Shape Model fitting using Non-Isotropic GMM

Claudia Arellano[†] **and Rozenn Dahyot**[*]

School of Computer Science and Statistics
Trinity College Dublin,
IRELAND

E-mail: [†] arellanc@scss.tcd.ie [*]Rozenn.Dahyot@scss.tcd.ie

Abstract — **We present a Mean Shift algorithm for fitting shape models. This algorithm maximises a posterior density function where the likelihood is defined as the Euclidean distance between two Gaussian mixture density functions, one modelling the observations while the other corresponds to the shape model. We explore the role of the covariance matrix in the Gaussian kernel for encoding the shape of the model in the density function. Results show that using non-isotropic covariance matrices improve the efficiency of the algorithm and allow to reduce the number of kernels to use in the mixture without compromising the robustness of the algorithm.**

Keywords — **Morphable Models, Fitting Algorithm, Gaussian Mixture Models, Mean Shift.**

I Introduction

Shape models have been widely used in image analysis for detection, reconstruction and recognition. Algorithms for fitting these shape models aim at being robust, accurate and automatic. In our previous work [1] we introduced a robust Mean-shift algorithm that fits a morphable shape model without correspondences. The cost function is defined in a Bayesian framework where the likelihood is chosen proportional to the Euclidean distance between two density functions, one modelling the observations and the other corresponding to the shape model. Both density functions were modelled as mixtures of Gaussians with isotropic covariance matrices. In this paper, we explore the role of the covariance matrices when modelling the density function of the shape model. We show that the shape model with non-isotropic covariance matrices better represents the class of shapes of interest. Moreover the resulting fitting algorithm improves robustness. Finally we show that the number of Gaussians in the model can be reduced efficiently without loss of accuracy.

II Related work

It has been argued that algorithms for reconstruction and recognition of shapes improve their performance when using prior knowledge such as shape models. Cootes et al. [2], for instance, have proposed the 2D Active Shape Model for feature detection and image segmentation. Blanz et al. [3] have introduced a 3D Model of faces. This 3D model has been successfully used as a prior for inferring 3D faces from images [4, 5, 6]. The problem of fitting the model to a set of observations is often solved by maximising the posterior probability in a Bayesian framework where the likelihood requires a known correspondence between the shape model and the observations [7, 8]. We recently proposed a new algorithm [1] that does not require correspondence for fitting a shape model to a point cloud. It is also based on a Bayesian framework but uses a robust likelihood [9] that is defined as the Euclidean distance between two mixtures of Gaussian density functions modelled using the observations and the shape model respectively. Each kernel is centred at each point of the data set and uses an isotropic covariance matrix. While the framework is robust and automatic, the computational time for the fitting is proportional to the number of kernels used for modelling both density functions $O(nm)$ (for n observation and m vertices in the model). Jian et al. [10] have previously used the Euclidean distance between density functions for robust registration. They have proposed to reduce the computation time by approximating the analytical solution of the Euclidean distance by the Gauss Transform. However, this approximation affects the robustness of the algorithm since it is only accurate when the points in the observation sets are uniformly sampled. In this paper, we extend our Bayesian framework by changing from isotropic Gaussian kernels to non-isotropic covariance matrices. This new modelling approach allows us to reduce the number of kernels to use and therefore to improve efficiency of the algorithm without compromising robustness. The paper is organised as follows: section III a) presents the Bayesian framework for robust shape fitting [1]. In section III b), we define the new approach for modelling the density function of the shape model. In section III c), we present the cost function and the Mean Shift Algorithm used for optimisation. Results and discussion are presented in section IV where a 2D hand model is fitted to point clouds extracted from images.

