International Conference on Computer Vision,
September 1999, pages 648-653.

Tracking Articulated Objects in Real-Time Range Image Sequences

Michael H. Lin
Interval Research Corporation
1801C Page Mill Road
Palo Alto, CA 94304
michelin@cs.stanford.edu

Abstract

A simple algorithm for tracking the pose of articulated objects in real-time range image sequences is proposed. This method models each target segment as a planar patch bounded by the convex hull of two circles, and utilizes both edge-like and region-like information in matching the model to the target. It uses hard constraints for joint attachment, and is designed to be robust to occlusions and missing data. Experimental results are presented in which a human arm is successfully tracked over 26 frames of real, video-rate range imagery.

1. Introduction

Tracking the position and pose of articulated objects in image sequences is an important problem in computer vision with many potential applications. Numerous algorithms have been developed for doing so in the usual domain of color or intensity images. Among these, some use only sparse feature points or edges, while others use dense image data. In the less common domain of range images, however, although much attention has been paid to estimating and tracking rigid-body motion, less has been done specifically regarding articulated motion. This may be because tracking articulated motions in range images is much more interesting when large quantities of range data are available, but obtaining such data quickly and accurately has typically been at least as difficult as estimating object motion directly from color or intensity images.

With the advent of real-time stereo hardware [9, 19, 12], however, there are now sources of fairly accurate range data that is dense in both space and time. Since range data provides precisely that information which is missing from 2D color or intensity data, incorporating range data as well as color/intensity data could be extremely helpful in attacking the articulated motion estimation problem.

This work begins to explore the extent of this helpfulness by looking at the problem of tracking an articulated object with known structure and initial configuration solely via a range image sequence, without the use of color or intensity information. The contribution of this work is the introduction of a simple yet novel model to represent the visible surface of the target, in which each articulated segment is described as a planar patch bounded by the convex hull of two circles, and the application of this model to high-resolution, video-rate range imagery.

Section 2 reviews some previous work in target tracking, and motivates the current work. Section 3 explains the proposed method. Section 4 presents some encouraging experimental results on a range image sequence of a real human arm. Section 5 suggests directions for future work.

2. Background

The majority of work on 3D object tracking or motion estimation has been directed at the use of intensity or color images. For example, Aggarwal and Nandhakumar [2] and Huang and Netravali [8] give surveys of several such methods for estimating rigid-body motion, Aggarwal et al. [1] review work for estimating articulated motion, and Rehg [14] and Bregler [6] further demonstrate two qualitatively different approaches to tracking articulated motion. These methods implicitly or explicitly calculate from the image data the depth/range of some or all of the target's image pixels in determining its 3D pose. If range information is available, however, one would like to use it directly, rather than recalculating it from intensity or color information. Some methods for color or intensity in fact do not depend much on the interpretation of pixel attributes, and could perhaps be modified so that range could be substituted instead, but this would not take advantage of the geometric meaning of range information.

Sabata and Aggarwal [15] and Huang and Netravali [8] review some motion estimation methods explicitly designed to use range data. Most of these methods are based on the *a priori* knowledge of 3D point correspondences, using

them to solve for an affine or rigid-body transformation mapping one frame into the next [13, 4]. Others do not require that the set of point correspondences be known, but do require that it exist and be one-to-one [13, 3]. This requirement is reasonable when range information is available at a sparse set of points, but is less appropriate for spatially dense range images.

Matching surface patches instead of points eliminates the need for pointwise correspondences, and can furthermore reduce the sensitivity of the motion estimation to occlusions and measurement noise. Methods have been developed to segment dense range data into surface patches, to find correspondences among them, and to use these correspondences to estimate 3D rigid body motion [10, 17, 16]. The extracted patches are generally restricted to be planar or quadric, however, and in any case the segmentation is performed according to surface curvature and thus could be highly sensitive to surface warping (e.g. as occurs in clothing). Vemuri and Skofteland [18] estimate a single surface from sparse range data, and Horn and Harris [7] essentially treat an entire dense range image as one surface patch, so they do not have a problem with patch correspondence. However, for bumpy or convoluted surfaces, their methods of patch alignment (and thus motion estimation) are susceptible to local minima and may not converge correctly for large interframe motions.

