

Technical Report 2003-02-CIRL-CS-JHU

Stereo-Based Direct Surface Tracking with Deformable Parametric Models

Jason J. Corso, Nicholas A. Ramey, and Gregory D. Hager
Computational Interaction and Robotics Laboratory
The Johns Hopkins University
Baltimore, MD 21218
{jcorso|nramey|hager}@cs.jhu.edu

Abstract

We present a general set of techniques for directly estimating and tracking deformable surfaces from a stream of rectified stereo pairs. These techniques are based on 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 specific case of tensor product surfaces. We also show that it is possible to naturally incorporate dynamic surface segmentation into the algorithm.

These algorithms have been implemented and run at or near frame rate. We evaluate their performance on a variety of image data, including cloth and bending paper.

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, such 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 provided depth information of limited accuracy (and sometimes questionable) accuracy [15].

Our goal is to develop effective multi-camera processing algorithms that can reconstruct and *track* the evolution of the set of rigid or deforming surfaces that comprise a scene. Our specific motivation is to track deformation of biological surfaces using video from stereo microscopes and endoscopes now commonly employed in operating rooms. However, more broadly effective stereo surface tracking can be used in scene reconstruction for mobile systems [3], realistic graphical rendering of deforming objects, human-machine interaction, or as a front end to algorithms for tracking and identification of moving objects.

Our approach is motivated by previous work in image registration [13, 11, 17, 18] and visual tracking [6, 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. In previous work, we used locally planar descriptions for disparity [4, 3]; in this paper we extend this work to deformable surfaces represented by B-spline surfaces (Appendix A).

This work extends 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 monoc-

ular 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 [12, 7] and compute interframe 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 [9, 16]. Finally, since we do not track motion, our algorithms can benefit from projected scene texture to improve local surface discriminability and accuracy [10]. 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 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 $\mathbf{p} \in \mathfrak{R}^n$ that describe a parametric disparity map $D : \mathfrak{R}^n \times \mathfrak{R}^2 \rightarrow \mathfrak{R}^1$. 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. In this realm, candidate functions include sum of squared differences (SSD), zero-mean SSD (ZSSD), and normalized cross-correlation (NCC) to name a few. Robust objective functions [8] might also be considered. As we show below, we achieve similar effects using a reweighting loop in the optimization [5].

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 $\bar{L}(u, v, t) = L(u, v, t) - (L * M)(u, v, t)$ and $\bar{R}(u, v, t) = R(u, v, t) - (R * M)(u, v, t)$ where $*$ denotes convolution and M is an appropriate averaging filter kernel in the spatial-temporal domain. We can then write our chosen optimization criterion as

$$O(\mathbf{p}) = \sum_{(u_i, v_i)} w_i (\bar{L}(u_i, v_i, t) - \bar{R}(u_i + D(\mathbf{p}; u_i, v_i), v_i, t))^2 \quad (1)$$

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 $\bar{L}(t)$ to denote the $N \times 1$ column vector $(\bar{L}(u_1, v_1, t), \bar{L}(u_2, v_2, t), \dots, \bar{L}(u_N, v_N, t))'$. Likewise, we define $\bar{R}(\mathbf{p}, t) = (\bar{R}(u_1 + D(\mathbf{p}; u_1, v_1), v_1, t), \dots, \bar{R}(u_N + D(\mathbf{p}; u_N, v_N), v_N, t))'$. Finally, for convenience we will also write $D(\mathbf{p}) = (D(\mathbf{p}; u_1, v_1) \dots D(\mathbf{p}; u_N, v_N))'$.

We now adopt the same method as in [13, 11, 6] and expand $\bar{R}(\mathbf{p}, t)$ in a Taylor series about a nominal value of \mathbf{p} . In this case, we have

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

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

$$J(\mathbf{p}, t) = \text{diag}(\bar{L}_x(t))J_D(\mathbf{p}) \quad (3)$$

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

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

¹Here, 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.

$$[J(\mathbf{p}, t)^t W J(\mathbf{p}, t)] \Delta \mathbf{p} = J(\mathbf{p}, t)^t W \mathbf{E}(\mathbf{p}, t) \quad (4)$$

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.

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 [3]. In this case, it is not hard to show that disparity is an affine function of image location, that is:

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

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

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

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}

$$\mathcal{B} \equiv \begin{bmatrix} B_{1,1}, B_{1,2} \dots B_{1,mn} \\ B_{2,1}, B_{2,2} \dots B_{2,mn} \\ \vdots \\ B_{N,1}, B_{N,2} \dots B_{N,mn} \end{bmatrix}$$

and write

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

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 that 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 $W(t+1) = NCC(\bar{L}(t), \bar{R}(\mathbf{p}_t, t))$. 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 Experimental Results

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 frame-rate and is used as a demonstration system (in particular, the mpeg8 for this paper were generated with the latter). 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. We run the real-time system on a Pentium III 700Mhz laptop running Linux with an IEEE 1394 stereo camera. The stereo vision system provides a rectified stream of images at a maximum of 26Hz.

