ACCV2002: The 5th Asian Conference on Computer Vision, 23–25 January 2002, Melbourne, Australia

Curvature-Based Algorithms for Non-Rigid Motion and Correspondence Estimation*

Pavel Laskov
Fraunhofer Institut FIRST
Kekulestr. 7
12489 Berlin, Germany
laskov@first.gmd.de

Chandra Kambhamettu
Dept. of Computer and Information Sciences
University of Delaware
Newark, DE 19716, USA
chandra@cis.udel.edu
http://www.cis.udel.edu/∼vims

Abstract

Two new algorithms for estimation of non-rigid motion in range data in the absense of correspondence information are presented. We derive a new relationship between Gaussian curvatures and other differential-geometric parameters before and after small deformation. This relationship depends linearly on derivatives of a motion model, which provides a closed-form, least squares solution for motion estimation. The first algorithm built solely on the new relationship demonstrates significant improvement of motion and correspondence estimation accuracy on certain artificial shapes; however, its poor numerical conditioning results in higher error in the presense of inaccurate values of differential-geometric parameters. The second algorithm combines the Gaussian curvature relationship with the previously known relationship between unit normals before and after deformation. The combined algorithm achieves higher accuracy of motion and correspondence estimation.

1 Introduction

In this work we address the problem of non-rigid motion and correspondence estimation in range images in the absense of any prior information other than the images before and after motion. Even though this problem has attracted considerable attention in the recent decade, no robust method, suitable to a variety of applications, is currently known. Some well-known algorithms are suitable for particular applications, e. g. left ventricular motion tracking [1, 2, 3, 13], or cerebral cortical surface correspondence estimation [15]; however, they contain application-specific features that cannot be used for different problems. Other methods impose certain restrictions on the type of non-rigid motion they can handle. The examples of such methods are the seminal work of Chaudhuri and Chatterjee [4, 5], which present a solution for estimation of symmetric or positive-definite non-rigid deformations in the Helmholtz motion model, or the work on restricted classes of non-rigid motion [8, 9] applicable to homothetic or conformal motion.

Our objective is to develop algorithms that can work under significantly more stringent conditions. The specific requirements we impose are the following:

a. The algorithm must handle general elastic non-rigid motion.

b. Correspondence between points in images is assumed unknown.

c. Only the 3D shape data, but no other prior information, is available.

d. The algorithm must be able to estimate motion of *any* point in the image – not necessarily the points with some favorable features.

For this scenario the selection of available methods is rather sparse. We cannot use the physically-based modeling because it requires prior knowledge of the physical properties of objects. Neither can we use feature-based non-rigid registration methods, such as [6, 7, 14], because of their search for specific features. The most attractive algorithms under our assumptions are the two differential-geometric algorithms based on relationships between unit normals and Gaussian curvatures before and after deformation [10].

The algorithms developed in this paper are founded on the new relationship between Gaussian curva-

*Funding for this work was provided under NSF grants CAREER IRI-9984842 and CISE CDA-9703088.

tures[1]. Our interest in Gaussian curvature stems from the fact that this quantity is an intrinsic property of a surface. For this reason we believe that the change in Gaussian curvature is a good characterization of surface motion. Unlike the Gauss's equation used in [10], our new relationship is *linear* with respect to the motion model. Consequently, minimization of the squared violation of this relationship results in standard linear system of normal equations. This system has a unique closed-form solution, which constitutes a significant computational improvement over the solution based on non-linear optimization proposed in [10].

Furthermore, linearity of our error function allows it to be combined, in a weighted fashion, with the error function of the unit normal algorithm. The new hybrid unit normal/Gaussian curvature algorithm, proposed in this paper, combines the favourable properties of its constituents.

2 Background

We assume the reader is familiar with basic concepts of differential geometry. The parametric surface is denoted as $\mathbf{r}(u, v)$, its derivatives as $\mathbf{r}_u(u, v)$ etc. In the differential-geometric approach the motion function is assumed to be a function $\mathbf{s}(u, v)$ of *parameters* rather than that of coordinates of the point on a surface. Throughout this work we assume the motion model is affine:

$$\mathbf{s} = \mathbf{a}u + \mathbf{b}v. \tag{1}$$

The goal of the motion estimation algorithm is, given the surfaces $\mathbf{r}$ and $\mathbf{r}'$ before and after motion, to estimate the parameters of the motion function such that it transforms the surfaces one into another: $\mathbf{r}' = \mathbf{r} + \mathbf{s}$. Obviously, affine motion is not powerful enough to model complicated global motions, but it is a good enough estimate locally in a neighborhood of a point of interest. The algorithms presented in this paper can be also applied for higher-order motion models.