III Mean Shift for Bayesian shape fitting

a) Bayesian framework for model fitting using a robust likelihood

Let us define a shape model by its mean shape $\boldsymbol{\mu}$ and a set of J eigenvectors $\{\mathbf{T}_j\}_{j=1,..,J}$ associated with the eigenvalues $\{\sigma_j\}_{j=1,..,J}$ computed by Principal Component Analysis using a representative set of exemplar shapes. We assume that any shape from the same class can be well approximated as a linear combination of $(\boldsymbol{\mu}, \{\mathbf{T}_j\}_{j=1,..,J})$:

$$\mathbf{y}(\alpha) = \boldsymbol{\mu} + \sum_{j=1}^{J} \alpha_j \ \mathbf{T}_j \tag{1}$$

The reconstructed shape $\mathbf{y}(\alpha)$ depends on the latent parameters $\alpha = \{\alpha_j\}_{j=1,..,J}$. Given a set of observations $\mathcal{U} = \{\mathbf{u}_k\}_{k=1,\cdots,n}$, we aim at estimating the parameters $\{\alpha_j\}_{j=1,..,J}$ such that $\mathbf{y}(\alpha)$ best fits the observations. Using a Bayesian framework the parameters are estimated by maximising the posterior:

$$\hat{\alpha} = \arg\max_{\alpha} p(\mathcal{U}|\alpha) \ p(\alpha) \tag{2}$$

with $p(\alpha)$ the prior and $p(\mathcal{U}|\alpha)$ the likelihood. The prior is chosen here as a multivariate Gaussian and it is expressed as follows:

$$p(\alpha = (\alpha_1, \cdots, \alpha_J)) \propto \exp\left(-\frac{1}{2}\sum_{j=1}^{J}\frac{\alpha_j^2}{\sigma_j^2}\right) \tag{3}$$

We have proposed in [1] to model the likelihood using the distance between two density function. Here, one density function is modelled using the shape model f_α and the other using the observations f_u. Loosely speaking, f_u and f_α can be understood as infinite dimensional vectors and the likelihood can be expressed using the Euclidean distance between those two density functions:

$$p(\mathcal{U}|\alpha) = p(f_u|f_\alpha) \propto \exp\left(-\frac{(\|f_u - f_\alpha\|)^2}{2\sigma_d^2}\right) \tag{4}$$

The Euclidean distance is here defined by:

$$\begin{aligned} \|f_u - f_\alpha\|^2 &= \textstyle\int_{\mathbb{R}^D} (f_u(\mathbf{x}) - f_\alpha(\mathbf{x}))^2 \ d\mathbf{x} \\ &= \textstyle\int_{\mathbb{R}^D} \left(f_u(\mathbf{x})^2 - 2f_u(\mathbf{x})f_\alpha(\mathbf{x})\right. \\ &\quad \left. +f_\alpha(\mathbf{x})^2\right) \ d\mathbf{x} \end{aligned} \tag{5}$$

Using equation (4) in equation (2), our parameter estimation is formulated as:

$$\hat{\alpha} = \arg\min_{\alpha}\left\{\frac{\|f_u - f_\alpha\|^2}{2\sigma_d^2} + \sum_{j=1}^{J}\frac{\alpha_j^2}{2\sigma_j^2}\right\} \tag{6}$$

The variance σ_d^2 is set experimentally and allows us to control the influence of the likelihood with the prior.

b) Modelling the density functions f_u and f_α

We previously modelled the density function of the shape model f_α as a Gaussian mixture where each kernel is centred at each point of the data set with an isotropic bandwidth (see Figure 1b) [1]. Since the shape model is a set of connected points, the density function can be redefined by using the spatial relation between vertices. Therefore, changes in direction of the contour shape can be modelled by adapting Gaussian kernels according to the direction given by consecutive points. We propose to model the density function using $m-1$ non-isometric Gaussian kernels as follows.

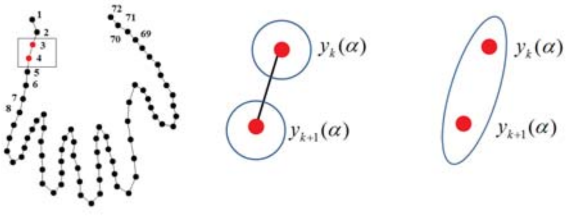
$$f_\alpha(\mathbf{x}) \propto \sum_{k=1}^{m-1} w_k \ G(\mathbf{x}, \mathbf{r}_k(\alpha), \textstyle\sum_k) \tag{7}$$

