Multiscale 3D Feature Extraction and Matching

Hadi Fadaifard
Department of Computer Science
Graduate Center, City University of New York
New York, USA
hfadaifard@gc.cuny.edu

George Wolberg
Department of Computer Science
City College of New York
New York, USA
wolberg@cs.ccny.cuny.edu

***Abstract*—Partial 3D shape matching refers to the process of computing a similarity measure between partial regions of 3D objects. This remains a difficult challenge without *a priori* knowledge of the scale of the input objects, as well as their rotation and translation. This paper focuses on the problem of partial shape matching among 3D objects of unknown scale. We consider the problem of face detection on arbitrary 3D surfaces and introduce a multiscale surface representation for feature extraction and matching. This work is motivated by the scale-space theory for images. Scale-space based techniques have proven very successful for dealing with noise and scale changes in matching applications for 2D images. However, efficient and practical scale-space representations for 3D surfaces are lacking. Our proposed scale-space representation is defined in terms of the evolution of surface curvatures according to the heat equation. This representation is shown to be insensitive to noise, computationally efficient, and capable of automatic scale selection. Examples in face detection and surface registration are given.**

***Keywords* - 3D shape matching, mesh signal processing, heat equation, 3D face detection, surface registration**

I. Introduction

3D shape matching refers to the process of computing a similarity measure between 3D objects [1]. The main applications of shape matching are 3D shape registration, recognition, retrieval, and classification. These applications, in turn, are used in higher-level processing tasks, such as 3D search engines [2] and automatic 3D model generation from physical objects [3].

Partial 3D shape matching refers to a more difficult subproblem that deals with measuring the similarity between partial regions of 3D objects. Despite a great deal of attention drawn to 3D shape matching in the fields of computer vision and computer graphics, partial shape matching applied to objects of arbitrary scale remains a difficult problem. In this case, the similarity is measured between partial regions of input objects, where the relative position, orientation, scale, and the extent of overlap may be unknown. Although various algorithms exist in the literature for 3D partial shape matching [3], [4], they do not handle shapes of arbitrary scale. We address this problem by introducing a new 3D surface matching approach based on the scale-space theory of signals.

The scale-space *representation* of a signal in $\mathbb{R}^n$ is defined in terms of its evolution according to the heat (diffusion) equation (Sec. II). The scale-space *theory* is concerned with the study and analysis of the properties of this representation of signals. The need for scale-space representations of signals arises when estimating derivatives for shape matching. Most shape matching applications require the estimation of the first few derivatives of the input signals [5]. A major difficulty with this estimation problem is that differentiation is highly sensitive to noise. Most techniques attempt to achieve resilience to noise by defining the differential operators in a multiscale fashion, with the scale determining the amount of low-pass filtering applied to the input. A difficulty with this approach is determining the proper spatially-varying scale for the operators [5], which is a process referred to as *automatic scale selection* [6]. This process served as the principle motivation for the development of the scale-space theory [5].

Scale-space techniques are now widely used for signals in $\mathbb{R}^n$ [7], with the theory having become quite mature over the past few decades, especially for 2D images. Beyond having nice theoretical properties, the scale-space representation of images has been shown to be realized efficiently with impressive practical results [8], [9]. Currently, scale-space matching techniques, such as SIFT [8] and SURF [10], are the *de facto* standards in many 2D matching applications. Despite finding widespread use in 2D signal processing, scale-space techniques have not been widely applied to 3D surfaces.

There are two major difficulties with extending scale-space representations to 3D surfaces. These difficulties are due to representation issues and the estimation of the scale parameter necessary for automatic scale selection. The lack of grid-like structures that are present in 2D images and the non-Euclidean geometry of surfaces make development of precise and efficient representations difficult. The scale parameter, which in the case of signals in $\mathbb{R}^n$ is defined in terms of the variance of the smoothing kernel, may not be readily available for 3D surfaces.

The goal of this work is to extend the use of scale-space theory to 3D surfaces for the purpose of partial shape recognition. The main contribution is a new scale-space representation for 3D surfaces that addresses the two major difficulties outlined above. The new representation is shown to be insensitive to noise, computationally efficient, and capable of automatic scale selection.

The few current scale-space based surface representations can be categorized into two classes based on how a signal is derived from the surface and consequently evolved. First, surface positions may be treated as the signal and therefore the surface geometry is modified during the evolution process.

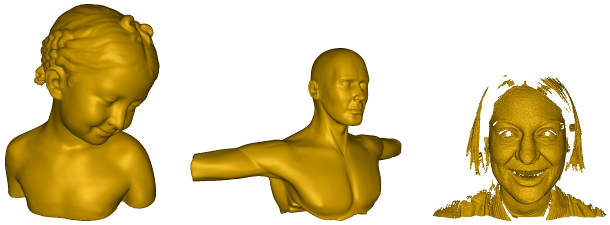


Fig. 1. Examples of possible inputs to our face detection system.

Second, a signal may be defined and evolved over the surface while the geometry of the surface remains unchanged. It is well-known that the evolution of surface positions generally leads to geometric problems such as shrinkage and foldovers [11], [12]. Therefore, we opt for the second approach whereby we define our scale-space representation in terms of the evolution of the surface curvatures.

