Vision, Modeling, and Visualization (2016), pp. 1–3
M. Hullin, M. Stamminger, and T. Weinkauf (Eds.)

# LaTeX Author Guidelines for EUROGRAPHICS Proceedings Manuscripts

D. W. Fellner[† 1,2] and S. Behnke[2]

[1]TU Darmstadt & Fraunhofer IGD, Germany
[2]Institut für ComputerGraphik & Wissensvisualisierung, TU Graz, Austria

**Abstract**

*The ABSTRACT is to be in fully-justified italicized text, between two horizontal lines, in one-column format, below the author and affiliation information. Use the word "Abstract" as the title, in 9-point Times, boldface type, left-aligned to the text, initially capitalized. The abstract is to be in 9-point, single-spaced type. The abstract may be up to 3 inches (7.62 cm) long.*
*Leave one blank line after the abstract, then add the subject categories according to the ACM Classification Index (see http://www.acm.org/class/1998/).*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

## 1. Introduction

Please follow the steps outlined in this document very carefully when submitting your manuscript to Eurographics.

You may as well use the LaTeX source as a template to typeset your own paper. In this case we encourage you to also read the LaTeX comments embedded in the document.

## 2. Related work

Please read the following carefully.

## 3. Review of method in [1]

First, they compute 2D landmarks for all images using the approach in [5]. With the 2D landmarks, Warp the 3D model on the images by project 3D landmarks on 2D landmarks on image. Estimate the weak prospecive projection matrix $P_i$ for each image. Deform the 3D template using laplacian surface editing and adapt for the lanmark constraints until convergence. Update the vertex verctor $X$ by minimizing the energy function:

$$E_{warp}(X,P_i) = \|\mathcal{L}X - \mathcal{L}X_0\|^2 + \lambda_l \sum_i \|P_i D_i X - W_i\|^2$$

where $\mathcal{L}$ is a discretization of the Laplace-Beltrami operator, with the weight $\mathcal{L}_{ij} = \frac{1}{2}(cot\alpha_{ij} + cot\beta_{ij})$ where $\alpha_{ij}$ and $\beta_{ij}$ are the opposite angles of edge if in the two incident triangles. $D_i$ is the selection matrix picking out the landmarks that have a correspondence in image $i$, it's a diagonal matrix with 1 on the diagonal for the vertices corresponding to such a landmark and 0 everywhere else. The deformed 3D template will be used as the initial template for the following steps.

Then they use photometric stereo technique to estimate normals. Refer to approach in [3], store each warped image reflectance intensity in a matrix $M$, where each row represents the pixels in one image. For the non-frontal images, some vertices are not visible. They set the intensity of the non-visible vertices as zeros and use matrix completion [26] to fulfill the missing value to obtain the full $M$. Factorize the matrix $M$ using SVD and take the rank-4 approximation to estimate the shape $S$ and light $L$, where $S$ contains the normals $n$ and albedo $\rho$, and $L$ contains the light directoin. Select the images that are modeled well by the rank-4 approximation, i.e.,$\|M - LS\| < \varepsilon$. Since the factorization is not unique, it leads to one ambiguity in $S$ and $L$, i.e., $LS = (\tilde{L}A^{-1})(A\tilde{S})$. Recover the ambiguity by solving for $\arg\min_A \|S^t - A\tilde{S}\|^2$, where $S^t$ is the shape vector of the template.

Given the normal vectors, they reconstruct the shape $X$ from the normals $n$ through mean curvature $H$ by minimizing $\|\mathcal{L}X - Hn\|^2$, where $H$ is the mean curvature. The mean curvature $H$ is estimated as $H_i = \frac{1}{4A_i}\sum_{j\in N(i)}(cot\alpha_{ij} + cot\beta_{ij}))e_{ij}\cdot(n_i - n_j)$, where $N(i)$ is the set of incident neighboring vertices of vertex $i$, $A_i$ is the sum of the triangles' areas incident to $i$, $e_{ij}$ is the edge from $i$ to $j$. The cotan weights are the same as those in the Laplacian operator $\mathcal{L}$. Update the cotan weights in each global iteration.

On the boundary, the mean curvature formula degenerates into a

† Chairman Eurographics Publications Board

1D version, the shape vectore $X$ is updated by minimizing $\|\mathcal{L}_b X - \mathcal{K} b\|^2$, where $\mathcal{L}_{b,ij} = \frac{1}{e_{ij}}$ and $\mathcal{K}$ is the geodesic curvature along the boundary and $b$ is the cross product between the surface normal and the boundary tangent.

