3D non-rigid point cloud based surface registration based on mean shift

Thomas Fabry Dirk Smeets Dirk Vandermeulen Paul Suetens
K.U.Leuven, Faculty of Engineering, Department of Electrical Engineering,
Center for Processing Speech and Images, Medical Imaging Research Center
thomas.fabry@uz.kuleuven.be

ABSTRACT

In this article, we present a new 3D non-rigid surface registration algorithm for unstructured point clouds. The algorithm has very low data requirements and can easily be adapted to use additional information other than vertex positions such as texture information or prior knowledge about local deformations. It is robust to noise, outliers and to some extent to missing data. The mathematical theory is based on probability density estimation. Furthermore, we use the mean shift formula for fast computation. The algorithm is able to use different regularisation models for the deformation. Quantitative and qualitative experiments are conducted on artificial surfaces and on real 3D face data.

Keywords: 3D non-rigid Registration, Mean Shift, Probability Density Estimation

1 INTRODUCTION

1.1 3D shape registration

Registration, defined as finding a mapping between two (3D) images or other geometric data structures, is an important computer vision research topic with many applications, such as 2D automatic panorama stitching [7] or object detection in 2D camera surveillance [2]. In medical imaging, registration is of key importance to combine the complementary properties of different imaging modalities like CT (Computed Tomography) and MRI (Magnetic Resonance Imaging) [18], for comparing medical images of the same person but on different time points (where tumors might have grown or shrunk, contrast has been injected [16]...), or for the alignment of intra-operative surfaces to pre-operative images.

The subfield of registration we are concerned with in this paper is 3D surface registration: finding a mapping between two 3D shapes, in our case represented as point clouds. As 3D capturing devices become less expensive and prove to be of value for face recognition, registration of 3D surfaces also becomes important. It is a problem of considerable interest in, e.g., computer graphics, 3D shape aquisition and reconstruction and statistical shape analysis in computer vision and medical imaging. For instance, the morphable statistical face model of Blanz and Vetter [6] is a method based on 3D face scans. For this method, all the faces have to be represented by one topologically consistent mesh configuration, meaning that all faces have to be represented by a mesh with a constant number of vertices and triangles. As this cannot be done at capturing time, a 3D surface registration is needed.

1.2 Related work

Most of the 3D registration literature covers rigid registration, in which only translations and rotations can be dealt with (a transformation is called rigid if the Euclidean distance between any two points remains constant). The most popular example is the Iterative Closest Point (ICP) algorithm of Besl and McKay [5], which has shown to be very successful and for which many variants have been developed. In a nutshell, ICP is a two-step iterative algorithm. In step one, the closest points between two matches are sought, and in the second step, the rigid transformation parameters that best account for the ensemble of these correspondences are calculated, after which this transformation is applied. These two substeps are repeated until convergence.

The problem of non-rigid registration, in which more general deformations besides translations and rotations are admissible, is far more difficult. Recently, a number of 3D non-rigid registration algorithms have been presented in the literature. Most of these algorithms are based on the ICP-algorithm. One example is the optimal step non-rigid ICP algorithm of Amberg et al. [1] in which the ICP framework is extended to non-rigid registration, while keeping the advantageous properties of the original ICP (Iterative Closest Point) algorithm [5]. It is robust to missing data, but as it is based on the ICP framework, it is not expected to be robust to outliers.

Another popular non-rigid registration algorithm related to ICP is Robust Point Matching introduced by Chui and Rangarayan in [9]. RPM is also a two-step iterative algorithm. The first step is the calculation of

a soft-assign (fuzzy) correspondence matrix including outlier handling. In the second step, the transformation that parameterizes the non-rigid registration is updated by solving a least-squares problem including a specific non-rigid deformation parameterization (more specifically, the *Thin Plate Spline* deformation model is chosen). The iterations are governed by an annealing scheme where the annealing parameter controls the fuzziness of the correspondences computed in the first step.