We show an application of our approach to a partial shape matching task involving the detection of human faces on 3D models without any assumptions about the scale of the models. To provide insight into our approach, we first present the problem of automatic 3D face detection on arbitrary models. Fig. 1 shows examples of input surfaces on which we seek to find the location and extent of a human face. Note that these surfaces come from different sources, have arbitrary scale, and contain various facial expressions. In Fig. 2, we show the general steps involved in the detection task. First, a set of interesting keypoints are extracted on the surface of the model (Fig. 2(b)). Using these extracted keypoints, the relative location, orientation, and scale of the model (Fig. 2(c)) is obtained with respect to a reference face model (Fig. 2(f)). Using this information, the locations of other important facial features are identified (Fig. 2(d)). Finally, using the new set of facial features in Fig. 2(d), the face region is extracted from the input model (Fig. 2(e)). The face extraction process generates a remeshed version of the cropped face in a manner consistent between all models. Therefore, the output of the face detection process can be directly fed into a 3D face recognition algorithm. Unlike the majority of automatic 3D face detection systems [13], [14], the proposed detection scheme does not make any assumptions about the relative pose or scale of the input models.

A. Related Work

Various scale-space representations have been proposed over the past decade. The most straightforward approaches for 3D surfaces include parameterization [15] and voxelization [16]. These two approaches, however, result in new surface representations that suffer from distortions or loss of precision, respectively.

As mentioned earlier, smoothing signals defined on the surface may be performed instead of smoothing surface geometry. For example, in [17], surface mean curvatures on a triangulated surface are repeatedly smoothed with a Gaussian filter. The proposed representation is then used to define a measure of *mesh saliency* over the surface and its applications in mesh simplification and viewpoint selection are shown in that paper.

Mean curvature flow, which is closely related to surface diffusion, may also be used to smooth the surface. Under mean curvature flow, each surface point is moved along its normal proportional to its mean curvature. [18] uses a modification of this approach to obtain a scale-space representation for surfaces and shows how it can be used to perform feature extraction and automatic scale selection on closed 3D models. A major problem with this approach, however, is the geometric degeneracies that generally arise from smoothing. In addition, computation times of more than 2 hours were reported for meshes with more than $2K$ vertices.

More recently, the Heat Kernel Signature (HKS) [19] has been used in global shape matching tasks involving 3D models that may have undergone isometric deformations. In this approach, the properties of the heat kernel of a surface are used to infer information about the geometry of the surface. A scale-invariant version of HKS was also introduced in [20] and used for non-rigid 3D shape retrieval. The main drawbacks of HKS-based techniques are computation times and their inability to perform automatic scale selection, which is required in most partial shape matching tasks. In [19], it is reported that the overall time required to compute the HKS on a surface with $100K$ vertices is approximately 90 minutes on a 2.4 GHz CPU. This seriously limits the practical applications of the approach.

The scale-space approach presented in this paper is capable of automatic scale selection and is shown to be efficient to compute. For instance, the scale-space representation of a surface with approximately $113.5K$ vertices can be obtained in 32 seconds on a 2.4 GHz CPU.

II. Scale-space Representation of Signals in $\mathbb{R}^n$

The scale-space representation of a continuous signal $f : \mathbb{R}^n \to \mathbb{R}$ is defined as the solution to the heat (diffusion) equation [7]:

$$\partial_t F = \Delta F \ , \tag{1}$$

where Δ denotes the Laplacian, and $F(\mathbf{x}; 0) = f(\mathbf{x})$ is the initial condition. It can be shown that the Gaussian is the fundamental solution to Eq. (1) [7]. The scale-space representation of f can therefore be expressed as

$$F(\mathbf{x}; t) = g(\mathbf{x}; t) * f(\mathbf{x}) \ , \tag{2}$$

where $*$ denotes convolution, $g : \mathbb{R}^n \to \mathbb{R}$ is the n-dimensional normalized Gaussian: $g(\mathbf{x}; t) = \frac{1}{(\pi t)^{n/2}} e^{-\|\mathbf{x}\|^2/t}$, and t is known as the *scale parameter*.

The non-enhancement property [7] of the scale-space representation of signals, in general, guarantees that the values of the local maxima (minima) decrease (increase) as the signal is smoothed. However, the amplitude of the spatial derivatives of the signal may be *scale-normalized* using the change of variable $\mathbf{v} = \frac{\mathbf{x}}{t^{\gamma/2}}$, for $\gamma > 0$. This results in the following scale-normalized spatial derivatives of the signal:

$$\partial_{\mathbf{v}^m} F^*(\mathbf{v}; t) = t^{m\gamma/2} \partial_{\mathbf{x}^m} F(\mathbf{x}; t) \ , \tag{3}$$

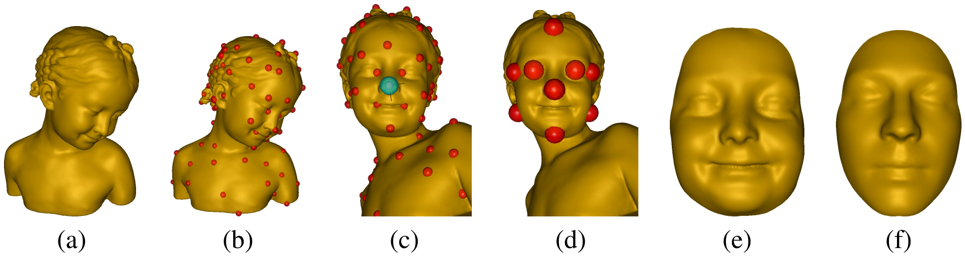


Fig. 2. Multiscale 3D face detection pipeline. (a) original model; (b) extracted keypoints; (c) detected location, orientation, and scale of the face region; (d) auxiliary facial features extracted from the face region; (e) cropped and remeshed face region; (f) reference face.

where m denotes the order of differentiation. It can be shown that the amplitudes of the scale-normalized derivatives of a signal first increase and then decrease, and the scale at which the maximum amplitude is reached is proportional to the frequency of the signal. The process of finding this scale is known as automatic scale selection [7]. We seek the same type of scale-normalization in a scale-space representation of a surface signal, and employ it to infer information about the size of structures on the surface.