Since point correspondence between two range images is assumed unknown it must be estimated along with the motion. This is done by *hypothesizing* correspondence between the point of interest $p_0 \in \mathbf{r}$ and some $p'_j \in \mathbf{r}'$. For the fixed correspondence one can estimate the motion and compute the error $E(p, p'_j)$ of this motion model. By repeating this procedure for a number of candidate points p'_j one can select the correspondence that minimizes the motion estimation error.

[1]Due to space constraints only a schematic derivation is presented. The details will be provided in the forthcoming journal paper.

Within this generic motion/correspondence estimation framework the difference between individual algorithms lies in the estimation and evaluation of the motion model $\mathbf{s}$. Let E,F,G denote the coefficients of the first fundamental form, L, M, N the coefficients of the second fundamental form, $D = \sqrt{EG - F^2}$ the discriminant. Gaussian curvature can be expressed in terms of coefficients of the second fundamental form and the discriminant as:

$$K = \frac{LN - M^2}{D^2}, \tag{2}$$

or in terms of coefficients of the first fundamental form only as:

$$K = -\frac{1}{2D}\left[\left(\frac{G_u}{D}\right)_u + \left(\frac{E_v}{D}\right)_v\right]. \tag{3}$$

Unlike the previous Gaussian curvature algorithm, derived from Eq. (3), we will use Eq. (2) and will defer substitution of particular surface and motion models until later stages in the derivation. This will allow us to see the linear structure of the relationship, and will also make our algorithm extensible for other surface and motion models.

3 Derivation of the Algorithm

3.1 Preliminaries

Let $\mathbf{ab}$ denote an *open product* product between two vectors[2], which is a linear operator. The *gradient operator* of a vector-valued function represents the 3 directions of the greatest increase of each of the function's components along a given surface. Assuming that parametrization is orthogonal, i.e., $F = 0$ and $D^2 = EG$, the gradient of the motion function $\mathbf{s}$ is the following operator:

$$\nabla\mathbf{s} = \frac{1}{E}\mathbf{r}_u\mathbf{s}_u + \frac{1}{G}\mathbf{r}_v\mathbf{s}_v. \tag{4}$$

With the notation and definitions above, the relationship between the Gaussian curvatures before and after motion can be stated as follows [16]:

$$K' = K(1 - 2\,\mathrm{div}\,\mathbf{s} + (\nabla^* \cdot \nabla\mathbf{s}) \cdot \mathbf{n}), \tag{5}$$

where

$$\nabla^* = \frac{\left[\mathbf{r}_u\left(N\frac{\partial}{\partial u} - M\frac{\partial}{\partial v}\right) + \mathbf{r}_v\left(L\frac{\partial}{\partial v} - M\frac{\partial}{\partial u}\right)\right]}{LN - M^2}.$$

One can see that Eq. (5) defines a 2-component motion model consisting of the divergence and the gradient components. Both components are linear in the motion function: divergence is linear by definition, linearity of the gradient component follows from simple examination of the expression for ∇^*.

[2]It can also be seen as a 3×3 matrix $\mathbf{ab}^T$.

3.2 Derivation of the Error Function

We define the error function as the error of violation of Eq. (5):

$$\epsilon = \Delta + 2K \operatorname{div} \mathbf{s} + \frac{(\Xi \cdot \nabla \mathbf{s}) \cdot \mathbf{n}}{D^2}, \tag{6}$$

where

$$\begin{aligned} \Delta &= K - K' \\ \Xi &= \mathbf{r}_u \left(N \frac{\partial}{\partial u} - M \frac{\partial}{\partial v} \right) + \mathbf{r}_v \left(L \frac{\partial}{\partial v} - M \frac{\partial}{\partial u} \right). \end{aligned} \tag{7}$$

Let us further investigate the quantity $(\Xi \cdot \nabla \mathbf{s})$. First we need the expressions for derivatives of the gradient operator $\nabla \mathbf{s}$. The open products of vectors can be differentiated using the usual product rule (see [17], for example). After some algebra we obtain

$$\begin{aligned} \frac{\partial}{\partial u}(\nabla \mathbf{s}) = \frac{\mathbf{r}_{uu}\mathbf{s}_u + \mathbf{r}_u \mathbf{s}_{uu}}{E} - 2\mathbf{r}_u \mathbf{s}_u \frac{\mathbf{r}_u \cdot \mathbf{r}_{uu}}{E^2} \\ + \frac{\mathbf{r}_{uv}\mathbf{s}_v + \mathbf{r}_v \mathbf{s}_{uv}}{G} - 2\mathbf{r}_v \mathbf{s}_v \frac{\mathbf{r}_v \cdot \mathbf{r}_{uv}}{G^2} \end{aligned}$$

and similarly $\frac{\partial}{\partial v}(\nabla \mathbf{s})$. Using associativity of scalar and open products $\mathbf{c} \cdot \mathbf{ab} = (\mathbf{c} \cdot \mathbf{a})\mathbf{b}$, and the orthogonality assumption, whereby all terms including $F = (\mathbf{r}_u \cdot \mathbf{r}_v)$ vanish, we then obtain the following expressions:

$$\mathbf{r}_u \cdot \left(\frac{\partial}{\partial u} \nabla \mathbf{s} \right) = -\frac{\mathbf{r}_u \cdot \mathbf{r}_{uu}}{E} \mathbf{s}_u + \frac{\mathbf{r}_u \cdot \mathbf{r}_{uv}}{G} \mathbf{s}_v + \mathbf{s}_{uu} \tag{8}$$

and similarly for other 3 terms.