In the same paper ([9]), also another algorithm is developed and used as baseline: a non-rigid *Thin Plate Spline* (TPS) ICP algorithm. Here, at each iteration, *hard* point correspondences are calculated and then TPS transformation parameters are optimized. The hard point correspondences are found using the nearest neighbor heuristic. To handle outliers, points too far away are not considered. The second step is the calculation of the transformation parameters for the TPS transformation by minimizing

$$E_{TPS}(f) = \sum_i ||y_i - f(x_i)||^2 + \lambda \iint \left[(\frac{\partial^2 f}{\partial x^2})^2 + 2(\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2 \right] \mathrm{d}x\mathrm{d}y, \quad (1)$$

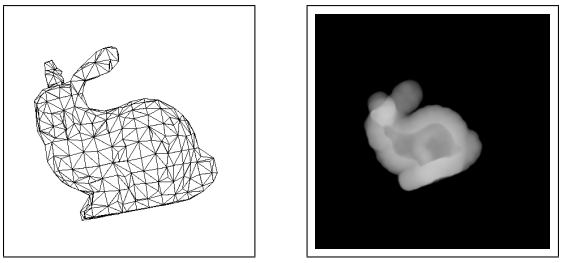
with $f(x_i) = x_i \cdot d + \phi(x_i) \cdot w$ and $\phi(x_i) = ||x - x_i||^2 log||x - x_i||$ the TPS kernel, d a matrix representing the affine transformation and w a warping coefficient matrix representing the non-affine transformation. The parameter λ is set using an annealing procedure.

An algorithm that is not based on ICP but bears close resemblance to ours is the simultaneous non-rigid registration method for multiple point sets for atlas construction of Wang et al. [19]. Here, the 3D model point clouds that need to be registered are represented by a Probability Density Estimate (PDE), which is also a key feature in our algorithm. The algorithm of Wang et al. proceeds by quantifying the distance between these PDE's using an information theoretic measure, which is optimized over the parameters of a deformation model.

2 MEAN SHIFT REGISTRATION

2.1 Probability Density Estimation and Mean Shift

In the majority of 3D surface processing algorithms, 3D objects and surfaces are represented as a mesh: a bunch of vertices (3D points) on the object's surface connected by polygons, mostly triangles. An example of a mesh is shown in figure 1(a). But, as has already been mentioned, we will, in this work, represent 3D surfaces as a Probability Density Estimate (PDE) based on a discrete point cloud version of the surface. These points



(a) (b)

Figure 1: The Stanford bunny represented as (a) a mesh; (b) a volume rendered Probability Density Estimate.

are most readily available and are the most basic representation of surfaces, and are sometimes the only available information. By representing the 3D surface as a PDE, every point in 3D space has a probability of lying on the surface of the 3D object we are working with. This statistical way of representing 3D objects will – to a certain amount – give our algorithm the ability to deal with missing data and outliers. An example of a 3D PDE can be seen in figure 1(b).

We construct the PDE of the 3D surface by using the Kernel Density Estimation (KDE) technique. The PDE is thus constructed in the following way:

$$\hat{p}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} K(\mathbf{x} - \mathbf{x_i}), \quad (2)$$

where $\hat{p}$ is the PDE itself, $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$ are the 3D points of the discretized surface, and $K(\mathbf{x})$ is a scalar function satisfying

$$\int_{\mathbb{R}^3} \mathbf{x}\mathbf{x}^T K(\mathbf{x}) \mathrm{d}\mathbf{x} = c_K I, \quad (3)$$

$$\int_{\mathbb{R}^3} \mathbf{x} K(\mathbf{x}) \mathrm{d}\mathbf{x} = 0, \quad (4)$$

$$\lim_{\|\mathbf{x}\| \to \infty} \|\mathbf{x}\|^3 K(\mathbf{x}) = 0, \quad (5)$$

$$\int_{\mathbb{R}^3} K(\mathbf{x}) \mathrm{d}\mathbf{x} = 1. \quad (6)$$

Here, $\|\cdot\|$ is the Euclidean norm, and c_K is a constant. We follow the method of Comaniciu and Meer [10] for Kernel Density Estimation and are thus only interested in the special class of radially symmetric kernels satisfying $K(\mathbf{x}) = c_{k,d} k(\|\mathbf{x}\|^2)$ with $c_{k,d}$ a normalization constant to make $K(\mathbf{x})$ integrate to one (d stands for the dimensionality of the problem). $k(\cdot)$ is called the profile of the kernel. Using a kernel with bandwidth h, the kernel density estimate is:

$$\hat{p}(\mathbf{x}) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k(||\frac{\mathbf{x} - \mathbf{x_i}}{h}||^2). \quad (7)$$

If we define $g(\mathbf{x}) = -k'(\mathbf{x})$, we can write the density gradient estimate as:

$$\hat{\nabla} p_(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} \mathbf{x_i} g(||\frac{\mathbf{x}-\mathbf{x_i}}{h}||^2) \right] \cdot \left[\frac{\sum_{i=1}^{n} \mathbf{x_i} g(||\frac{\mathbf{x}-\mathbf{x_i}}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{\mathbf{x}-\mathbf{x_i}}{h}||^2)} - \mathbf{x} \right] \quad (8)$$