III. Scale-Space Representation for 3D Surfaces

In this section, we formulate a similar representation for surfaces that is as close as possible to the scale-space representation of signals in $\mathbb{R}^n$. Our proposed approach is similar to the HKS-based techniques, in the sense that we derive the scale-space formulation of the surface in terms of the evolution (diffusion) of signals on the surface with the help of the Laplace-Beltrami operator. However, we analyze the surface structures by directly studying the behavior of the signal as it evolves on the surface. More specifically, we take the signal to be the surface curvatures, which are derived from the surface geometry. The main advantages of this approach over HKS are gains in computational efficiency and the ability to estimate the size of the surface structures. Additionally, our representation enables us to robustly and efficiently estimate the Laplacian of surface curvatures that results in a rich set of features, which is useful in subsequent matching tasks.

Therefore, the scale-space representation, $F : \mathcal{M} \times \mathbb{R} \to \mathbb{R}$, of 3D surface $\mathcal{M}$, is defined as the solution to the diffusion equation:

$$\partial_t F = \Delta_{\mathcal{M}} F \ , \tag{4}$$

with the initial condition $F(\mathbf{p}; 0) = f(\mathbf{p})$, where $f(\mathbf{p})$ denotes the mean or Gaussian curvature at point $\mathbf{p} \in \mathcal{M}$, and $\Delta_{\mathcal{M}}$ is the Laplace-Beltrami operator.

From the above formulation, a stack of Gaussian-smoothed surface curvatures is obtained, which can be used directly in multiscale feature extraction and descriptor computations. However, to make the best use of the representation for automatic scale selection, the value of the scale parameter at each level must also be estimated. The smoothed curvatures together with the associated scales at each level define our multiscale surface representation, which we refer to as the Curvature Scale-Space 3D (CS3), as depicted in Fig. 3.

In Sec. III-A, we describe how a discrete surface signal may be efficiently smoothed in a manner consistent with the scale-space representation of signals. In Sec. III-C, we show how the representation may be used for feature point extraction with an automatic scale selection mechanism.

A. Gaussian Smoothing of a Surface Signal

Let our discrete surface be represented by the polygonal mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$, and $\mathcal{E} = \{e_{ij} | v_i \text{ is connected to } v_j\}$ are the vertex and edge sets, respectively. Let $F^l : \mathcal{V} \to \mathbb{R}$ denote the smoothed discrete surface signal (curvatures) at level l, and define $\mathbf{F}^l = \begin{pmatrix} F^l(v_1) & \cdots & F^l(v_N) \end{pmatrix}^\top$. We employ the implicit surface smoothing scheme of [12] to obtain the smoothed surface signal $\mathbf{F}^{l+1}$, at level $l+1$, by solving the following sparse system of linear equations

$$(\mathbf{I} - \lambda_l \mathbf{L}) \mathbf{F}^{l+1} = \mathbf{F}^l \ , \tag{5}$$

where $\lambda_l > 0$ is a time step, and $\mathbf{L}$ and $\mathbf{I}$ denote the $N \times N$ Laplacian and identity matrices, respectively. The elements of the Laplacian matrix $\mathbf{L} = (w_{ij})_{N \times N}$ are given as

$$w_{ij} = \begin{cases} -1 & \text{for } i = j \ , \\ \frac{1}{|\mathcal{N}(i)|} & \text{for } j \in \mathcal{N}(i) \ , \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $\mathcal{N}(i)$ denotes the 1-ring neighbor set of vertex v_i. The Laplacian matrix may also be populated with other types of weights, such as cotan weights [12]. The linear system in Eq. (5) can be efficiently solved using the Biconjugate Gradient method.

The scale-space representation of the surface signal $\mathbf{f}$ is then given by the sequence $(\mathbf{F}^0, \ldots, \mathbf{F}^{L-1})$, which is obtained recursively using

$$\mathbf{F}^l = \begin{cases} (\mathbf{I} - \lambda_{l-1} \mathbf{L})^{-1} \mathbf{F}^{l-1} & \text{if } l > 0 \\ \mathbf{f} & \text{if } l = 0 \ , \end{cases} \tag{7}$$

for $l = 0, \ldots, L-1$.

The resulting transfer function of the implicit Laplacian smoothing in Eq. (5) is

$$h(\omega) = (1 + \lambda_n \omega^2)^{-1} \ , \tag{8}$$

where ω denotes surface signal frequency [12]. When a stack of smoothed signals with L levels is constructed according

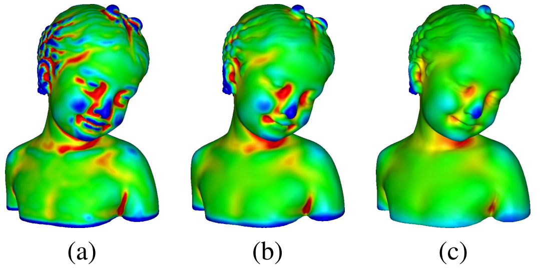


Fig. 3. The CS3 representation of the Bimba model at scales (b) $t = 3.0$, (c) $t = 7.5$, (d) $t = 13.8$.

to Eq. (7), with corresponding time steps $(\lambda_0, \ldots, \lambda_{L-2})$, the transfer function of the filter at level $L-1$ is given by

$$h_{L-1}(\omega) = \prod_{l=0}^{L-2} (1 + \lambda_l \omega^2)^{-1} \ . \tag{9}$$

Note that the representation needs to be defined in a recursive manner, since the transfer function of the filter defined by Eq. (5) is not a Gaussian. On the other hand, the transfer functions of our recursive formulation approach Gaussians, as L grows.

