Facial Contour Labeling via Congealing

Xiaoming Liu, Yan Tong, Frederick W. Wheeler, and Peter H. Tu

Visualization and Computer Vision Lab
GE Global Research, Niskayuna, NY 12309
{liux,tongyan,wheeler,tu}@research.ge.com

Abstract. It is a challenging vision problem to discover non-rigid shape deformation for an image ensemble belonging to a single object class, in an automatic or semi-supervised fashion. The conventional semi-supervised approach [1] uses a congealing-like process to propagate manual landmark labels from a few images to a large ensemble. Although effective on an inter-person database with a large population, there is potential for increased labeling accuracy. With the goal of providing highly accurate labels, in this paper we present a parametric curve representation for each of the seven major facial contours. The appearance information along the curve, named *curve descriptor*, is extracted and used for congealing. Furthermore, we demonstrate that advanced features such as Histogram of Oriented Gradient (HOG) can be utilized in the proposed congealing framework, which operates in a dual-curve congealing manner for the case of a closed contour. With extensive experiments on a 300-image ensemble that exhibits moderate variation in facial pose and shape, we show that substantial progress has been achieved in the labeling accuracy compared to the previous state-of-the-art approach.

Key words: Facial contour, congealing, semi-supervised, HOG.

1 Introduction

This paper addresses the problem of estimating semantically meaningful *facial contours* from an image ensemble using *semi-supervised congealing*. The shape of an object can be described by object contours, which include both the overall object boundary and boundaries between key components of the object. By facial contour, in particular, we refer to the boundary of chin and cheek, as well as the facial features including eyes, eyebrows, nose, and mouth. Given a large set of face images, semi-supervised congealing [2, 1] is defined as a process of propagating the labeling, which is the facial contour in this work, across the entire ensemble from a few labeled examples (See Fig. 1).

There are many applications of semi-supervised congealing. In computer vision, landmark labeling is necessary for learning models of the object shape, such as Active Appearance Models (AAM) [3, 4] and Boosted Appearance Model [5] for faces, which is often conducted manually for a large set of object instances/images. However, this is a labor-intensive, time-consuming, and error-prone process. Our semi-supervised approach will dramatically alleviate this problem. Furthermore,

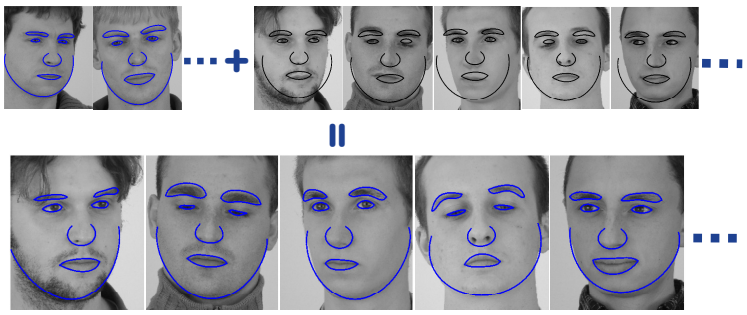

Fig. 1. Given an image ensemble with an overlayed initial contour via face detection (top right), together with manual labels of contours on a few images (top left), our proposed algorithm estimates the contour parameters for all images in the ensemble (bottom), regardless of the variations of facial shape, pose, and unseen subjects.

our approach can be used to discover the non-rigid shape deformation of a real-world object, when applied to an image ensemble of an object class.

Given the wide application space of semi-supervised congealing, there is a surprisedly limited amount of prior work concerning ensemble-based non-rigid shape estimation for objects with greatly varying appearance, such as faces. The work by Tong *et al.* [1] might be the most relevant one to ours. They use least-square-based congealing to estimate the set of landmarks for all images in an ensemble given the labels on a few images. The least square term between any image pair is evaluated on a common rectangle region, which is where the image pair warps toward based on the landmark location. By gradually reducing the size of rectangle, the precision of landmark estimation is improved.

