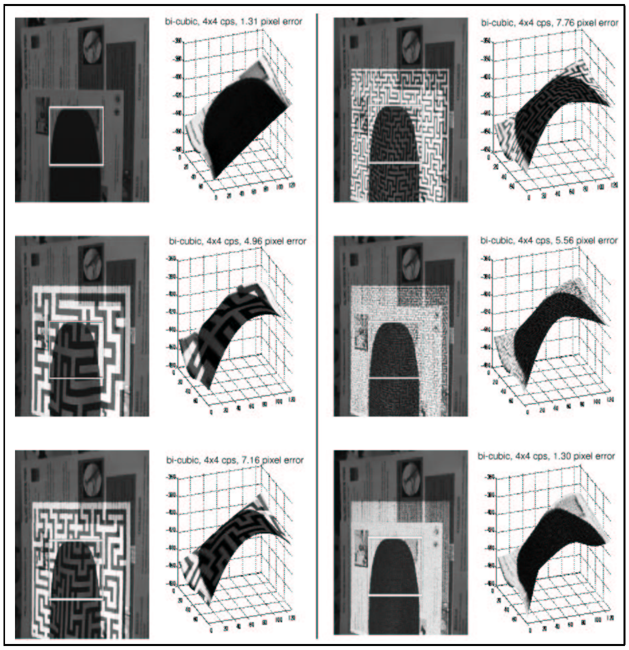
bi-cubic, 4x4 cps, 1.31 pixel error
bi-cubic, 4x4 cps, 7.76 pixel error
bi-cubic, 4x4 cps, 4.96 pixel error
bi-cubic, 4x4 cps, 5.56 pixel error
bi-cubic, 4x4 cps, 7.16 pixel error
bi-cubic, 4x4 cps, 1.30 pixel error

Figure 6: Tracking with Structured Light.

tions of the surface.

Figure 7 shows two contrived examples illustrating the ability of low degree splines to approximate $\mathcal{C}^0$ and $\mathcal{C}^1$ discontinuities. These approximations incorporate no knot-multiplicities. It is evident that the low degree splines can approximate the discontinuities well.

Although splines can handle $\mathcal{C}^0$ discontinuities, in most cases such discontinuities are representative of off-surface occlusion and would interrupt the stability of the occluded surface's approximation.

As mentioned earlier in Section 2.2, we incorporate a weighting matrix (a mask) into our scheme in order to make our tracking robust to such occlusions. We calculate the weight as the normalized cross correlation of the spline surface at the end of each frame. The computed mask of each frame is dilated and propagated forward for the next frame.

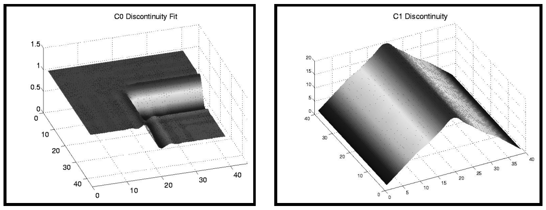

Figure 7: Fitting Discontinuities.

For each key-frames in Figure 8, L (top), $\mathbf{E}$ (top-middle), mask (bottom-middle), and the reconstructed surface (bottom) are provided with and without masking (above and below the heavy line, respectively). Note that without the mask the surface demonstrates exaggerated deformation in the face of occlusions.

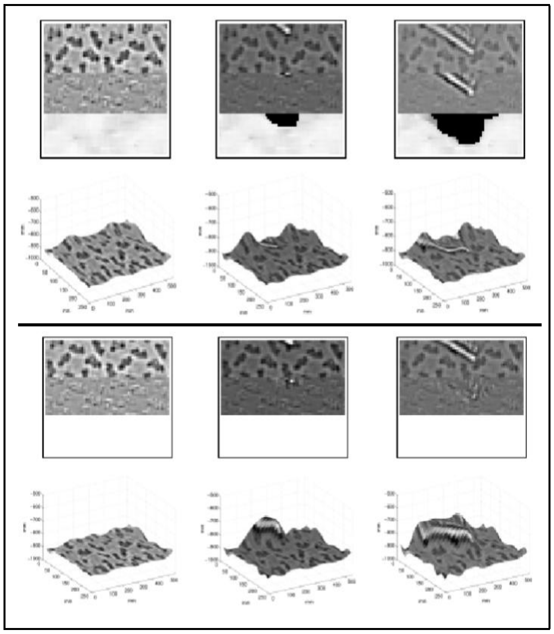

Figure 8: Tracking with/without occlusion robustness.

