Stereo Tracking using ICP and Normal Flow Constraint

Louis-Philippe Morency Trevor Darrell
MIT AI Lab
Cambridge, MA, 02139, USA
{lmorency, trevor} @ai.mit.edu

Abstract

This paper presents a new approach for 3D view registration of stereo images. We introduce a hybrid error function which combines constraints from the ICP (Iterative Closest Point) algorithm and normal flow constraint. This new technique is more precise for small movements and noisy depth than ICP alone, and more robust for large movements than the normal flow constraint alone. Finally, we present experiments which test the accuracy of our approach on sequences of real and synthetic stereo images.

1. Introduction

The problem of estimating 3D rigid body motion has been studied extensively in the computer vision and graphics fields. The well-known Iterative Closest Point (ICP) algorithm, introduced by Chen and Medioni [4] and Besl and McKay [2], has been used extensively in the graphics literature to merge 3D laser range scans. In the vision literature much progress has been made on gradient-based parametric motion estimation techniques which aggregate pointwise normal flow constraints [3, 9, 11].

ICP finds corresponding points between two 3D point clouds and tries to minimize the error (usually the euclidian distance) between the matched points. Chen and Medioni minimize this error based on a point-to-plane distance, while Besl and McKay minimize the direct euclidian distance between the matched points (point-to-point). Rusinkiewicz and Levoy [16] present a extensive survey of many variants of ICP. Godin *et al.*[7] first used color to filter matched points during ICP. While other methods [6, 17] have incorporated color information in the distance function of the matching process, no solution has been suggested that uses color/brightness during the error minimization process.

The normal flow is 3D vector field which can be defined as the component of the 2D optical flow that is in the direction of the image gradient[20]. When 3D observations are directly available, such as from optical stereo or laser range finders, a normal flow constraint can be expressed directly to estimate rigid body motion [19]. Harville *et al.*[8] combined normal flow constraint with a depth gradient constraints to track rigid motion. Gradient-based approaches use color/brightness information during the minimization process and have proved to be accurate for subpixel movements[1].

In this paper, we present an integrated tracking approach which jointly aligns images using a normal flow gradient constraint and an ICP algorithm. This new framework has the precision of the former with the robustness of the latter. To date, most ICP algorithms have been tested on very precise 3D data sets from a laser scanners [14] or other range scanning methods. Approaches based on depth gradients usually presume high-frame rate stereo observations (e.g., [8]). We are interested in tracking data from relatively noisy optical stereo range data at modest frame rates.

The following section describes the iterative framework used for 3D view registration. Section 3 presents the closest point matching process and point-to-plane error function, two important components of ICP. Section 4 reviews the normal flow constraint and shows how inverse calibration parameters can be used to do find correspondence. Then, section 5 describes the hybrid error functions. Finally, in section 6, we show how this integrated approach can reliably track sequences from optical stereo data that neither technique alone could track.

2. Integrated View Registration Framework

In our new framework, we integrate an ICP 3D euclidian error function with a normal flow constraint, creating a hybrid registration error metric yielding a tracker which is both robust and precise. The ICP approach matches points in 4 dimensions (3D + brightness) and minimizes the euclidian distance between corresponding points. Empirically, we have found that ICP robustly handles coarse motion. The NFC (Normal Flow Constraint) approach matches points based on the inverse calibration parameters and find

the transformation between corresponding points based on their appearance and their 3D position. As shown in Section 5, this method is more precise for small movement since it searches the pose parameter space using a gradient method which can give sub-pixel accuracy.

2.1. Preprocessing and Pose Update

Our tracker takes two image sets as input: the new image set $\{I_t, Z_t\}$ grabbed at time t and the reference image set $\{I_r, Z_r\}$. The reference image set can be either the image set grabbed at time t-1, the first image set, or any relevant image set between time 0 and time t-1 [15].

