MEAN SHIFT ALGORITHM FOR ROBUST RIGID REGISTRATION BETWEEN GAUSSIAN MIXTURE MODELS

Claudia Arellano and Rozenn Dahyot

School of Computer Science and Statistics
Trinity College Dublin, Ireland

ABSTRACT

We present a Mean shift (MS) algorithm for solving the rigid point set transformation estimation [1]. Our registration algorithm minimises exactly the Euclidean distance between Gaussian Mixture Models (GMMs). We show experimentally that our algorithm is more robust than previous implementations [1], thanks to both using an annealing framework (to avoid local extrema) and using variable bandwidths in our density estimates. Our approach is applied to 3D real data sets captured with a Lidar scanner and Kinect sensor.

Index Terms— Mean Shift, Registration, Gaussian Mixture Models, Rigid Transformation

1. INTRODUCTION

Point set registration refers to the process of finding the spatial transformation between two sets of points. It is an essential task in computer vision for applications that require shape alignment such as tracking, recognition and indexing among others. Jian and Vermuri [1] proposed to estimate the rigid transformation parameters by minimizing the Euclidean distance between two density functions estimated with the two point sets to register. They solved the estimation using numerical optimization [1]. As a first contribution, we present a Mean Shift Algorithm for robustly registering two point sets by modelling them as two mixtures of Gaussians (section 3). Our algorithm is implemented with an annealing strategy to converge toward the global minimum of the distance [2] and it is well suited for parallel programming implementations [3]. Secondly, both density functions representing the point sets use Gaussian kernels with variable bandwidths, and contrary to previous modelling [1, 4], none of the densities are simplified to the empirical density function. As a result, we show experimentally that our modelling is more robust than Jian and Vermuri's [1] (section 4) and we apply our approach to align 2D and 3D data sets captured by a Lidar scanner and a Kinect sensor.

Thanks to Trinity College Dublin and The Government of Chile for funding.

2. STATE OF THE ART

One popular method to perform registration between two point sets is the Iterative Closest Point (ICP) algorithm [5, 6]. It is based on a point-to-point correspondence between two data sets performed using the nearest neighbour criteria. Many improvements have been made to the basic ICP algorithm [7] but the approach is sensitive to both outliers and its initialization which requires some manual labelling [8].

Probabilistic methods that redefined the registration problem as a Maximum Likelihood estimation problem have also been proposed [9] but they assume that one set (observations) is a dense sample of the reference set which is not always true in real applications. As an alternative to Maximum Likelihood estimation of the transformation, a more robust approach consists in modelling the two point sets as two probability density functions and finding the transformation that minimises a cost function between these two pdfs [4]. Several cost functions have been proposed such as cross correlation [10] and the Cauchy-Schwarz divergence [11]. Jian and Vermuri [1] have shown that most algorithms for registration can be interpreted as a particular case of the minimization of the power divergence between two probability density functions [12]. Jian and Vermuri's modelling is based on choosing the cost function as the Euclidian L_2 distance:

$$\hat{\Theta} = \arg\min_{\Theta} \int_{\mathbb{R}^{d_{\mathbf{x}}}} \left(p_u(\mathbf{x}) - p_v(\mathbf{x}|\Theta)\right)^2 \, d\mathbf{x} \tag{1}$$

with Θ being the transformation parameters to be estimated. When the probability density functions $p_u(\mathbf{x})$ and $p_v(\mathbf{x})$ are modelled as Gaussian Mixtures, the cost function has a closed form expression. Moreover, when Θ is the parameter of a rigid transformation, the estimation (eq. 1) is equivalent to [1]:

$$\widehat{\Theta} = \arg\max_{\Theta} \int_{\mathbb{R}^{d_{\mathbf{x}}}} p_u(\mathbf{x}) \, p_v(\mathbf{x}|\Theta) \, d\mathbf{x} \tag{2}$$

To use this estimation scheme for registration, the two pdfs p_u and p_v need to be estimated from the two point sets. One choice is to model one of the distributions with its empirical density function, e.g. [4, 1]:

$$\hat{p}_u(\mathbf{x}) = \frac{1}{n_u} \sum_{k=1}^{n_u} \delta(\mathbf{x} - \mathbf{u}^{(k)})$$