Since the computational cost 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-quadratic surfaces approximated with 6 to 8 control points. For such cases we achieve rates between 12 and 15 Hz. We would expect to see significant improvements if we took further advantage of the banded nature of the linear system and other simple algorithm improvements.

To initiate processing, we employ a standard stereo algorithm with a complete correspondence search. Figure 1 shows the reconstruction from the disparity versus reconstruction from the parametric surface disparity function. This is how our algorithm is initially seeded. Note that underlying disparities vary within the range of a single disparity. In fact, most of the results we will present involve computing disparities to far less than one pixel of accuracy.

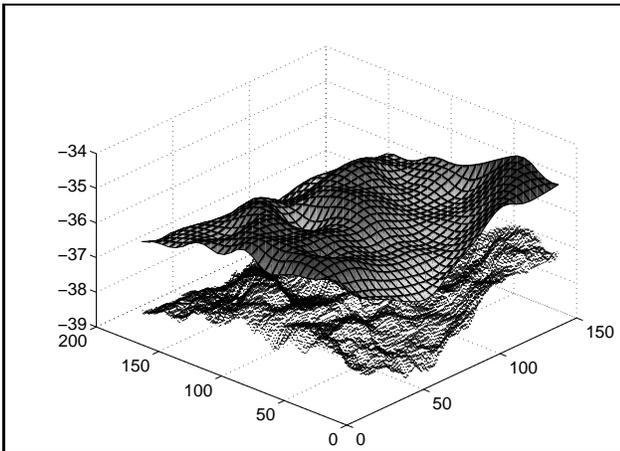


Figure 1: A raw disparity map and an initial spline fit to that map. The fitted surface is offset for presentation purposes.

In Figure 2 we show an example of a bending piece of paper being tracked and reconstructed. We used a bi-quadratic surface with 6 control points in each direction in this experiment. Three key-frames are shown from the sequence that starts with a planar-calibration pattern and the ripples the pattern. Notice

in the third image that there is a drop-off on the left side of the image, this is not an error; it is because no mask was used in this sequence and the surface is trying to fit the sharp drop in disparity from the discontinuity in the tracked region.

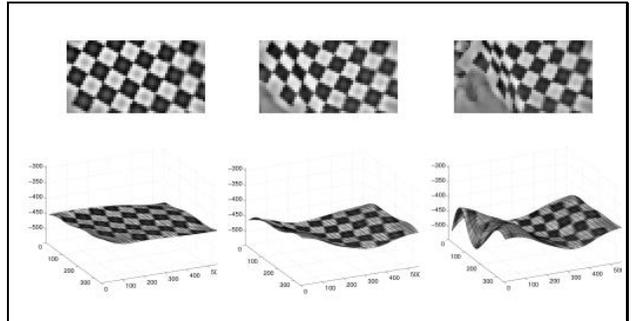


Figure 2: Tracking and reconstruction of a piece of bending paper.

3.2 Fit vs. Control Points

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 the image difference between the left and the warped right image on a representative sequence for three different control point resolutions (Figure 3). The graphs shows 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} . As expected, for a low control point density, the average pixel error is slightly higher than for higher control point densities. On the contrary, the convergence speed is slower for higher control point densities.

We evaluate variance of both the residual error var_R and the parameter estimate var_J as a function of the number of control points. The variance of the residual error var_R is computed as

$$\text{var}_R(t) = \text{var}(\mathbf{E}(\mathbf{p}, \mathbf{t}) - J(t) * \Delta \mathbf{p}), \quad (8)$$

Intuitively, as we increase the number of control points, we expect var_R to decrease because we can better approximate the warping function. In Figure 4, we show this phenomenon in four twenty-frame sequences. The variance of the residuals is computed for

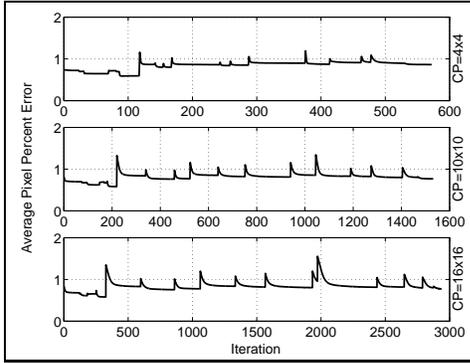


Figure 3: Difference between left and warped right image for a sequence of images with varying control point lattice resolution.

every frame in the sequence once the surface has converged, and we plot the mean residual variance for the complete sequence as we vary the number of control points.