Although [1] has shown some promise, the accuracy of the labeling has potential for further improvement. First of all, the coarse-to-fine scheme and measurement in the warped space poses fundamental limitation on the accuracy. Also, the intensity feature is not salient enough to capture edge information, which is where all landmarks reside. To alleviate these problems, we propose a novel approach in this paper. With the goal of providing high accuracy in labeling, we use a parametric curve to represent the facial contour, rather than a landmark set. Hence, the appearance feature along the curve, named *curve descriptor*, can be extracted and drives the congealing process. Since two curve functions are used to represent a closed contour such as the eye, we present a dual-congealing algorithm operating jointly on both curves, with the help of a geometric constraint term. We demonstrate that advanced features such as HOG [6] can be utilized in the proposed congealing framework. With extensive experiments, we show that large progress has been achieved in the labeling accuracy compared to the state-of-the-art approach.

2 Prior Work

There is a long history of unsupervised group-wise registration in computer vision [7], particularly in the area of medical image analysis. Learned-Miller [2, 8] names this process "congealing", where the basic idea is to minimize a cost

function by estimating the warping parameters of an ensemble. The work by Cox *et al.* [9] is a recent advance in least-squares-based congealing (LSC) algorithm. However, these approaches estimate only affine warping parameters for each image, rather than the non-rigid deformation addressed here.

There is also work on unsupervised image alignment that allows more general deformation models, such as [10–18]. However, almost all approaches report results on images with small intra-class appearance variation, such as brain image, digits, and faces of a small population. In contrast, the semi-supervised congealing algorithm of [1] demonstrates promising performance on an ensemble of over 1000 images from hundreds of subjects, which motivates us to use the semi-supervised approach for facial contours.

There is a rich literature concerning contour and edge detection [19]. We should note that in dealing with real-world images, the dominant edge from low-level image observations might not be consistent with the high-level semantic-meaningful contour. For example, the double eyelid can have stronger edge information compared to the inner boundary between the conjunctiva and the eyelid, which is often what we are interested in extracting for describing the shape of eyes. Thus, semi-supervision seems to be a natural way to allow the human expert to label the to-be-detected contours on a few examples, so as to convey the *true* contour that is of real interests for the application at hand.

3 Semi-supervised Least-Squares Congealing

First we will describe the basic concept and objective function of the conventional semi-supervised least-square congealing (SLSC) by using image warping [1].

Congealing approaches operate on an ensemble of K unlabeled images $\mathbf{I} = \{\mathbf{I}_i\}_{i\in[1,K]}$, each with an unknown parameter $\mathbf{p}_i$, such as the landmark set in [1], that is to be estimated. Semi-supervised congealing also assumes there is a small set of $\tilde{K}$ labeled images $\tilde{\mathbf{I}} = \{\tilde{\mathbf{I}}_n\}_{n\in[1,\tilde{K}]}$, each with a known parameter $\tilde{\mathbf{p}}_n$. We denote the collection of all unknown parameters with $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_K]$. The goal of SLSC is to estimate $\mathbf{P}$ by minimizing a cost function defined on the entire ensemble:

$$\varepsilon(\mathbf{P}) = \sum_{i=1}^{K} \varepsilon_i(\mathbf{p}_i). \tag{1}$$

The total cost is the summation of the cost of each unlabeled image $\varepsilon_i(\mathbf{p}_i)$:

$$\varepsilon_i(\mathbf{p}_i) = \frac{1-\alpha}{K-1} \sum_{j=1, j\neq i}^{K} \|f(\mathbf{I}_j, \mathbf{p}_j) - f(\mathbf{I}_i, \mathbf{p}_i)\|^2 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} \|f(\tilde{\mathbf{I}}_n, \tilde{\mathbf{p}}_n) - f(\mathbf{I}_i, \mathbf{p}_i)\|^2, \tag{2}$$

where $f(\mathbf{I}, \mathbf{p})$ is the feature representation of image $\mathbf{I}$ evaluated at $\mathbf{p}$. Hence, $\varepsilon_i(\mathbf{p}_i)$ equals the summation of the pairwise feature difference between $\mathbf{I}_i$ and all the other images in the ensemble, including both the unlabeled images (the 1^{st} term of Eqn. (2)) and the labeled image (the 2^{nd} term of Eqn. (2)).