Finally, committing ourselves to the affine model (1) and using the definition of divergence (e.g., [16]), we obtain the following error function :

$$\epsilon = \Delta + \frac{1}{D^2}[G(\mathbf{r}_u \cdot \mathbf{a}) + E(\mathbf{r}_v \cdot \mathbf{b}) + \alpha^1(\mathbf{n} \cdot \mathbf{a}) + \alpha^2(\mathbf{n} \cdot \mathbf{b})]. \tag{9}$$

where

$$\begin{aligned} \alpha^1 &= \frac{1}{E}(-N\Gamma_{uuu} + L\Gamma_{vuv} + M\Gamma_{uuv} - M\Gamma_{vuu}) \\ \alpha^2 &= \frac{1}{G}(N\Gamma_{uuv} - L\Gamma_{vvv} - M\Gamma_{uvv} + M\Gamma_{vuv}). \end{aligned} \tag{10}$$

and Γ's are the Christoffel symbols.

3.3 Minimization of the Error Function

One can easily see that the error function (9) is underconstrained, since it contains only one constraint and 6 parameters in $\mathbf{a}$ and $\mathbf{b}$. To overcome this problem we collect the error contributions ϵ_i for a number of points in the neighborhood.

To avoid tedious calculations of derivatives, we collect the point-specific data in the respective matrices:

$$\begin{aligned} \boldsymbol{\epsilon} &\equiv [\epsilon_1, \ldots, \epsilon_k]^T && [k \times 1] \\ \boldsymbol{\Delta} &\equiv [\Delta_1, \ldots, \Delta_k]^T && [k \times 1] \\ N &\equiv [\mathbf{n}_1, \ldots, \mathbf{n}_k] && [3 \times k] \\ A^1 &\equiv \operatorname{diag}(\alpha_1^1, \ldots, \alpha_k^{1,2}) && [k \times k] \\ A^2 &\equiv \operatorname{diag}(\alpha_1^2, \ldots, \alpha_k^{1,2}) && [k \times k] \\ D &\equiv \operatorname{diag}(D_1, \ldots, D_k)^{-1} && [k \times k] \\ E &\equiv \operatorname{diag}(E_1, \ldots, E_k) && [k \times k] \\ G &\equiv \operatorname{diag}(G_1, \ldots, G_k) && [k \times k] \\ K &\equiv \operatorname{diag}(K_1, \ldots, K_k) && [k \times k] \\ R_u &\equiv [(\mathbf{r}_u)_i, \ldots, (\mathbf{r}_u)_k] && [3 \times k] \\ R_v &\equiv [(\mathbf{r}_v)_i, \ldots, (\mathbf{r}_v)_k] && [3 \times k] \\ B &= [A^1 N^T - 2KGR_u,\ A^2 N^T - 2KER_v] && [k \times 3] \\ \mathbf{x} &\equiv [\mathbf{a}, \mathbf{b}]^T && [6 \times 1]. \end{aligned}$$

Then the constraints representing Eq. (5) for *all* points of interest can be expressed as the following matrix equation, linear in $\mathbf{x}$:

$$\boldsymbol{\epsilon} = \boldsymbol{\Delta} + D^{-2} B \mathbf{x}. \tag{11}$$

Consequently the squared-error function to be minimized is:

$$\boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = (\boldsymbol{\Delta} + D^{-2} B \mathbf{x})^T (\boldsymbol{\Delta} + D^{-2} B \mathbf{x}). \tag{12}$$

Using the chain rule for the first differentials [12] one can find the differential of the function $\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}$:

$$\mathsf{D}(\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}) = 2(\boldsymbol{\Delta} + D^{-2} B \mathbf{x})^T D^{-2} B.$$

Then we equate to zero the gradient of the error function—which is the transpose of the first differential—and thus obtain the familiar normal system of linear equations:

$$B^T D^{-4} B \mathbf{x} = -B^T D^{-2} \boldsymbol{\Delta}. \tag{13}$$

3.4 Hybrid Unit Normal / Gaussian Curvature Algorithm

Let $\boldsymbol{\epsilon}_i^N$ denote the the point-wise error function of the unit normal algorithm, defined as[3].