The new image set $\{I_t, Z_t\}$ is preprocessed in concert with known camera calibration information to obtain the 3D vertex set Ψ_t of $i := 1..m$ vertices $\vec{v}_{ti} = \{\vec{p}_{ti}, \vec{n}_{ti}, I_{ti}\}$ where $\vec{p}_{ti}$ is the 3D point coordinates in the camera reference, $\vec{n}_{ti}$ is the normal vector of the surface projected by Z_t at point $\vec{p}_{ti}$ and I_{ti} is the brightness value of the point $\vec{p}_{ti}$ as specified by the intensity image I_t. The normal vector $\vec{n}_{ti}$ is computed from the depth image gradients:

$$\vec{n}_{ti} = \begin{bmatrix} \frac{\partial Z_t}{\partial u_{ti}} & \frac{\partial Z_t}{\partial v_{ti}} & 1 \end{bmatrix} \quad (1)$$

where u_{ri} and v_{ri} are the 2D image coordinates of Z_t.

The goal of the tracker is to find the rigid pose change $\{\mathbf{R}, \vec{\mathbf{t}}\}$ between the two image sets, where $\mathbf{R}$ is a 3x3 rotation matrix and $\vec{t}$ is a 3D translation vector . At each iteration, a transformation $\vec{\delta}$ represented by 6 parameters vector $[\ \vec{\omega} \quad \vec{t}\ ]^t$ is computed. In this vector, $\vec{\omega}$ is the instantaneous rotation (3 parameters) and $\vec{t}$ is the translation (3 parameters). The current pose estimation is updated as follow:

$$\mathbf{R^{k+1}} = \mathbf{R^k R}^{(\delta)} \quad (2)$$

$$\vec{t}^{\,k+1} = \vec{t}^{\,k} + \vec{t}^{\,(\delta)} \quad (3)$$

where k is the iteration number and $\mathbf{R}^{(\delta)}$ is the 3x3 matrix representing the rotation $\omega^{(\delta)}$. Initially, $\mathbf{R^0}$ is set to the identity matrix and $\vec{t}^{\,0}$ is set to 0.

2.2. Hybrid Tracker

As shown in figure 1, our hybrid tracker iterates a joint error minimization process until convergence. At each iteration two error function are minimized in the same linear system. The iteration process can be divided into 5 distinct steps: Match, Error Function, Minimization, Warping and Convergence check.

- The Match stage finds corresponding points between the 3D image sets. In the hybrid tracker we use two matching techniques: closest point and inverse calibration. These techniques are described in more details in sections 3.1 and 4.1.

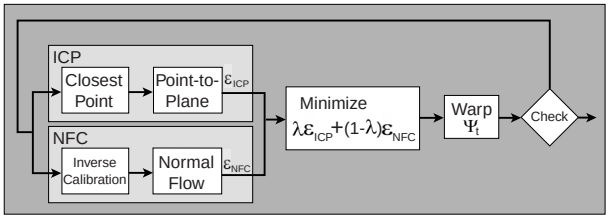

Figure 1. Hybrid tracker structure.

- Given the two sets of correspondences, we compute two error functions: point-to-plane and normal flow constraint. These two error functions relate the corresponding point sets to the pose parameters. As shown in Section 3.2 and 4.2, each error function can be locally approximated as linear problems in terms of the motion parameters:

$$\varepsilon_{ICP} = \|\mathbf{A_{ICP}}\vec{\delta} - \vec{b}_{ICP}\|^2 \quad (4)$$

$$\varepsilon_{NFC} = \|\mathbf{A_{NFC}}\vec{\delta} - \vec{b}_{NFC}\|^2 \quad (5)$$

- The Minimization stage estimates the optimal transformation $\vec{\delta}^*$ between the matched points using the combined error function:

$$\vec{\delta}^* = \arg\min_{\vec{\delta}}[\lambda(\overline{d})\varepsilon_{ICP} + (1 - \lambda(\overline{d}))\varepsilon_{NFC}] \quad (6)$$

where $\overline{d}$ is the average distance between matched points and $\lambda(\overline{d})$ is a sigmoid function which arbitrates the importance of the ICP error function over the normal flow error function as alignment improves (see figure 2). Section 5 discusses in more details how the sigmoid function $\lambda(\overline{d})$ is computed.