In [1], the feature representation is defined as,

$$f(\mathbf{I}, \mathbf{p}) \doteq \mathbf{I}(\mathbf{W}(\mathbf{x}; \mathbf{p})), \tag{3}$$

where $\mathbf{W}(\mathbf{x};\mathbf{p})$ is a warping function that takes $\mathbf{x}$, which is a collection of pixel coordinates within the common rectangle region, as input, and outputs the corresponding pixel coordinates in the coordinate space of image $\mathbf{I}$. Given this warping function, $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}))$ denotes the corresponding warped image vector obtained by bilinear interpolation of the image $\mathbf{I}$ using the warped coordinates $\mathbf{W}(\mathbf{x};\mathbf{p})$. Note that in [1], $\mathbf{W}(\mathbf{x};\mathbf{p})$ is a simple 6-parameter affine warp, rather than a complex non-rigid warp such as the piecewise affine warp [4]. This is due to the high dimensionality in the non-rigid warp, as well as the needs to optimize $\mathbf{p}$ for all images simultaneously. Hence, by applying affine-warp-based optimization multiple times, each at a different rectangle region with decreasing size, the non-rigid natural of the warp can be approximated.

Since the total cost $\varepsilon(\mathbf{P})$ is difficult to optimize directly, [1] chooses to iteratively minimize the individual cost $\varepsilon_i(\mathbf{p}_i)$ for each $\mathbf{I}_i$. The well-known inverse warping technique [20] is utilized and after taking the first order Taylor expansion, Eqn. (2) can be simplified to:

$$\frac{1-\alpha}{K-1}\sum_{j=1,j\neq i}^{K}\|\mathbf{b}_j+\mathbf{c}_j\Delta\mathbf{p}_i\|^2+\frac{\alpha}{\tilde{K}}\sum_{n=1}^{\tilde{K}}\|\tilde{\mathbf{b}}_n+\tilde{\mathbf{c}}_n\Delta\mathbf{p}_i\|^2, \tag{4}$$

where

$$\mathbf{b}_j=f(\mathbf{I}_j,\mathbf{p}_j)-f(\mathbf{I}_i,\mathbf{p}_i),\ \mathbf{c}_j=\frac{\partial f(\mathbf{I}_j,\mathbf{p}_j)}{\partial\mathbf{p}_j}. \tag{5}$$

The least-square solution of Eqn. (4) can be obtained by setting the partial derivative of Eqn. (4) with respect to $\Delta\mathbf{p}_i$ to be equal to zero. We have:

$$\Delta\mathbf{p}_i=-\left[\frac{1-\alpha}{K-1}\sum_{j=1,j\neq i}^{K}\mathbf{c}_j^T\mathbf{c}_j+\frac{\alpha}{\tilde{K}}\sum_{n=1}^{\tilde{K}}\tilde{\mathbf{c}}_n^T\tilde{\mathbf{c}}_n\right]^{-1}\left[\frac{1-\alpha}{K-1}\sum_{j=1,j\neq i}^{K}\mathbf{c}_j^T\mathbf{b}_j+\frac{\alpha}{\tilde{K}}\sum_{n=1}^{\tilde{K}}\tilde{\mathbf{c}}_n^T\tilde{\mathbf{b}}_n\right]. \tag{6}$$

4 Facial Contour Congealing

In this section, we will present our facial contour congealing approach in detail. Three key technical components will be covered: parametric curve representation, curve descriptor, and contour congealing.

4.1 Parametric Curve Representation

In computer vision, it has been very popular to use a set of landmarks to describe the shape of an object by placing the landmarks along the object contour, such as the Point Distribution Model (PDM) applied to faces. However, there are disadvantages to using the landmark representation. First, an excessive number of landmarks are needed in order to approximate the true contour of facial images, especially for high quality images. Second, for the semi-supervised congealing application, little constraint can be applied on the distribution of landmarks since there are very few labeled images, which poses a challenge for landmark estimation on unlabeled images.