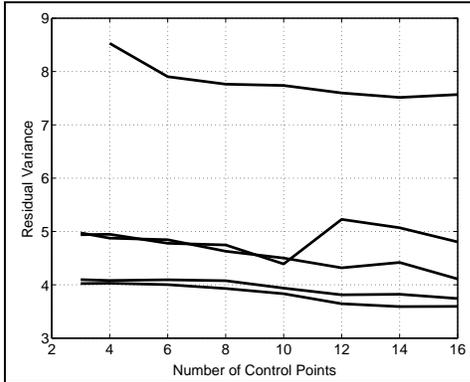


Figure 4: Residual variance (var_R) versus number of control points.

If we view the input data as noise contaminated, then we can establish a relationship between parameter variance and data variance. It is not hard to show that parameter variance is inversely related to the square of the singular values of the Jacobian. Thus we define var_J as $1/s_{min}^2$, where s_{min} is the smallest singular value of J . As the number of control points increases, the surface becomes less determined by the data and therefore var_J will go up. In Figure 5 we plot var_J versus the number of control points for the same sequences as Figure 4. The upward trend is clearly present in the graph.

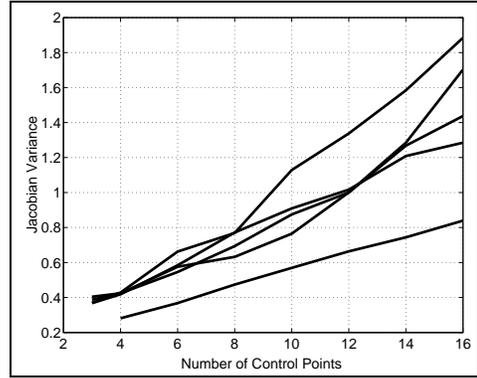


Figure 5: Jacobian variance (var_J) versus number of control points.

3.3 Masking and Occlusion

Figure 6 shows two contrived examples illustrating the ability of low degree splines to approximate C^0 and C^1 discontinuities. These approximations incorporate no knot-multiplicities. It is evident that the low degree splines can approximate the discontinuities well.

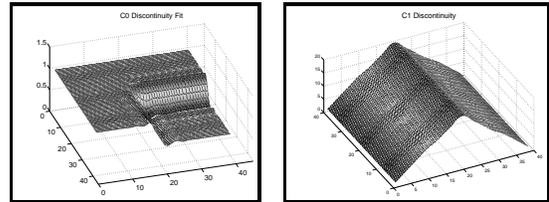


Figure 6: Synthetic C^0 (left) and C^1 (right) discontinuities approximated with a quadratic spline.

Although splines can handle 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. The sequences of key-frames in Figures 7 and 8 depict L (top), \mathbf{E} (top-middle), mask (bottom-middle), and the reconstructed surface (bottom). Note that without the mask (more properly, a mask of all ones), the surface demonstrates exaggerated deformation in the face of occlusions.

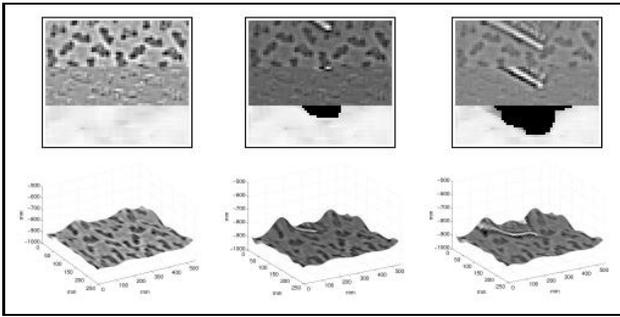


Figure 7: A sequence of key-frames shows the tracking performance in the face of occlusion with masking. In all images, L (top), \mathbf{E} (top-middle), mask (bottom-middle), and the reconstructed surface (bottom).

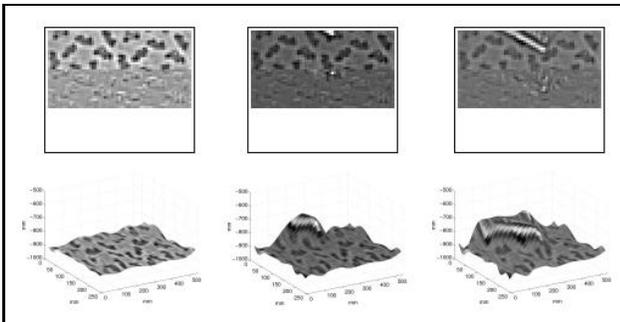


Figure 8: The same sequence without masking enabled. Notice that the surface compensates for the occluder.