© EURASIP, 2012 - ISSN 2076-1465

for the point set $\mathcal{U} = \{\mathbf{u}^{(k)}\}_{k=1,\cdots,n_u}$. The estimation (2) becomes [4, 1]:

$$\widehat{\Theta} = \arg\max_{\Theta} \frac{1}{n_u} \sum_{k=1}^{n_u} p_v(\mathbf{u}^{(k)}|\Theta) \tag{3}$$

The density p_v is then approximated with a Gaussian kernel density estimate using the second set of observations $\{\mathbf{v}^{(i)}\}_{i=1,\cdots,n_v}$ [1], with K defining the kernel and h its bandwidth:

$$\widehat{\Theta} = \arg\max_{\Theta} \frac{1}{n_u n_v} \sum_{k=1}^{n_u} \sum_{i=1}^{n_v} K\left(\frac{\|\mathbf{u}^{(k)} - \Theta(\mathbf{v}^{(i)})\|}{h}\right) \tag{4}$$

Jian and Vermuri [1] solved equation (4) by using a standard non linear optimisation function from Matlab.

One problem with this modelling is that the empirical density function for p_u is assumed to be a good estimate of the true density function. This is not quite true for instance when the observations are sparse or when there is missing data. Indeed in many applications such as Lidar scan alignment, the recorded point sets are not uniformly sampled from the scene since they correspond only to what is seen in the scene from a particular point of view. To compensate for this drawback, we propose to explicitly estimate both of the density functions by using Gaussian kernel estimates with the idea that adjusting bandwidths can help to deal with missing observations. Section 3 presents an explicit algorithm to perform this optimisation and we show experimentally (section 4) that our modelling combined with our algorithm is more robust than Jian and Vermuri's [1].

3. MEAN SHIFT ALGORITHM FOR 2D POINT SETS

Let us define the random variable $\mathbf{x} \in \mathbb{R}^2$ for which we collected the observations $\mathcal{U} = \{\mathbf{u}^{(k)}\}_{k=1,\cdots,n_u}$. A kernel estimate of the probability density function of $\mathbf{x}$ can be proposed by using $\mathcal{U}$:

$$\hat{p}_u(\mathbf{x}) = \sum_{k=1}^{n_u} G(\mathbf{x}, \mathbf{u}^{(k)}, (h_u^{(k)})^2\mathbf{I})\, \pi_u^{(k)} \tag{5}$$

where G is the Gaussian density function with three arguments: the variable $\mathbf{x} \in \mathbb{R}^2$, the mean (i.e. the data point $\mathbf{u}^{(k)}$), and the covariance matrix chosen isotropic (with variable bandwidth $h_u^{(k)}$ and $\mathbf{I}$ the identity matrix). $\{\pi_u^{(k)} \geq 0\}_{k=1,\cdots,n_u}$ are the weights of the Gaussian kernels such that $\sum_{k=1}^{n_u} \pi_u^{(k)} = 1$. Lets assume that a second set of 2D points $\{\mathbf{v}^{(i)}\}_{i=1,\cdots,n_v}$ is collected such that $\mathcal{V} = \{\mathrm{R}\mathbf{v}^{(i)} + \mathbf{t}\}_{i=1,\cdots,n_v}$ is also a set of observations for the random variable $\mathbf{x}$ with

$$\mathrm{R} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \text{ and } \mathbf{t} = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

We denote $\Theta = (\alpha, t_x, t_y)$ the latent variable of interest corresponding to the rigid transformation. Then a second kernel density estimate can be proposed to approximate the desnitisty of $\mathbf{x}$ as follows:

$$\hat{p}_v(\mathbf{x}|\Theta) = \sum_{i=1}^{n_v} G(\mathbf{x}, \mathrm{R}\mathbf{v}^{(i)} + \mathbf{t}, (h_v^{(i)})^2\mathbf{I})\, \pi_v^{(i)} \tag{6}$$