$$\boldsymbol{\epsilon}_i^N = \Delta \mathbf{n}_i + \mathbf{n}_i \times \operatorname{curl} \mathbf{s}, \tag{14}$$

and let ϵ_i^G denote the error function (9) of the Gaussian curvature algorithm. We propose that the combined point-wise error of the hybrid algorithm be computed as

$$\varepsilon_i = (\boldsymbol{\epsilon}_i^N \cdot \boldsymbol{\epsilon}_i^N) + \omega_i (\epsilon_i^G \cdot \epsilon_i^G), \tag{15}$$

[3]The unit normal algorithm was proposed in [10], with the point-wise error above used in [11].

where the weighting coefficient ω_i is equal to either the mean curvature H_i or the square root of the Gaussian curvature K_i at the point of interest. The choice of the weighting factor is motivated by a heuristic observation that the unit normal algorithm achieves better accuracy of correspondence estimation of relatively flat surfaces while the Gaussian curvature algorithm on more curved surfaces[4]. Therefore we would like the impact of the Gaussian curvature error function to have the effect proportional to curvedness of the surface.

Minimization of the combined error follows the same course as in section 3.3. The resuling linear system essentially contains a "weighted sum of two linear systems" of the respective algorithms:

$$(B_N^T B_N + B_G^T \Omega D^{-4} B_G)\,\mathbf{x} = \\ -\,(B_N^T \operatorname{vec} \Delta_N + B_G^T \Omega D^{-2} \Delta_G), \quad (16)$$

where $\Omega \equiv \operatorname{diag}(\omega_1 \ldots \omega_k)$, and the indices N, K indicate that the correspondent quantities are taken from the unit normal or the Gaussian curvature algorithms.

4 Experiments on Artificial Shapes

We first compare the performance of the unit normal, the Gaussian curvature and the hybrid algorithms on artificial shapes, on which magnitude of the deformation can be controlled. The experiment consists of applying the uniform affine deformation with parameters $\mathbf{a} = (0.001, 0.001, 0.001)$ and $\mathbf{b} = (0.001, 0.001, 0.001)$ to a set of 49 points equally spaced on a quadric. Different parameters of the quadric are considered. The magnitude of the affine motion is multiplied by the parameter δ in the range of 1 to 1000. The true correspondence used in the deformation model is one-to-one; however, in the algorithm it is assumed unknown, and the search for the correct correspondence is carried out in a 7×7 neighborhood around the point of interest.

The relative performance of the unit normal and the Gaussian curvature algorithms depends on the surface properties. On the surfaces with higher curvature, such as in Figures 1 and 2, the Gaussian curvature algorithm yields lower correspondence error, while on a flatter surface of Figure 3 the unit normal is a clear winner. The hybrid algorithm successfully attains the accuracy of the best of its components.

[4]The square root is taken of the Guassian curvature to compensate for the qudratic nature of this parameter, since intuitively a weighting factor should be a linear value.

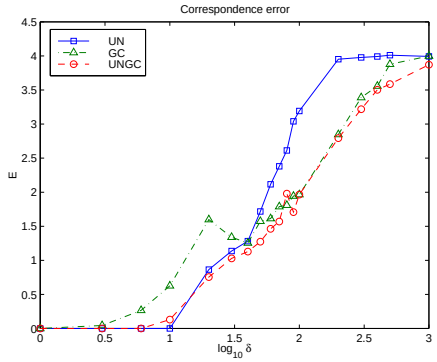


Figure 1: Correspondence error of the Gaussian curvature (GC), the unit normal (UN) and hybrid (UNGC) algorithms for the quadric with $a = c = 1$.

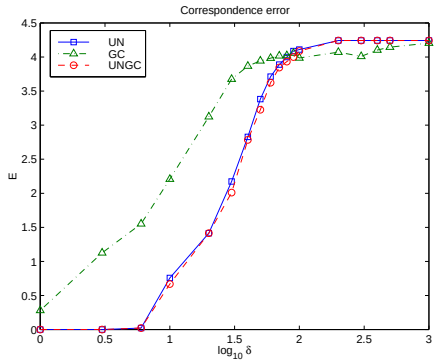


Figure 2: Correspondence error of the Gaussian curvature (GC), the unit normal (UN) and the hybrid (UNGC) algorithms for the quadric with $a = c = 0.1$.

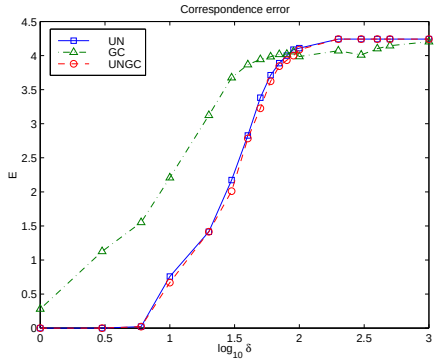


Figure 3: Correspondence error of the Gaussian curvature (GC), the unit normal (UN) and the hybrid (UNGC) algorithms for the quadric with $a = c = 0.01$.