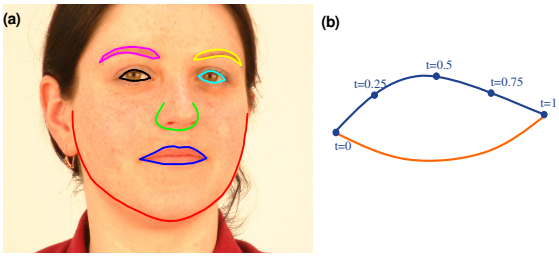

Fig. 2. (a) The entire facial shape is described by contours on 7 facial components; (b) The closed contour (such as the eye) is represented by two connected parametric curves, where each curve's parameter can be estimated via curve fitting on labeled landmarks (5 landmarks in this case).

In this paper, we propose to use a parametric curve representation to describe the facial contour. As shown in Fig. 2(b), we use two parametric curves to represent the closed contour of one of the seven facial components, such as eye. For simplicity of notation, we will initially focus on one of the two curves, which covers half of the contour. A 2D parametric curve is defined by the n-order polynomials:

$$x(t) = p_{x,n}t^n + p_{x,n-1}t^{n-1} \cdots + p_{x,1}t + p_{x,0}, \tag{7}$$

$$y(t) = p_{y,n}t^n + p_{y,n-1}t^{n-1} \cdots + p_{y,1}t + p_{y,0}, \tag{8}$$

where usually we consider $t \in [0, 1]$, and the collection of coefficients,

$$\mathbf{p} = [\,\mathbf{p}_x\ \mathbf{p}_y\,]^T = \left[\,p_{x,n}\ p_{x,n-1} \cdots\ p_{x,1}\ p_{x,0}\ p_{y,n}\ p_{y,n-1} \cdots\ p_{y,1}\ p_{y,0}\,\right]^T, \tag{9}$$

is called the *curve parameter*, which fully describes the shape of the curve. Given a known $\mathbf{p}$, we can generate any number of points on the curve by varying t.

In practice, when we manually label face images, we label landmarks rather than the curve directly. Suppose there are m landmarks being manually labeled along the contour, we have:

$$\mathbf{x} = [\,x(t_1)\ y(t_1) \cdots\ x(t_m)\ y(t_m)\,]^T = \mathbf{T}\mathbf{p}, \tag{10}$$

where

$$\mathbf{T} = \begin{bmatrix} t_1^n & t_1^{n-1} & \cdots & t_1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & t_1^n & t_1^{n-1} & \cdots & t_1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ t_m^n & t_m^{n-1} & \cdots & t_m & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & t_m^n & t_m^{n-1} & \cdots & t_m & 1 \end{bmatrix}. \tag{11}$$

By assuming the landmarks are evenly spaced, we have $[t_1, t_2, \cdots, t_m] = [0, \frac{1}{m-1}, \cdots, 1]$. Hence, the curve parameter can be directly computed from the landmark set:

$$\mathbf{p} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{x}. \tag{12}$$

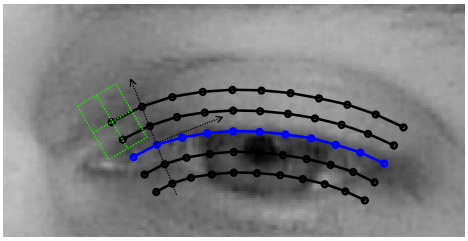

Fig. 3. An array of point coordinates are computed within the band following the target curve. Appearance information, such as pixel intensity or HOG, can be extracted from these coordinates and form a curve descriptor for the target curve.

4.2 Curve Descriptor

Having introduced the mapping between the curve parameter and the landmark set, we will present our method to extract the appearance feature for a curve, which is called *curve descriptor*, given the known curve parameter.

For landmark-based shape representation, e.g. Active Shape Model (ASM) [21], the appearance information is often extracted within a small rectangle region around the landmark. Similarly, for curve-based representation, the curve descriptor will be extracted from a *band*-like region along the curve.