- The Warping stage warps the 3D vertex set Ψ_t according to the new estimated transformation $\vec{\delta}$. The warping is done by updating the $\vec{p}_{ti}$ and $\vec{n}_{ti}$ of each vertex as follows:

$$\vec{n}_{ti}{}' = \mathbf{R}^{(\delta)}\vec{n}_{ti} \quad \vec{p}_{ti}{}' = \mathbf{R}^{(\delta)}\vec{p}_{ti} + \vec{t}^{\,(\delta)} \quad (7)$$

- The Convergence Check stage computes the convergence factor ϵ by averaging the distance D between warped 3D points $\vec{p}_{ti}{}'$ and referential 3D points $\vec{q}_{ri}$:

$$\epsilon = \frac{1}{n}\left(\sum_{i=1}^{n} D(\vec{p}_{ti}{}', \vec{q}_{ri})\right) \quad (8)$$

If the difference between the convergence factor ϵ of two consecutive iterations is smaller then a threshold

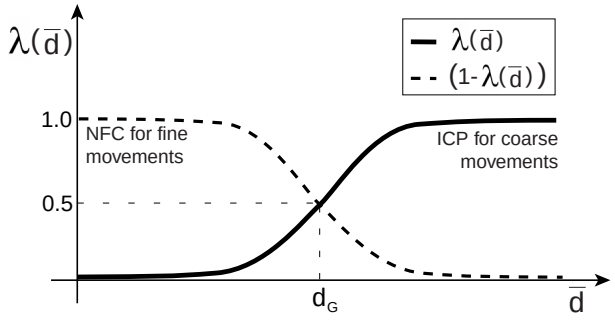

Figure 2. Plot of the sigmoidal function $\lambda(\overline{d})$ used in equation 6. Notice that as the average distance between matched points $\overline{d}$ decrease, NFC error function has more weight, and vice-versa.

value τ, then convergence is reached. The 3D view registration is completed when convergence is reached or, in the case of non-convergence, when a maximum number N_I of iterations is performed.

3. ICP Error Function

To compute the ICP error function, the matching stage searches for closest points in a 4-dimensional space composed of the 3D euclidian space and 1D for brightness. An exhaustive search for matching closest points makes large displacements easier to track. A k-d tree is used to accelerate the matching process [18]. As suggested by Rusinkiewicz and Levoy [16], we use a point-to-plane error function to align the matched points.

3.1. Closest Point with k-d Tree

Among the earliest ICP distance functions proposed was the 3D euclidian distance [2]. This function doesn't take into account color or intensity information which may be available. As Godin *et al.*[7], we take advantage of intensity information and use a 4D space (X,Y,Z,E) where E is the brightness value from a intensity image I_r. When I_r is a color image, Godin *et al.* [7] suggests using the hue channel as the brightness measure.

To accelerate the matching process we use a k-d tree and an Approximate Nearest Neighbor algorithm [12]. The k-d tree is created with the values $\{\vec{x}_r, \vec{y}_r, \vec{z}_r, \vec{I}_r\}$ of the referencial image set. The same k-d tree is used throughout all the iterations. The matching process finds, for each vertices $\vec{v}_{ti}$ of the current 3D vertex set Ψ_t, the closest node of the k-d tree $\{x_{ri}, y_{ri}, z_{ri}, I_{ri}\}$ that minimizes the 4D distance function:

$$\|\vec{q}_{ri} - \vec{p}_{ti}\| + k\|I_{ri} - I_{ti}\| \tag{9}$$

where k is a constant to normalize the brightness value.