Whether they use point correspondences or surface patch correspondences, all the aforementioned methods for use on range data are alike in that they assume a single rigid body motion, such as that which occurs when the camera moves within a static environment. One obvious way to apply these methods to tracking articulated motion would be first to partition the initial input frame into rigid subsets using some separate segmentation algorithm, and thereafter to use rigid-body methods to track each of those subparts independently. For methods using range data with known pointwise correspondences, this approach would suffice because given a correct initial segmentation, the *a priori* correspondences would track the segmentation perfectly.

However, rigid-body methods that are not provided with *a priori* correspondences are generally not easily adapted for estimating or tracking articulated motions. The aforementioned approach of initially segmenting then independently tracking subparts would usually not be effective, because an initially correct segmentation would still need to be maintained over time. Without maintenance, small errors in segmentation would lead to less accurate localization, and small errors in localization could lead to less accurate segmentation; independently tracking connected parts would ignore the very useful information provided by their interdependencies.

Furthermore, although some work on automatic model construction from range images [5] or otherwise [11] is explicitly designed for articulated objects, as they completely recalculate the articulation structure of the target with each image frame, the efficiency of such methods for tracking objects with constant and known articulation structure would be questionable.

This paper describes a simple algorithm for tracking the pose of articulated objects that attempts to address each of these issues. The proposed method directly uses the geometric information in range data. It does not require the predetermination of features or their correspondences, instead simultaneously performing target segmentation and localization. It is especially designed for articulated targets, and explicitly incorporates the joint attachment constraints to help track the positions and boundaries of the target's subparts, leading to greater robustness.

3. Algorithm

The proposed method estimates the sequence of poses of a target from a range image sequence[1] thereof, given a model of the target as well as its initial conditions. The range image sequence should be dense in both space and time, for example on the order of digital video camera resolutions and frame rates. The proposed method tracks the target by incrementally estimating its pose at each frame; that is, it does sequential state estimation.

State estimation can be thought of as trying to find a set of parameters for a given model that best accounts for a set of observations without exceeding too far the constraints of what is expected to be possible. Formally, it is sought to find the most probable sequence of states given a set of observations and a set of expectations; that is, to find the sequence of states with the maximum *a posteriori* probability. Without accurate probablistic models on which to base them, however, such statistical calculations are of questionable usefulness, and much simpler heuristics can often suffice.

Informally, the intuition of state estimation is to find the parameters that maximize the agreement between the model and the observations. However, because real image data is subject to noise and occlusions, the observations will generally have some erroneous or missing elements. In order to alleviate the effects of these imperfections on the results, this idea of maximizing the agreement is refined by also considering *a priori* expectations, such as smooth motions or bounded accelerations or velocities. Thus the idea is generalized to minimizing some residual, where the residual takes into account the agreement or correlation of the proposed state with the observations, as well as some measure of the unexpectedness or innovation of the

[1] Throughout this paper, the terms "range" and "depth" will be used interchangeably.

proposed state based on past history. That is, we take

$$\hat{\Phi}_f = \arg\widetilde{\min_{\Phi}} \text{ residual}(\Phi; \hat{\Phi}^{(f-1)}, I_f)$$

where $\hat{\Phi}_f$ is the estimated state for the current frame f, $\hat{\Phi}^{(f-1)}$ is the accumulated history of past estimated states, I_f is frame f of the observed image sequence, and the approximate minimization is described in Section 3.4; and where

$$\text{residual}(\Phi; \hat{\Phi}^{(f-1)}, I_f) = \text{innovation}(\Phi, \hat{\Phi}^{(f-1)}) - \text{correlation}(\Phi, I_f)$$

expresses the decomposition of the residual into dependencies on the past and on the present.

The remainder of this section describes the computation and optimization of this residual in more detail. The model of the visible surface of the target is a set of planar patches, each bounded by the convex hull of two circles. The correlation between model and observation is expressed in terms of the correspondence of individual pixels to each articulated subpart. The innovation of a state is based on the hypothesized changes in the subparts' sizes. The minimization of the residual function is simplified by instead minimizing a lower-dimensional projection of the residual function.

3.1. Model

The target is modeled by a set of connected planar patches, each in the shape of the convex hull of two circles. The radius and 3D location of each of those circles are variable parameters which the proposed method estimates, but the connectivity of the patches is fixed and must be provided by the user. This articulation model can be expressed as a graph, in which edges correspond to body segments and non-leaf nodes correspond to joints.