The last factor is called the *mean shift*:

$$m_h(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x_i} g(||\frac{\mathbf{x}-\mathbf{x_i}}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{\mathbf{x}-\mathbf{x_i}}{h}||^2)} - \mathbf{x}. \quad (9)$$

Because this can be written as

$$m_h(\mathbf{x}) = \frac{1}{2} h^2 c \frac{\hat{\nabla} p_{h,K}(\mathbf{x})}{\hat{p}_{h,G}(\mathbf{x})}, \quad (10)$$

it can be seen that, at location $\mathbf{x}$, the mean shift is proportional to the density gradient estimate with kernel K, normalized with the density estimate computed with kernel G, and thus always points at the direction of maximum increase in density.

Because of the normalization, the mean shift can be thought of as an adaptive gradient estimation for use in gradient ascent methods. The mean shift is adaptive in the way that it is bigger in regions with low density, and smaller in regions with high density. This is what we will need for our registration algorithm.

In this paper, we will use the Gaussian kernel function $G(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}} \exp(-\frac{1}{2}\|\frac{\mathbf{x}}{h}\|^2)$, leading to:

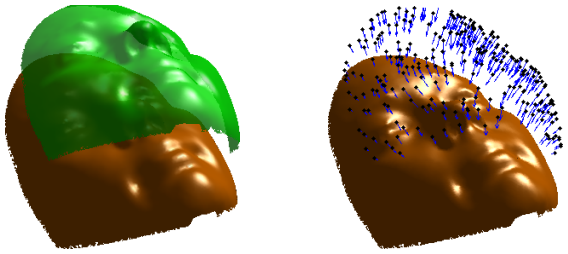
$$m_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x_i} \exp(||\frac{\mathbf{x}-\mathbf{x_i}}{h}||^2)}{\sum_{i=1}^{n} \exp(||\frac{\mathbf{x}-\mathbf{x_i}}{h}||^2)} - \mathbf{x}. \quad (11)$$

It should be noticed that this theory and the associated algorithms are also valid for other kernel functions.

2.2 Rigid mean shift registration

If the PDE of interest represents a 3D surface, the mean shift vector will always point in the direction that needs to be followed to get closer to the surface. This is exactly what we need for a registration algorithm. Also, as can be seen from equation (11), the mean shift vector in every point $\mathbf{x}$ is not only dependent on the nearest point $\mathbf{x_i}$ in the (target) point cloud, but every point in the (target) point cloud, weighted with its distance to point $\mathbf{x}$, is taken into account. This will assure the robustness of our method against noise and outliers.

As a start, the two objects that need to be registered are assumed to lie in the same 3D Euclidean space. From now on, we will talk about the *target surface*, the static surface to which the other surface, the moving *floating surface*, has to be registered. We can now for every point $\mathbf{x}$ in the floating point cloud compute the mean shift vector $m_{h,G}(\mathbf{x})$ in the PDE of the target surface by using equation (11). This is illustrated in figure 2.



(a) (b)

Figure 2: (a) The floating and the target surface in the same Euclidean space; (b) the mean shift vectors of the target surface, evaluated in the points of the floating surface (subsampled for clarity).

After the calculation of the mean shift vectors, we need to transform these vectors into one rigid transformation for the whole 3D model. We do this by calculating the optimal rigid transformation matrix that approximates these vectors in a least squares sense and applying it. This is the same procedure as used in the traditional ICP algorithm.

These two steps, the calculation of the mean shift vectors and the computation of the optimal rigid transformation matrix, are then iterated, until convergence.

The algorithm can thus be summarized as follows:

```
REPEAT
 - Calculate mean-shift vectors at the
    points of the floating surface.
 - Calculate the optimal rigid
    transformation matrix and
    apply it.
UNTIL convergence
```

An example of a Mean Shift Rigid Registration between two different facial surfaces can be seen in figure 3 .

