Neurocomputing 172 (2016) 134–142

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An efficient mesh-based face beautifier on mobile devices

Xin Fan [a,*], Zhi Chai [a], Yuyao Feng [a], Yi Wang [a], Shengfa Wang [a], Zhongxuan Luo [a,b]

[a] School of Software, Dalian University of Technology, Dalian, Liaoning 116620, China
[b] School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning 116021, China

ARTICLE INFO

Article history:
Received 31 October 2013
Received in revised form
30 July 2014
Accepted 23 September 2014
Available online 13 May 2015

Keywords:
Face beautifier
Triangular mesh
Laplacian
Mobile devices
Social networking service

ABSTRACT

The pursuit of beautiful appearance for human greatly inspires the studies on digital beautifiers especially for the popularity of social networking services (SNS) in this booming era of mobile phones. Unfortunately, existing beautification techniques are not efficient enough for mobile devices that have limited computation and storage resources. In this paper, we propose a beautifier that is efficient on both space and time, and develop a mobile APP available online. We focus on the local beautification of facial chin that has a great effect on facial attractiveness, and simply use an averaged chin shape as the template. We leverage Laplacian as the constraint to remap the original shape to the target, and recursively apply the Laplacian constraint to the contour points from the coarse to fine scales. Thus, we only need to inverse a small matrix for the remap, which requires quite low expenses on space and time. Finally, we apply an efficient warping algorithm based on triangular meshes to generate a beautified face image. The performance of the algorithm is validated by a survey on 70 facial images since no objective metric for beauty exists. Twenty volunteers are invited to grade the results labeled as 'worse', 'unchanged' and 'better'. The results show that 81.50% of inputs are believed becoming more beautiful, and another 10.07% of them are considered unchanged. We also provide the comparisons on the beautification quality and time expenses with the algorithm based on radial basis functions (RBF).

1. Introduction

Human's daily lives have been tied up with social network services (SNS) in this heyday of social media. People are more likely to share their best appearances to their social connections on Facebook, Twitter or Instagram in the cyberspace. Meanwhile, it is possible for users to conveniently modify their appearances by a *digital beautifier* developed upon modern digital multimedia processing techniques so that physical surgeries of high risks are not necessary in the context of SNS. One of the most straightforward way is to apply an APP installed on mobile devices in this decade when digital cameras of high resolution are merging with mobile phones. Unfortunately, mobile devices have limited resources on computational power and storage space compared with desktops. In this study, we address the issue of developing a digital beautifier of high efficiency on both space and time.

Facial beauty or attractiveness is such a subjective concept that early studies are typically performed from the psychological respect [14,22]. The study in [14] showed that certain regions of a human face including the chin, upper lip, and nose have the greatest effect on the judgment of facial appearance, and Thornhilla and Gangestad examined more factors for attractiveness, e.g., facial symmetry, averageness, and secondary sex characteristics [22]. Recently, researchers resort to the data driven approach to the attractive models when advanced computational methods and a large volume of data are available. Eisenthal et al. [6] proved that facial beauty is a universal concept and learnable from examples. Mu [16] investigated the prediction of facial attractiveness and established an aesthetics sensitive test. Gan et al. [9] leveraged convolutional deep networks (CDN) to learn hierarchical textural features for facial beauty, and [25] developed fully automatic facial beauty assessments in unstrained environments. Though these computational approaches provide flexible representation for attractive target faces, the storage of the learnt models typically require a huge space, e.g., millions of parameters in [9], and thus are not applicable to mobile APPs. Additionally, it is controversial to define the beauty only with databases, not mentioning that the learning results may vary with algorithms.

Great efforts have been also devoted to digital processing algorithms, specifically beautification techniques, when attractive models are given. Arakawa and Nomoto [2] developed a system for facial beautification by removing undesirable wrinkles and spots. Huang and Fuh [10] presented a prototype for facial color improvements, and [23] can further beautify facial images by adding cosmetic

* Corresponding author. Tel.: +86 411 87571638.
E-mail address: xin.fan@ieee.org (X. Fan).

http://dx.doi.org/10.1016/j.neucom.2014.09.093
0925-2312/

makeup effects. These works mainly focus on *textural* changes, and on the other hand, researches attempt to beautify facial *geometries*. Face attractiveness improvements are presented in [20] using beauty prototypes and statistical decisions. A beauty decision function is learned as a classifier that can discriminate whether a face is beautiful or not. Melacci et al. [13] proposed a method for face beautification by using a template that encodes the characteristics of attractive facial images. An input face given facial landmarks is compared with the template by means of Catmull–Rom splines (CRS), and moved toward the K nearest ones (KNN) by 2D warping. In [12], researchers explored a facial aesthetic enhancement with the help of attractiveness engine trained on a face dataset. They proposed two methods to build the map from the input to target, i.e., a support vector regressor (SVR) and KNN. The KNN algorithm is time consuming in the case of large number of landmarks when a complete (or global) face shapes are considered in both [13] and [12]. The SVR in [12] has to store the regressor parameters for global faces whose number is also relatively large for mobile devices. Regarding the warping algorithms given the geometry maps, the algorithms based on radial basis functions (RBF) [1,21,18] are widely used in these geometry based beautifiers. RBFs implicitly ensure smoothing interpolation, but the pixel-wise smoothness lags the efficiency of the warping.