Finally, the mesh is deformed by minimizing the energy as following:

$$E = \|\mathcal{L}X - H^k n\|^2 + \lambda_b \|\mathcal{L}_b X - \mathcal{L}X^k\|^2 + \lambda_l \sum_i \|P_i D_i X - W_i\|^2$$

From the results shown in [1], we can see that the frontal face of the 3D reconstructed model fits to the individual, but the profile doesn't fit well, especially the nose and chin.The nose is supposed to be taller and the chin should be more curved. So we propose to use silhouette of the faces to refine the reconstructed face shape.

## 4. Approach

Continue from the work in [1], we refine the face model to make the shape fit better with the real image. We extract the silhouettes on 3D model and in each 2D images and build correpondence between the silhouette vertex on face model and the silhouette points in images. Then add silhouette constraint in the mesh deformation.

The main step can be described as following:

1. Reconstruct a 3D model using the method in [1].

2. Use photometric stereo to estimate the normal and compute the mean curvature using the propsed formula in [1].

3. Estimate perspective projection matrix and rotation matrix. Extract 3D and 2D silhouette candidates for each image. Build up correspondence between 2D and 3D silhouette candidates. Discard the silhouette candidate points only show in few images.

4. Reconstruct face model with silhouette constraint.

5. Go back to step 2 until convergence.

In the following subsection, we will give the details of step 2 to step 4.

### 4.1. Silhouette Extraction

To extract silhouette on 3D model, first we detect the points on 3D mesh whose normal are parallel to the image plane. Given the estimated perspective projection matrix, we can estimate the rotation matrix $R_i$ for each image. The view direction then can be estimated from the rotation matrix. Suppose the direction perpendicular to the frontal face is z-axis, then for $i$-th image, view direction $v_i = R_i * [0;0;1]$. Find the silhouette candidate points on 3D model whose normals are perpendicular to the view direction, the cosine of the angle between view direction and normal should be near to zero, i.e, $\frac{|v_i \cdot n_j|}{\|v_i\| \cdot \|n_j\|} < \varepsilon$, where $n_j$ is the normal of vertex $j$ on 3D mesh and $\varepsilon$ is a small value near to zero.

To avoid noises, choose the points whose incident faces have faces that are front-facing to the view direction and faces that are back-facing to the view direction. Among the silhouette candidates, the points sheltered by the nose has no edge on the 2D image, so these points will be discarded. To guarantee proper extraction of silhouette from the images, only the "nonfrontal" images are used. From the rotation matrix, we can estimate the yaw, pitch and roll of the face pose for each image. Choose the images in which the yaw of the face is bigger than a threshold. The points satisfying the above constraints are considered as silhouette candidates on 3D model, we denote it as $X_{sil3D}$.

### 4.2. Build correspondence between 3D silhouette and 2D silhouette

Let $S_i$ denotes the silhouette candidates of 3D model for image $i$. Warp $S_i$ on image $i$. Warp the silhouette candidate $X_{sil3D}$ on the images and move the candidates to the nearest points on an edge in the image. From the edge of one image, for each silhouette candidates on the 3D face model $X^i_{sil3D}$ , find the closest edge points by minimizing the square distance between warped silhouette point and edge points, i.e.,

$$\arg\min_{X_{sil2D}} \| \; X^i_{sil3D} - X^i_{sil2D} \|^2$$

Since the reconstructed model is close to the real shape of the individual, the projected silhouette should not be too far away from the correspongding edge. We set a threshold for the square distance, if the distance is larger than the threshold, that point will be considered as outlier and rejected. This can also reduce the effect by the extreme expession.

For each image, we obtain the silhouette point sets and the corresponding vertex on 3D mesh. Then for each silhouette point on 3D model, we can know it's related to which images. Let $X_j$ denotes the silhouette vertex $j$, $I_j$ denotes the image set containing $X_j$ . $I_{ij}$ denotes image $i$ containing silhouette points correponding to $X_j$ .

In our apporach, the face model is reconstructed from the average normals of images. The silhouette points that only appear in few images may lead to pointy region on 3D mesh. Especially for the chin, the extreme expression such as big mouth, may result in a bad estimation of the shape $X$. So the points showing in few images will be discarded to decrease the influence of noise in the slihouette extraction and the various expressions.