3.2. Point-to-Plane

The point-to-plane method [4] minimizes the distance between a point $\vec{q}_{ri}$ and the tangential plane of the corresponding point $\vec{p}_{ti}$:

$$D_{Plane}(\vec{q}_{ri}, \vec{p}_{ti}) = \vec{n}_{ti}(\vec{q}_{ri} - (\mathbf{R}\vec{p}_{ti} - \vec{t})) \tag{10}$$

By approximating the rotation $\mathbf{R}$ with an instantaneous rotation ω and rearranging the equation 10 adequately, we obtain the following linear system:

$$\varepsilon_{ICP} = \|\mathbf{A_{ICP}}\vec{\delta} - \vec{b}_{ICP}\|^2 \tag{11}$$

where each line is defined as follow

$$\vec{A}_i = \begin{pmatrix} \vec{n}_{ti} \times \vec{q}_{ri} \\ -\vec{n}_{ti} \end{pmatrix} \tag{12}$$

$$b_i = \vec{n}_{ti} \cdot (\vec{p}_{ti} - \vec{q}_{ri}) \tag{13}$$

Compared with the point-to-point method [2], the point-to-plane converges faster but requires extra preprocessing to compute the normals (see [16] for more details).

4. NFC Error Function

The normal flow constraint is a gradient-based approach which can estimate sub-pixel movements accurately. During the matching stage, we use an inverse calibration method to find corresponding points which belong on the same projective ray. This provides the correspondence needed to compute the temporal gradient term of the normal flow constraint.

4.1. Inverse Calibration

The inverse calibration approach [13] searches for corresponding points of $\vec{p}_{ti}$ by projecting the 3D point from the 3D coordinate system of Υ_t to the referential depth image Z_r coordinate system:

$$\begin{bmatrix} \vec{u}_{ri} \\ 1 \end{bmatrix} = \mathbf{C} \begin{bmatrix} \vec{p}_{ti} \\ 1 \end{bmatrix} \tag{14}$$

where $\mathbf{C}$ is a 3x4 projection matrix that relate 3D coordinate system of $\vec{p}_{ti}$ to the 2D image coordinate $\vec{u}_{ri} = [\ u_{ri} \quad v_{ri}\ ]$. This matrix is based on the stereo camera or laser scanner parameters.

After projection, two match functions could be used: 1) interpolate the 3D coordinates $\vec{q}_{ri}$ of the corresponding point from the projection value $\vec{u}_{ri}$, or 2) search around the projected point $\vec{u}_{ri}$ in the Z_r image to find the closest point.

We used the first method to be compatible with the time gradient term of the normal flow constraint which assumes that the corresponding points are on the same projective ray.

The 3D coordinates $\vec{q}_{ri} = [\ x_{ri} \quad y_{ri} \quad z_{ri}\ ]$ are interpolated from the depth image Z_r as follows:

$$z_{ri} = Z_r(\vec{u}_{ri}) \quad , \quad x_{ri} = f\frac{u_{ri}}{z_{ri}} \quad , \quad y_{ri} = f\frac{v_{ri}}{z_{ri}} \tag{15}$$

4.2. Normal Flow Constraint

Given 3D input data, the normal flow is the component of the optical flow in the direction of the image gradient. As shown in [20], the normal flow can be expressed as:

$$-\frac{\partial I_{ri}}{\partial t} = \nabla I_{ri}\left[\frac{\partial \vec{u}_{ri}}{\partial \vec{q}_{ri}}\right]\vec{V} \tag{16}$$

where $\nabla I_{ri} = \left[\ \frac{\partial I_{ri}}{\partial u_{ri}} \quad \frac{\partial I_{ri}}{\partial v_{ri}}\ \right]$ is the image gradient, $\vec{V} = \left[\ \frac{\partial x_{ri}}{\partial t} \quad \frac{\partial y_{ri}}{\partial t} \quad \frac{\partial z_{ri}}{\partial t}\ \right]$ is the velocity of the object and $\frac{\partial I_{ri}}{\partial t}$ is the time gradient. $\frac{\partial I_{ri}}{\partial u_{ri}}$ and $\frac{\partial I_{ri}}{\partial v_{ri}}$ are computed directly from the referential image I_r. The time gradient is approximated by:

$$\frac{\partial I_{ri}}{\partial t} = I_{ti} - I_{ri} \tag{17}$$

For a perspective projection where $u_{ri} = f\frac{x_{ri}}{z_{ri}}$ and $v_{ri} = f\frac{y_{ri}}{z_{ri}}$, we can find the Jacobian matrix:

$$\frac{\partial \vec{u}_{ri}}{\partial \vec{q}_{ri}} = \begin{bmatrix} \frac{f}{z_{ri}} & 0 & -f\frac{x_{ri}}{z_{ri}^2} \\ 0 & \frac{f}{z_{ri}} & -f\frac{y_{ri}}{z_{ri}^2} \end{bmatrix} \tag{18}$$

