# Robust Bayesian Fitting of 3D Morphable Model

Claudia Arellano
School of Computer Science and Statistics
Trinity College Dublin
Dublin 2, Ireland
arellanc@scss.tcd.ie

Rozenn Dahyot
School of Computer Science and Statistics
Trinity College Dublin
Dublin 2, Ireland
Rozenn.Dahyot@tcd.ie

## ABSTRACT

We propose to fit automatically a 3D morphable face model to a point cloud captured with a RGB-D sensor. Both data sets, the shape model and the target point cloud are modelled as two probability density functions (pdfs). Rigid registration (rotation and translation) and reconstruction on the model is performed by minimising the Euclidean distance between these two pdfs augmented with a multivariate Gaussian prior. Our resulting process is robust and it does not require point to point correspondence. Experimental results on synthetic and real data illustrates the performance of this novel approach.

## General Terms

Computer Vision

## Keywords

Morphable Models, Shape Fitting, 3D Face Reconstruction, Registration, Divergence, L2E, RGB-D sensor.

## 1. INTRODUCTION

Face reconstruction is a very attractive research area as it forms the basis of a wide range of applications such as face animation, human-computer interfaces, face recognition or medical applications (e.g. plastic surgery). Methods for reconstructing 3D faces can be categorised depending on the nature of the input data (data capture system) and their applications. The most accurate system to capture a 3D face is the laser scanner, which records a dense 3D map. Unfortunately its high cost and the need for cooperation from the individual during the acquisition process has limited its usage. Some cheaper alternatives rely on the inference of the 3D face geometry from RGB images, where multiple cues from images or multiples views of the scene are used for inferring the 3D reconstruction. These approaches perform well but mainly in controlled environments. More realistic conditions are still challenging to deal with such as the variability of the head pose, the facial expressions or the illumination conditions.

A popular alternative for capturing a 3D map is by using an RGB-D sensor. Two technologies of RBG-D sensors co-exist: time of flight cameras (expensive) and structured light cameras (cheap). Both provide a noisy recording of the 3D environment, however they have become a hot topic of research in recent years as an alternative for 3D shape reconstruction. In particular, the availability of commercial devices such as the Microsoft Kinect (sensor using structured light) at a very low cost has triggered the development of many algorithms for improving its performance and extending its usage in many new applications other than video games for which the Kinect was originally designed as an extension of the Xbox console.

Scanning 3D faces with an RGB-D sensor does not provide accurate results. The recorded data is very noisy and the depth map contains holes or missing data due the limitations of the sensor. Two approaches have been recently explored in the literature for dealing with those problems. The first one relies on the use of multiple acquisitions and combines them in order to obtain a high resolution 3D shape. The second uses a 3D model that is fitted to the captured data. Results achieved when fitting shape models (Morphable Model of the face) have shown better performance since they are capable of inferring details of the face such as the mouth or eyes. These details are usually lost when inferring a 3D shape by averaging over all the registered depth scans.

The performance of most fitting algorithms depends on the initial correspondences between the observed data and the shape model. For instance, the representation of a particular feature in the observation (i.e tip of the nose) must be in correspondence with the vertex representing the same feature in the Shape Model. As a consequence most fitting algorithms depends on feature selection or landmarking (required for initialisation) and this is limiting their automation and with that, their applications.

We propose here to fit a morphable model by using a cost function that does not require any correspondences to be made. This cost function corresponds to the distance between two probability density functions, one modelled using the shape model and the other using the observed data [3, 5, 4]. Inferring the model parameter with this distance allows for the approach to be unsupervised and also robust. Our approach to 3D face morphable model fitting can be explained in two steps. The first step consists in aligning the observed point cloud with the average shape model. In other words this first step focuses on estimating the rigid transformation to match the observations with the average shape of the morphable model. The second step aims at fitting the morphable model by estimating the best reconstruction on the basis of its eigenvectors. The

mathematics used in both steps are based on pdf matching techniques (reviewed in paragraph 2.3.4) extended for Gaussian Mixtures (section 3.2) in a Bayesian framework (section 3.4). We do not perform any kind of filtering which would reduce details in the input data, and we do not need any correspondences in between the two shapes (observations and the model shape). The resulting system is unsupervised and robust.

This paper is structured as follows. In section 2 we review the state of the art in 3D face reconstruction. Sections 3 and 4 describe our modelling to solve 3D face inference using an RGB-D camera. Experimental results are reported in section 5 and section 6 discusses future improvements.

# 2. RELATED WORK

This section reviews several methods for reconstructing 3D faces from RGB and RGB-D images. We also present a brief overview on registration algorithms since it is a key process of most fitting algorithms and it is the basis of our approach to 3D reconstruction presented in section 3.

## 2.1 3D face reconstruction from RGB images

The inference of the 3D face shape from RGB images has been deeply explored in the past decade. Systems for capturing RGB images are cheap, fast and noninvasive. However, the reconstruction process from images involves a set of challenging difficulties such as head pose, illumination and face expressions among others. Several algorithms addressing those challenges have been reported in literature based on shape from X techniques and analysis by synthesis.

Shape from X methods are algorithms that use specific cues (X) to infer the 3D shape [12]. The cues used are usually motion, shading, stereo and silhouettes among others [17, 25, 48, 26]. Even though those methods have been successfully applied to 3D object reconstruction, most cannot provide a realistic estimation of the 3D face shape. Using an array of video cameras, Bradley et al. [11] succeeds, however, in capturing 3D textured faces by combining multi-view stereo with tracking techniques.

Analysis by synthesis on the other hand has proven to achieve better results due to the use of prior information. It uses a morphable model of the face built over a set of observations (training database) [10, 35, 36, 7]. The problem can then be defined in a Bayesian framework where the parameters of the model that best fit the observations are estimated by maximising its posterior probability given the input data (observations). Optimisation techniques used are reviewed in sections 2.3.1 and 2.3.2.

## 2.2 3D reconstruction from RGB-D images

3D reconstruction using structured light or Time-of-Flight cameras has been a hot research topic in the past few years [16, 21, 22, 30, 32, 34]. If the quality of the recorded depth map is noisy and suffers from missing data, it is still an attractive alternative for 3D reconstruction. Using a Time-of-Flight camera, Cui et al. [15] propose to merge a set of depth maps to infer a less noisy 3D shape. The key problem to solve is the alignment or registration of all the captured depth scans. Similarly using a turning table and a cheap Kinect camera, Ruttle et al. infer a 3D shape of an object by merging several point clouds recorded from different view points around that object [44]. Registration is performed by maximising a cross correlation between two pdfs (cf. paragraph 2.3.4), and the accuracy achieved is shown to be similar to the 3D surface recorded with an expensive Laser scanner [43].