Then Θ can be estimated robustly with:

$$\hat{\Theta} = \arg\max_{\Theta} \left\{ \mathcal{C}(\Theta) = \int_{\mathbb{R}^2} \hat{p}_u(\mathbf{x})\, \hat{p}_v(\mathbf{x}|\Theta)\, d\mathbf{x} = \sum_{k=1}^{n_u} \sum_{i=1}^{n_v} E(\Theta, \mathbf{u}^{(k)}, \mathbf{v}^{(i)}) \right\} \tag{7}$$

where the analytical solution for the kernel E is [13]:

$$E(\Theta, \mathbf{u}^{(k)}, \mathbf{v}^{(i)}) = \frac{\pi_u^{(k)}\pi_v^{(i)}}{2\pi((h_u^{(k)})^2 + (h_v^{(i)})^2)} \exp\left(\frac{-\|\mathbf{u}^{(k)} - \mathrm{R}\mathbf{v}^{(i)} - \mathbf{t}\|^2}{2\,((h_u^{(k)})^2 + (h_v^{(i)})^2)}\right) \tag{8}$$

3.1. Taylor Expansion

The kernel E is not linear w.r.t. Θ. Consider the function $B_{\mathbf{v}}(\Theta) = \mathrm{R}\mathbf{v} + \mathbf{t}$ defined for $\Theta \in [-\pi; \pi] \times \mathbb{R}^2$ to $B_{\mathbf{v}}(\Theta) \in \mathbb{R}^2$. The Taylor expansion of this vector valued function around $\Theta^{(0)}$ is:

$$B_{\mathbf{v}}(\Theta) = \underbrace{B_{\mathbf{v}}(\Theta^{(0)}) + DB_{\mathbf{v}}(\Theta^{(0)})\, \underbrace{(\Theta - \Theta^{(0)})}_{\delta\Theta}}_{B(\delta\Theta, \Theta^{(0)}, \mathbf{v})} + h.o.t \tag{9}$$

where $DB_{\mathbf{v}}(\Theta^{(0)})$ corresponds to the 2×3 matrix of the partial derivatives of $B_{\mathbf{v}}$ computed at $\Theta^{(0)}$ (for the point $\mathbf{v} = (v_x, v_y)$):

$$DB_{\mathbf{v}}(\Theta^{(0)}) = \begin{pmatrix} -v_x \sin\alpha^{(0)} - v_y \cos\alpha^{(0)} & 1 & 0 \\ v_x \cos\alpha^{(0)} - v_y \sin\alpha^{(0)} & 0 & 1 \end{pmatrix} \tag{10}$$

The first order approximation $B(\delta\Theta, \Theta^{(0)}, \mathbf{v})$ is now linear w.r.t. $\delta\Theta$ (eq. 9) and the kernel E is modified as follow:

$$E_{ki}(\delta\Theta, \Theta^{(0)}) = \frac{\pi_u^{(k)}\pi_v^{(i)}}{2\pi((h_u^{(k)})^2 + (h_v^{(i)})^2)} \times \exp\left(\frac{-\|\mathbf{u}^{(k)} - B(\delta\Theta, \Theta^{(0)}, \mathbf{v}^{(i)})\|^2}{2\,((h_u^{(k)})^2 + (h_v^{(i)})^2)}\right) \tag{11}$$

The modified cost function to maximise is now defined near $\Theta^{(0)}$ by:

$$\mathcal{C}(\delta\Theta, \Theta^{(0)}) = \sum_{k=1}^{n_u} \sum_{i=1}^{n_v} E_{ki}(\delta\Theta, \Theta^{(0)}) \tag{12}$$

and the estimation of Θ (eq. 7) is done iteratively by maximising $\mathcal{C}$ w.r.t $\delta\Theta$ (see algorithm 1).

Algorithm 1 Estimation of Θ (eq. 7).