Since the object is rigid, the velocity V can be expressed as:

$$\vec{V} = [\ \mathbf{I} \quad -\hat{q}_{ri}\ ]\,\vec{\delta} \tag{19}$$

where $\mathbf{I}$ is a 3x3 identity matrix and $\hat{q}_{ri}$ is the skew matrix of the vector $\vec{q}_{ri}$. By rearranging the equation, we get a linear system similar to the point-to-plane technique (section 3.2):

$$\varepsilon_{NFC} = \|\mathbf{A_{NFC}}\vec{\delta} - \vec{b}_{NFC}\|^2 \tag{20}$$

where each line is defined as follow

$$\vec{A}_i = \nabla I_{ri}\left[\frac{\partial \vec{u}_{ri}}{\partial \vec{q}_{ri}}\right][\ \mathbf{I} \quad -\hat{q}_{ri}\ ] \tag{21}$$

$$b_i = -\frac{\partial I_{ri}}{\partial t} \tag{22}$$

4.3. Accuracy comparison for small movements

We compared the performance of NFC and ICP sequential tracking approach on sequences with small movements. The top part of figure 3 shows the first and the last frame of a 31 synthetic frame sequence. A rotation of 0.5 degrees occurred between each consecutive frames. Since the sequence is synthetic, we could compare the result of each tracker with the real transformation (0.5 degrees). The average error was computed by warping the referential image by the found transformation and the real transformation and computing the average distance between the two sets of 3D points. The average error for normal flow constraint was 0.898mm, better then the ICP with 2.06mm . The graph in figure 3 presents the average error at each frame.

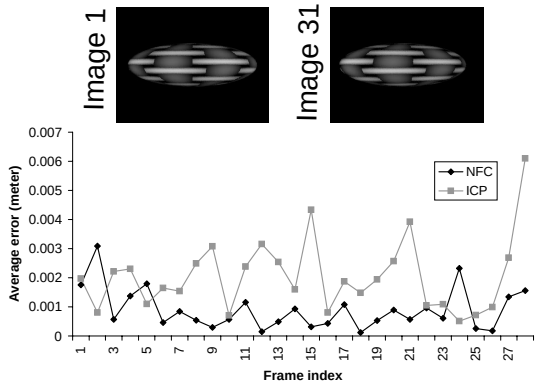

Figure 3. Small rotation sequence with synthetic images.

5. Hybrid Error Function

At each iteration, the tracking algorithm minimize the hybrid error function to find the optimal pose parameters δ^*. We can rewrite equation 6 as one linear system:

$$\arg\min_{\vec{\delta}} \left\| \begin{bmatrix} \lambda(\overline{d})\mathbf{A_{ICP}} \\ (1-\lambda(\overline{d}))\mathbf{A_{NFC}} \end{bmatrix}\vec{\delta} - \begin{bmatrix} \lambda(\overline{d})\vec{b}_{ICP} \\ (1-\lambda(\overline{d}))\vec{b}_{NFC} \end{bmatrix} \right\|^2$$

This linear system can be solved using a least-squares method or any robust estimator. To reduce the influence of outliers, we use a M-estimator to minimize the system [10].