The time steps are picked as $\lambda_l = \lambda_{l-1}\delta = \lambda_0 \delta^l$, where λ_0 denotes an initial time step and $\delta > 1$ is a constant. It is important to note that the time steps λ_l are not equivalent to the scale parameter t in the original scale-space representation of signals given by Eq. (2). Fig. 3 shows a 3D model and its corresponding CS3 representation at various scales.

B. Estimating the Scale Parameter

To recover the scale parameter t at each level l, we fit a Gaussian to the transfer function of the smoothing filter for that level, and define the scale of the smoothed signal as the scale of the fitted Gaussian. This is done by sampling the transfer function h_l in Eq. (9) over the range $[0, 2]$. This range is chosen since the frequency content of the signal is defined in terms of the eigenvalues of the Laplace-Beltrami operator $\mathbf{L}$ and the choice of weights used to construct $\mathbf{L}$ in Eq. (6) guarantees that $-\mathbf{L}$ has real eigenvalues $0 \leq \omega_0 \leq \cdots \leq \omega_{N-1} \leq 2$ [11]. As a result, we obtain a set of pairs $\Gamma = \{(\omega_j, h_l(\omega_j))\}_{j=0}^{J-1}$, which is used to estimate the scale t_l of a fitted Gaussian $g_l(\omega, t_l) = e^{-\omega^2 t_l}$, in the least-squares sense:

$$t_l = \frac{\sum_{j=0}^{j<|\Gamma|} \omega_j^2 \sum_{k=0}^{k<l-1} \ln(1 + \lambda_k \omega_j^2)}{\sum_{j=0}^{j<|\Gamma|} \omega_j^4} \ . \tag{10}$$

The scale parameter t_l for each level l can alternatively be defined in terms of the variance of the transfer function at that level. Since the transfer function at each level is analytic and only depends on the known sequence of time steps, λ_l, its variance can be precomputed numerically. The obtained sequence of scales, $(t_0, \ldots, t_{L-1})$, together with the stack of smoothed signals, $(\mathbf{F}^0, \ldots, \mathbf{F}^{L-1})$, define the CS3 representation of the surface.

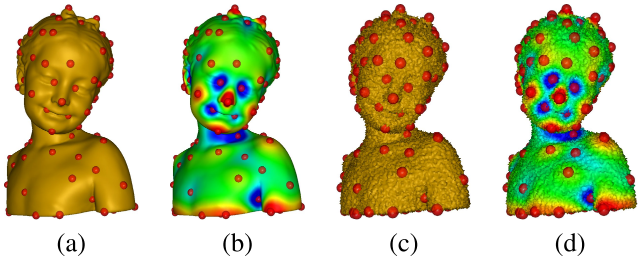
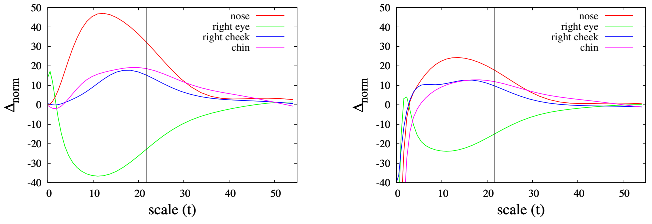


Fig. 4. Extracted features on (a) original, and (c) noisy Bimba models at $t = 21.7$; the false-colors in (b) and (d) reflect the response of the Δ^{si} (Eq. (16)) at each vertex on the original and noisy models, respectively. The models in (c) and (d) contain 80% Gaussian noise.



(a) original (b) noisy

Fig. 5. Plots of LoC values of a few vertices on the surfaces in Fig. 4. The vertical black lines indicate the location of the displayed scale ($t = 21.7$) in Fig. 4.

C. Feature Extraction with Auto Scale Selection

The CS3 representation of a 3D surface may be used directly for feature extraction. Let $\Phi(\mathcal{M}) = (\mathbf{F}^0, \ldots, \mathbf{F}^{L-1})$ and $\Psi(\mathcal{M}) = (t_0, \ldots, t_{L-1})$ correspond to the CS3 representation of surface $\mathcal{M}$. The difference between the smoothed signals at two consecutive levels l and $l+1$ can be used to approximate the Laplacian of the signal at level l. This difference can be expressed as a convolution of the original signal with Gaussian filters:

$$\mathbf{F}^{l+1} - \mathbf{F}^l \approx \mathbf{F}^0 * (g(\cdot; t_{l+1}) - g(\cdot; t_l)) \ , \tag{11}$$

where $*$ denotes convolution defined over the surface and $g(\cdot; t_l)$ is a Gaussian with scale t_l. Noting that $\frac{\partial g}{\partial t} = 0.5\Delta g$, we have

$$\frac{\partial g}{\partial t} = 0.5\Delta g \approx \frac{g(\cdot; t_{l+1}) - g(\cdot; t_l)}{t_{l+1} - t_l} \ , \tag{12}$$

and consequently,

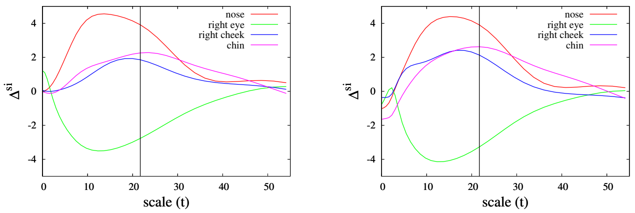
$$\mathbf{F}^{l+1} - \mathbf{F}^l \approx 0.5(t_{l+1} - t_l)\mathbf{F}^0 * \Delta g \ . \tag{13}$$

Therefore, the estimated Laplacian of $\mathbf{F}^0$, at level l, which we denote by $\Delta \mathbf{F}^l$, is approximated by

$$\Delta \mathbf{F}^l \approx \frac{2(\mathbf{F}^{l+1} - \mathbf{F}^l)}{t_{l+1} - t_l} \ . \tag{14}$$