Newcombe et al. [33] propose the KinectFusion system for merging sequential depth scans recorded with a Kinect camera and inferring in real time a 3D mesh of a scene. The noise is reduced by filtering each scan before registration. Focusing on faces, Hernandez et al. [29] extend the KinectFusion approach to infer laser scan quality 3D faces. However, small details are lost due to spatiotemporal smoothing used in the process. Weise et al. [53] use kinect depth images to infer the facial expressions and dynamics of an actor to animate accordingly an avatar in real time. Their approach uses a user-specific expression model that is fitted to the observed scans with the Iterative Closest Point (ICP) algorithm. Similarly using a Kinect camera, Zollhofer et al. [56] propose an automatic method for 3D face reconstruction using Morphable Models. Their algorithm relies on feature landmarks that can be detected from the face (eyes, nose and chin).

Schneider et al. [45] proposed two algorithms based on the ICP framework (c.f. paragraph 2.3.2) for registering laser scans of human heads. Holes may occur in the resulting mesh but these can be filled in using a prior model for heads. Using a morphable model also helps for resampling efficiently the inferred mesh.

## 2.3 Registration & correspondences

In many reconstruction methods using RGB and RGB-D images, a common problem to solve is the registration between datasets. Indeed merging several point clouds or fitting a morphable model, require to perform an accurate and robust registration. There is a vast literature dedicated to registration algorithms. However, the most relevant methods rely on the ICP algorithm [9, 54] and probabilistic modelling for parameters estimation [49, 24, 14, 18, 31]. We review standard optimisation algorithms in paragraph 2.3.1 and registration methods in sections 2.3.2 and 2.3.4.

### *2.3.1 Optimisation*

Many optimisation algorithms used for RGB images (section 2.1) are based either on Stochastic Newton Optimization (SNO) [10], Linear Shape and Texture Fitting (LiST) [38] or Inverse Compositional Image Analysis (ICIA) [39, 6] algorithms. The evaluation of these methods depends on their application. For instance, SNO is reported to be more accurate but it lacks efficiency compared with ICIA [27]. Improvements to these algorithms have been mainly achieved by adding additional information to the cost function such as multiple features from a single image [40] or using multiple images [41, 1, 55, 52]. This multiple feature/image strategy provides a fitting algorithm more robust to local minimum, perhaps due to the smoothness of the overall cost function achieved by the extra information used. However, assumptions about known correspondences in between the input images and the model are introduced in order to match the extra features. In practice, this correspondence is difficult to achieve and mismatches during the starting of the fitting process affect the robustness and accuracy of the fitting algorithm.

### *2.3.2 Iterative Closest Point (ICP)*

For RGB-D images (section2.2), registration methods such as the ICP algorithm [9, 54, 42] have been proposed for solving the correspondence problem and the fitting process itself [45]. The ICP algorithm was introduced by Besl et al. [9] and Zhang [54]. It

is based on a point-to-point correspondence between the two data sets performed using the nearest neighbour criteria. Once the correspondences have been found the transformation is calculated. These two steps (correspondence-transformation) are iterated until convergence criterion is reached. Many improvements have been made to the basic ICP algorithm (refer to [42]). However, they are still sensitive to outliers and initialisation. ICP requires the initial position of the two point sets to be adequately close. This is usually achieved by matching manually labelled points in both sets [53, 2].

### *2.3.3 EM-like algorithms*

Robustness to outliers and initialisation has been improved by using probabilistic methods. The registration problem can be redefined as a density estimation problem. Chui et al. [14] for instance, proposed to treat one of the data sets (observations) as samples of the density function modelled with the reference data set. This modelling comes from the assumption that observations are uniformly distributed around points in the reference dataset. The problem can be solved as a two step optimisation (correspondence-transformation) but in an EM-like fashion where the centroids of the density function are estimated with the transformation parameters. The sensitivity to outliers inherent to Expectation Maximization algorithms is compensated by adding an extra kernel to model the outliers. Myronenko et al. [31] and Ganger et al. [18] also proposed algorithms following this strategy. The main drawback of those algorithms is that they rely on the availability of a dense set of observations. This implies a many-to-one correspondence which is not true when the number of observations is roughly the same (or less) than the number of points in the reference set.

### *2.3.4 Matching probability density functions*

A different approach is to consider the two sets as separate probability density functions. The parameter estimation is then performed by optimising a similarity measure between the two density functions [13]. Tsin et al. [49] for instance, used the cross correlation between the two density functions as a measure of similarity while Hasanbelliu et al. [19] proposed a registration algorithm based on the Cauchy-Schwarz divergence. A good overview is proposed by Jian et al. [24] showing the relationship between divergence functions used to measure the similarity between two pdfs [8].

The Euclidean distance has the advantage of having a closed form solution when density functions are modelled as Gaussian Mixtures. Having two density functions $p'(\mathbf{x}|\Theta)$ (model) and $p(\mathbf{x})$ (observations), the Euclidean distance is then defined as:

$$\begin{aligned}\mathcal{L}_2(\Theta) &= \int_{\mathbb{R}^{d_\mathbf{x}}} \left(p(\mathbf{x}) - p'(\mathbf{x}|\Theta)\right)^2 \, d\mathbf{x} \\ &= \int_{\mathbb{R}^{d_\mathbf{x}}} p^2(\mathbf{x}) - 2\, p(\mathbf{x})\, p'(\mathbf{x}|\Theta) + p'^2(\mathbf{x}|\Theta)\, d\mathbf{x}\end{aligned} \tag{1}$$

Tsin et al. [49] maximises the correlation between the pdfs to estimate the parameters of interest $\Theta$:

$$\mathcal{C}(\Theta) = \int_{\mathbb{R}^{d_\mathbf{x}}} p(\mathbf{x})\, p'(\mathbf{x}|\Theta)\, d\mathbf{x} \tag{2}$$

Jebara et al [23] coined this correlation as the *expected likelihood kernel* since it is the expectation of one distribution (e.g. $p$) w.r.t. the other (e.g. $p'$). When the parameters $\Theta$ correspond to the rigid affine transformation parameter (rotation and translation) then the estimate with the cost function $\mathcal{C}$ and $\mathcal{L}_2$ is the same:

$$\widehat{\Theta} = \arg\max_\Theta \mathcal{C}(\Theta) = \arg\min_\Theta \mathcal{L}_2(\Theta) \tag{3}$$

This expression comes from the fact that the term $\int_{\mathbf{R}^{d_\mathbf{x}}} p(\mathbf{x})^2 d\mathbf{x}$ does not depend on $\Theta$ and $\int_{\mathbb{R}^{d_\mathbf{x}}} p'(\mathbf{x}|\Theta)^2 d\mathbf{x}$ remains the same for all rigid transformation $\Theta$. However, this equivalence is not true when $\Theta$ does not correspond to a rigid transformation.