5 Real Motion Experiments

An ultimate test of any motion estimation algorithm is how this algorithm can handle motion between real shapes. Since several assumptions have been made in the derivation of the algorithms presented in this paper, most notably, the assumptions of small motion, orthogonal parametrization and the particular motion model, validity of these assumptions needs to be verified on real data.

5.1 Cyberware Data

The facial 3D data collected from the Cyberware range scanner contains ground truth correspondence information between selected points of interest. This information is obtained from visual images of a subject with paint markers applied to his/her face. Our evaluation parameters include:

- *correspondence error*: $\sqrt{(u_c - \bar{u})^2 - (v_c - \bar{v})^2}$.
- Let p_0 denote the point of interest before motion, $\bar{p}'$ – true corresponding point after motion, p'_c – computed corresponding point after motion. Then the *relative image error* is defined as

$$\frac{||\bar{p}' - p'_c||}{||\bar{p}' - p_0||}. \tag{17}$$

  The relative image error (averaged over all points of interest) measures the 3D error due to erroneous correspondence estimation, relative to the magnitude of motion.

- Let p'_{1-1} denote the point after motion under one-to-one correspondence. Then the *image error improvement ratio* is computed as

$$\frac{||\bar{p}' - p'_{1-1}|| - ||\bar{p}' - p'_c||}{||\bar{p}' - p'_{1-1}||}. \tag{18}$$

  This quantity reveals improvement in image error relative to the one-to-one correspondence (which does not need correspondence estimation and can be assumed by default). This measure can be negative, implying that correspondence estimation has brought in more confusion than a simple guess.

We now compare the Gaussian curvature and the hybrid algorithm with the two previous versions of the unit normal algorithm: the "old" (as in [10]) and the modified (as in [11]). Figure 4 shows the correspondence error, the relative image error and the image error improvement ratio for 7 data sets used in our study.

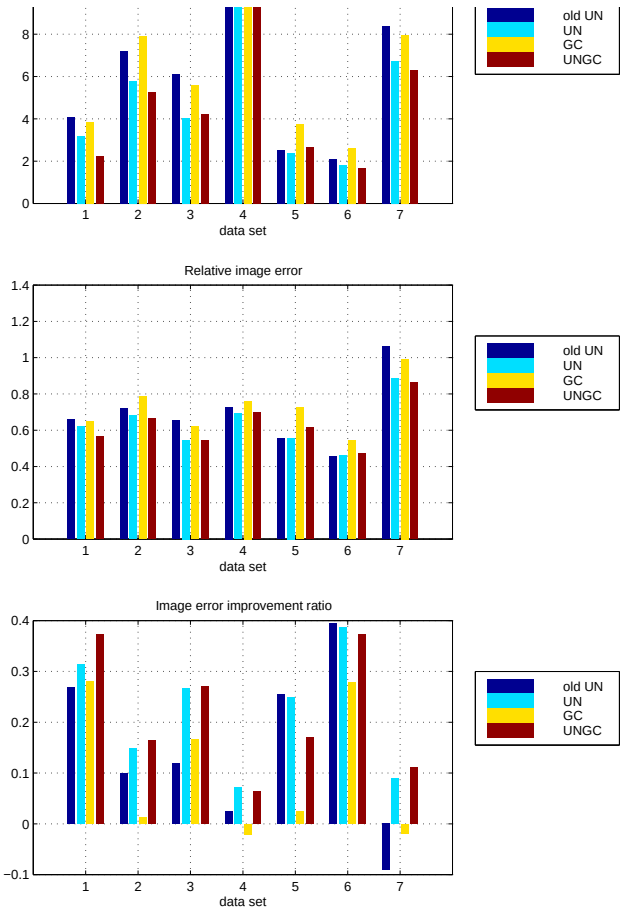


Figure 4: Comparison of the new algorithms on the real motion, 7 data sets. Unit normal (UN), Gaussian curvature (GC) and the hybrid (UNGC) algorithms.

The accuracy of the Gaussian curvature algorithm is not as good in comparison with the unit normal algorithm as it was observed for artificial shapes. It may have something to do with the flatness of a surface, since the unit normal algorithm featured better for flat surfaces; however, a more serious problem was discovered with the Gaussian curvature algorithm during the experiments. The numerical conditioning of the left-hand side matrix of the Gaussian curvature algorithm is noticeably worse than that of its counterpart of the unit normal algorithm. In general, conditioning of all differential-geometric algorithms is mediocre. The ratio between the largest and the smallest singular values was observed to be at least 10^5, which makes singular-value decomposition a preferred solu-

tion method. Thresholding of excessively small singular values is also required. Conditioning of the Gaussian curvature algorithm is, however, in the range of $10^8 - 10^9$ compared to $10^5 - 10^6$ of the unit normal algorithm. This difference does not manifest itself in the artificial shape experiments where the accuracy of estimation of the differential-geometric parameters is high (at least $15 - 16$ significant digits for double-precision floats), well below the smallest singular value of a scaled left-hand side matrix. On the real shapes the accuracy of estimation of differential-geometric parameters is much lower, at about $5 - 6$ significant digits for Gaussian curvature. As a result the solution to the linear system can have strong numerical errors and correctness of motion and correspondence estimation is doomed.