We define the *scale-normalized* Laplacian of the surface signal at scale t_l as

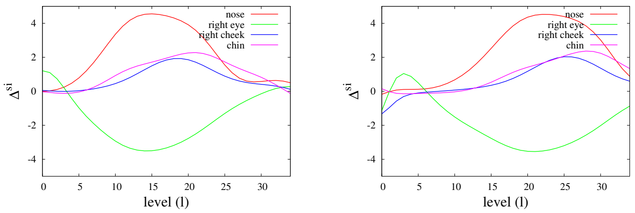
$$\Delta_{norm} \mathbf{F}^l = t_l \Delta \mathbf{F}^l = \frac{2t_l(\mathbf{F}^{l+1} - \mathbf{F}^l)}{t_{l+1} - t_l} \ . \tag{15}$$



(a) original

(b) noisy

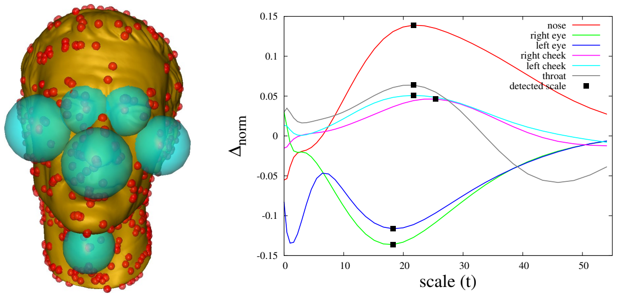
Fig. 6. Plots of the *scale-invariant* LoC values of a few vertices on the surfaces in Fig. 4.



(a) spatial scaling: ×100

(b) sampling resolution: ×4

Fig. 7. Comparison of scale-invariant LoC plots of the Bimba model (Fig. 4(a)) with different spatial scales and sampling resolutions. Plot in (a) is identical to the plot for the original model, shown in Fig. 6(a), while (b) has been shifted to the right by approximately 7 levels. Note that, unlike the plots in Figs. 5 and 6, the x-axis denotes the level in the CS3 stack, and not the scale.



(a)

(b)

Fig. 8. Automatic scale selection on the Caesar model. (a) Estimated scales at a few locations; the radii of the blue spheres are computed using Eq. (18). (b) Plots of the scale-normalized Laplacian of the surface mean curvatures at the selected vertices as functions of scale; the locations of the filled squares on the scale-axis indicate the detected scale for the keypoints.

Throughout this work, we assume that the surface signal corresponds to the surface mean curvatures. $\Delta_{norm}\mathbf{F}$ then corresponds to the scale-normalized Laplacian of mean Curvatures (LoC).

The local extrema of $\Delta_{norm}\mathbf{F}$ could be used to define feature points on a 3D model. For example, Fig. 4 depicts the computed scale-normalized Laplacian of mean curvatures on a 3D model and its noisy counterpart, at level $l = 20$ (scale $t = 21.7$); the red spheres indicate the locations where LoC is locally maximum or minimum at the displayed level. As seen in the figure, the detected locations of the extrema of LoC, despite their high differential order, are robust to noise and may be used for extraction of stable and well-localized feature points.

The plots in Fig. 5 show the computed LoC values at a few selected vertices on the original and noisy models in Fig. 4 as a function of scale. As expected, the values for both of these models converge at the higher scales. However, the corresponding LoC values of the vertices at the scale shown in Fig. 4 are not the same between the two models due to the noise. To alleviate this, we introduce the *scale-invariant* LoC as

$$\Delta^{si}\mathbf{F}^l = \frac{\Delta\mathbf{F}^l - \bar{\mathbf{F}}^l}{\sigma_l} \ , \tag{16}$$

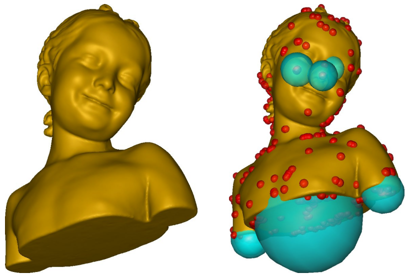
where

$$\bar{\mathbf{F}}^l = \frac{1}{N}\mathbf{1}^\top\Delta\mathbf{F}^l\mathbf{1}, \quad \sigma_l = \frac{1}{\sqrt{N}}\|\Delta\mathbf{F}^l - \bar{\mathbf{F}}^l\| \ , \tag{17}$$

denote the vector-form mean, and standard deviation of the LoC values at level l, respectively; N is the total number of vertices in $\mathcal{M}$, and $\mathbf{1}$ is an N-dimensional vector of all 1's.

Fig. 6 shows the scale-invariant LoC plots of the same vertices as in Fig. 5. As can be seen, the LoC curves of the two surfaces begin to converge at a much finer scale, and look more similar. The scale-invariant LoC is resilient to changes in sampling resolution, spatial scaling, and additive i.i.d. noise. Additionally, Fig. 7 provides a comparison between the Δ^{si} plots on the original, scaled, and higher resolution versions of the same model as in Fig. 4(a). The higher resolution version of the model was obtained by applying one iteration of Loop's subdivision scheme, which increases the number of mesh vertices by a factor of 4. As can be seen, spatial scaling of the model has no effect on the plotted Δ^{si} curves, while the increase in the resolution of the surface shifts the curves to the right.