The visible surface of each articulated subpart of the target is modeled by a single patch, as shown in Figure 1. Each surface patch $S_k = S_{(ij)}$ is defined by the two nodes $\mathbf{n}_i, \mathbf{n}_j$ at its ends; each node can be associated with one or more patches (those associated with more than one correspond to joints). Each node $\mathbf{n}_i$ is fully specified by four scalar values, (x_i, y_i, z_i, r_i), which specify its location and size. Given these values for two adjacent nodes $\mathbf{n}_i, \mathbf{n}_j$, the connecting model patch $S_{(ij)}$ is a region of a plane with range map $R_{(ij)}$ that passes through the two points (x_i, y_i, z_i) and (x_j, y_j, z_j) but that is otherwise maximally parallel to the image plane, with boundary such that its projection $P_{(ij)}$ onto the image plane is the convex hull of the two circles defined by (x_i, y_i, r_i) and (x_j, y_j, r_j).

Thus for each segment of the target there corresponds a windowed depth map; the set of these maps over all segments S_k, where k ranges over those pairs (i, j) for which a segment exists, forms the complete prediction against which the observations must be compared.

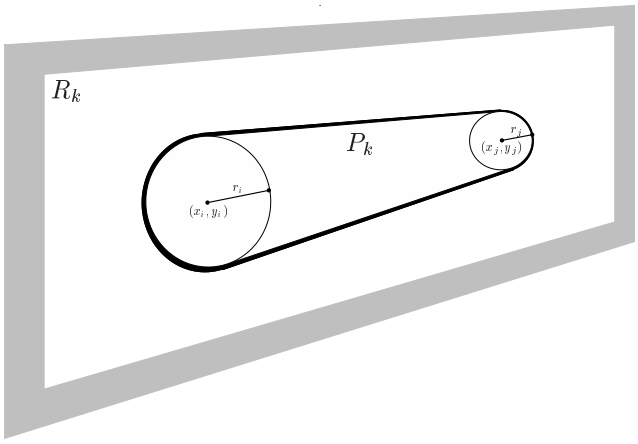

Figure 1. Each segment of the target is modeled by a planar patch.

3.2. Correlation

Given a hypothesized state, the correlation measures the amount of agreement between the prediction and the observation, and is qualitatively similar to log likelihood ratios in that positive values indicate supporting evidence, negative values indicate refuting evidence, and zero values indicate neutral evidence. The calculation of the correlation is based on the idea that given a hypothesized surface of infinite extent, pixels can be *observed* to be either far from or close to it, and given a hypothesized boundary thereof, pixels can be *expected* to be far from or close to it. If being far or close are represented by numbers between $+1$ and -1, respectively, then the product of observed and expected farnesses can be taken to be a measure of the conformity between a given pixel and surface. This measure can then be summed over all pairings of a pixel with a surface to evaluate the appropriateness of a hypothesized set of surfaces. That is,

$$\text{correlation}(\Phi, I) = \sum_k \sum_u \text{observe\_far}(u; R_k, I) \, \text{expect\_far}(u; P_k)$$

where k sums over all segments and u sums over all pixels.

For each segment S_k and pixel u, the observed farness between them depends on the difference between the range $I(u)$ observed in the input and the range $R_k(u)$ predicted by that segment. Because there will be errors in both the measurement and the model, the magnitude of this difference is then compared against a finite threshold; pixels

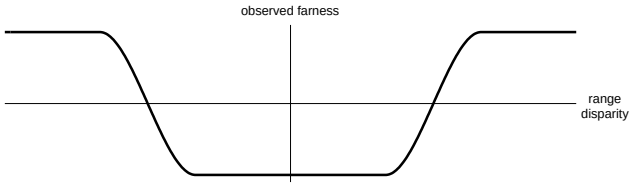

Figure 2. $\rho(\cdot)$ function for observed farness

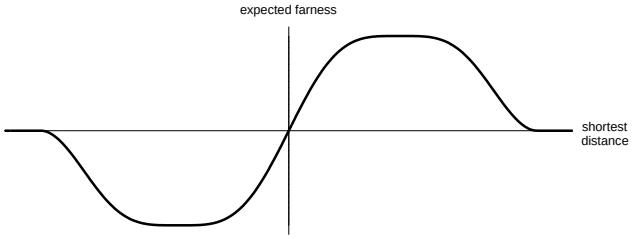

Figure 3. $\psi(\cdot)$ function for expected farness

falling outside or inside of this threshold are respectively declared far from or close to the surface. That is,

$$\text{observe\_far}(u; R_k, I) = \rho\big(R_k(u) - I(u)\big)$$

where $\rho(\cdot)$ is shown in Figure 2.