Input: $m = 0, \Theta^{(0)}$, e, M
  repeat
    $\widehat{\delta\Theta} = \arg\max_{\delta\Theta} \mathcal{C}(\delta\Theta, \Theta^{(m)})$ (algorithm 2)
    $\Theta^{(m+1)} = \Theta^{(m)} + \widehat{\delta\Theta}$
    $m \leftarrow m + 1$
  until $\|\Theta^{(m+1)} - \Theta^{(m)}\| \leq e$ or $m > M$
Output: $\hat{\Theta} = \Theta^{(m)}$

Algorithm 2 Estimation of $\delta\Theta$.

Input: $n = 0, \Theta^{(m)}$, $\delta\Theta^{(0)} = 0$, e, N
  repeat
    $\delta\Theta^{(n+1)} = \mathrm{A}(\delta\Theta^{(n)}, \Theta^{(m)})\ \mathbf{b}(\delta\Theta^{(n)}, \Theta^{(m)})$
    $n \leftarrow n + 1$
  until $\|\delta\Theta^{(n+1)} - \delta\Theta^{(n)}\| \leq e$ or $n > N$
Output: $\widehat{\delta\Theta} = \delta\Theta^{(n)}$

3.2. Mean shift optimisation

An iterative Mean Shift algorithm can then be calculated to optimise the cost function $\mathcal{C}$ in (12) w.r.t. $\delta\Theta$ (algorithm 2). The 3×3 matrix A is defined as:

$$\mathrm{A}(\delta\Theta^{(n)}, \Theta^{(m)}) = \left(\sum_{i=1}^{n_v} \sum_{k=1}^{n_u} \frac{E_{ki}(\delta\Theta^{(n)}, \Theta^{(m)})}{2((h_u^{(k)})^2 + (h_v^{(i)})^2)} DB_{\mathbf{v}^{(i)}}(\Theta^{(m)})^T DB_{\mathbf{v}^{(i)}}(\Theta^{(m)}) \right)^{(-1)} \quad (13)$$

The vector $\mathbf{b}$ is defined as:

$$\mathbf{b}(\delta\Theta^{(n)}, \Theta^{(m)}) = \sum_{i=1}^{n_v} \sum_{k=1}^{n_u} \frac{E_{ki}(\delta\Theta^{(n)}, \Theta^{(m)})}{2((h_u^{(k)})^2 + (h_v^{(i)})^2)} DB_{\mathbf{v}^{(i)}}(\Theta^{(m)})^T \left(\mathbf{u}^{(k)} - B(0, \Theta^{(m)}, \mathbf{v}^{(i)})\right) \quad (14)$$

3.3. Simulated annealing

In order to avoid algorithm 1 to be dependent on the initial guess $\Theta^{(0)}$ and to prevent the estimate of Θ to be trapped in a local maximum, an annealing strategy is also implemented using the bandwidths as a temperature [2]. Starting with large bandwidths, these are decreased iteratively using a geometric rate up to a minimum value (algorithm 3). The limits for the bandwidths, $\{lh_u^{(k)}, lh_v^{(i)}\}$, can be set automatically using nearest neighbours, e.g.

$$lh_u^{(k)} = \arg\min_{k' \neq k} \|\mathbf{u}^{(k')} - \mathbf{u}^{(k)}\|$$

or manually chosen.

3.4. Remarks

In the simpler case when bandwidths are not variable, e.g. $h_u^{(k)} = h_u, \forall k$ and $h_v^{(i)} = h_v, \forall i$, then the bandwidth in the kernel E (eq.11) is also not variable $h = \sqrt{h_u^2 + h_v^2}$ making the simulated annealing algorithm (3) easier to use. Moreover in this case, our cost function to minimise is exactly equivalent to Jian and Vermuri's. (eq. (4)). We show experimentally in the next section that our annealing algorithm converges better than Jian and Vermuri's numerical algorithm for this same cost function. Using variable bandwidths can help in two ways. First, choosing locally larger bandwidths can compensate for missing data and give a better pdf estimate to represent the shape from a set of points. Second, this can also improve the convergence of the Mean Shift algorithm [14].

Algorithm 3 Estimation of Θ with simulated annealing.