As shown in Fig. 3, let us denote the U points along the central target curve as $\{[\breve{x}_{u,0}, \breve{y}_{u,0}]\}_{u=1,\cdots,U}$, where $[\breve{x}_{u,0}, \breve{y}_{u,0}] = [x(t_u), y(t_u)]$. We can allocate V synthetic curves on both sides of the target curve, where the distance between any neighboring curves is r. Specifically, for the u^{th} point on the curve, we have a point $[\breve{x}_{u,v}, \breve{y}_{u,v}]$ on its normal direction with a distance $|v|r$, which is then located on the v^{th} synthetic curve,

$$\begin{bmatrix} \breve{x}_{u,v} \\ \breve{y}_{u,v} \end{bmatrix} = \begin{bmatrix} x(t_u) - vrsin\theta_u \\ y(t_u) + vrcos\theta_u \end{bmatrix}, \tag{13}$$

where θ_u is the tangent angle for the u^{th} point on the curve:

$$\theta_u = arctan(\frac{dy}{dx}|_{t_u}) = arctan(\frac{\mathbf{T}'_u \mathbf{p}_y}{\mathbf{T}'_u \mathbf{p}_x}), \tag{14}$$

and $\mathbf{T}'_u$ is the derivative of polynomial evaluated at t_u:

$$\mathbf{T}'_u = \begin{bmatrix} nt_u^{n-1} & (n-1)t_u^{n-2} & \cdots & 1 & 0 \end{bmatrix}. \tag{15}$$

Hence, with a set of point coordinates $\breve{\mathbf{x}} = \{[\breve{x}_{u,v}, \breve{y}_{u,v}]\}_{u=1,\cdots,U, v=-V,\cdots,V}$, as well as their corresponding angles $\theta = \{\theta_u\}_{u=1,\cdots,U}$, we can extract the curve descriptor. The simplest descriptor is to use the pixel intensity evaluated at $\breve{\mathbf{x}}$, i.e., $f(\mathbf{I}, \mathbf{p}) \doteq \mathbf{I}(\breve{\mathbf{x}})$. Motivated by the work of [6, 22, 23], we can also use the powerful HOG feature as the curve descriptor:

$$f(\mathbf{I}, \mathbf{p}) \doteq \mathbf{h}(\breve{\mathbf{x}}, \theta) = [\hat{\mathbf{h}}(\breve{x}_{u,v}, \breve{y}_{u,v}, \theta_u)]_{u=1,\cdots,U, v=-V,\cdots,V}, \tag{16}$$

which is a concatenation of $U(2V+1)$ L^2-normalized HOG vectors, each centered at $[\breve{x}_{u,v}, \breve{y}_{u,v}]$ with angle θ_u. Note that the HOG feature we employ makes use

of the tangent angle θ. Hence it will better capture the appearance information along the curve, as well as on either side of the curve.

4.3 Contour Congealing

With the presentation on contour representation and curve descriptor, we now introduce how to conduct contour congealing for an ensemble of facial images. The basic problem setup is the same as the SLSC in Section 3. That is, given the unlabeled image set $\{\mathbf{I}_i\}$ and its initial label $\{\mathbf{p}'_i\}$, as well as a small number of labeled images $\{\tilde{\mathbf{I}}_n\}$ and their known labels $\{\tilde{\mathbf{p}}_n\}$, we need to estimate the true curve parameters $\{\mathbf{p}_i\}$.

In this work, our semi-supervised contour congealing is applied on each of the seven components independently. Notice that 5 out of the 7 components are closed contours, where two curve functions are needed to represent the entire contour. In contrast to the SLSC in Section 3, now we face a new challenging problem of congealing two sets of curve parameters simultaneously, where simply applying Eqn. 2 is not sufficient.