According to the principle of automatic scale selection [7], the scale(s) where $\Delta_{norm}\mathbf{F}_i$ becomes a local extremum across scales can be expected to correspond to the size of surface structures at vertex v_i. This is visually verified in Figures 8 and 9, where the size of the blue spheres indicate the computed spatial scale (neighborhood size) at a few selected keypoints. An approach similar to Lowe's [8] was used to select the keypoints (shown as red spheres) on the models, in the two figures. The keypoints were selected as the vertices that were



(a)

(b)

Fig. 9. Automatic scale selection on the Bimba model. (a) Original model; (b) estimated scales at a few locations on the model.

local extrema among their immediate neighbors, both on the current level and two adjacent levels on the stack: let set $Q^l(i) = \{\mathbf{F}_j^{l+k}\} \cup \{\mathbf{F}_i^{l-1}, \mathbf{F}_i^{l+1}\}$, for $k = -1, 0, 1$, and $j \in \mathcal{N}(i)$. Then, vertex v_i, at level l, is selected as a keypoint if $\mathbf{F}_i^l > q_j, \forall q_j \in Q^l(i)$ or $\mathbf{F}_i^l < q_j, \forall q_j \in Q^l(i)$. Let t_l be the scale associated with level l. t_l then defines the scale of the detected keypoint v_i. The radius of each blue sphere in Figures 8 and 9 was computed using

$$r = t_l \bar{e} \ , \tag{18}$$

where $\bar{e}$ is the average edge length in the surface mesh.

The graph in Fig. 8(c) shows the plots of LoC values at the few selected keypoints (blue spheres) on the model in Fig. 8(a). The filled squares on the curves indicate the location of the detected scale for each keypoint.

IV. Application: Automatic Face Detection

The CS3 representation together with the feature extraction procedure described in the previous section can be used in 3D matching tasks, such as surface registration and 3D object recognition. In this section, we describe how our proposed representation can be used to detect the face region on 3D surfaces containing human faces with arbitrary scale, orientation, and translation.

The detection system consists of two main stages. In the offline (training) stage, a classifier is trained with a set of 3D human faces. The training set consists of faces with and without expressions. The online (matching) stage involves the actual detection of the human face on an input 3D surface. Sec. IV-A describes the training stage, while the matching stage of the algorithm is described in Sec. IV-B.

A. Face Detection: Training

The objectives of the training phase are: (1) facial feature identification and AFM computation, and (2) face region descriptor computations.

Facial Feature Identification and AFM Computation. In this step, a user manually selects a predefined set of facial features (*e.g.*, the nose tip, eyes, chin) on each 3D face in the training set. These features are then used to register all the surfaces with a reference face model. These features are additionally used to obtain a cropped version of each of the surfaces such that only the face region is included. Finally, all the cropped faces in the training set are averaged together to obtain an Average Face Model (AFM). Example of an AFM is shown in Fig. 2(d).

Face Region Descriptor Computations. In this step, local shape descriptors and feature vectors (described in Sec. IV-B) are computed on the nose tips of the faces in the training set. These descriptors are used to define the distance (dissimilarity) measure between 3D surfaces that is needed to detect the face region in the matching stage.

We define the distance of a feature vector $\mathbf{z} \in \mathbb{R}^D$ from the distribution of the feature vectors in the training set as the squared Mahalanobis distance

$$d_{c_f}^2(\mathbf{z}) = (\mathbf{z} - \bar{\mathbf{x}}_{c_f})^\top \mathbf{\Sigma}_{c_f}^{-1} (\mathbf{z} - \bar{\mathbf{x}}_{c_f}) \ , \tag{19}$$

where $\bar{\mathbf{x}}_{c_f}$ and $\mathbf{\Sigma}_{c_f}$ correspond to the mean and covariance matrix of the feature vectors in the training set. To estimate Eq. (19) efficiently and reliably, we use a similar approach to [21], where Principal Component Analysis (PCA) is used to reduce the dimensionality of the data. The approach, however, retains some information about the less significant principal components of the data, which is generally discarded in other PCA-based approaches.

B. Face Detection: Matching

The main goal of the algorithm's online stage is to identify the location, orientation, and scale of the face region. This region is centered at the nose tip because the nose bridge helps establish orientation and symmetry, and its high curvature is easily detected using our keypoint extraction method. This is accomplished in the following steps: keypoint extraction / local scale estimation, global scale adjustment, local descriptor computation, and detection.

Keypoint Extraction/Local Scale Estimation. We use a slight modification of the approach discussed in Sec. III-C to obtain the location and scale of a set of keypoints on the input surface. Note that since the signal that is being smoothed in our representation is the surface mean curvature, the extracted keypoints correspond to locations of blob-like features on the surface; this is verified in Fig. 4. Since we are interested in identifying the location of the nose tip, these features provide an ideal choice. Additionally, the size of each blob is estimated by the associated scale of the keypoint at that location using Eq. (18).

Global Scale Adjustment. One advantage of estimating the size of the surface structures, as described in Sec. III-C, is that the computed values are intrinsic to the surface. The relative scale of an input 3D surface with respect to the 3D face models in the training set can therefore be estimated by comparing the detected sizes of the surface structures. To do this, the median radius of the surface structures on the surfaces in the training set are computed using Eq. (18), offline. Similarly, the median radius of the structures on the input 3D surface are computed. The ratio of these radii are then used to estimate and then adjust the relative scale of the input model with respect to the models in the training set.

Local Shape Descriptor Computation. Given a keypoint and the estimated size of the face region (obtained from the faces in the training set), we construct a local coordinate system at the point and represent the positions of its neighboring vertices in the form of a height map. We use the Multilevel B-Spline Algorithm [22], to obtain a continuous map of the local neighborhood around the keypoint, which is then turned into an image. This image defines the local descriptor employed by our detection system.