On the other hand, the idea of combining the unit normal and Gaussian curvature error functionals in a hybrid algorithm is promising. On five data sets this algorithm exhibits the smallest correspondence error and the best values of other measures. Since the values of the Gaussian curvature are usually much smaller than 1 for the observed shapes, the hybrid linear system is very similar to the system of the unit normal algorithm, yet the small adjustment leads to better results.

We further validate the proposed algorithms by means of visualization. In these experiments motion and correspondence estimation is performed on every fourth point on the image before motion[5]. The estimation results are used to compute the surfaces on 8 intermediate images. Estimated correspondence is used to determine motion field on the surfaces, and points on intermediate surfaces are taken at equal intervals along the motion field vectors. The remaining points of the intermediate shapes are obtained by interpolation. The sequences of the computed intermediate shapes for the first data set are shown in Figure 5. Each frame contains the images reconstructed by four algorithms: the "old" unit normal algorithm in the upper left, the modified unit normal algorithm in the upper right, the Gaussian curvature algorithm in the lower left and the hybrid algorithm in the lower right. The sequence of frames should be read top to bottom, left to right. The first (upper left) and the last (lower right) frame in each image contains true images before and after motion.

If the correspondences are perfectly estimated, the sequence of intermediate images displays a smooth transition from the first to the last. Incorrectly esti-

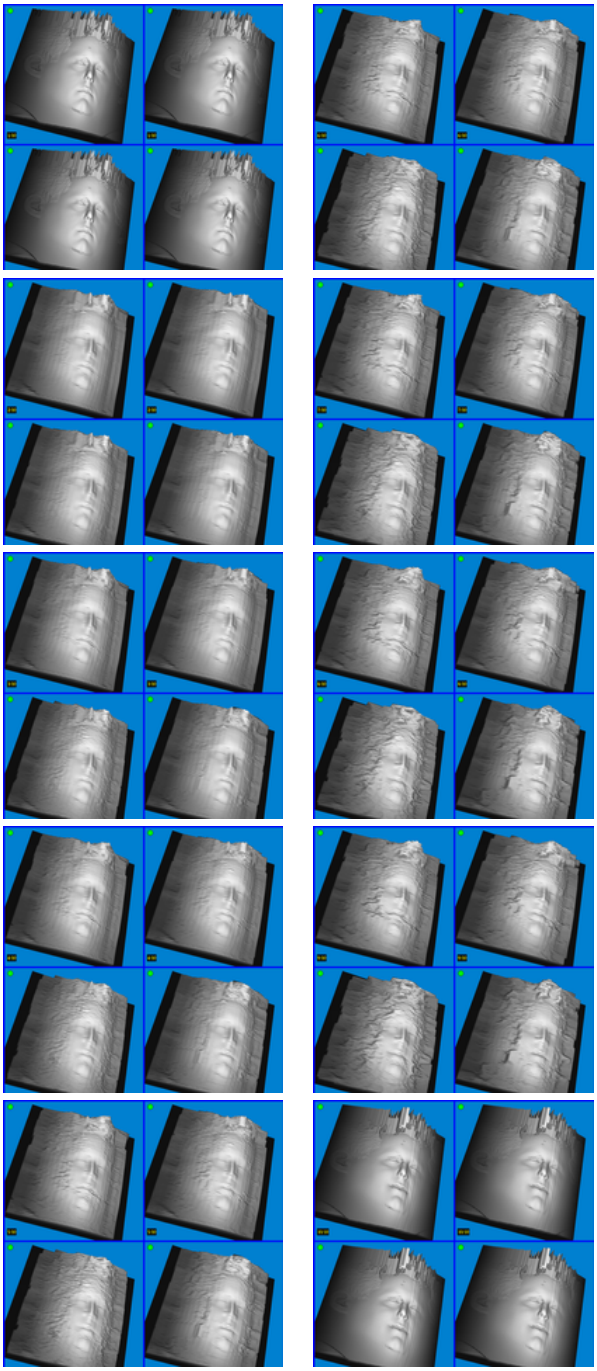
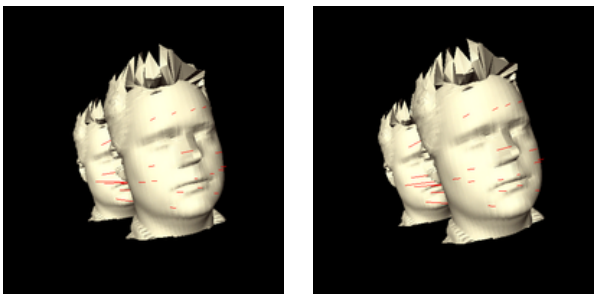


Figure 5: Transformation between two 3D images based on correspondence computed different algorithms. Cyberware data.