Using observations $\{\mathbf{x}^{(i)}\}_{i=1,\cdots,n}$ such that $\forall i,\ \mathbf{x}^{(i)} \sim p(\mathbf{x})$, the empirical density function defined with the Dirac kernel as

$$\hat{p}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \delta(\mathbf{x} - \mathbf{x}^{(i)})$$

can be used to substitute the pdf $p$ in the cost function $\mathcal{C}$ [24]:

$$\mathcal{CE}(\Theta) = \frac{1}{n}\sum_{i=1}^{n} p'(\mathbf{x}^{(i)}|\Theta) \tag{4}$$

Similarly, Scott [46] defines the cost function $\mathcal{L}_2E$ approximating $\mathcal{L}_2$:

$$\mathcal{L}_2E(\Theta) = \int_{\mathbb{R}^{d_\mathbf{x}}} p'^2(\mathbf{x}|\Theta)\, d\mathbf{x} - 2\,\mathcal{CE}(\Theta) \tag{5}$$

Jian et al. [24] uses the $\mathcal{L}_2E$ distance with a standard non-linear optimization function from Matlab to estimate $\Theta$.

Note that approximations of $\mathcal{C}$ and $\mathcal{L}_2$ with $\mathcal{CE}$ and $\mathcal{L}_2E$ respectively are good approximations as long as the empirical density function $\hat{p}$ is a good approximation of $p$. This may not be the case when the observations are sparse and/or not well uniformally sampled from $p$.

# 3. MORPHABLE MODEL REGISTRATION

Section 3.1 presents an overview of our approach and introduces the notations used in this paper. The details about the explicit expression of the $\mathcal{L}_2$ distance for Gaussian mixtures is reported in section 3.2. This distance is used to estimate the rigid transformation (rotation and translation) between the observations and the morphable model (section 3.3) and it is extended to estimate the morphable parameters in a Bayesian framework (section 3.4).

## 3.1 Overview & notations

Our 3D face reconstruction process can be described in four steps: acquisition, preprocessing, rigid registration (shape alignment) and morphable shape model fitting (see Figure 1). A depth map is captured with a kinect sensor and the region of the depth map corresponding to the face is detected and converted to a point cloud. This process is automatically done by applying a face detector [51, 28] and a skin detector algorithm in the region of the image. We assume that the person is in the foreground in between a specific range of distances from the sensor (50 and 120 cm). The resulting point cloud is noted $\{u_i\}_{i=1,\cdots,n}$ and each vertex $u_i$ is a point in $\mathbb{R}^3$. We define the random variable $x \in \mathbb{R}^3$ and its pdf is modelled using a Gaussian Kernel density estimate fitted on the observations:

$$p(x) = \sum_{i=1}^{n} \pi_i\, G(x; u_i, \Sigma_i) \tag{6}$$

$G(x; u_i, \Sigma_i)$ is the Gaussian probability density function centered on the vertex $u_i$ with a $3\times 3$ covariance matrix $\Sigma_i$. The weights $\pi_i$ are chosen equiprobable $\pi_i = \frac{1}{n}$. The pdf $p(x)$ corresponds to the target distribution.

The 3D shape Face Model[1] provided by Basel University [36] provides the average 3D shape noted $\mu = \{\mu_i\}_{i=1,\cdots,n'}$ that is com-

[1] `http://faces.cs.unibas.ch/bfm/`.

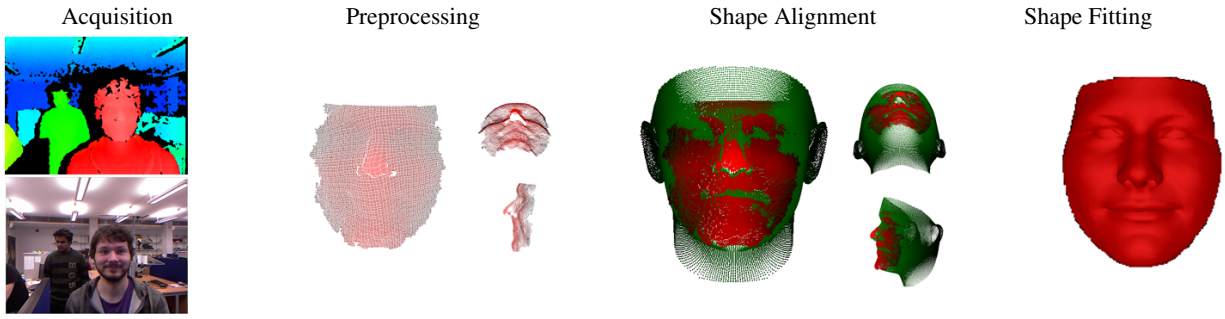


**Figure 1: Overview: a 3D morphable model is fitted automatically to a point cloud captured using a Kinect Sensor.**

posed of $n'$ vertices $\mu_i \in \mathbb{R}^3 \ \forall i$, a set of $J$ principal shape components noted $\{\mathbf{e}_j\}_{j=1,\cdots,J}$, associated with the standard deviations $\{\sigma_j\}_{j=1,\cdots,n'}$. If the target pdf $p(x)$ corresponds to a face, the model provides a suitable family of pdfs noted $p'(x|\Theta)$, that can explain the target $p$:

$$p'(x|\Theta) = \sum_{i=1}^{n'} \pi_i' \ G(x, u_i', \Sigma_i') \tag{7}$$

where the vertices $\{u_i'\}_{i=1,\cdots,n'}$ are created with the morphable model considering a rigid affine transformation (rotation R and translation t):

$$u_i' = \mathrm{R}\ (\mu_i + \sum_{j=1}^{J} \alpha_j \ e_{ji}) + \mathrm{t} \tag{8}$$

The problem addressed in this paper is the robust estimation of $\Theta$ corresponding to the rigid transformation (R and t) and the coordinates $\{\alpha_j\}_{j=1,\cdots,J}$ on the model. This is performed by minimising the $\mathcal{L}_2$ distance between $p$ and $p'$. Section 4 gives more details on the design of $p$ and $p'$.

## 3.2 Matching Gaussian mixtures with $\mathcal{L}_2$

To compute the $\mathcal{L}_2$ distance between the two Gaussian mixtures $p$ and $p'$, we use the closed form solution for the integral of the product of two Gaussian density functions [46, 20]:

$$\begin{aligned} < G_i|G_k' > &= \int_{\mathbb{R}^D} G(x, u_i, \Sigma_i)\ G(x, u_k', \Sigma_k')\ dx \\ &= \frac{1}{(2\pi)^{\frac{D}{2}} \sqrt{|\Sigma_i + \Sigma_k'|}} \ \exp\left(-\frac{q(u_i, u_k')}{2}\right) \end{aligned} \tag{9}$$

where,

$$\begin{aligned} & q(u_i, u_k') = u_i^T \Sigma_i^{-1} u_i + u_k'^T \Sigma_k'^{-1} u_i' - u^T \Sigma^{-1} u \\ & \text{with} \\ & \Sigma^{-1} = \Sigma_i^{-1} + \Sigma_k'^{-1} \\ & u = \Sigma\ (\Sigma_i^{-1} u_i + \Sigma_k'^{-1} u_i') \end{aligned} \tag{10}$$

The $\mathcal{L}_2$ distance can then be computed explicitly:

$$\begin{aligned} \mathcal{L}_2(\Theta) = & \sum_{i=1}^{n} \sum_{k=1}^{n} \pi_i \pi_k < G_i|G_k > \\ & + \sum_{i=1}^{n'} \sum_{k=1}^{n'} \pi_i' \pi_k' < G_i'|G_k' > \\ & - 2 \sum_{i=1}^{n} \sum_{k=1}^{n'} \pi_i \pi_k' < G_i|G_k' > \end{aligned} \tag{11}$$

Note that the parameters $\Theta$ only occur in the terms $< G_i'|G_k' >$ and $< G_i|G_k' >$.