4 Extensions

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-merged camera, we can write

$$D(\mathbf{p}; u, v) = s/z(\mathbf{p}; u, v) \quad (9)$$

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

$$\nabla_{\mathbf{p}} D(\mathbf{p}; u, v) = s/z(\mathbf{p}, u, v)^2 \nabla_{\mathbf{p}} z(\mathbf{p}, u, v) \quad (10)$$

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

$$J_D(\mathbf{p}) = \text{sdiag}(1/z(\mathbf{p}, u, v))^2 \mathcal{B}. \quad (11)$$

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.

Tracking without Rectification In some cases, it may be advantageous to compute a solution without prior rectification. In this case, we would compute a disparity function with two coordinate offsets at each pixel. In terms of a spline basis, this would be:

$$D(\mathbf{p}_u, \mathbf{p}_v) = \mathcal{B}[\mathbf{p}_u, \mathbf{p}_v]$$

Assuming a known calibration between left and right image, we also know the epipolar constraint between coordinates of the left and right images

$$(u_r, v_r, 1)' E(u_l, v_l, 1) = 0.$$

However, note that we can write the right-hand coordinates for pixel location i in terms of the i th row of spline basis as $B_i[\mathbf{p}_u, \mathbf{p}_v]$. As a result, we can write the epipolar constraint as a linear function of the spline parameters, and thus we now only need to solve a quadratic optimization problem with a linear equality constraint. The solution is again given by solving an associated linear system.

This is particularly useful if we are computing a disparity surface using 3 or more cameras. In this case, we can an optimization in terms of a base reference image (where disparity is computed) and matching with epipolar constraints for the other cameras.

5 Conclusion

We have presented an approach to tracking deforming surfaces by directly estimating parametric disparity maps on a sequence of rectified images. This technique uses no disparity search and has been formulated as a general linear in parameters optimization. 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 the surface as a function of control point density. A normalized cross-correlation based mask was implemented for discontinuities, including occlusions and drop-offs. This method seeds future work with tracking in depth and without rectification.

A B-splines

For more complete discussions of B-splines, B-spline surfaces, and their properties, see [14, 2]. B-splines are defined by their degree p , number $(m + 1)$ and position (\mathbf{p}) of control points, and their knot vector U , which spans the parameter space. Bsplines (12) are constructed as the superposition of weighted basis functions $N_{i,p}(u)$ (13):

$$\mathbf{C}(u) = \sum_{i=0}^m N_{i,p}(u) \mathbf{p}_i \quad (12)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (13)$$

$$U = \{ \underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{k-p-1}, \underbrace{b, \dots, b}_{p+1} \}$$

with $(k + 1)$ knots, where a and b are repeated $p + 1$ times to clamp the curve to the first and last control points. The degree p , number of control points $m + 1$, and the number of knots $k + 1$ satisfy $k = m + p + 1$. This relationship of control points, knots, and degree has important implications for the shape of the spline.

The degree determines the shape of the basis functions, and by extension, the global shape of the curve. The control points affect scaling of these basis functions (there are at most $p + 1$ non-zero basis functions for any knot span). Finally, multiplicity of a knot can have tremendous local effects, eliminating C^0 continuity for multiplicities of $p + 1$, C^1 continuity for multiplicities of p , and so on.

These definitions extend easily to tensor surfaces by incorporating p th and q th degree basis functions in the horizontal, u , and vertical, v , directions (14), along with corresponding knot vectors U and V of $r + 1$ and $s + 1$ knots, respectively.

$$\mathbf{S}(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) \mathbf{p}_{i,j} \quad (14)$$

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.

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] R. Bartels, J. Beatty, and B. Barsky. *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufman, 1987.
- [3] 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*, 2003. To appear.
- [4] Jason Corso and Gregory D. Hager. Planar surface tracking using direct stereo. Technical report, The Johns Hopkins University, 2002. CIRL Lab Technical Report.
- [5] R. Dutter and P.J. Huber. Numerical methods for the nonlinear robust regression problem. *J. Statist. Comput. Simulation*, 13(2):79–113, 1981.
- [6] 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.
- [7] G. Hager and P. Belhumeur. Tracking in 3d: Image variability decomposition for recovering object pose and illumination. *Pattern Analysis and Applications*, March 1999. Accepted to appear.
- [8] Peter J. Huber. *Robust Statistics*. Wiley, 1981.
- [9] M. Irani and P. Anandan. About direct methods. In *Vision Algorithms: Theory and Practice (International Workshop on Vision Algorithms)*, 1999.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings DARPA Image Understanding Workshop*, 1981.
- [14] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer, 1995.
- [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.