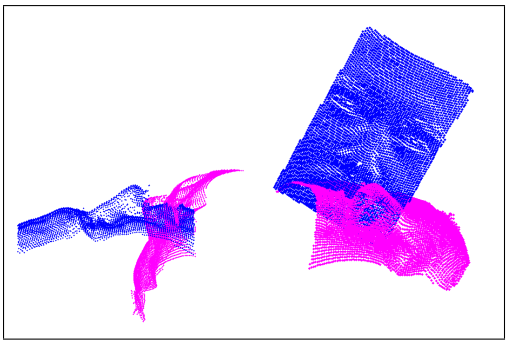
2.3 Non-rigid mean shift registration

The classical non-rigid registration algorithm scheme is a two-step scheme very similar to the rigid registration: first, compute a displacement vector for every point in the floating surface, and then optimize the parameters of a non-rigid transformation model with relation to these displacement vectors.

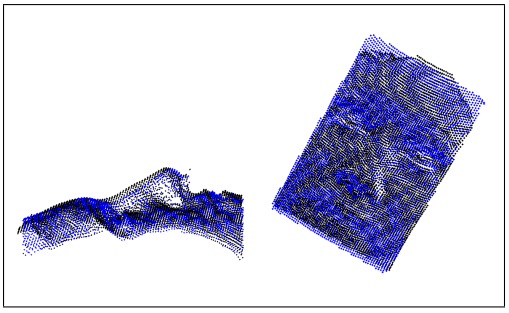
We on the other hand use the mean shift vectors as local estimates of a non-parametric deformation, and in such a way we don't need any explicit deformation model . These displacement vectors are then regularized using smoothing or quasi-interpolation and scaling before they are applied to the points of the floating surface.

For the regularization, we implemented two different strategies, but of course there are a plethora of other possibilities.

Radial Basis Function regularization The first strategy is *quasi-interpolation with radial basis functions*



(a) before registration



(b) after registration

Figure 3: A Sample Mean Shift Rigid Registration of two different faces.

(RBFs). A radial basis function interpolant $s(\mathbf{x})$ is a function of the form:

$$s(\mathbf{x}) = \sum_{i=1}^{n} \mu_i \Phi(||\mathbf{x} - \mathbf{x_i}||) - p(\mathbf{x}), \qquad (12)$$

where $\mathbf{x_i}$ are the interpolation data points, $\Phi(\mathbf{x})$ are radially symmetric basis functions, $p(\mathbf{x})$ is a low degree polynomial and μ_i are the unknown RBF coefficients. For interpolation, the RBF coefficients μ_i are computed assuring

$$s(\mathbf{x_i}) = m_i, \quad i = 1, \ldots, N, \qquad (13)$$

with m_i the values at the data points $\mathbf{x_i}$. For quasi-interpolation, the RBF coefficients can be determined by minimizing

$$\rho||s||^2 + \frac{1}{N}\sum_{i=1}^{N}(s(\mathbf{x_i}) - m_i)^2, \qquad (14)$$

where $||s||^2$ denotes the smoothing penalty defined by

$$\|s\|^2 = \int_{\mathbb{R}^2} ((\frac{\partial^2 s(\mathbf{x})}{\partial x^2})^2 + (\frac{\partial^2 s(\mathbf{x})}{\partial y^2})^2 + (\frac{\partial^2 s(\mathbf{x})}{\partial z^2})^2 + 2(\frac{\partial^2 s(\mathbf{x})}{\partial x \partial y})^2 + 2(\frac{\partial^2 s(\mathbf{x})}{\partial x \partial z})^2 + 2(\frac{\partial^2 s(\mathbf{x})}{\partial y \partial z})^2)\mathrm{d}\mathbf{x} \qquad (15)$$

and ρ is a smoothing constant: the smaller ρ, the more the interpolant is passes through the data points $\mathbf{x_i}$, and vice versa. This approach is known as *spline smoothing* [8]. For the regularization of the mean shift vectors using radial basis functions, we make use of three radial basis function interpolants s_x, s_y, s_z: one for each component dimension of the mean shift vector. This means that we have to compute three RBF interpolants;

$$s(\mathbf{x}) = \begin{cases} s_x(\mathbf{x}) = \sum_{i=1}^{n} \mu_{i,x} \Phi(||\mathbf{x} - \mathbf{x_i}||) - p_x(\mathbf{x}) \\ s_y(\mathbf{x}) = \sum_{i=1}^{n} \mu_{i,y} \Phi(||\mathbf{x} - \mathbf{x_i}||) - p_y(\mathbf{x}) \\ s_z(\mathbf{x}) = \sum_{i=1}^{n} \mu_{i,z} \Phi(||\mathbf{x} - \mathbf{x_i}||) - p_z(\mathbf{x}) \end{cases}, \qquad (16)$$

with constraints

$$s(\mathbf{x_i}) = \mathbf{m_{h,i}} = \begin{cases} m_{h,i,x} \\ m_{h,i,y} \\ m_{h,i,z} \end{cases}, \qquad (17)$$

where $\mathbf{m_{h,i}}$ are the mean shift vectors from equation (11). The regularized mean shift vectors are then computed as

$$\tilde{\mathbf{m}}_{\mathbf{h,i}}(\mathbf{x_i}) = \lambda \cdot (s_x(\mathbf{x_i}), s_y(\mathbf{x_i}), s_z(\mathbf{x_i})), \qquad (18)$$

with $\mathbf{x_i}$ the points of the floating point cloud and λ a scaling parameter.