## 3.3 Rigid transformation estimation

We first estimate the rigid transformation parameters using the mean face (e.g. $\alpha_j = 0, \ \forall j$ in equation (8)). As stated in equation (3), the estimated rigid transformation is the same when minimising the $\mathcal{L}_2$ distance and maximising the cross-correlation $\mathcal{C}$. For Gaussian mixtures, this cross-correlation is defined as:

$$\mathcal{C}(\Theta) = \sum_{i=1}^{n} \sum_{k=1}^{n'} \pi_i \pi_k' < G_i|G_k' > \tag{12}$$

and the estimation is performed such that

$$(\hat{\mathrm{R}}, \hat{\mathrm{t}}) = \arg\max_{\Theta=(\mathrm{R},\mathrm{t},\alpha_1=0,\cdots,\alpha_J=0)} \mathcal{C}(\Theta) \tag{13}$$

This optimisation can be solved using a Mean Shift algorithm [3]. An example of the alignment performed between the kinect data and the average shape of the face model using this algorithm is shown in Figure 1.

## 3.4 Bayesian Morphable shape fitting

Given the estimate $(\hat{\mathrm{R}}, \hat{\mathrm{t}})$, we need now to estimate the parameters $\{\alpha_j\}_{j=1,\cdots,J}$ and this can be done by minimising the $\mathcal{L}_2$ distance:

$$\hat{\alpha} = \arg\min_{\Theta=(\hat{\mathrm{R}},\hat{\mathrm{t}},\alpha)} \mathcal{L}_2(\Theta) \tag{14}$$

As the $\alpha$ parameters do not encode a rigid transformation, we do not substitute the objective function $\mathcal{L}_2$ by $\mathcal{C}$ for this estimation.

As seen in the review (paragraph 2.1), Bayesian estimation has been popular for fitting morphable models using a Gaussian prior on the parameters $\{\alpha_j\}_{j=1,\cdots,J}$:

$$p(\alpha = (\alpha_1, \cdots, \alpha_J)) \propto \exp\left(-\frac{1}{2} \sum_{j=1}^{J} \frac{\alpha_j^2}{\sigma_j^2}\right) \tag{15}$$

where $\sigma_j^2$ is the eigenvalue associated with the principal component $\mathbf{e}_j$ both provided with the morphable model. We have already shown [5] that for fitting 2D morphable model of hands, prior information was a valuable addition to the $\mathcal{L}_2$ distance for performing an accurate estimation. To this end, we interpret the distance $\mathcal{L}_2$ as log-likelihood so we can write the posterior as:

$$p(\Theta|\{u_i\}_{i=1,\cdots,n}) \propto p(\alpha) \cdot \exp\left(\frac{-\mathcal{L}_2(\Theta)}{2\sigma_d^2}\right) \tag{16}$$

And the estimation is then performed by:

$$\hat{\alpha} = \arg\min_{\Theta=(\hat{\mathrm{R}},\hat{\mathrm{t}},\alpha)} \left\{ \frac{\mathcal{L}_2(\Theta)}{\sigma_d^2} + \sum_{j=1}^{J} \frac{\alpha_j^2}{\sigma_j^2} \right\} \tag{17}$$

The variance $\sigma_d^2$ is set experimentally and allows us to control the influence of the likelihood with the prior. In section 5.2 we illustrate the relation used for computing this variance, how it is related to the bandwidth of the kernels in the density function and the error tolerance in the optimisation.

# 4. SHAPES WITH GAUSSIAN MIXTURES

Modelling surfaces in the 3D space using Gaussian mixtures is addressed in sections 4.1 and 4.2. Section 4.3 proposes to reduce the computational cost of using the $\mathcal{L}_2$ distance and paragraph 4.4 explains the algorithm used for its optimisation.

## 4.1 Model Gaussian Mixture

The shape model is a set of vertices connected by edges defining triangles (cf. figure 2). The means $\{u_k'\}_{k=1,\cdots,n'}$ are selected as the barycentre of these triangles and the covariance matrices $\{\Sigma_k'\}_{k=1,\cdots,n'}$ are also computed automatically using the model (cf. equation (7)). Each covariance matrix $\Sigma_k'$ is a positive definite matrix that can be decomposed into the form $\Sigma_k' = \mathrm{QDQ}^T$ where Q is the matrix containing the eigenvectors and D is a diagonal matrix of eigenvalues. The eigenvectors represent the principal directions $\overrightarrow{n}_2, \overrightarrow{n}_3$ and the normal to the surface $\overrightarrow{n}_1$. The eigenvalues control the fuzziness of the kernel in each direction:

$$\Sigma_k' = \left( \overrightarrow{n}_1 | \overrightarrow{n}_2 | \overrightarrow{n}_3 \right) \begin{pmatrix} a^2 & 0 & 0 \\ 0 & b^2 & 0 \\ 0 & 0 & c^2 \end{pmatrix} \left( \overrightarrow{n}_1 | \overrightarrow{n}_2 | \overrightarrow{n}_3 \right)^T \tag{18}$$

Figure 2 shows the directions $\overrightarrow{n}_1, \overrightarrow{n}_2$ and $\overrightarrow{n}_3$. The values for $b$ and $c$ can be easily computed as the semi axes of the ellipse that best fit the triangle. The fuzziness along the normal ($\overrightarrow{n}_1$) on the other hand, is defined as $a = h$ where $h$ is the bandwidth we choose for all the kernels and it is related to the error we are willing to tolerate between the observations and the model. The average shape of the model ($\mu$) and the eigenvectors ($e_j$) are updated according to the new position of the kernels (the barycentre of the triangles instead of the vertex). The use of non-isometric covariance matrices helps in modelling a density function that represents better the shape in the point cloud [4].

## 4.2 Target Gaussian Mixture

The edges between vertices captured with the kinect are assumed unknown and some observations are outliers (points that do not belong to the shape of interest). Hence the pdf $p$ computed from the observations $\{\mathbf{u}_k\}_{k=1,\cdots,n}$ (eq. (6)) is modelled with isotropic kernels centred at each data point $\Sigma_i = h^2 I$ ($I$ the identity matrix). The parameter $h$ is the same as defined earlier with the model (section 4.1) and has a specific role in our algorithm (section 4.4).