where $\mathbf{r}_k(\alpha) = \frac{\mathbf{y}(\alpha)_k + \mathbf{y}(\alpha)_{k+1}}{2}$. The covariance matrix $\sum_k$ of each kernel is a positive definite matrix that can be decomposed into the form $\sum_k = \mathbf{QAQ}^T$. Where $\mathbf{Q}$ is the matrix containing the eigenvectors and $\mathbf{A}$ is a diagonal matrix of eigenvalues. The eigenvectors represent the principal direction $\overrightarrow{n}_1$ and its normal $\overrightarrow{n}_2$. The eigenvalues control the fuzziness of the kernel in both directions:

$$\textstyle\sum_k = (\ \overrightarrow{n}_1 | \overrightarrow{n}_2\ ) \begin{pmatrix} a^2 & 0 \\ 0 & b^2 \end{pmatrix} (\ \overrightarrow{n}_1 | \overrightarrow{n}_2\ )^T \tag{8}$$

From Figure 2a, we can compute $\overrightarrow{n}_1$ and $\overrightarrow{n}_2$ as follows:

$$\overrightarrow{n}_1 = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \text{ and, } \ \overrightarrow{n}_2 = \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \end{pmatrix} \tag{9}$$



(a) Shape Model (b) Isotropic (c) Non-isotropic

Fig. 1: Exemplar shape (a) when modelling isotropic Kernels (b) and when using non-isotropic kernels (c)

with,

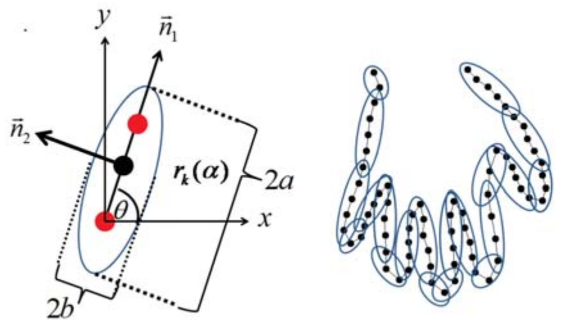
$$\cos(\theta) = \frac{\mathbf{y}_k^{(x)}(\alpha) - \mathbf{y}_{k+1}^{(x)}(\alpha)}{d} \tag{10}$$

$$\sin(\theta) = \frac{\mathbf{y}_k^{(y)}(\alpha) - \mathbf{y}_{k+1}^{(y)}(\alpha)}{d} \tag{11}$$

where $d = \|\mathbf{y}_k(\alpha) - \mathbf{y}_{k+1}(\alpha)\|$. The parameter a controls the overlap between consecutive kernels and it can be defined as a function of the distance d, such as $a = \tau d$, with τ controlled by the user. The fuzziness along the normal ($\overrightarrow{n}_2$) is defined as $b = h$, where h is the bandwidth we choose for all the kernels and it is related to the error we are willing to tolerate in the optimisation problem. In order to achieve a homogeneous shape for the density function, we defined the weight of each kernel as $w_k = \sqrt{(2\pi)^D \, |\sum_k|}$ and the final density function is normalised by $\sum_{k-1}^{m-1} w_k$. The resulting density function of the shape model is now defined over the mean of consecutive points. The mean of the model becomes $\mathbf{r}_k(\alpha = 0)$ and the eigenvectors are updated as follows:

$$\mathbf{v}_{jk} = \frac{\mathbf{T}_{jk} + \mathbf{T}_{jk+1}}{2}, \forall j \text{ and for } k = 1, ...n-1 \tag{12}$$

Reducing the number of kernels in the modelling of f_α: Thanks to the introduction of the non-isotropic covariance in the model, we can reduce the number of vertices (number of kernels to use) without altering the shape of the density function (see figure 2b). The modelling for each kernel given a set of vertices to consider $\{\mathbf{y}_l\}_{l=l_1,..,l_2}$ is then defined by its mean $\mathbf{r}_k(\alpha)$ and the covariance matrix ($\sum_k$). The mean is computed as the mean of the set of points ($\{\mathbf{y}_j\}_{j=l_1,..,l_2}$). The covariance matrix follows the same definition given in equation (8). Where $\overrightarrow{n}_1$ and $\overrightarrow{n}_2$ correspond to the eigenvectors of the sets of points $\{\mathbf{y}_l\}$ and can be found using PCA. The eigenvector of the model $\mathbf{v}_{jk}$ associates with the new vertex $\mathbf{r}_k(\alpha)$ is updated as the mean of the contribution of the eigenvector of each point in the set $\{\mathbf{y}_l\}$. This can be expressed as follows:



(a) Geometry (b) Reducing Kernels

Fig. 2: In (a) an scheme of the geometry used for computing the main direction of the ellipse and its normal is shown. Figure (b) shows an example of reducing number of kernels to represent the encoded shape in the data set

$$\mathbf{v}_{jk} = \frac{\sum_{l=l_1}^{l=l_2} \mathbf{T}_{jl}}{(l_2 - l_1 + 1)} \tag{13}$$

Figure 3 shows a set of density functions modelled using the average shape when reducing the number of Gaussian kernels. In the top row as reference we compute the density function using isometric covariance matrix for bandwidths of $30, 20, 10$ and 5. In the second row and for the same bandwidths, we have the density function of the model computed using non-isometric covariance matrix and considering all the vertices of the model (72 vertices, 71 kernels). From the third row until the bottom, we show the density function computed when using 56, 39 and 31 kernels respectively (for bandwidth of $30, 20, 10$ and 5).

Modelling f_u: The density function computed from the observations $\mathcal{U} = \{\mathbf{u}_k\}_{k=1,\cdots,n}$ is modelled with isotropic kernels centred at each data point. Indeed the connections between vertices are unknown in the observations and some observations are outliers (points that do not belong to the shape of interest). The density function f_u is modelled using n Gaussian kernels centred at each point of the data set, with the isotropic covariance matrix $\sum_i^{'} = h_i^2 I$ (I the identity matrix):

$$f_u(\mathbf{x}) = \sum_{i=1}^{n} w_i' \; G(\mathbf{x}, \mathbf{u}_i, \textstyle\sum_i'), \;\forall i \;\; w_i' = \frac{1}{n} \tag{14}$$

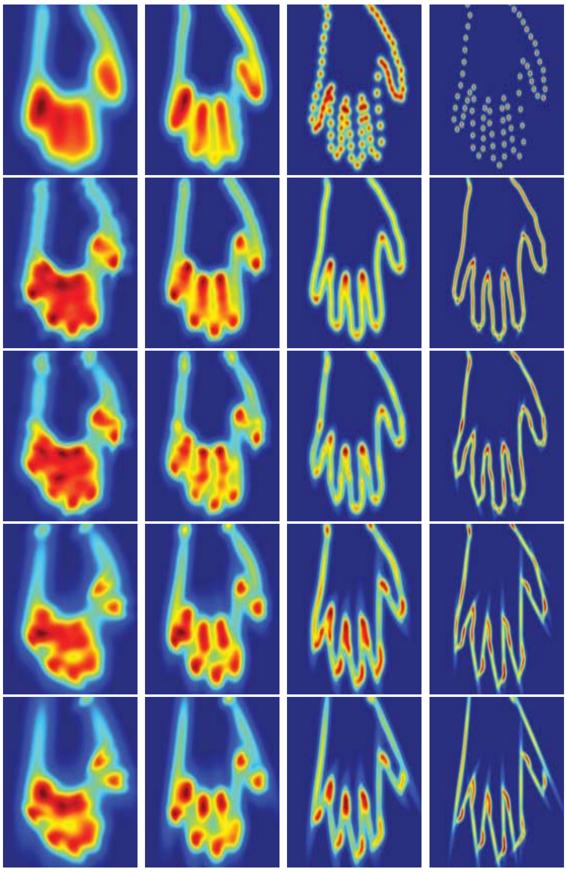


Fig. 3: Density functions computed from the hand model when using a isotropic kernels (top row) and using a non-isotropic kernels (second row). From the third row to the bottom, density functions modelled when using 56, 39 and 31 kernels respectively.