As shown in figure 3, the NFC error function is more accurate for the estimation small movement. Since the normal flow constraint approximate the pixel by a plane to compute the intensity gradients $\frac{\partial I_{ri}}{\partial u_{ri}}$ and $\frac{\partial I_{ri}}{\partial v_{ri}}$ of equation 16, its accuracy is directly related to the variance of the Gaussian d_G used to compute to compute these gradients. We want a function that increases the importance of NFC when the average distance $\overline{d}$ between matched points decreases, and vice versa. Figure 5 shows the sigmoid function that we use:

$$\lambda(\overline{d}) = \frac{1}{1 + e^{-c(\overline{d} - d_G)}} \tag{23}$$

where c is a constant that determine the slope of the sigmoid function and $\overline{d}$ is the average distance of matched points found during the closest point matching process (see Sec-

tion 3.1). We define the average distance as follow:

$$\overline{d} = \frac{1}{N} \sum_{i=1}^{N} D(\vec{q}_{ri}, \vec{p}_{ri}) \quad (24)$$

where N is the number of matched points and D is the euclidian distance between two points.

6. Results with Real Images: Face Tracking

We tested our hybrid tracker with sequences obtained from a stereo camera using the SRI Small Vision System [5]. Tracking was initiated automatically by using a face detector [21]. Without special optimizations, the hybrid sequential tracker can update poses based on observations of 2500 points per frame at 2Hz on a Pentium III 800MHz.

Figures 4 and 5 presents some key frames of two user moving in front of the camera. During the first sequence (180 frames), the user turned his head approximately 40 degrees down, up, left, and right. Then, the user translated his head 30cm, which was equivalent to 25 image pixels. During the second sequence (160 frames), the user turned his head left and right approximately 25 degrees and then translated his head left and right rapidly, three times.

In both figures, we observe that ICP alone performs well for translation, but has trouble with rotation. We observe the opposite results for NFC alone which handles rotation well, but translation poorly. The hybrid tracker is able to track all the sequence reliably. The figure 6 shows the average convergence factor of each registration technique. The convergence factor is computed as described in section 2.2. The three techniques converge in less then 3 iterations. The hybrid error function converge to an average distance error 20% smaller then ICP alone and 5% smaller then NFC alone. Movies of the above results can be found at http://www.ai.mit.edu/people/lmorency/ .

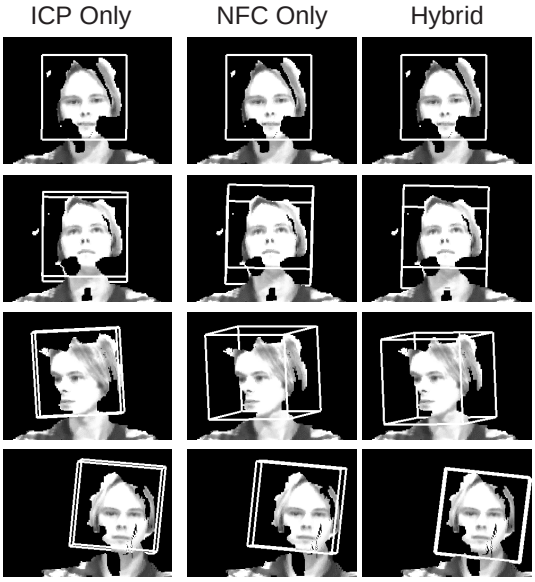

Figure 4. Face tracking results with stereo images. Each row represents tracking results at different frames: 0, 66, 140, and 180.

7. Conclusions

We presented a new hybrid 3D view registion framework for tracking 3D pose from noisy 3D stereo images. Our approach integrated the fine tracking ability of a gradient-based normal flow constraint with the robust coarse tracking ability of the ICP algorithm. The stability of our tracker was shown on synthetic sequences with known ground truth and on sequences grabbed from a low-cost stereo camera. Our results indicated that the hybrid approach outperformed either algorithm alone.

References

[1] J. Barron, D. Fllet, and S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994.

[2] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(2):239–256, February 1992.

[3] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *ICCV*, pages 374–381, 1995.

[4] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. In *Proc. of the IEEE Int. Conf. on Robotics and Authomation*, pages 2724–2728, 1991.

[5] V. Design. *MEGA-D Megapixel Digital Stereo Head*. http://www.ai.sri.com/ konolige/svs/, 2000.