Gaussian smoothing as regularization Another possibility for regularizing the mean shift vectors is using Gaussian smoothing. Because the interpolation points do not lie on a regular grid, we perform Gaussian smoothing using the Gauss transform. Here, the regularized mean shift vectors are computed using (18), with $\mathbf{x_i}$ the points of the floating point cloud, and

$$s(\mathbf{x}) = \begin{cases} \sum_{i=1}^{N} m_{h,i,x} \exp(||\frac{\mathbf{x}-\mathbf{x_i}}{h_s}||^2) \\ \sum_{i=1}^{N} m_{h,i,y} \exp(||\frac{\mathbf{x}-\mathbf{x_i}}{h_s}||^2) \\ \sum_{i=1}^{N} m_{h,i,z} \exp(||\frac{\mathbf{x}-\mathbf{x_i}}{h_s}||^2) \end{cases}. \qquad (19)$$

Here, h_s is the smoothing parameter. The regularized mean shift vector at point $\mathbf{x}$ is thus computed as an inverse distance weighted sum of the mean shifts in all points of the point cloud.

Algorithm The non-rigid registration algorithm can thus be summarized as follows:

```
REPEAT
 - Calculate (weighted) mean-shift
   vectors at the points of the
   floating surface.
 - Regularize and scale the
   displacement vectors.
 - Move the points of the floating
   surface using the computed
   displacement vectors.
UNTIL convergence
```

2.4 Implementation

The algorithm was implemented in MATLAB. Both the mean shift computation and the Gaussian smoothing are essentially a series of Gauss transforms:

$$\sum_{i=1}^{N} q_i \exp(\| \frac{\mathbf{x_j} - \mathbf{x_i}}{h} \|^2), \forall j = 1 \ldots M \qquad (20)$$

When solved naively, and $M = N$ the Gauss transform's complexity is quadratic ($\mathcal{O}(N^2)$). The Fast Gauss Transform (FGT) of Greengard and Strain [14] reduced the complexity of the Gauss Transform to linear ($\mathcal{O}(N)$). The Improved Fast Gauss Transform (IFGT) of Yang et al. [20] has further increased the speed of the Gauss transform, mainly for high dimensional problems.

In another class of fast implementations, the Gauss Transform is computed using a space-partitioning data structure for organizing the point clouds. These methods are often referred to as *kd-tree* methods (short for *k*-dimensional tree methods) [13, 15].

The FIGTree algorithm of Morariu et al. [17] is an implementation that combines various fast algorithms, including a IFGT and *kd*-tree algorithm, automatically selecting the best parameters for each problem. As such, the speed of our registration algorithms can benefit from using this package.

The RBF calculation is of a similar complexity. Conventional methods for RBF calculation, using 3D biharmonic (thin-plate) splines are $\mathcal{O}(N^3)$. Hierarchical and fast multipole methods [3] can reduce this complexity to $\mathcal{O}(N \log N)$. These fast results can for instance be obtained using the FastRBF Toolbox [12].

To increase the robustness of our algorithm, the iterations are done in an annealing or multiscale way. In optimization, annealing is a way of iterative improvement of the cost function by letting a temperature parameter decrease in a controlled way. In our algorithm, the Gauss transform's kernel bandwidth parameter for the mean shift computation is large in the beginning and is decreased over the iterations.

3 EXPERIMENTS

In this section, we present some results for our registration algorithm. In this paper, we will focus on the non-rigid registration. For results of the rigid registration, we refer to [11].

3.1 Proof of concept

To proof the concept of the mean shift based non-rigid registration algorithm, we make use of meshes of superquadric surfaces. Deformed meshes were then created by non-rigidly transforming them using a thin plate spline deformation field and resampling the deformed meshes. Afterwards, the original configuration was estimated by applying the mean shift non-rigid registration algorithm from the deformed (floating) surface to the original (static) surface. To make the challenge fair, since the surfaces were deformed using a TPS deformation field, we used Gaussian smoothing as regularizer.

landmark number	landmark name
1	nose tip
2	left inner eye corner
3	right inner eye corner
4	left outer eye corner
5	right outer eye corner
6	left mouth corner
7	right mouth corner

Table 1: The landmarks used in the registration of intra-subject face scans experiment.