c) *Mean Shift Algorithm for Gaussian Mixtures*

The integral of the product of two Gaussian density function modelled using non-isotropic covariance matrix has a closed form solution. This solution is given by the following expression:

$$\int_{\mathbb{R}^D} G(\mathbf{x}, \mu_A, \Sigma_A) G(\mathbf{x}, \mu_B, \Sigma_B) d\mathbf{x} = \frac{1}{(2\pi)^{\frac{D}{2}} \sqrt{|\Sigma_A + \Sigma_B|}} \exp\left(-\frac{q(\mu_A, \mu_B)}{2}\right) \quad (15)$$

where,

$$\begin{aligned} & q(\mu_A, \mu_B) = \mu_A^T \Sigma_A^{-1} \mu_A + \mu_B^T \Sigma_B^{-1} \mu_B - \mu_C^T \Sigma_C^{-1} \mu_C \\ & \Sigma_C^{-1} = (\Sigma_A^{-1} + \Sigma_B^{-1}) \\ & \mu_C = \Sigma_C (\Sigma_A^{-1} \mu_A + \Sigma_B^{-1} \mu_B) \end{aligned} \quad (16)$$

Using these expressions (15 and 16), we can solve the Euclidean distance between $f_u(\mathbf{x})$ and $f_\alpha(\mathbf{x})$ proposed in equation (5) as follows:

$$\begin{aligned} & \|f_u - f_\alpha\|^2 = \\ & \sum_{i=1}^{n} \sum_{p=1}^{n} \underbrace{\frac{w'_i w'_p}{(2\pi)^{\frac{D}{2}} \sqrt{|\Sigma'_i + \Sigma'_p|}} \exp\left(-\frac{q(\mathbf{u}_i, \mathbf{u}_p)}{2}\right)}_{E_1(\mathbf{u}_i, \mathbf{u}_p)} \\ & -2 \sum_{k=1}^{m-1} \sum_{i=1}^{n} \underbrace{\frac{w_k w'_i}{(2\pi)^{\frac{D}{2}} \sqrt{|\Sigma_k + \Sigma'_i|}} \exp\left(-\frac{q(\mathbf{r}(\alpha)_k, \mathbf{u}_i)}{2}\right)}_{E_2(\mathbf{r}(\alpha)_k, \mathbf{u}_i)} \\ & + \sum_{k=1}^{m-1} \sum_{p=1}^{m-1} \underbrace{\frac{w_k w_p}{(2\pi)^{\frac{D}{2}} \sqrt{|\Sigma_k + \Sigma_p|}} \exp\left(-\frac{q(\mathbf{r}(\alpha)_k, \mathbf{r}(\alpha)_p)}{2}\right)}_{E_3(\mathbf{r}(\alpha)_k, \mathbf{r}(\alpha)_p)} \end{aligned} \quad (17)$$

Replacing the Euclidean distance (17) in equation (6) we rewrite the energy function $E(\alpha)$ as follows:

$$\begin{aligned} E(\alpha) = & \sum_{i=1}^{n} \sum_{p=1}^{n} E_1(\mathbf{u}_i, \mathbf{u}_p) - 2 \sum_{k=1}^{m-1} \sum_{i=1}^{n} E_2(\mathbf{r}(\alpha)_k, \mathbf{u}_i) \\ & + \sum_{k=1}^{m-1} \sum_{p=1}^{m-1} E_3(\mathbf{r}(\alpha)_k, \mathbf{r}(\alpha)_p) + \lambda \sigma_d^2 \sum_{j=1}^{J} \frac{\alpha_j^2}{2\sigma_j^2} \end{aligned} \quad (18)$$

where the parameter λ is introduced to let the user control the influence of the likelihood with the prior. The parameters α of the model that best fit the observation can then be estimated by minimising equation (18). The Mean Shift Algorithm is computed by differentiating the energy function E with respect to α and equalling the result to zero. Starting from an initial guess $\alpha^{(t)}$, the update is computed by:

$$\alpha^{(t+1)} = \mathrm{A}(\alpha^{(t)})^{-1} \mathbf{b}(\alpha^{(t)}) \quad (19)$$

with A a $J \times J$ matrix defined as:

$$A_{j,s}(\alpha) = \begin{cases} L_{j,s}(\alpha), & \text{if } j \neq s \\ L_{j,s}(\alpha) + \frac{\lambda \sigma_d^2}{\sigma_j^2}, & \text{if } j = s \end{cases} \quad (20)$$

The expression for L and b are presented in Table 1. For simplicity $E_2(\mathbf{r}(\alpha)_k, \mathbf{u}_i)$ and $E_3(\mathbf{r}(\alpha)_k, \mathbf{r}(\alpha)_p)$ are expressed as E_2^{ki} and E_3^{kp} respectively. The parameter $\mathbf{r}_o$ corresponds to the average shape of the model $\mathbf{r}(\alpha = 0)$ and the covariance matrices $\Sigma_{c(ki)}$ are computed for each pair of distributions (k, i) as it was illustrated in equation (16).