The expected farness between a pixel u and a segment S_k depends on the position (x, y) of u relative to the 2D projection P_k of S_k. A true segment boundary would typically have pixels of the segment inside the boundary, and pixels of something else outside the boundary. Thus the depth of pixels just inside the boundary of the predicted support map should fit the segment model well, and the depth of those just outside the boundary should fit the extrapolated segment model poorly. Thus,

$$\text{expect\_far}(u; P_k) = \psi\big(u \ominus P_k\big)$$

where $u \ominus P_k$ is the minimum distance between the point u and its nearest neighbor in the region P_k, and $\psi(\cdot)$ is shown in Figure 3.

Thus the correlation between hypothesized state and input is

$$\text{correlation}(\Phi, I) = \sum_k \sum_u \rho\big(R_k(u) - I(u)\big)\ \psi\big(P_k \ominus u\big)$$

3.3. Innovation

In addition to maximizing the correlation between the state and the input, optimizing the residual should also minimize the unexpectedness or innovation of the state. Although the target is expected to move, it can generally be expected not to change size quickly or significantly. Because of this, the proposed method penalizes hypotheses in which it does. Specifically, the subparts of the target are assumed to be of roughly constant shape and size. Then at every time step, the estimated radius of each node is biased towards its expected radius, by taking the innovation to be

$$\text{innovation}(\Phi, \hat{\Phi}^{(f-1)}) = \lambda \sum_k \big(r_k - \bar{r}_k^{(f)}\big)^2$$

where $\bar{r}_k^{(f)}$ represents the expected radius of each circle, is a function of the past estimated states $\hat{\Phi}^{(f-1)}$, and is recursively defined as

$$\bar{r}_k^{(f)} = (1 - \alpha)\, \bar{r}_k^{(f-1)} + \alpha\, \hat{r}_k^{(f-1)}$$

where α is a small constant. Segment lengths are not considered because to do so properly would require the calibration of the unit of range measurement with respect to pixel size, and this information is not always available.

3.4. Minimization

Formulating a good objective function is important, but so is being able to optimize it. The residual described above is a non-convex function of $4n$ variables (where n is the number of nodes), which is not trivial to minimize for interesting n. The proposed method alleviates this problem by explicitly solving for some of the variables in terms of the others, thereby reducing the dimensionality of the iterative portion of the optimization problem.

If the 2D support map of a segment is known perfectly, and if the surface depth model is exact and linear, then the correct set of depth parameters can be found directly, for example via least squares. Accordingly, the proposed method performs a least squares fit of a plane to the observed depth data for any hypothesized support map. Specifically, it estimates the set of node depths $\{z_i\}$ as a group from the set of 2D parameters $\{x_i, y_i, r_i\}$, by choosing $\{R_k\}$ to minimize

$$\text{error}(\{R_k\}; \{P_k\}, I) = \sum_k \sum_{u \in P_k} \big(R_k(u) - I(u)\big)^2$$

for any given $\{P_k\}$ and I. This eliminates one out of every four dimensions of the optimization problem. Note, however, that this only approximately minimizes the residual as formulated earlier in this section.

Even after this reduction in dimension, the non-convexity of the residual function still makes it difficult to minimize. However, particularly because of the real-time nature of the intended input, there is likely to be a high degree of temporal coherence in the motion of the target. This coherence can be used to hasten convergence of the minimizer by providing it with an intelligent initial guess,