[5]More precisely, on every second point of an image subsampled by 2.



(a) unit normal algorithm

(b) hybrid algorithm

Figure 6: Visualization of correspondences computed by the unit normal and hybrid algorithms. Data set 1.

mated correspondences give rise to the "ripple". The smoother the images, the more accurate correspondence estimation. One can see from the images in Figure 5 that the accuracy of correspondence estimation of the two unit normal algorithms is very similar, the pure Gaussian curvature algorithm exhibits somewhat worse, and the hybrid algorithm the best correspondence estimation accuracy.

Another way to visualize correctness of correspondence estimation is to to draw connections between the corresponding points between the shapes before and after motion. Such visualization is shown in Figure 6. One can see where correspondence is estimated incorrectly. Both algorithms miss the point that is initially located on a nose. This point moves to the left cheek on the shape after motion. On the two points on the upper lip the hybrid algorithm does a much better job than the unit normal algorithm. One can see that the new points computed by the hybrid algorithm are located approximately above the points on the lower lip, as on the shape before motion. The new points computed by the unit normal algorithm are noticeably skewed in the opposite direction. On the remaining points correspondence is estimated correctly.

5.2 Data from Structure-from-Motion Algorithm

The data obtained from the structure-from-motion algorithm of Zhou and Kambhamettu allows one to perform a different kind of quantitative evaluation of correspondence estimation. Since this data was originally recorded on a 30 frames/s video camera the true intermediate images are available.

The setup of the experiment is similar to the experiments on Cyberware data presented in section 5.1.

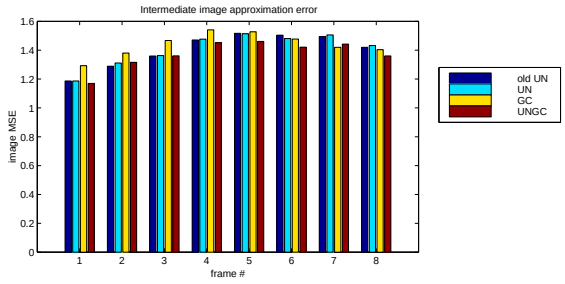


Figure 7: Intermediate image approximation error of the unit normal (UN) and the hybrid (UNGC) algorithms.

The surfaces represent a 460×460 depth image with a large portion of it being background. The background points are excluded from consideration; points of interest equally spaced on a grid of step 8 are selected from the remaining points. After correspondence has been computed the counterparts of the points of interest in intermediate images are selected at regular intervals along the motion vector field. These points are placed in locations on a surface selected at regular intervals along the field of parameter displacements $(u, v) \longrightarrow (u', v')$. Finally, the intermediate image error is computed as mean-squared error of the intermediate points of interest and the points at corresponding locations of the true intermediate images. For the purpose of visualization the remaining non-background points on intermediate shapes are computed by interpolation.

The intermediate image approximation error of the unit normal and the hybrid algorithms is shown in Figure 7. One can see that on this data the hybrid algorithm displays a small but consistent improvement over other algorithms. The absolute values of the error are rather large, but this is due to a large amount of noise in the data.

Visualization of the intermediate images is shown in Figure 8, for the four algorithms. The lower image error of the hybrid algorithm manifests itself in a smoother surface. The overall motion is estimated surprisingly well for such a noisy shape.

6 Conclusions

We have presented two new algorithms for non-rigid motion and correspondence esimation in 3D shapes. Both algorithms are build on a new, linear relation between the Gaussian curvatures before and after motion. The second algorithm, which is the most successful in our practical application, combines the objective functions based on the Gaussian curvature and the