3.4. Tracking Biomedical Surfaces

Three-dimensional biomedical images can provide important information about the properties of the objects from which the images are derived. An understanding of the overall surface anatomy may provide immense new opportunities for medical diagnosis and treatment especially in robotic assisted surgeries. This technique can provide more accurate intra-operative image guidance when registration to preoperative images is no longer valid due to movements and deformations of tissues.

We used an anesthetized Wistar rat as an animal model. Images of the rat's chest's movement were acquired by a stereo microscope (Zeiss OPMI1-H) mounted with two CCD cameras (SONY XC77). The rats fur provided a natural texture. An eight second sequence was processed offline by our Matlab implementation. In Figure 9 we graph the respiration (75 breaths per minute) of the rat which was computed by recording a fixed point on the tracked surface. This disparity representing respiration varies by 1/10 of a pixel. Close inspection suggests another periodic signal riding the respiratory signal that is of the order of 1/20 of a pixel. We believe this second variation to be the heart beat, although further studies are needed confirmation.

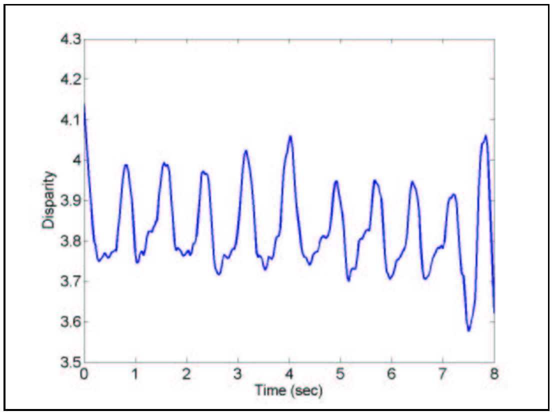

Figure 9: Graph of rat respiration.

In a second experiment, a cross-bred domestic pig (weight, 19.5 kg) was anesthetized with telazol-ketamine-xylazine (TKX, 4.4 mg T/kg, 2.2 mg K/kg, and 2.2 mg X/kg) and mechanically ventilated with a mixture of isoflurane (2%) and oxygen. Heart rate was continuously monitored by a pulse oximeter (SurgiVet, Waukesha, WI). The da Vinci tele-manipulation system (Intuitive Surgical, Sunnyville, CA) was used for endoscopic visualization. Three small incisions were made on the chest to facilitate the insertion of a zero-degree endoscope and other surgical tools. The pericardium was opened and video sequences of the beating heart from the left and right cameras were recorded at 30 frames/sec. The recording lasted approximately two minutes.

The system captured both the beating of the heart and the respiration of the subject (Figure 10). The results are consistent with the other measurements we took during the surgery. In Figure 10, the blue line is a plot of the motion of a fixed point on the surface. The respiration (red-dotted line) is computed using Savitzy-Golay Filtering. For a more detailed discussion of the experiment please see [10].

4. Extensions

Multigrid Enhancements The results in this paper specifically track surfaces in pre-specified regions of the images. However, to handle general imagery and potentially track multiple surfaces, further research is required. In [2], we track multiple planes using a masking technique, but this does not generalize to all linear in parameters surfaces as the tracking method in this paper does. Thus, we plan to explore the use of adaptive multigrid techniques [13]. Multigrids provide a systematic method of managing multiple regions in the image each having different surface parameters.

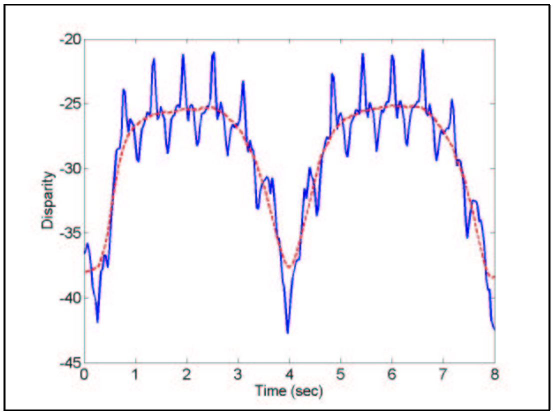

Figure 10: Graph of pig respiration and heart beats.