The results of these registrations as illustrated in figure 4, and prove qualitatively that the new non-rigid registration algorithm can lead to good registrations. The non-rigid registration took less then 2 minutes on one cluster node with a dual-core AMD Opteron 2220 processor.

3.2 Registration of intra-subject face scans with face expressions

To test the ability of our algorithm to deal with real-world data, we make use of the Binghamton University 3D Facial Expression (BU-3DFE) Database [21]. This is a database containing 100 subjects, with for each subject 25 scans for different face expressions. In this experiment, we tried to register the neutral face scan of one person to eight of the expression face scans, on which seven anatomical landmarks were manually indicated. A set of face scans with different expressions are visualised in figure 5. To construct reliable statistical models, it is required that the landmark locations of one face instance are mapped as closeby as possible to the landmark locations of the target face, when registered. The result of the non-rigid registration can then be validated by the statistical distribution of the Euclidean distances of the landmarks on the registered face to the real landmark locations:

$$d = \sqrt{(l_{o,x} - l_{r,x}) + (l_{o,y} - l_{r,x}) + (l_{o,z} - l_{r,x})}, \qquad (21)$$

where $l_o = l_{o,x}, l_{o,y}, l_{o,z}$ are the landmark locations before registration and $l_r = l_{r,x}, l_{r,y}, l_{r,z}$ are the landmark locations after registration . This is what can be found in figure 6. For comparison, also results of the non-rigid TPS-ICP registration described in the introduction are included.

The landmark numbers are explained in table 1. The experimental results show that the mean shift non-rigid registration algorithm outperforms the TPS-ICP algorithm, especially in regions where large expression-induced shape variations are to be expected, at the mouth corners. For these experiments, we tuned the parameters of both algorithms as well as possible.