Input: $\Theta^{(0)}$, $\{h_u^{(k)}, h_v^{(i)}\}$ large $\forall(k,i)$, $0 < \beta < 1$
  repeat
    $\hat{\Theta} = \arg\max_{\Theta} \mathcal{C}(\Theta)$ (algorithm 1)
    for $k = 1 \rightarrow n_u$ **do**
      if $h_u^{(k)} > lh_u^{(k)}$ **then**
        $h_u^{(k)} \leftarrow \beta\, h_u^{(k)}$
      end if
    end for
    for $i = 1 \rightarrow n_v$ **do**
      if $h_v^{(i)} > lh_v^{(i)}$ **then**
        $h_v^{(i)} \leftarrow \beta\, h_v^{(i)}$
      end if
    end for
  until $h_u^{(k)} < lh_u^{(k)}\ \forall k$ and $h_v^{(i)} < lh_v^{(i)}\ \forall i$
Output: $\hat{\Theta}$

Our algorithm has been extended to 3D point sets. Applications using 3D data captured by a Lidar scan and a Kinect sensor are also reported in the following section.

4. EXPERIMENTAL RESULTS

4.1. 2D data sets

We use for the experiments (as observations) the following data sets: Fish, Contour, Chinese Character and the Road[1]. As reference sets we use the same data but rotated by 50^o, 80^o, 50^o, and 180^o respectively. The problem to solve then is to estimate the transformation parameters that align the two sets of points (observation and reference). In figure 1 we show the results obtained when using Jian and Vermuri's algorithm (top row) and when using our proposed MS algorithm (bottom row).

We first compare in figure 1 (a) the performance of our MS algorithm when using the same setting used in Jian and Vermuri's algorithm (fixed and equals bandwidth for all kernels in the two density functions). We compute the error between the ground truth and the estimated parameters. Results

[1] Available at `http://www.cise.ufl.edu/~anand/students/chui/research.html` [15].