By denoting $\mathbf{p}^1$ and $\mathbf{p}^2$ as the curve parameters for the top curve and bottom curve respectively, we can utilize one simple geometric constraint. That is, the points on both ends of the 1^{st} curve should overlap with those of the 2^{nd} curve. With that, our semi-supervised congealing for a closed contour utilizes the following objective function:

$$\varepsilon_i(\mathbf{p}_i^1, \mathbf{p}_i^2) = \varepsilon_i(\mathbf{p}_i^1) + \varepsilon_i(\mathbf{p}_i^2) + \beta\|\mathbf{x}_i^1 - \mathbf{x}_i^2\|^2, \tag{17}$$

where $\mathbf{x}_i^1 = \mathbf{T}^{01}\mathbf{p}_i^1$, $\mathbf{x}_i^2 = \mathbf{T}^{01}\mathbf{p}_i^2$, and $\mathbf{T}^{01}$ is a sub-matrix of $\mathbf{T}$ including its first two rows and last two rows. This objective function is basically the summation of the error terms from two curves, and their geometric constraint weighted by β.

By employing inverse warping technique [20] and similar simplification as Eqn. 4, we have:

$$\varepsilon_i(\Delta\mathbf{p}_i^1, \Delta\mathbf{p}_i^2) = \frac{1-\alpha}{K-1}\sum_{j=1,j\neq i}^{K}(\|\mathbf{b}_j^1 + \mathbf{c}_j^1\Delta\mathbf{p}_i^1\|^2 + \|\mathbf{b}_j^2 + \mathbf{c}_j^2\Delta\mathbf{p}_i^2\|^2) +$$
$$\frac{\alpha}{\tilde{K}}\sum_{n=1}^{\tilde{K}}(\|\tilde{\mathbf{b}}_n^1 + \tilde{\mathbf{c}}_n^1\Delta\mathbf{p}_i^1\|^2 + \|\tilde{\mathbf{b}}_n^2 + \tilde{\mathbf{c}}_n^2\Delta\mathbf{p}_i^2\|^2) + \beta\|\mathbf{x}_i^1 - \mathbf{x}_i^2 - \mathbf{e}_i(\Delta\mathbf{p}_i^1 - \Delta\mathbf{p}_i^2)\|^2, \tag{18}$$

where $\mathbf{e}_i = \frac{\partial\mathbf{x}_i^1}{\partial\mathbf{p}_i^1} = \frac{\partial\mathbf{x}_i^2}{\partial\mathbf{p}_i^2} = \mathbf{T}^{01}$, and $\mathbf{b}_j^*$ and $\mathbf{c}_j^*$ can be defined similarly as Eqn. 5.

The curve parameter updates $\Delta\mathbf{p}_i^1$ and $\Delta\mathbf{p}_i^2$ can be estimated by solving a linear equation system as:

$$\begin{cases} \frac{\partial\varepsilon_i(\Delta\mathbf{p}_i^1, \Delta\mathbf{p}_i^2)}{\partial\Delta\mathbf{p}_i^1} = 0, \\ \frac{\partial\varepsilon_i(\Delta\mathbf{p}_i^1, \Delta\mathbf{p}_i^2)}{\partial\Delta\mathbf{p}_i^2} = 0. \end{cases} \tag{19}$$

Substituting Eqn. (18) to Eqn. (19), we have:

$$\begin{bmatrix} \mathbf{A}_1, & \mathbf{B} \\ \mathbf{B}, & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{p}_i^1 \\ \Delta\mathbf{p}_i^2 \end{bmatrix} = - \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}, \tag{20}$$

where

$$\mathbf{A}_1 = \frac{1-\alpha}{K-1} \sum_{j=1, j\neq i}^{K} (\mathbf{c}_j^1)^T \mathbf{c}_j^1 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} (\tilde{\mathbf{c}}_n^1)^T \tilde{\mathbf{c}}_n^1 + \beta \mathbf{e}_i^T \mathbf{e}_i, \tag{21}$$

$$\mathbf{A}_2 = \frac{1-\alpha}{K-1} \sum_{j=1, j\neq i}^{K} (\mathbf{c}_j^2)^T \mathbf{c}_j^2 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} (\tilde{\mathbf{c}}_n^2)^T \tilde{\mathbf{c}}_n^2 + \beta \mathbf{e}_i^T \mathbf{e}_i, \tag{22}$$