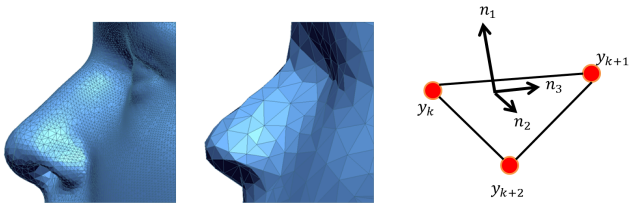


**Figure 2: Modelling the pdf, reducing the number of vertices that describe the shape model**

## 4.3 Reducing computations

Our cost function $\mathcal{L}_2$ is a function of $\Theta$ with $n \times n'$ Gaussian kernels. Downsampling the mesh model (i.e. reducing $n'$) can be done off-line to reduce computation cost. For that purpose, we used a probabilistic method based on the statistical self organising map that estimates the modes of a Gaussian mixture while preserving the spatial relationship between them [50]. The resulting position for the kernels are used in our modelling for defining the density function of the shape model. Downsampling the depth map can also be used to reduce $n$ in the target distribution.

## 4.4 Optimisation

In practice we estimate iteratively the rigid transformation and the morphable model parameters in a loop such that the final solution $\hat{\Theta}$ solves this optimisation:

$$\hat{\Theta} = \arg\min_{\Theta} \left\{ \frac{\mathcal{L}_2(\Theta)}{\sigma_d^2} + \sum_{j=1}^{J} \frac{\alpha_j^2}{\sigma_j^2} \right\} \tag{19}$$

For 3D faces, the mean face is a good enough representation of the class for aligning the model with the observations. Only one iteration of the loop is often required for aligning and fitting the model with the observations. In the case of estimating the rigid transformation, the Mean Shift algorithm is used with an annealing strategy for optimisation [37]. The parameter $h$ is used as the temperature [47]: starting with a large bandwidth $h = h_{max}$, the bandwidth is decreased using a geometric rate $\beta$ until the minimum value $h_{min}$ is reached. When estimating the parameters $\alpha$ of the morphable model, we also use $h$ as a temperature to introduce gradually non-convexity in the cost function (eq. (19)) since the prior term is convex.

# 5. EXPERIMENTAL RESULTS

Our algorithm is assessed with the 3D morphable shape face model provided by Basel University [36] (truncated to keep only the face region). This model has been computed using PCA on 3D meshes of neutral faces (100 males, 100 females) captured with a high-end 3D scanner, and it provides users with a mean mesh face and all eigenmeshes with their eigenvalues. The coordinates $\alpha$ on these eigenmeshes are estimated with our algorithm (as well as the rotation and translation for alignment) and it is assumed here that any neutral face can be well reconstructed with this model. Our approach is tested on synthetic data generated from the model (section 5.1) and experiments using real data captured using the Kinect sensor are reported in section 5.2 (note that in this case, the target faces are different from the faces used to compute the PCA model).

## 5.1 Fitting 3D face model to synthetic data

A set of synthetic target faces are created from the model by randomly generating a set of parameters $\alpha$s using $J = 10$ eigenvectors

(these constitute the ground truth parameters $\alpha_{GT}$). The rotation and translation between the model and the target is known and only the parameters $\alpha$ are estimated. In this experiment, the covariance matrices are chosen spherical as $\Sigma_i = h^2 I \ \forall i$ and $\Sigma_k = h^2 I \ \forall k$ because the two point clouds have been generated from the same synthetic 3d model in this experiment. This is not the case when we use the Kinect camera to capture the observations for which we will use adaptive variable covariance matrices (see section 5.2) as explained in section 4.1.

The bandwidth $\sigma_d$ for the likelihood term (see section 3.4) is computed (for all experiments in this paper) by:

$$\sigma_d = \frac{\lambda \mathcal{L}_2(\Theta_o, h)}{J} \tag{20}$$

where $\lambda$ is a parameter that we use to control the influence of the prior over the likelihood. We set $\lambda = 0.05$ for all the experiments. Starting from an initial guess randomly selected, we estimate $\alpha$ starting with a large bandwidth $h = h_{max} = 1.5cm$ that is iteratively decreased with geometric rate $\beta = 0.8$ until $h = h_{min} = 2.5mm$. We initialise $\sigma_d$ using $\Theta_o$ and $h_{max}$. Once the algorithm converges for $\hat{\Theta}$ we update $\sigma_d$ according to the new values for $h$ and $\Theta_o = \hat{\Theta}$.

Figure 3 shows several results of 3D face reconstruction. We can visually recognise that the estimated face using the fitting algorithm is similar to the target face in all four examples.

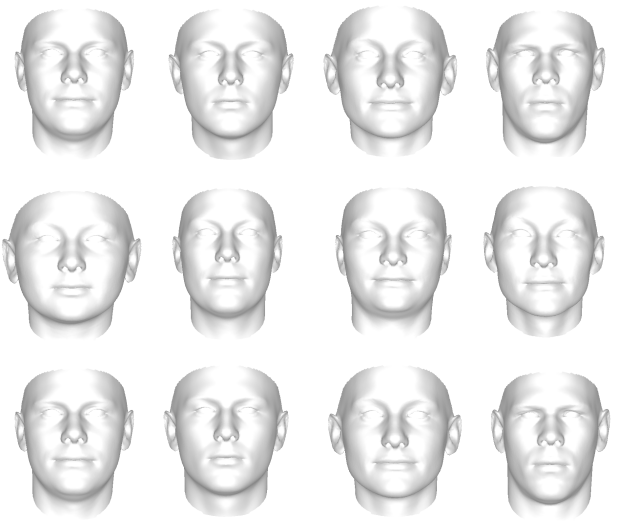

**Figure 3: Fitting synthetic faces: target face (observations, top row), random starting guess for initialising the algorithm (middle row) and estimated face with our algorithm (bottom row).**

Figure 4 shows the error surface between the target and the estimated reconstruction. Note that we use a truncated model: neck and ears areas have not been matched. The errors on the face (in light blue) correspond to areas that are not well captured in the $J = 10$ eigenvectors that we have used for reconstruction in this experiment. Convergence of the algorithm to the optimal solution is tested by running experiments with different starting points (randomly chosen) and the error is computed between the estimated $\alpha$ and the ground truth. In all the experiments the estimates converge to the ground truth with a root mean square error smaller than 0.0054.

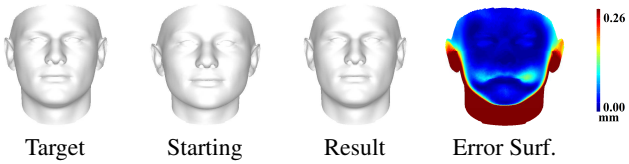


**Figure 4: Surface error computed between the target face and our reconstruction.**

## 5.2 Fitting 3D face model to Kinect

We have just shown that the algorithm converges well in controlled conditions. In real applications however, the observations originate from a different process than the model and more care needs to be given to modelling the covariance matrices. When the data are not uniformly sampled (or are sparse), the covariance matrices cannot be chosen isotropic but need to be chosen such that the pdfs approximate well the surface shapes. Covariance matrices are next set as explained in sections 4.1 and 4.2. The number of kernels is reduced as described in section 4.3.