The local coordinates at the keypoints need to be constructed in a manner which are invariant to any transformation the 3D model may undergo. We use PCA of the local neighborhood around each point of interest to construct the local coordinate system. Let p_0 denote the keypoint where the local coordinate system is being constructed, and $k_1 \geq$

$k_2 \geq k_3 \geq 0$, u_1, u_2, u_3 denote the eigenvalues and unit eigenvectors of the covariance matrix of the vertex positions in the neighborhood around p_0, respectively. u_3 approximates the normal direction, while u_1 and u_2 span the tangent plane at p_0. u_3 defines the z direction of the constructed local coordinate system, and u_2 and u_1 define the directions of the x and y axes, respectively.

Since the eigenvectors provide only information about direction and not orientation, we make the following adjustments: u_3 is adjusted so that it has the same orientation as the surface normal at p_0, while the orientations of u_1 and u_2 are changed such that $u_2 \times u_1 = u_3$. u_1 and u_2 may still be rotated by 180 degrees. To overcome this ambiguity, we create two height maps using the two possible orientations. In the detection stage, both maps are used, and the one with the worst performance is ignored.

In our implementation, the covariance matrix was weighted by the LoC values at each vertex. The weighting scheme helped the two major eigenvectors of the covariance matrix to be better-aligned with the underlying elongated structures of the surface at each keypoint.

Detection. The goal of the detection stage is to find the location of the face region on an input 3D surface using the computed local descriptors at the keypoints. Each local descriptor is converted into a feature vector by concatenating the rows in its height map image. Let $\mathbf{z}_i$ denote the feature vector associated with keypoint i, and let $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{I}$ be the set of all such feature vectors extracted on the input surface. The ideal objective of detection would then be to identify all $\mathbf{z}_i$'s which correspond to a face region—and as a result, having the ability of identifying multiple faces on the same model. However, in this work, we seek to solve a simpler problem, namely, finding the most likely keypoint that belongs to the face region. The detection task is therefore to find $\mathbf{z}^*$, minimizing $d_{cf}^2(\mathbf{z}_i)$ among all $\mathbf{z}_i \in \mathbf{Z}$, where d_{cf}^2 denotes the estimated squared Mahalanobis distance as defined by Eq. (19).

Once the most likely location of the face region is identified on the input surface, we use its location and associated local coordinate system to obtain an initial guess for the relative pose of the input with respect to the AFM in the training set. The estimated position and orientation are then iteratively improved using the ICP algorithm [23]. In each iteration, we also improve the initial guess for the scale by finding s that minimizes

$$E = \sum_{\mathbf{p}_i \in M_{afm}} \min_{\mathbf{q}_j \in M_{in}} \|\mathbf{p}_i - s\mathbf{q}_j\|^2 \ , \tag{20}$$

where M_{in} and M_{afm} denote the point sets corresponding to the input surface and the AFM, respectively. Both surfaces are assumed to have been translated so that the nose tip is at the origin. This prevents scaling from changing the relative positions of the nose tips between the two surfaces.

Once the input surface has been aligned with the AFM, the locations of auxiliary facial features, which were specified in the training phase, are identified on the surface. Fig. 2(d) shows the locations of these features, which were automatically detected by our system, on Fig. 2(a). These features are then used to crop and remesh the face region using an approach similar to [24]. Fig. 2(e) shows an example of the final result produced by our face detection procedure. Note that in [24] the locations of the facial features on the input model were specified manually, whereas our system detects them automatically.

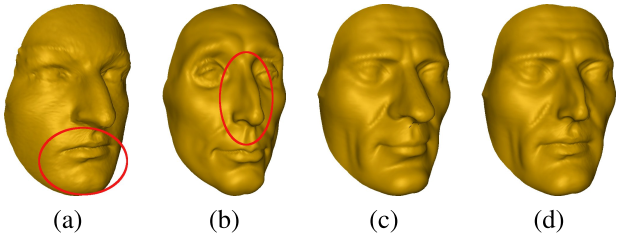


Fig. 10. Automatic face warping results. (a) target mouth; (b) target nose; (c) source face; (d) resulting hybrid face.

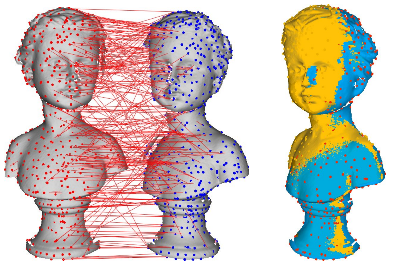
V. Face Detection Results

We tested the performance of our detection system on a database of 1068 models, which included artificial models with human faces and more than 1000 3D scanned human faces from the FRGC database [25]. The majority of the FRGC scans where from Spring 2003, which included large portions of clothing. All models were randomly scaled (by a factor in the range $(0, 1000]$) and rotated prior to detection. The correct detection rate of the system was 92.13%. The required time to build the CS3 representation with 35 levels for a model with approximately $113.5K$ vertices was 32secs. The overall detection time on the same model was 180secs on a 2.4 GHz CPU.

As mentioned previously, the output of the detection system, which results in normalized and compatibly remeshed 3D surfaces, can directly be used in a 3D face recognition system. Additionally, the extracted 3D faces may be used in automatic 3D processing tasks such as swapping of facial features between 3D faces. We employed the differential coordinates approach of [26] to enable a user to automatically specify and swap facial features between various face models. For example, Fig. 10 shows an example of a warping task involving three different faces. Our system allowed the user to easily specify which facial features on the source face (Fig. 10(c)) were to be replaced by the ones on the target faces (Fig. 10(a) and Fig. 10(b)); Fig. 10(d) shows the resulting hybrid face. Since the faces were normalized and the meshing was consistent between the surfaces, the user only needed to specify the location and region of influence of the features on the source face. The resulting hybrid faces may, in turn, be used to generate new faces and extend the size of a training set.