The Mean Shift algorithm is presented in algorithm 1 with its annealing strategy. We use the parameter h defined for f_α and f_u in section b) as temperature. Starting from a maximum value h_{max}, the bandwidth is decreased using a geometric rate β until the minimum value h_{min} is reached. Note that the covariance matrix of each kernel in the density function of the model is updated at each iteration.

Table 1: Expressions for computing $L_{(j,s)}$ and $b_{(j)}$

$L_{(j,s)}$
$= 2\sum_{k=1}^{m-1}\sum_{i=1}^{n} E_2^{ki}(\mathbf{v}_{jk}^T\Sigma_k^{-1} - \mathbf{v}_{jk}^T\Sigma_k^{-1}\Sigma_{c(ki)}^{-1T}\Sigma_k^{-1T})\mathbf{v}_{sk}$
$- \sum_{k=1}^{m-1}\sum_{p=1}^{m-1} E_3^{kp}(\mathbf{v}_{jk}^T\Sigma_k^{-1}\mathbf{v}_{sk} + \mathbf{v}_{jp}^T\Sigma_p^{-1}\mathbf{v}_{sp})$
$+ \sum_{k=1}^{m-1}\sum_{p=1}^{m-1} E_3^{kp}(\mathbf{v}_{jk}^T\Sigma_k^{-1} + \mathbf{v}_{jp}^T\Sigma_p^{-1})\Sigma_{c(kp)}^{-1T}\Sigma_p^{-1T}\mathbf{v}_{sp}$
$+ \sum_{k=1}^{m-1}\sum_{p=1}^{m-1} E_3^{kp}(\mathbf{v}_{jk}^T\Sigma_k^{-1} + \mathbf{v}_{jp}^T\Sigma_p^{-1})\Sigma_{c(kp)}^{-1T}\Sigma_k^{-1T}\mathbf{v}_{sk}$

$\mathbf{b}_{(j)}$
$= -2\sum_{k=1}^{m-1}\sum_{i=1}^{n} E_2^{ik}(\mathbf{r}_{ok}^T\Sigma_k^{-1})\mathbf{v}_{jk}$
$+2\sum_{k=1}^{m-1}\sum_{i=1}^{n} E_2^{ik}((\mathbf{u}_i^T\Sigma_i'^{-1} + \mathbf{r}_{ok}^T\Sigma_k^{-1})\Sigma_k^{-1}\Sigma_{c(ki)}^{-1T})\mathbf{v}_{jk}$
$- \sum_{k=1}^{m-1}\sum_{p=1}^{m-1} E_3^{kp}(\mathbf{r}_{ok}^T\Sigma_k^{-1} + \mathbf{r}_{op}^T\Sigma_p^{-1T})\Sigma_{c(kp)}^{-1T}\Sigma_k^{-1T}\mathbf{v}_{jk}$
$- \sum_{k=1}^{m-1}\sum_{p=1}^{m-1} E_3^{kp}(\mathbf{r}_{ok}^T\Sigma_k^{-1} + \mathbf{r}_{op}^T\Sigma_p^{-1T})\Sigma_{c(kp)}^{-1T}\Sigma_p^{-1T}\mathbf{v}_{jp}$
$+ \sum_{k=1}^{m-1}\sum_{p=1}^{m-1} E_3^{kp}(\mathbf{r}_{ok}^T\Sigma_k^{-1}\mathbf{v}_{jk} + \mathbf{r}_{op}^T\Sigma_p^{-1T}\mathbf{v}_{jp})$