Tracking Depth Rather Than Disparity One might object that locally polynomial (e.g. locally quadratic) surface patches do not project, in general to locally quadratic disparity functions. In this regard, we note two facts. First, if we consider parameterized range as a function of image coordinates, then for a non-verged camera, we can write

$$D(\mathbf{p}; u, v) = s/z(\mathbf{p}; u, v) \tag{8}$$

where s combines scaling due to baseline and focal length. It follows immediately that

$$\nabla_{\mathbf{p}} D(\mathbf{p}; u, v) = -s/z(\mathbf{p}, u, v)^2 \nabla_{\mathbf{p}} z(\mathbf{p}, u, v) \tag{9}$$

If we approximate z as a tensor B-spline surface, and we define $z(\mathbf{p}) = \mathcal{B}\mathbf{p}$, this we have immediately that

$$J_D(\mathbf{p}) = -s \; \mathrm{diag}(1/z(\mathbf{p}, u, v))^2 \mathcal{B}. \tag{10}$$

Thus, we can track a range map rather than a disparity map with little extra cost.

One might further object that this formulation still does not adequately address locally polynomial surfaces. In this case, the logical solution is to use *rational* b-splines. This is a subject of our ongoing research.

5. Conclusion

We presented an approach to real-time 3D surface tracking and demonstrated its application to a number of fields including mobile-robot navigation, general deformable surface tracking, and biomedical surface tracking. This technique has been formulated as a general linear in parameters optimization without disparity searching. In performing a continuous optimization over these parameters, we compute the disparity surface directly from image intensity data. We offer results demonstrating the converged fit of multiple surfaces in a variety of robotic, general, and medical schemes.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0112882. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

We thank Intuitive Surgical (Sunnyville, CA) for providing the stereo endoscope, and Dr. Randy Brown and Dr. David Yuh for surgical assistance.

References

[1] J. Banks, M. Bennamoun, K. Kubik, and P. Corke. Evaluation of new and existing confidence measures for stereo matching. In *Proc. of the Image & Vision Computing NZ conference (IVCNZ98)*, 1998.

[2] Jason Corso, Darius Burschka, and Gregory D. Hager. Direct Plane Tracking in Stereo Image for Mobile Navigation. In *Proceedings of International Conference on Robotics and Automation*, pages 875–880, 2003.

[3] R. Dutter and P.J. Huber. Numerical methods for the nonlinear robust regression problem. *J. Statist. Comput. Simulation*, 13(2):79–113, 1981.

[4] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 20(10):1125–1139, 1998.

[5] G. Hager and P. Belhumeur. Tracking in 3d: Image variability decomposition for recovering object pose and illumination. *Pattern Analysis and Applications*, March 1999.

[6] Peter J. Huber. *Robust Statistics*. Wiley, 1981.

[7] M. Irani and P. Anandan. About direct methods. In *Vision Algorithms: Theory and Practice (International Workshop on Vision Algorithms)*, 1999.

[8] Sing Bing Kang, Jon A. Webb, C. Lawrence Zitnick, and Takeo Kanade. An active multibaseline stereo system with real-time image acquisition. Technical Report CMU-CS-94-167, School of Computer Science, Carnegie Mellon University, 1994.

[9] D. Keren, S. Peleg, and R. Brada. Image sequence enhancement using sub-pixel displacements. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1988.

[10] W. Lau, N. Ramey, J. Corso, N. Thakor, and G. Hager. Stereo-based endoscopic tracking of cardiac surface deformation. In *Proc. of Medical Image Computing and Computer Assisted Intervention*, In review.

[11] L. Lu, Z. Zhang, H.-Y. Shum, Z. Liu, and H. Chen. Model- and exemplar-based robust head pose tracking under occlusion and varying expression. In *In Proc. IEEE Workshop on Models versus Exemplars in Computer Vision, (CVPR'01)*, 2001.

[12] B. Lucas and T. Kanade. An iterative image registratoin technique with an application to stereo vision. In *Proceedings DARPA Image Understanding Workshop*, 1981.

[13] E. Memin and P. Perez. A Multigrid Approach for Hierarchical Motion Estimation. In *In Proceedings of International Conference of Computer Vision*, pages 933–938, 1998.

[14] J. Rissanen. Modeling by shortest data description. *Automatica*, 14, 1978.

[15] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.

[16] G. Stein and A. Shashua. Direct estimation of motion and extended scene structure from a moving stereo rig. *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[17] Richard Szeliski and James Coughlan. Hierarchical spline-based image registration. In *In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994.

[18] Richard Szeliski and Heung-Yeung Shum. Motion estimation with quadtree splines. Technical report, DEC Cambridge Research Lab, 1995.

[19] Roberto Cipolla Tat-Jen Cham. Automated b-spline curve representation incorporating mdl and error-minimizing control point insertion strategies. *PAMI*, 21, 1999.