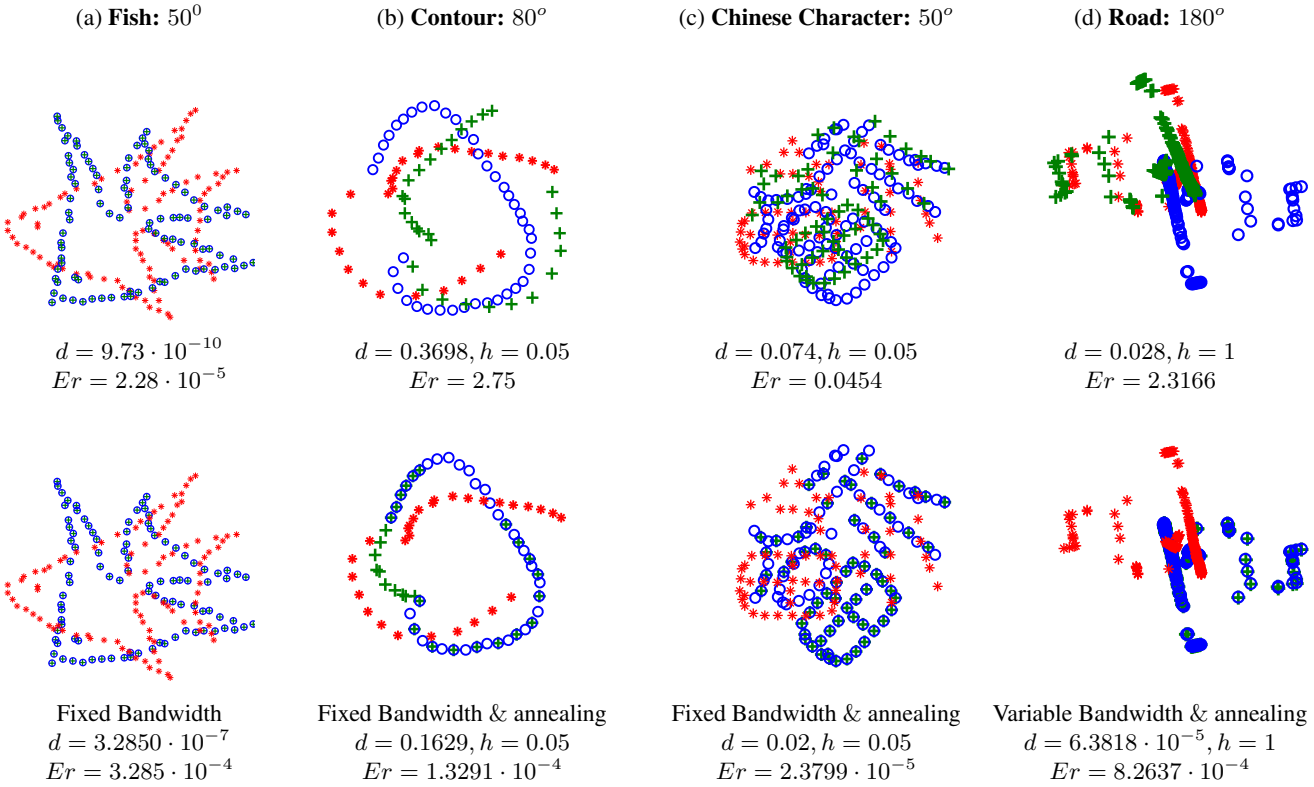


Fig. 1. 2D data sets: alignment obtained when testing the 2D data sets Fish, Contour, Chinese Character and The Road: Reference point set (blue circles), observation (red asterisk) and estimated solution (green cross). The top row of the figure shows results obtained using Jian and Vermuri's algorithm while in the bottom row we show the convergence obtained using our proposed MS algorithm.

are approximately zero in both cases. For comparison we also compute the Euclidean distance d between the two density functions which is close to zero in both cases since the sets are perfectly aligned. Results shows that our algorithm performs similarly to Jian and Vermuri's algorithm when using the same modelling for the cost function.

Figure 1(b and c) shows results when the data sets (observation and reference sets) are not the same set of points (which is the standard scenario in real applications). Moreover, the sets are chosen as non-uniform subsamples from the original data set. In these cases Jian and Vermuri's algorithm fails in estimating the right solution for the transformation parameters while our algorithm converges to the global solution thanks to the annealing strategy implemented (settings: $h_{max} = 2$ and $h_{min} = 0.01$).

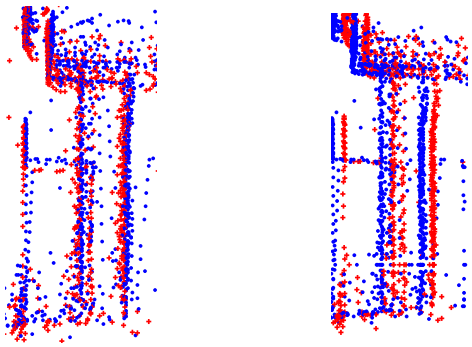
When the shape on the data set is not well encoded in the modelling of the density function, the MS algorithm may not converge to the right solution (even when using the annealing strategy). However, this can be solved by using variable bandwidth when modelling the two density functions. An example is shown in (Figure 1d)) when using the Road data set rotated by 180^o. In this case the dense points on the sets are modelled using a larges bandwidth than the sparse point on the set. This variability of the bandwidth helps in the convergence of the algorithm (Figure 1d) bottom). Both the Euclidean distance d and errror Er are small and this shows how accurate our algorithm is with respect to Jian and Vermuri's.

4.2. Applications using Real 3D Data sets

3D Lidar scan alignment: We use 93 Lidar scans from the *Hannover1* database[2] and estimate the transformation parameters between consecutive pairs. We compare the value of the Euclidean distance between the density functions when using our estimated solution $\widehat{\Theta}$ and when using the ground truth Θ_{GT}. For the 92 pairs computed we conclude with 99.9% of confidence that our estimate $\widehat{\mathbf{\Theta}}$ minimizes the distance better between the two pdfs even when relaxing the bandwidth. Figure 2 presents a close up when aligning two scans using our estimate of $\mathbf{\Theta}$ and using the ground truth.

3D Kinect scan alignment: We use our algorithm for aligning data captured from multiple views using the Kinect sensor. The resulting mesh when many views are used (Fig.