VI. Application: Surface Registration

Automatic surface registration is another application of our proposed CS3 representation and feature extraction procedure.



(a) (b)

Fig. 11. Surface registration results: (a) point-point correspondences established between the two surfaces using scale-invariant LoC shape descriptors; (b) final registration results

We implemented a registration system, which used the CS3 representation for establishing correspondences between two input surfaces. Given the point-point correspondences, the unknown transformation between the two surfaces was estimated, and consequently the surfaces were registered. The following summarizes the general steps involved in registering two input surfaces ($\mathcal{A}$ and $\mathcal{B}$) using the CS3 representation. An example is shown in Fig. 11.

1) **CS3 Computation.** Obtain the CS3 representations of each of the surfaces (Sec. III-A).
2) **Keypoint Extraction.** Use the keypoint extraction procedure described in Sec. III-C to extract keypoints on the two surfaces.
3) **Local Descriptor Computation.** Use the scale-invariant LoC curves described in Sec. III-C to construct feature vectors at all extracted keypoints on the two surfaces.
4) **Registration.** For each keypoint k on $\mathcal{A}$ find its correspondence on $\mathcal{B}$ by finding the keypoint on $\mathcal{B}$ whose feature vector has the smallest L_2 distance from the feature vector of k. Finally, use a branch and bound approach similar to [3] to eliminate incorrect point-point correspondences and register $\mathcal{B}$ with $\mathcal{A}$.

VII. Conclusion

In this work, we presented a new scale-space based representation for 3D surfaces, which is useful for feature extraction and partial shape matching. Our proposed representation is defined in terms of the evolution of the surface curvatures according to the heat equation. This representation was shown to be insensitive to noise. In addition, other major benefits of our method over the most relevant approaches, such as [19] and [20], are the capability of automatic scale selection and improved computational efficiency. We presented an application of our approach to partial 3D shape matching involving detection of 3D faces on input surfaces with arbitrary scale, orientation, and translation. The output of our detection system could directly be fed into a 3D face recognition system. Additionally, we showed other applications involving automatic processing of 3D faces, such as the generation of hybrid faces, and general surface registration. In future work, we will improve the performance of our detection system and also use the CS3 representation for 3D face recognition.

References

[1] Tangelder and Veltkamp, "A survey of content based 3d shape retrieval methods," in *SMI '04: Proceedings of the Shape Modeling International 2004*, 2004, pp. 145–156.

[2] T. Funkhouser, S. Rusinkiewicz, and M. Kazhdan, "Princeton 3D Model Search Engine," http://shape.cs.princeton.edu/.

[3] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann, "Reassembling fractured objects by geometric matching," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 569–578, 2006.

[4] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Proc. Eurographics symposium on Geometry processing*, 2005, p. 197.

[5] A. P. Witkin, "Scale-space filtering," in *Proceedings of the Eighth international joint conference on Artificial intelligence*, 1983, pp. 1019–1022.

[6] Tony Lindeberg, "Scale-space theory in computer vision," *Monograph 1994*, 1994.

[7] T. Lindeberg, "Scale-space theory in computer vision," 1994.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[9] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vision*, vol. 74, no. 1, pp. 59–73, 2007.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008, similarity Matching in Computer Vision and Multimedia. [Online]. Available: http://www.sciencedirect.com/science/article/B6WCX-4RC2S4T-2/2/c2c03b6165996e30312e5b7c7b681155

[11] G. Taubin *et al.*, "Geometric signal processing on polygonal meshes," *Eurographics State of the Art Reports*, 2000.

[12] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *SIGGRAPH '99*, 1999, pp. 317–324.

[13] A. Mian, M. Bennamoun, and R. Owens, "Automatic 3d face detection, normalization and recognition," in *Proc. 3DPVT*, 2006, pp. 735–742.

[14] V. Ayyagari, F. Boughorbel, A. Koschan, and M. Abidi, "A New Method for Automatic 3D Face Registration," *Comp. Vision and Pattern Recog. (CVPR) Workshop*, pp. 119 –119, June 2005.

[15] J. Novatnack, K. Nishino, and A. Shokoufandeh, "Extracting 3D shape features in discrete scale-space," in *3DPVT06*, 2006, pp. 946–953. [Online]. Available: http://dx.doi.org/10.1109/3DPVT.2006.60

[16] M. Novotni, P. Degener, and R. Klein, "Correspondence generation and matching of 3d shape subparts," Universitt Bonn, Tech. Rep., 2005.

[17] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 659–666, Jul. 2005.

[18] M. Schlattmann, P. Degener, and R. Klein, "Scale space based feature point detection on surfaces," *Journal of WSCG*, vol. 16, no. 1-3, February 2008.

[19] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Eurographics Symposium on Geometry Processing (SGP)*, 2009.

[20] M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," *Comp. Vision and Pattern Recog. (CVPR)*, pp. 1704 –1711, June 2010.

[21] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 696 –710, jul. 1997.

[22] S. Lee, G. Wolberg, and S. Y. Shin, "Scattered data interpolation with multilevel b-splines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, pp. 228–244, 1997.

[23] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and machine Intelligence*, vol. 14, no. 2, pp. 239–258, Feb. 1992.

[24] X. Li, T. Jia, and H. Zhang, "Expression-insensitive 3d face recognition using sparse representation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[25] P. J. Flynn, "FRGC database v2.0, 2003," http://bbs.bee-biometrics.org/.

[26] M. Alexa, "Differential coordinates for local mesh morphing and deformation," *The Visual Computer*, vol. V19, no. 2, pp. 105–114, May 2003. [Online]. Available: http://dx.doi.org/10.1007/s00371-002-0180-0106