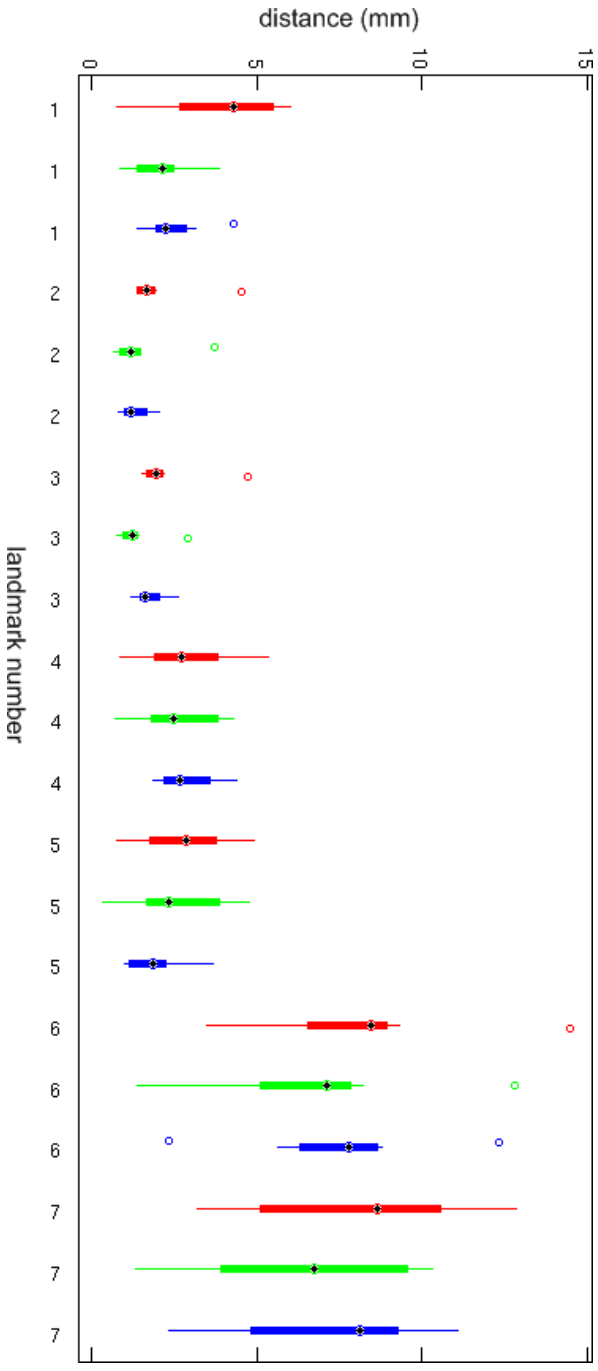


Figure 6: Registration of intra-subject face scans with face expressions. The red bars (1,4,7,10,13,16 & 19) represent the results after only rigid registration, green bars (2,5,8,11,14,17 & 20) indicate results after mean shift non-rigid registration, and blue bars (3,6,9,12,15,18 & 21) represent results after non-rigid ICP registration. (Color version online.)

4 CONCLUSIONS AND FUTURE WORK

We developed a novel non-rigid point cloud based surface registration algorithm based on mean shift. The algorithm shows to be able to perform good non-rigid registrations. Quantitative experiments prove the power of our algorithm.

The algorithm's theory is stated in 3D, and we make only use of the 3D point locations of the discretisized surface representation. The framework can however be expanded to N dimensions. A straightforward application that can benefit from a 4D registration is recognition for 3D facial gestures, which can be seen as a 4D surface [4]. We however plan to use the extendibility to N dimensions for improving the performance of the 3D registration. In its current implementation, the algorithm is expected to be more suitable for expressionless inter-person registration than for intra-person registration where it has to deal with face expressions. We want to tackle this problem by building a N dimensional space consisting of 3 Euclidean shape space dimensions and additional dimensions containing a.o. texture and curvature information.

Furthermore, we plan to review variable kernel methods, to deal with unevenly sampled surfaces and the non-uniform nature of face surfaces. A broad validation of the algorithm on a 3D face database is also planned.

ACKNOWLEDGEMENTS

This work is supported by the Flemish Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT Vlaanderen), the Research Programme of the Fund for Scientific Research - Flanders (Belgium) (FWO) and the Research Fund K.U.Leuven.

REFERENCES

[1] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid ICP algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[2] C. Anagnostopoulos, T. Alexandropoulos, S. Boutas, V. Loumos, and E. Kayafas. A template-guided approach to vehicle surveillance and access control. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 534–539, Sept. 2005.

[3] R.K. Beatson and L. Greengard. A short course on fast multipole methods. *Wavelets, multilevel methods and elliptic PDEs*, pages 1–37, 1997.

[4] L. Benedikt, D. Cosker, P.L. Rosin, and D. Marshall. 3D facial gestures in biometrics: from feasibility study to application. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–6, 29 2008-Oct. 1 2008.

[5] P. J. Besl and H. D. Mckay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992.

[6] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive*

techniques, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[7] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *Journal International Journal of Computer Vision*, 74(1):59–73.

[8] J. C. Carr, R. K. Beatson, B. C. McCallum, W. R. Fright, T. J. McLennan, and T. J. Mitchell. Smooth surface reconstruction from noisy range data. In *ACM GRAPHITE*, pages 119–126, Melbourne, Australia, February 2003.

[9] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.*, 89(2-3):114–141, 2003.

[10] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, May 2002.

[11] Thomas Fabry, Dirk Vandermeulen, and Paul Suetens. 3D face recognition using point cloud kernel correlation. In *BTAS '08: Proceedings of the IEEE Second International Conference on Biometrics Theory, Applications and Systems*, Arlington, Virginia, USA, September 2008.

[12] FarField Technology. *FastRBF MATLAB Toolbox Manual*, August 2004.

[13] A.G. Gray and A.W. Moore. Nonparametric density estimation: Toward computational tractability. In *SIAM Data Mining*, 2003.

[14] L. Greengard. The fast Gauss transform. *SIAM J. Sci. Stat. Comput.*, 12(1):79–94, 1991.

[15] Dongryeol Lee, Alexander Gray, and Andrew Moore. Dual-tree fast gauss transforms. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 747–754. MIT Press, Cambridge, MA, 2006.

[16] Dirk Loeckx, Walter Coudyzer, Frederik Maes, Dirk Vandermeulen, Guido Wilms, Guy Marchal, and Paul Suetens. Nonrigid registration for subtraction CT angiography applied to the carotids and cranial arteries. *Academic Radiology*, 14(12):1562 – 1576, 2007.