[6] J. Feldmar and N. Ayache. affine and locally affine registration of free-form surfaces. *IJCV*, 18(2):99–119, 1996.

[7] G. Godin, M. Rioux, and R. Baribeau. Three-dimensional registration using range and intensity information. In *Proceedings of SPIE Videometric III*, volume 2350, pages 279–290, 1994.

[8] M. Harville, A. Rahimi, T. Darrell, G. Gordon, and J. Woodfill. 3d pose tracking with linear depth and brightness constraints. In *Proceedings of ICCV 99*, pages 206–213, Corfu, Greece, 1999.

[9] B. Horn and E. Weldon, Jr. Direct methods for recovering motion. *IJCV*, 2(1):51–76, June 1988.

[10] P. Huber. *Robust statistics*. Addison-Wesley, New York, 1981.

[11] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *In ICCV*, September 1999.

[12] D. M. Mount and S. Arya. *ANN: Library for Approximate Nearest Neighbor Searching*. http://www.cs.umd.edu, 1998.

[13] P. Neugebauer. Geometrical cloning of 3d objects via simultaneous rgistration of multiple range images. In *Proc.*

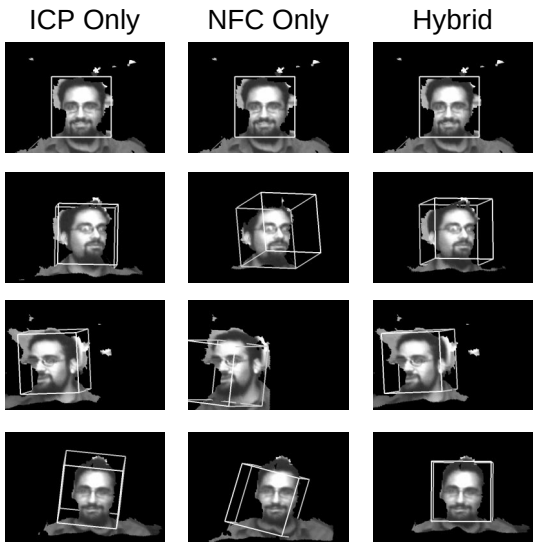

Figure 5. Face tracking results with stereo images. Each row represents tracking results at different frames: 0, 24, 100, and 158.

Int. Conf. Shape Modeling and Applications, pages 130–139, 1997.

[14] K. Pulli. Multiview registration for large data sets. In *Proc. 3DIM*, pages 160–168, 1999.

[15] A. Rahimi, L.-P. Morency, and T. Darrell. Reducing drift in parametric motion tracking. In *ICCV*, pages 315–322, June 2001.

[16] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proc. 3DIM*, pages 145–152, 2001.

[17] C. Schtz, T. Jost, and H. Hgli. Multi-featured matching algorithm for free-form 3d surface registration. In *ICPR*, pages 982–984, 1998.

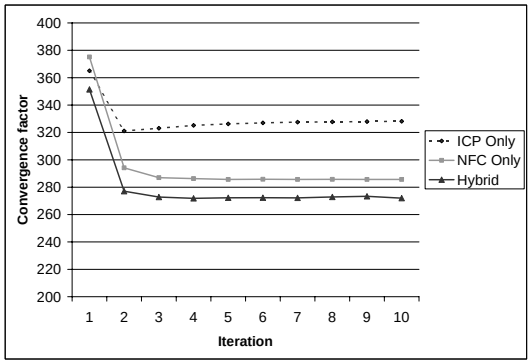

Figure 6. Comparison of average convergence factor for 80 frames (same sequence of figure 5).

[18] Simon. *Fast and Accurate Shape-Based Registration.* Ph.D. Dissertation, Carnegie Mellon University, 1996.

[19] G. Stein and A. Shashua. Direct estimation of motion and extended scene structure from moving stereo rig. In *Proc. of CVPR*, June 1998.

[20] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *ICCV (2)*, pages 722–729, 1999.

[21] P. Viola and M. Jones. Robust real-time face detection. In *ICCV01*, page II: 747, 2001.