### *5.2.1 Data Capture and Preprocessing*

The Microsoft Kinect sensor provides a depth map and a colour image of the scene with a resolution $480 \times 640$ pixels. The field of view captured is within a range of $50cm$ and $4m$ approximately. To obtain the point cloud of the face, we first select the region of the image within a range of $50cm$ to $120cm$ from the sensor (see example in Fig. 5b). The face region is then detected and converted to a point cloud by using a face and skin detector (Fig. 5c).

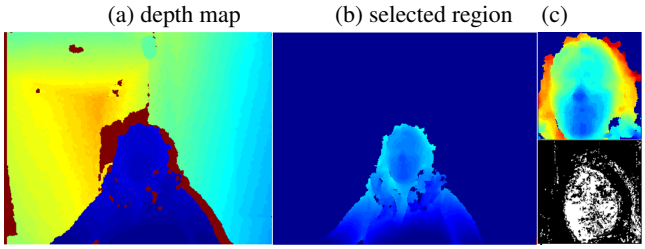


**Figure 5: Preprocessing for generating the point cloud of the face (target): depth map (a) as captured by the RGB-D sensor, selection of the scene (b) in close range (between $0.5m$ and $1.2m$ from the sensor), extracted face and skin region (c).**

### *5.2.2 Kinect point cloud alignment*

A crude estimate of the translation between the target and the model point clouds is computed by matching their barycenters in the 3D space. This estimation is used as a starting guess for our algorithm for rigid transformation. The settings used for $h$ in the optimisation are $h_{max} = 1cm$ and $h_{min} = 5mm$. Both datasets are downsampled with a ratio of $1:10$ in order to reduce the number of kernels in the density functions and speed up the optimisation process.

Figure 6 shows an example of the alignment process. Before alignment (bottom row) and after alignment (top row). We evaluate numerically the performance of the alignment by comparing the error between the observed point cloud and the average shape before and after the alignment. The error is computed as the Euclidean distance of each point in the observed data set to its closest triangle in the mesh of the average shape. Figure 7 shows the histogram of

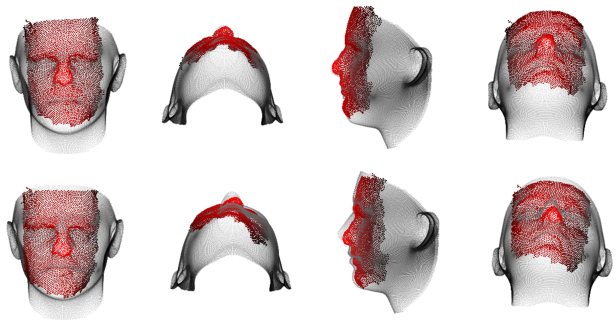

**Figure 6: Shape Alignment: different views of the point clouds (model (grey) and observations (red)) after alignment (top row). Same point clouds displayed before alignment (bottom row).**

the error of the aligned shape (red line) and before alignment (blue dash). Note the number of pixels closer to the shape model (error close to 0) increases after the alignment.

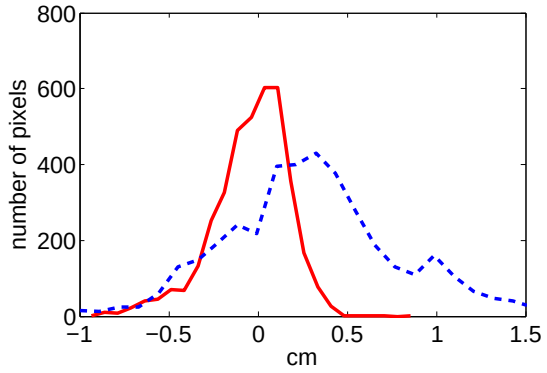


**Figure 7: Histogram of the errors between the observations to the average shape: before alignment (blue) and after (red). The sum of the absolute error is $6.3456 \times 10^6$ (after alignment) compare with $1.7054 \times 10^7$ (before alignment).**

### *5.2.3 3D Morphable Shape fitting on Kinect point cloud*

Once the target face is aligned to the shape model, the $\alpha$s are estimated using average shape (cf. Figure 8) as a starting guess in our algorithm (e.g. $\alpha_j = 0,\ \forall j \in \{1, \cdots, J\}$) and with the following settings: $h_{max} = 1.5cm$, $h_{min} = 5mm$ and $J = 20$.

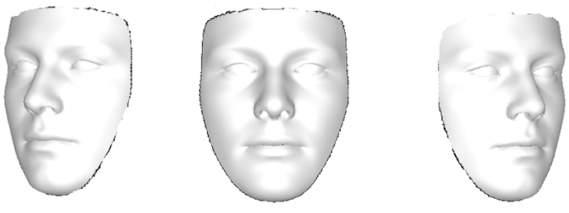

**Figure 8: Average Shape used as initial guess.**

Figure 9 shows the reconstructed faces for several people (none were used to train the morphable model). The last two faces (F5 and F6) correspond to the same person appearing behind occluding objects. The general shape of the faces is well recovered while some detailled areas are sometimes not accurate (e.g. eyes or mouths) that are noisy in the kinect data, and also may not well be described by the first $J = 20$ eigenvectors for these people in this experiment. Figure 10 compares the reconstruction of F5 with a capture using

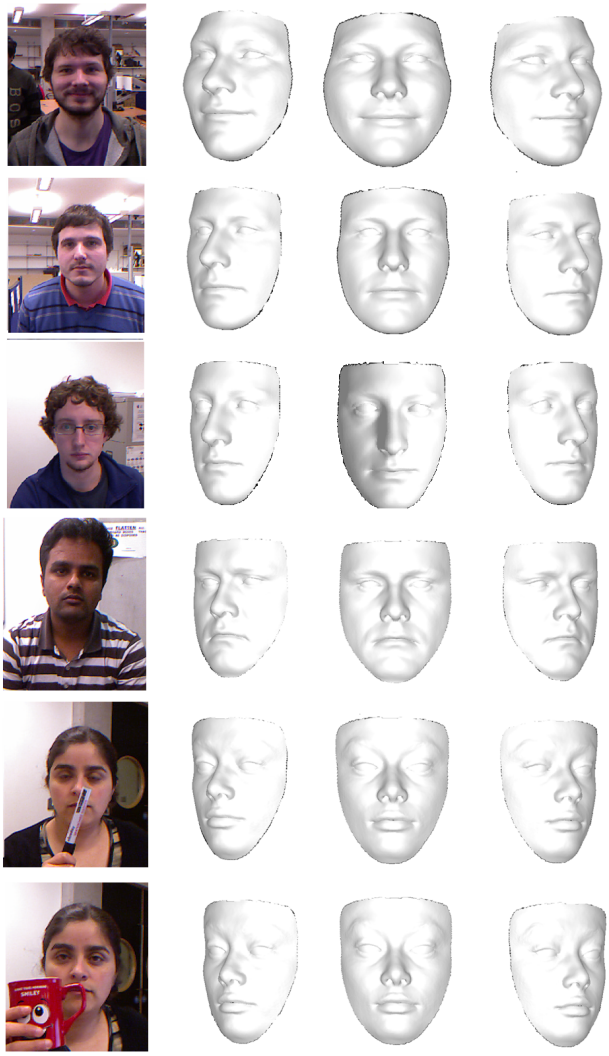

**Figure 9: Estimated reconstructed faces (labelled F1 (top), to F6 (bottom)) from 3 viewpoints shown with the colour image captured with the Kinect (left).**