Algorithm 1 Estimation of α

```
Require: $\alpha_j^{(0)} = 0, \forall j$,   $h_{min}$, $h_{max}$, $e_o$,  $\tau = 0.5$
  $\lambda = 0.05$ and $\beta = 0.8$
  $h = h_{max}$
  repeat
    $\sigma_d^2 = \|f_u - f_\alpha\|^2$
    repeat
      Compute $A(\alpha^{(t)})$ and $\mathbf{b}(\alpha^{(t)})$
      $\alpha^{(t+1)} = A(\alpha^{(t)})^{-1}\mathbf{b}(\alpha^{(t)})$
      Compute $\Sigma_k$  $\forall k$
    until $|\alpha^{(t+1)} - \alpha^{(t)}| \leq$ $e_o$
    $h \leftarrow \beta h$
  until $h \leq h_{min}$
```

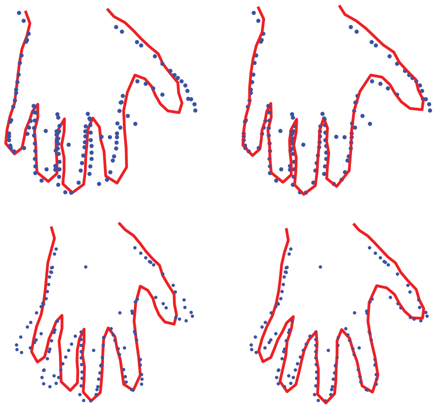
IV Experimental results

a) *Experimental setting*

A hand model has been trained using the annotated data sets of 18 hands provided by Tim Cootes[1]. Using our Mean Shift algorithm, the model is fitted to 2D point clouds extracted from images using an edge detector (e.g. figure 5). The rigid registration between the observations and the model is first performed as a pre-process [11] in all experiments.

b) *Non-Isotropic Vs Isotropic Kernels*

The 2D point cloud used as observation consists in 120 points taken randomly from the edge map of 2D images of hands. The isotropic shape model uses 72 isotropic kernels and the non-isotropic shape model uses 71 non-isotropic kernels. The settings for the MS algorithm are $h_{max} = 25$ and $h_{min} = 5$. Figure 4 compares the fitting using both shape models on two examples. The estimated shapes (red line) are better using the non-isotropic model as they are closer to the observations (blue dots).



(a) (b)

Fig. 4: Estimated shapes (in red solid line) using the isotropic shape model (a) and the non-isotropic shape model (b). The observations are shown as blue dots.

c) *Impact of the number of kernels in the models*

The performance of the algorithm is assessed when reducing the number of kernels in the shape models. Figure 5 shows the estimated shapes using the isotropic and non-isotropic shape models with 71, 54 and 43 kernels. As can be seen the non-isotropic shape model performs better than the isotropic one, as its solution is closer to the observations. As the number of kernels decrease, the isotropic shape model loses robustness. We compute the Euclidean distance between the estimated shapes and the observations as a quantitative measure for similarity in between the shapes. Results are reported in figure 6. The non-isotropic model minimises better the Euclidean distance. The robustness of the algorithm is improved by using non-isotropic kernels, and the number of kernels can be chosen to reduce the computation time without loss of accuracy.

V Conclusions

We have presented a Mean Shift Algorithm that fits a shape model to a point cloud in a Bayesian framework. The likelihood is defined with the Euclidean distance between two Mixtures of Gaussians corresponding to the model and the observation. While the observation uses Gaussians with isotropic covariance matrices, we have shown that the model can be better represented by non-isotropic covariances. This non isotropic modelling increases the robustness of the fitting algorithm.

[1] `http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/data/hand_data.html`