[17] Vlad I. Morariu, Balaji Vasan Srinivasan, Vikas C. Raykar, Ramani Duraiswami, and Larry S. Davis. Automatic online tuning for fast gaussian summation. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[18] Pieter Slagmolen, Dirk Loeckx, Sarah Roels, Xavier Geets, Frederik Maes, Karin Haustermans, and Paul Suetens. Nonrigid registration of multitemporal CT and MR images for radiotherapy treatment planning. In *Biomedical Image Registration*, volume 4057/2006, pages 297–305. Publisher Springer Berlin / Heidelberg, 2006.

[19] F. Wang, B.C. Vemuri, A. Rangarajan, and S.J. Eisenschenk. Simultaneous nonrigid registration of multiple point sets and atlas construction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):2011–2022, Nov. 2008.

[20] C. Yang, R. Duraiswami, N.A. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 664–671 vol.1, Oct. 2003.

[21] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J. Rosato. A 3d facial expression database for facial behavior research. In *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, pages 211–216, Washington, DC, USA, 2006. IEEE Computer Society.

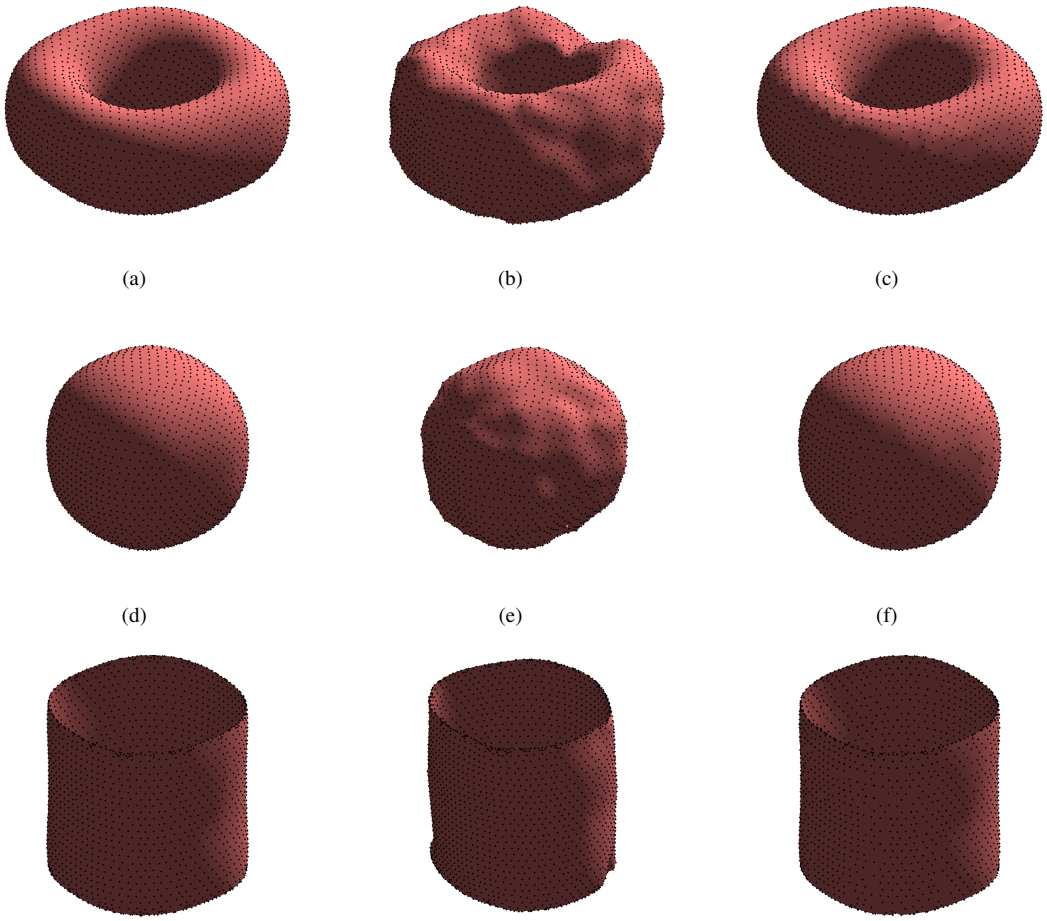


Figure 4: Results of the mean shift non-rigid registration algorithm. (a), (d) and (g) are the original surfaces. (b), (e) and (h) are deformed, noisy version hereof. These were registered to the originals, and the result is shown in (c), (f) and (g). Only the point clouds were used, but for clarity of visualization, the point clouds are displayed as shaded meshes with black dots on the point locations.

Figure 5: Nine face surfaces captured from the same person but with different facial expressions.