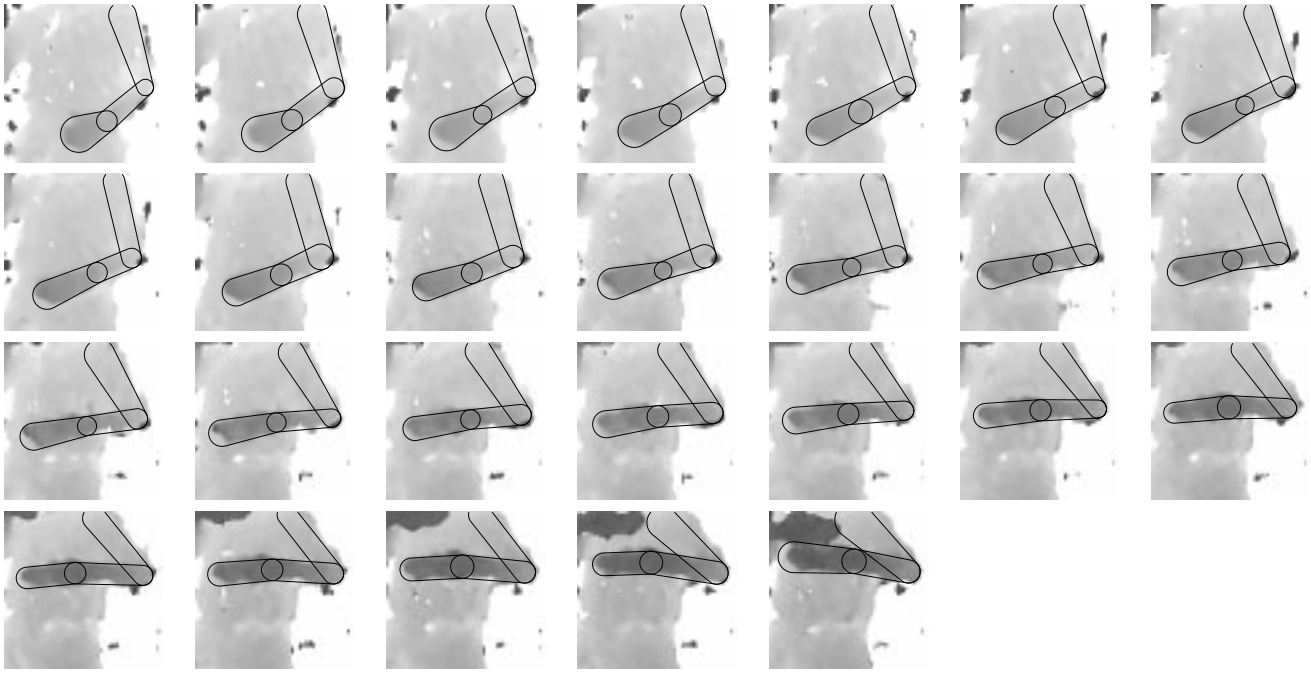

Figure 4. These images of the experimental results show the estimated target pose overlaid on the input range image sequence. Darker shades of gray represent smaller distances, and outlined regions represent the estimated positions of the hand, forearm, and upper arm.

and by limiting its search in parameter space to a neighborhood thereof. The proposed algorithm initializes the search at each time step with the estimated state from the previous time step.

4. Implementation and Results

The proposed algorithm was implemented in Matlab. Residual minimization was done using the `fminu` function in the Optimization Toolbox; this function implements the BFGS (Broyden-Fletcher-Golfarb-Shanno) quasi-Newton method with a mixed quadratic and cubic line search procedure. Derivatives of the residual function were calculated using finite differences instead of analytically.

The method was applied to track a human hand, forearm, and upper arm in a range image sequence derived from real stereo footage. Because the model of such a three-segment structure would expect the shoulder as well as the hand to be an end of the articulated chain, whereas in actuality the shoulder is connected to the torso, the expected radius $\bar{r}_m^{(f)}$ of the shoulder used in the innovation term was modified to encourage the estimated shoulder size to be similar to the elbow size, to compensate for the lack of an depth boundary on the torso side of the shoulder. Specifically,

$$\bar{r}_m^{(f)} = (1-\alpha)\,\bar{r}_m^{(f-1)} + \alpha(1-\beta)\,\hat{r}_{m'}^{(f-1)} + \alpha\beta\,\hat{r}_m^{(f-1)}$$

where β is a small constant, and m and m' refer to the shoulder and elbow nodes, respectively.

Figure 4 shows results for a video-rate image sequence of a real human. The range images were produced by recording the video streams from a pair of synchronized digital cameras, then processing each stereo pair off-line using the census transform [20] with subpixel interpolation to produce depth data. The clipping window of the sequence shown in Figure 4 has a resolution of 120×120, and the frame rate is the standard video rate. Running times were on the order of 30 seconds per frame on a 300 MHz Pentium II computer, using unoptimized Matlab code throughout. Initialization of the tracker was done by hand.