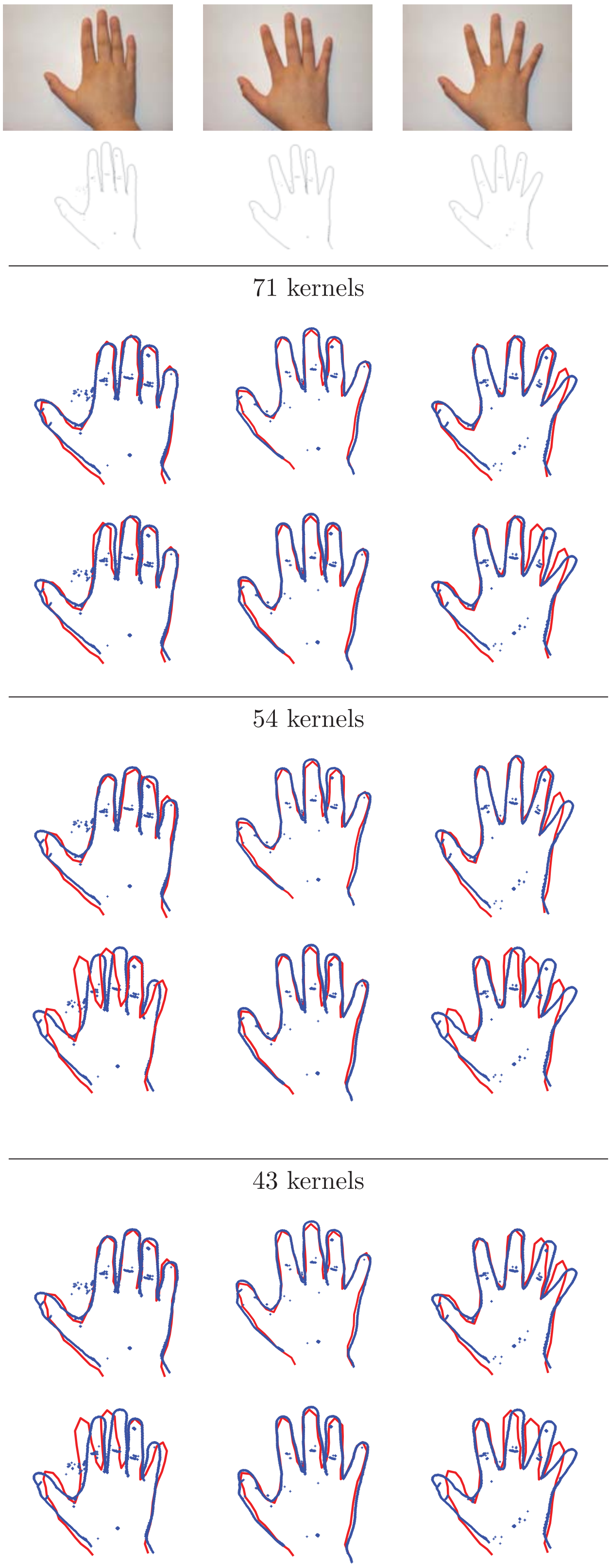


Fig. 5: From top to bottom: colour images, extracted edges (observations appearing as blue dots next), estimated shapes (in solid red line) using 71, 54 and 43 kernels. Solutions from the non isotropic shape model are shown on top of the ones from the isotropic shape model in each case.

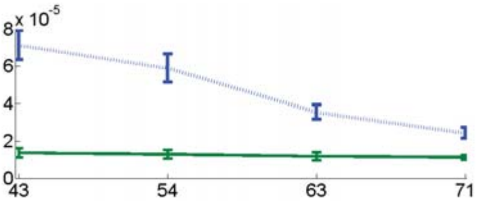


Fig. 6: Average and standard deviation of the Euclidean distance between the estimated shape and the observations computed using the three examples shown in Figure 5. The green line correspond to the non-isotropic model and the blue dots when using the isotropic model. The abscissa corresponds to number of kernels used in the model.

References

[1] C. Arellano and R. Dahyot. Shape model fitting without point correspondence. *20th European Signal Processing Conference (Eusipco)*, 2012.

[2] T. Cootes, C. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61:38 – 59, 1995.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. *Proceedings of the annual conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, pages 187–194, 1999.

[4] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. *European Conference on Computer Vision, ECCV*, pages 3–19, 2002.

[5] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1063–1074, 2003.

[6] X. Wang, W. Liang, and L. Zhang. Morphable face reconstruction with multiple views. *International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC*, pages 250–253, 2010.

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

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

[9] D.W. Scott. Parametric statistical modeling by minimum integrated square error. *Technometrics*, 43:274–285, 2001.

[10] B. Jian and B. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[11] C. Arellano and R. Dahyot. Mean shift algorithm for robust rigid registration between gaussian mixture models. *20th European Signal Processing Conference (Eusipco)*, 2012.