It is worth noting that slight and local refinements on the appearance are desired in the SNS context since the loop of a person on SNS is typically expanded from his/her families, alumni and acquaintances who have the appearance of the subject *a priori*. In this paper, we develop an efficient face beautifier with *local* adjustment for mobile devices.[1] Specifically, we focus on the local chin shape that has greater effect on facial attractiveness [14] but less on face identification [19]. We use a simple contour as the target shape so that no extra space is required to store the parameters of a complex target model as in [9,16,25]. We leverage the Laplacian as the constraint to remap the original to the target, and recursively apply the Laplacian to the control points from the coarse to fine scales. Thus, we only need to inverse a small matrix for the deformation, which requires much lower expenses on space and time than SVR or KNN in [12,13]. Finally, we apply an efficient warping algorithm based on triangular meshes instead of pixel-wise RBFs to generate a beautified face image. These economic techniques for attractive template, landmark deformation (or remap) and image warping render the efficient beautification both on time and space.

The rest of paper is organized as follows. We describe our method in detail in Section 2 and give comparisons and experiments in Section 3. Section 4 concludes the paper.

2. Efficient face beautifier

As shown in Fig. 1, our beautifier starts from facial feature extraction and triangulations on the feature points of the face. We obtain the Laplacians of vertices lying on the face contour, and relocate the vertices with the Laplacian constraint. Finally, we generate the beautified images by texture mapping on triangular meshes. All the components of the procedure are detailed in this section.

2.1. Feature points extraction and triangular mesh establishment

As shown in Fig. 2, we extract 69 feature points from input images. Nineteen of them lie on the outer contour while the rest locates on eyes, nose, and mouth. Because our purpose is to slim the face, we choose the 19 points lying on the facial contour to adjust the facial geometry. These points have a great impact on facial attractiveness [14], and slightly affect the identification of a subject [19]. Hence, we are able to beautify a face with few modifications to the global geometry, which is desired for a facial beautifier in the context of SNS. We label these 19 points from 1 to 19 as shown in Fig. 2. The rest points are used to preserve the geometries of facial organs. We employ the regression based algorithm developed by [4] for the extraction in this study, while the classical active shape models (ASM) [5], probabilistic contour extraction [7], and the latest extraction algorithms including [28,24] are also applicable.

Then, we connect these feature points into polygons that represent the geometries of eyes, nose, mouth and facial contour as shown in Fig. 3(a). All these polygons are placed into a rectangle face region (polygon 6 in Fig. 3(a)) in order to make the shape remap more smoothing and natural. We take the polygons 1, 2, 3 and 4 as the holes of the polygon 5, and triangulate the polygon 5 into Delaunay triangles [3]. Similarly, we triangulate the polygon 6 by taking the polygon 5 as a single hole. The final triangular methods are shown in Fig. 3(b).

2.2. Efficient vertex remap with Laplacian

We employ the Laplacian constraints in [17] in order to deform vertices to a target, and also present an acceleration algorithm that efficiently remaps facial contours.

2.2.1. Vertex remap

The meshes built above can be represented as a graph $\mathbf{G}=(\mathbf{V},\mathbf{E})$ where $\mathbf{V}=[v_1^T, v_2^T, \ldots, v_n^T]^T$, $v_i=[v_{ix}, v_{iy}]^T \in \mathbb{R}^2$. The symbols v_1 to v_{19} correspond to points 1 to 19 ($P_1 \ldots P_{19}$) shown in Fig. 2. $\mathbf{V}$ is the set of vertices and $\mathbf{E}$ is the set of edges. We represent sub meshes $\mathbf{G}_o=(\mathbf{V}_o,\mathbf{E}_o)$, where $\mathbf{G}_o \subset \mathbf{G}, \mathbf{V}_o \subset \mathbf{V}, \mathbf{E}_o \subset \mathbf{E}$. The set $\mathbf{V}_o$ denotes the vertices to be remapped and $\mathbf{V}'$ denotes the new positions of $\mathbf{V}_o$. Edges in $\mathbf{E}_o$ are connected by vertices in $\mathbf{V}_o$. We use δ_i to represent the Laplacian of v_i

$$\delta_i = \sum_{\{i,j\} \in \mathbf{E}_o} w_{ij}(v_j - v_i) = \left[\sum_{\{i,j\} \in \mathbf{E}_o} w_{ij} v_j\right] - v_i, \tag{1}$$

where $\sum_{\{i,j\} \in \mathbf{E}_o} w_{ij}=1$, and the weights w_{ij} can be calculated as follows:

$$w_{ij} = \frac{\omega_{ij}}{\sum_{\{i,k\} \in \mathbf{E}_o} \omega_{i,k}}. \tag{2}$$

In this study, we choose $\omega_{i,j}=1$. There is no need to calculate the Laplacian of P_1 and P_{19} since they are the ending points of the facial contour having only one neighbor in $\mathbf{V}_o$. Thus, we use a $(n-2)\times n$ Laplacian matrix to obtain the Laplacian of the facial contour, where n is the number of elements in $\mathbf{V}_o$. The elements of Laplacian matrix is described as follows:

$$\mathbf{L}_{ij} = \begin{cases} -1, & i=j \\ w_{ij}, & i,j \in \mathbf{E}_o, \ \ i \neq 1 \text{ and } i \neq 19. \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Our algorithm is applied on each coordinate component separately. We denote $\mathbf{V}_d=[v_{1d}, v_{2d}, \ldots, v_{nd}]^T, d \in \{x,y\}, v_n \in \mathbf{V}_o, \mathbf{V}'_d$ is adjusted $\mathbf{V}_d$. and Δ_d is the Laplacians where $\Delta_d=[\delta_{1d}, \delta_{2d}, \ldots, \delta_{nd}]^T, d \in \{x,y\}$. We calculate Δ_d as

$$\Delta_d = \mathbf{L}\mathbf{V}_d. \tag{4}$$

The Laplacian constraints characterize the geometrical structures necessary to accurately describe the face. It is efficient to use Laplacians of the mesh to depict the key to the beautify. When we

[1] The APP is available online at http://www.windowsphone.com/en-us/store/app/change-your-face/a920c467-cd6a-4c62-a952-cf405400fade.