a more accurate laser scanner (Minolta vivid 700). As can be seen, the laser scan is not perfect either and the reconstruction with the model has the advantage of recovering a full mesh without any hole.

The Euclidean distance (error) between each point in the target to its closest triangle in the reconstructed shape is computed, and the histogram of these errors are shown in Fig. 11. In all the experiments, the error is significantly reduced after the fitting: the variance of the distribution of the error is smaller for the data computed using the reconstructed face. In both cases, the 90% of observations are within a distance of $3mm$ after the fitting is done (compare with 50% before the fitting is done, see Figure 12).

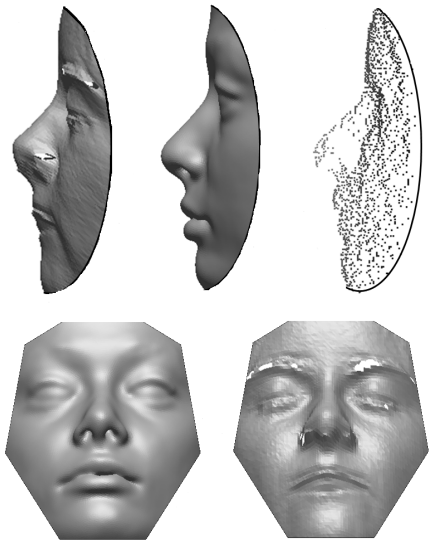

**Figure 10: At the top, profil view of F5 : laser scan (left), reconstruction (middle) and Kinect point cloud (right). At the bottom, frontal view of F5: reconstruction (left) and laser scan (right).**

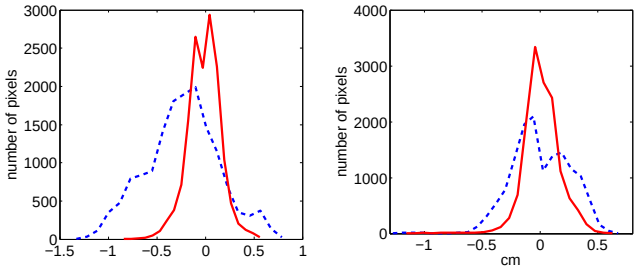


**Figure 11: Histograms of the errors between the observations and the reconstructed face (red line) and the observation and the average shape of the model (blue dash) for F1 (left) and F2 (right).**

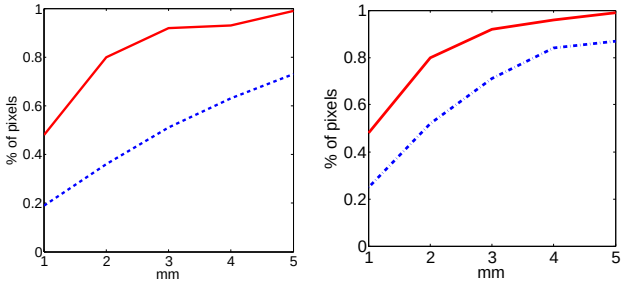


**Figure 12: Percentage of observations below a distance between 1 and 5mm (reported on the absissa) for F1 (left) and F2 (right), before fitting (blue dash) and after fitting (red line).**

Figure 13 shows the estimated $J = 20$ coordinates normalised with the eigenvalues for faces F1 to F6 (cf. figure 9). We can note that the estimates for F5 and F6 are close to each other corroborating the fact that the same person appears in both captures. Despites the occlusions, the algorithm converges towards the same solution for F5 and F6.

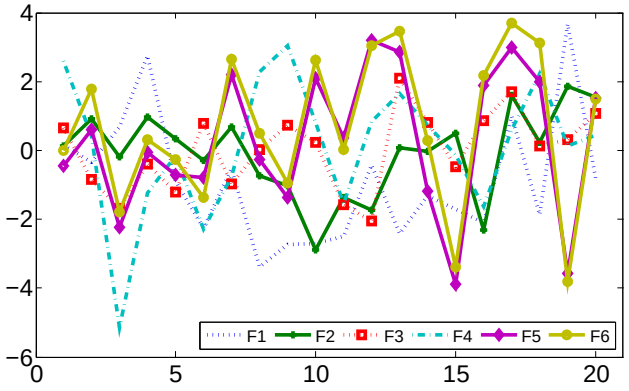


**Figure 13: Values of the $J = 20$ coordinates of parameter $\alpha$ normalised with the eigenvalues computed for faces F1 to F6.**

We have computed the Mahalanobis distance (with $\Lambda$ diagonal matrix of the eigenvalues):

$$d_{i,j} = \sqrt{(\hat{\alpha}_{Fi} - \hat{\alpha}_{Fj})^T \Lambda (\hat{\alpha}_{Fi} - \hat{\alpha}_{Fj})}$$

for all faces F1 to F6. The results are shown in Table 1. The distance $d_{i,j}$ is smaller when the parameters correspond to the same individual (e.g. F5 and F6, $d_{5,6} = 0.0888$) than when considering different people (e.g. F1 and F2, $d_{1,2} = 0.3596$). Although these experiments are preliminary and require further analysis, they suggest the feasibility of robust identification using noisy depth sensors.

**Table 1: Malahanobis distance $d_{i,j}$ between the estimated parameters of faces F1 to F6 (Figure 9).**

| Faces | F1 | F2 | F3 | F4 | F5 | F6 |
|---|---|---|---|---|---|---|
| F1 | 0 | 0.3596 | 0.5374 | 0.6925 | 0.7609 | 0.7815 |
| F2 | | 0 | 0.3396 | 0.5041 | 0.6141 | 0.6286 |
| F3 | | | 0 | 0.3885 | 0.4978 | 0.5155 |
| F4 | | | | 0 | 0.5508 | 0.5475 |
| F5 | | | | | 0 | 0.0888 |
| F6 | | | | | | 0 |

## 6. CONCLUSION

We have proposed a new cost function to perform 3D morphable model alignment and fitting. This cost function is composed with the robust $\mathcal{L}_2$ distance between pdfs, and that does not require any correspondence to be defined between the two point sets to be matched. In addition, the cost function is augmented with prior information that can be available with the model. Note that the algorithm was tested using individuals that not were included in the original model. Our approach is robust and unsupervised, however it is computationally expensive when the two point clouds to match are far from each other. Several directions can be explored to improve efficiency: prior information would help to initialise the optimisation with a good initial guess (e.g. the past estimate in a tracking context), and multi-resolution approaches reducing the number of kernels to compute could also be used (cf. section 4.3),

## 7. ACKNOWLEDGMENTS

This work has been supported by scholarships from Trinity College Dublin Ireland and the Goverment of Chile.