[2]Captured by Oliver Wulf from the Leibniz University and available at `http://kos.informatik.uni-osnabrueck.de/3Dscans/`.



(a) (b)

Fig. 2. Details of the alignment between buildings. a) using our proposed algorithm for aligning the scans and b) using ground truth.

3 b) shows less holes and appears smoother than the mesh generated from only one Kinect scan (Fig. 3 a).

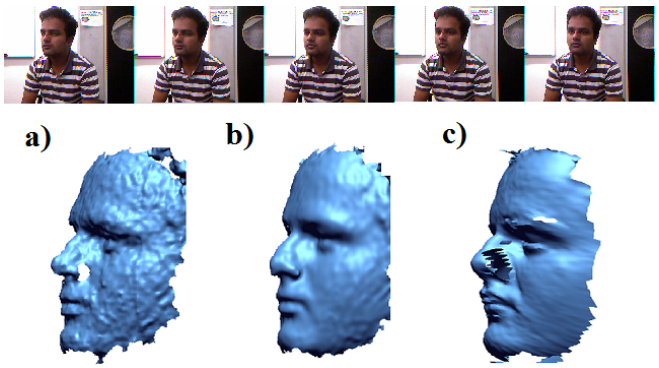


Fig. 3. **Kinect scan alignment for a face.** The top row shows a sequence of images captured from different points of view. Bottom row: Mesh created from (a) a single acquisition, (b) multiples views and (c) face captured using a laser scanner.

5. CONCLUSIONS

We have presented a Mean Shift algorithm for solving the robust point set registration proposed by Jian and Vermuri [1]. We use as a cost function the closed form solution for the Euclidean distance between the Mixture of Gaussians. The resulting algorithm is shown to be more robust to Jian and Vermuri's thanks to the annealing framework implemented and the use of variable bandwidth for modelling the density functions. The use of variable bandwidth compensates the limitation of the isotropic covariance in the modelling of the density function allowing better representations of the encode shape in the data set. Additional experiments in 3D real data were also reported, demonstrating the performance of our algorithm in real applications.

6. REFERENCES

[1] B. Jian and B. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1633–1645, 2011.

[2] C. Shen, M.J. Brooks, and A. van den Hengel, "Fast global kernel density mode seeking: Applications to localization and tracking," *IEEE Transactions on Image Processing*, vol. 16, 2007.

[3] B. Srinivasan, H. Qi, and R. Duraiswami, "Gpuml: Graphical processors for speeding up kernel machines," *Workshop on High Performance Analytics - Algorithms, Implementations, and Applications, Siam Conference on Data Mining*, 2010.

[4] D.W. Scott, "Parametric statistical modeling by minimum integrated square error," *Technometrics*, vol. 43, pp. 274–285, 2001.

[5] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992.

[6] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. Journal Comput. Vision*, vol. 13, pp. 119–152, 1994.

[7] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," *International Conference on 3-D Digital Imaging and Modeling*, 2001.

[8] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," *ACM Trans. Graph.*, vol. 30, pp. 1–10, 2011.

[9] A. Myronenko and Xubo S., "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2262–2275, 2010.

[10] Y. Tsin and T. Kanade, "A Correlation-Based Approach to Robust Point Set Registration," *European Conference in Computer Vision ECCV*, pp. 558–569,2004.

[11] E. Hasanbelliu, L. Sanchez Giraldo, and J. Principe, "A robust point matching algorithm for non-rigid registration using the cauchy-schwarz divergence," *IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6,2011.

[12] A. Basu, I. Harris, N. Hjort, and M. Jones, "Robust and efficient estimation by minimising a density power divergence," *Biometrika*, vol. 85, no. 3, pp. 549–559, 1998.

[13] M. Helén and T. Virtanen, "Audio query by example using similarity measures between probability density functions of features," *EURASIP Journal on Audio, Speech, and Music Processing*, 2010.

[14] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *IEEE International Conference on Computer Vision, ICCV*, vol. 1, pp. 438–445,2001.

[15] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 190 –197,2000.