### 4.3. Reconstruction using silhouette points

Given the silhouette points on 3D model and the corresponding sihouette in 2D images, we add the silhouette constraint to the energy mentioned in [1] and we can get:

$$\begin{aligned} E = \|LX_{new} - H^k n\|^2 + \lambda_l \sum_i \|P_i D_i X_{new} - W_i\|^2 \\ + \lambda_b \|L_b X_{new} - LX\|^2 + \lambda_s \sum_{j \in V_{sil}} \sum_{s \in I_{sil}} \|P_{sj} D_{sj} X_{new} - S_{sj}\|^2 \end{aligned} \tag{1}$$

where $I_{sil}$ is the image set containing silhouette points, $V_{sil}$ is all the silhouette vertex on 3D model. $P_{sj}$ is weakly prospective projected matrix between silhouette vertices $X_j$ and 2D silhouette points in $s$ -th images, $D_{sj}$ is a diagonal matrix selecting out the silhouette vertex that have a correspondence in image $s$, . $\lambda_s$ is the silhouette constraint weight.

$X$ can be solved in a linear system as:

$$\begin{aligned}&(\mathcal{L}^2+\lambda_b\mathcal{L}_b^2+\lambda_l\sum_i D_iP_i^TP_iD_i+\lambda_s\sum_j\sum_s D_{sj}P_{sj}^TP_{sj}D_{sj})X\\&=\mathcal{L}H+\lambda_b\mathcal{L}_b^2X+\lambda_l\sum_i(P_i^T)W_i+\lambda_s\sum_j\sum_s P_{sj}^TD_{sj}S_{sj}\end{aligned} \tag{2}$$

## 5. Experiment

In this section we present the visualization result of the improvement of our algorithm.

**Dataset** We collect the photo of celebrities on the website. First we tried to use Google API to collect the photo by searching celebrities' name, but less than half of the result can be use for the experiment, since many of the returned images are not portrait or are related to the celebrity. So we collect the images on the websites that own specific webpage for the celebrities, Mtime and Douban. And we write a python script to download images for different individuals. For the initial template used to reconstruct the model, we use the face model in [].

Extract silhouette points in a number of images of the dataset. Geoge Clooney(169/1152), Kevin Spacey(85/468), Edward Norton(93/707), Tom Cruise(95/695) and James McAvoy(149/1038), WentworthMiller(106/737). The former number is the images used for extracting silhouette points, the latter number is the total amount of images for each celebrity.

**Silhouette**

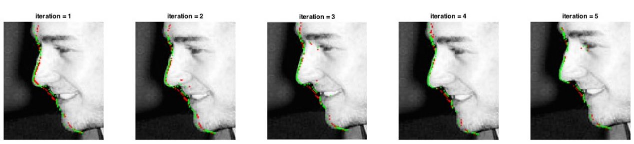


**Figure 1:** *the warped silhouette in each iteration. Red: warped silhouette. Green: extracted 2D silhouette*

Fig.2 plots the warped silhouette points on the images. We can see that the after adding silhouette constraint to the reconstruction step, the silhouette of the 3D mesh (yellow points) match better with the real silhouette.

From the result in Fig.3 we can see, compared with the previous method, our result has obvious improvement on the nose and the chin. Also the eyes in the profile. When the 3D model has good correspondence to the images, it helps the photometric stereo approach to estimate more accurate normals.

## 6. Conclusions

Please direct any questions to the production editor in charge of these proceedings.

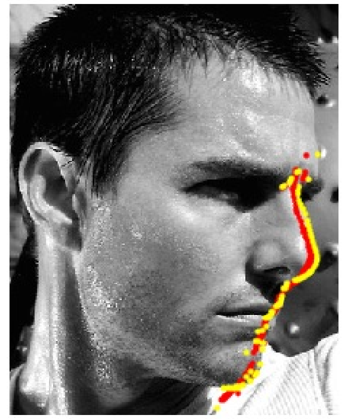

**Figure 2:** *the warped silhouette before and after using silhouette constraint. Red: before using silhouette. Yellow: after adding silhouette constraint*


**Figure 3:** *For publications with color tables (i.e., publications not offering color throughout the paper) please* ***observe****: for the printed version – and ONLY for the printed version – color figures have to be placed in the last page.*