5. Discussion and Future Work

The images of the experimental results suggest that the proposed method works reasonably well. Moreover, although the current implementation is rather slow, if the method were implemented in a compiled language such as C or Fortran, and if a more suitable function optimization routine were used (for example a derivative-free, trust-region method), it should be significantly faster.

As described, this work makes many simplifying assumptions, the most inaccurate of which is perhaps that of

planar surfaces. However, this simplification is not intrinsic to the method, and in fact can readily be generalized to many classes of curved surfaces. As long as the depth maps of the set of modeled surfaces are representable by linear combinations of the depth maps of a set of basis surfaces, the same method of least-squares surface fitting will suffice to estimate the surface parameters. The main disadvantage of a more general surface model would be an increased difficulty in minimizing the residual. It would be worthwhile to further investigate the tradeoffs involved in choosing the complexity of the model.

The present method also does not handle surface projection boundaries of any shape other than the convex hull of two circles, nor does in handle occlusions. Both of these limitations could be addressed by using an iteratively-reweighted least squares estimation of the surface depth parameters. This would be another area for further work that although technically involved, should conceptually be fairly straightforward.

6. Acknowledgements

The author would like to thank Interval Research Corporation for its support of this work, and Harlyn Baker and John Woodfill in particular for their invaluable ideas, discussions, and feedback. The author would also like to thank Carlo Tomasi for his many helpful comments and suggestions.

References

[1] J. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Articulated and elastic non-rigid motion: A review. In *Proceedings of the 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 2–14, 1994.

[2] J. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images—a review. *Proceedings of the IEEE*, 76(8):917–935, Aug. 1988.

[3] J. Aloimonos and I. Rigoutsos. Determining the 3-d motion of a rigid planar patch without correspondence, under perspective projection. In *Proceedings of the IEEE Computer Society Workshop ond Motion: Representation and Analysis*, pages 167–174, May 1986.

[4] K. Arun, T. Huang, and S. Blostein. Least squares fitting of two 3-d point sets. *IEEE Transactions on PAMI*, 9:698–700, 1987.

[5] A. Ashbrook et al. Segmentation of range data into rigid subsets using surface patches. In *Proceedings of the ICCV*, pages 201–206, 1998.

[6] C. Bregler and J. Malik. Video motion capture. Technical Report UCB//CSD-97-973, University of California, Berkeley, 1997.

[7] B. Horn and J. Harris. Rigid body motion from range image sequences. *CVGIP*, 53(1):1–13, Jan. 1991.

[8] T. Huang and A. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 1994.

[9] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A video-rate stereo machine and its new applications. In *Proceedings of CVPR*, 1996.

[10] N. Kehtarnavaz and S. Mohan. A framework for estimation of motion parameters from range images. *CVGIP*, 45(1):88–105, Jan. 1989.

[11] I. Kompatsiaris et al. 3-d model-based segmentation of videoconference image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):547–561, Sept. 1998.

[12] K. Konolige. Small vision systems: Hardware and implementation. In *Eighth International Symposium on Robotics Research*, Nov. 1997.

[13] Z. Lin, T. Huang, S. Blostein, H. Lee, and E. Margerum. Motion estimation from 3-d point sets with and without correspondences. In *Proceedings of CVPR*, pages 194–201, 1986.

[14] J. Rehg and T. Kanade. Visual tracking of self-occluding articulated objects. Technical Report CMU-CS-94-224, Carnegie Mellon University, 1994.

[15] B. Sabata and J. Aggarwal. Estimation of motion from a pair of range images: A review. *CVGIP*, 54(3):309–324, Nov. 1991.

[16] B. Sabata and J. Aggarwal. Surface correspondence and motion computation from a pair of range images. *CVIU*, 63(2):232–250, Mar. 1996.

[17] B. Sabata, F. Arman, and J. Aggarwal. Segmentation of range images using pyramidal data structures. In *Proceedings of the ICCV*, pages 662–666, 1990.

[18] B. Vemuri and G. Skofteland. Invariant surface and motion estimation from sparse range data. *JMIV*, 1:43–64, 1992.

[19] J. Woodfill and B. Von Herzen. Real-time stereo vision on the parts reconfigurable computer. In *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 242–250, Apr. 1997.

[20] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the ECCV*, pages 151–158, 1994.