# 8. REFERENCES


[1] B. Amberg, A. Blake, A. Fitzgibbon, S. Romdhani, and T. Vetter. Reconstructing high quality face-surfaces using model based stereo. *IEEE International Conference on Computer Vision, ICCV*, pages 1–8, 2007.

[2] 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.

[3] C. Arellano and R. Dahyot. Mean shift algorithm for robust rigid registration between gaussian mixture models. In *20th European Signal Processing Conference (Eusipco)*, pages 1154–1158, Bucharest, Romania, August 2012.

[4] C. Arellano and R. Dahyot. Shape model fitting using non-isotropic gmm. In *23rd IET Irish Signals and Systems Conference*, 2012.

[5] C. Arellano and R. Dahyot. Shape model fitting without point correspondence. In *20th European Signal Processing Conference (Eusipco)*, pages 934–938, Bucharest, Romania, August 2012.

[6] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1090–1097, 2001.

[7] C. Basso, T. Vetter, and V. Blanz. Regularized 3d morphable models. In *IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, pages 3–12, 2003.

[8] A. Basu, I. Harris, N. Hjort, and M. Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.

[9] 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.

[10] 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.

[11] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. *ACM Transactions Graphics*, 29(4):41:1–41:10, July 2010.

[12] H. Bulthoff. *Shape from X: psychophysics and computation*. In: Landy M, Movshon A, editors. Computational models of visual processing. Cambridge: MIT Press, 1991.

[13] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1:300–307, 2007.

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

[15] Y. Cui, S. Schuon, C. Derek, S. Thrun, and C. Theobalt. 3d shape scanning with a time-of-flight camera. In *IEEE conference on Computer Vision and Patern Recognition, CVPR*, 2010.

[16] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. Depth mapping using projected patterns. *Patent Publication number: US 2010/0118123 A1*, 2008.

[17] P. Fua. Regularized bundle-adjustment to model heads from image sequences without calibration data. *International Journal of Computer Vision*, volume 38:153–171, 2000.

[18] S. Granger and X. Pennec. Multi-scale em-icp: A fast and robust approach for surface registration. *European Conference on Computer Vision ECCV*, pages 418–432, 2002.

[19] E. Hasanbelliu, L. S. 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*, pages 1–6,2011.

[20] 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.

[21] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgbd mapping: Using depth cameras for dense 3d modeling of indoor environments. In *In RGB-D: Advanced Reasoning with Depth Cameras Workshop in conjunction with RSS*, 2010.

[22] B. Huhle, T. Schairer, P. Jenke, and W. Straßer. Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *Computer Vision and Image Understanding*, 114(12):1336 – 1345, 2010. Special issue on Time-of-Flight Camera Based Computer Vision.

[23] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research*, volume 5:819–844, 2004.

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

[25] J. Lee, B. Moghaddam, H. Pfister, and R. Machiraju. Silhouette-based 3d face shape recovery. In *Graphics Interface*, pages 21–30, 2003.

[26] R. Lengagne, P. Fua, and O. Monga. 3d stereo reconstruction of human faces driven by differential constraints. *Image and Vision Computing*, volume 18(4):337–343, 2000.

[27] M. D. Levine and Y. (Chris) Yu. State-of-the-art of 3d facial reconstruction methods for face recognition based on a single 2d training image per person. *Pattern Recogn. Lett.*, volume 30(10):908–913, 2009.

[28] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *International Conference on Image Processing (ICIP)*, volume 1, pages I–900 – I–903, 2002.

[29] H. M, J. Choi, and G. Medioni. Laser scan quality 3-d face modelling using a low cost depth map. In *20th European Signal Processing Conference (Eusipco)*, pages 1995–1999, Bucharest, Romania, August 2012.

[30] P. Merrell, A. Akbarzadeh, L. Wang, J. michael Frahm, and R. Y. D. NistÃl'r. Real-time visibility-based fusion of depth maps. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

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

[32] R. Newcombe and A. Davison. Live dense reconstruction with a single moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1498 –1505, june 2010.

[33] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface

mapping and tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society.

[34] K. Ouji, M. Ardabilian, L. Chen, and F. Ghorbel. A space-time depth super-resolution scheme for 3d face scanning. In *Proceedings of the 13th international conference on Advanced concepts for intelligent vision systems*, ACIVS'11, pages 658–668, Berlin, Heidelberg, 2011. Springer-Verlag.

[35] A. Patel and W. A. Smith. 3d morphable face models revisited. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1327 –1334, 2009.

[36] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, 2009.

[37] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Verlag, 1999.

[38] 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.

[39] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, pages 59–66, 2003.

[40] S. Romdhani and T. Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 986–993, 2005.

[41] F. Romeiro and T. Zickler. Model-based stereo with occlusions. In *Proceedings of the 3rd international conference on Analysis and modeling of faces and gestures, AMFG*, pages 31–45, 2007.

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

[43] J. Ruttle. *Statistical Framework for Multi-Sensor Fusion and 3D Reconstruction*. PhD thesis, School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland, 2012.

[44] J. Ruttle, C. Arellano, and R. Dahyot. Extrinsic camera parameters estimation for shape-from-depths. In *20th European Signal Processing Conference (Eusipco)*, pages 1985–1989, Bucharest, Romania, August, 27-31 2012.

[45] D. Schneider and P. Eisert. Algorithms for automatic and robust registration of 3d head scans. *Journal of Virtual Reality and Broadcasting*, 7, 2010.

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

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

[48] W. A. Smith and E. R. Hancock. Recovering face shape and reflectance properties from single images. In *IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–8, 2008.

[49] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *European Conference in Computer Vision ECCV*, pages 558–569,2004.

[50] J. J. Verbeek, N. Vlassis, and B. J. A. Kröse. Self-organizing mixture models. *Neurocomput.*, 63:99–123, Jan. 2005.

[51] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[52] 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.

[53] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Transactions Graphics*, 30:1–10, 2011.

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

[55] M. Zhao, T.-S. Chua, and T. Sim. Morphable face reconstruction with multiple images. In *IEEE International Conference on Automatic Face and Gesture Recognition, FGR*, pages 597–602, 2006.

[56] M. Zollhöfer, M. Martinek, G. Greiner, M. Stamminger, and J. Süßmuth. Automatic reconstruction of personalized avatars from 3d face scans. 22(3-4):195–202, 2011.