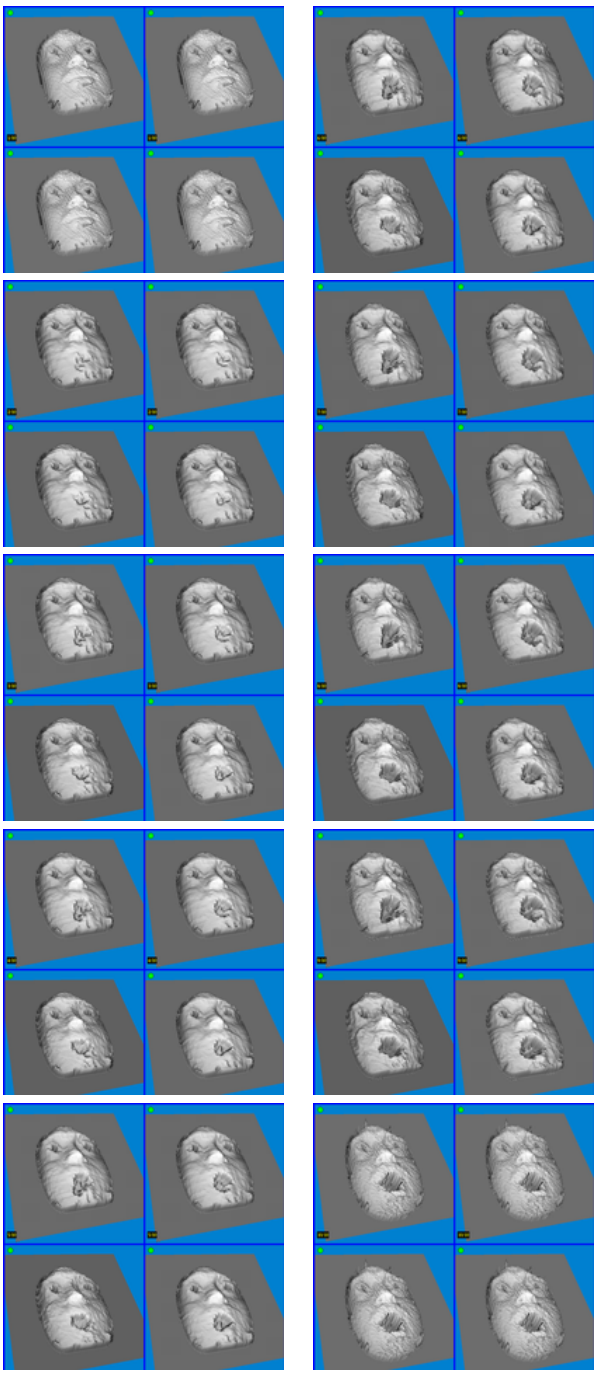


Figure 8: Transformation between two 3D images based on correspondence computed different algorithms. Structure-from-motion data.

unit normal information. Both algorithms are computationally efficient and provide stable solutions.

References

[1] A. A. Amini and J. S. Duncan. Point-wise tracking of left-ventricular motion in 3D. In *Proceedings of IEEE Workshop on Visual Motion*, pages 294–299, 1991.

[2] A. A. Amini and J. S. Duncan. Bending and stretching models for LV wall motion analysis from curves and surfaces. *IVC*, 10(6):418–430, 1992.

[3] A. A. Amini and J. S. Duncan. Differential geometry for characterizing 3D shape change. In *Proceeding of SPIE, Vol. 1786: Mathematical Methods in Medical Imaging*, pages 170–181, 1992.

[4] S. Chaudhuri and S. Chatterjee. Estimation of motion parameters for a deformable object from range data. In *Proc. CVPR*, pages 291–295, 1989.

[5] S. Chaudhuri and S. Chatterjee. Motion analysis of a homogeneously deformable object using subset correspondences. *Pat. Rec.*, 24(8):739–745, 1991.

[6] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In *Proc. ECCV*, pages 458–466, 1992.

[7] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. Technical Report 2200, INRIA, 1994.

[8] C. Kambhamettu and D. B. Goldgof. Point correspondence recovery in non-rigid motion. In *Proc. CVPR*, pages 222–227, 1992.

[9] C. Kambhamettu and D. B. Goldgof. Curvature-based approach to point correspondence recovery in conformal non-rigid motion. *CVGIP*, 60(1):26–43, 1994.

[10] C. Kambhamettu, D. B. Goldgof, and M. He. Determination of motion parameters and estimation of point correspondences in small non-rigid deformations. In *Proc. CVPR*, pages 943–946, 1994.

[11] P. Laskov. *Extensions of differential-geometric algorithms for estimation of 3D non-rigid motion and correspondence.* PhD thesis, University of Delaware, 2001.

[12] J. R. Magnus and H. Neudecker. *Matrix differential calculus with applications in statistics and econometrics.* John Wiley and Sons, 1988.

[13] P. Shi, A. J. Sinusas, R. T. Constable, E. Ritman, and J. S. Duncan. Point-tracked quantitative analysis of left-ventricular surface motion from 3D images sequences. *PAMI*, 19(1):36–50, Jan. 2000.

[14] J.-P. Thirion. New feature points based on geometric invariants for 3D image registration. *IJCV*, 18(2):121–137, May 1996.

[15] Y. Wang, B. S. Peterson, and L. H. Staib. Shape-based 3D surface correspoondence using geodesics and local geometry. In *Proc. CVPR*, pages 644–651, 2000.

[16] C. E. Weatherburn. *Differential geometry of three dimensions.* Cambridge University Press, 1927.

[17] R. C. Wrede. *Introduction to vector and tensor analysis.* John Wiley and Sons, 1963.