$$\mathbf{B} = -\beta \mathbf{e}_i^T \mathbf{e}_i, \tag{23}$$

$$\mathbf{C}_1 = \frac{1-\alpha}{K-1} \sum_{j=1, j\neq i}^{K} (\mathbf{c}_j^1)^T \mathbf{b}_j^1 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} (\tilde{\mathbf{c}}_n^1)^T \tilde{\mathbf{b}}_n^1 - \beta \mathbf{e}_i^T (\mathbf{d}_i^1 - \mathbf{d}_i^2), \tag{24}$$

$$\mathbf{C}_2 = \frac{1-\alpha}{K-1} \sum_{j=1, j\neq i}^{K} (\mathbf{c}_j^2)^T \mathbf{b}_j^2 + \frac{\alpha}{\tilde{K}} \sum_{n=1}^{\tilde{K}} (\tilde{\mathbf{c}}_n^2)^T \tilde{\mathbf{b}}_n^2 + \beta \mathbf{e}_i^T (\mathbf{d}_i^1 - \mathbf{d}_i^2). \tag{25}$$

The above solution is straightforward to implement as long as we know how to compute $\mathbf{b}_j^*$ and $\mathbf{c}_j^*$, among which $\frac{\partial f(\mathbf{I},\mathbf{p})}{\partial \mathbf{p}}$ will likely take the most effort to compute. Hence, from now on we will focus on the computation of $\frac{\partial f(\mathbf{I},\mathbf{p})}{\partial \mathbf{p}}$ when the curve descriptor is the HOG feature. For the case of the intensity feature, $\frac{\partial f(\mathbf{I},\mathbf{p})}{\partial \mathbf{p}}$ is relatively easier and will be omitted from this discussion.

Note that our HOG feature is a L^2-normalized version, $\hat{\mathbf{h}} = \frac{\mathbf{h}}{\|\mathbf{h}\|_2}$, due to the proven superior performance over the non-normalized version [6]. Hence,

$$\frac{\partial \hat{\mathbf{h}}}{\partial \mathbf{p}} = \frac{\partial \hat{\mathbf{h}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{p}} \tag{26}$$

$$= \left(\frac{\mathbf{I}^{32}}{\|\mathbf{h}\|_2} - \frac{1}{(\|\mathbf{h}\|_2)^{3/2}} \mathbf{h}\mathbf{h}^T\right)\left(\frac{\partial \mathbf{h}}{\partial \breve{x}_{u,v}} \frac{\partial \breve{x}_{u,v}}{\partial \mathbf{p}} + \frac{\partial \mathbf{h}}{\partial \breve{y}_{u,v}} \frac{\partial \breve{y}_{u,v}}{\partial \mathbf{p}} + \frac{\partial \mathbf{h}}{\partial \theta_u} \frac{\partial \theta_u}{\partial \mathbf{p}}\right), \tag{27}$$

where $\mathbf{I}^{32}$ is a 32×32 identity matrix,

$$\frac{\partial \breve{x}_{u,v}}{\partial \mathbf{p}} = \frac{\partial \breve{x}_{u,0}}{\partial \mathbf{p}} - vrcos\theta_u \frac{\partial \theta_u}{\partial \mathbf{p}}, \tag{28}$$

and

$$\frac{\partial \theta_u}{\partial \mathbf{p}} = \frac{1}{1 + (tan\theta_u)^2} \frac{\partial \frac{\mathbf{T}'_u \mathbf{p}_y}{\mathbf{T}'_u \mathbf{p}_x}}{\partial \mathbf{p}} \tag{29}$$

$$= \frac{1}{1 + (tan\theta_u)^2} \left[-\frac{\mathbf{T}'_u \mathbf{p}_y}{(\mathbf{T}'_u \mathbf{p}_x)^2} \mathbf{T}'_u \; 0 \; \frac{1}{\mathbf{T}'_u \mathbf{p}_x} \mathbf{T}'_u \; 0 \right]. \tag{30}$$

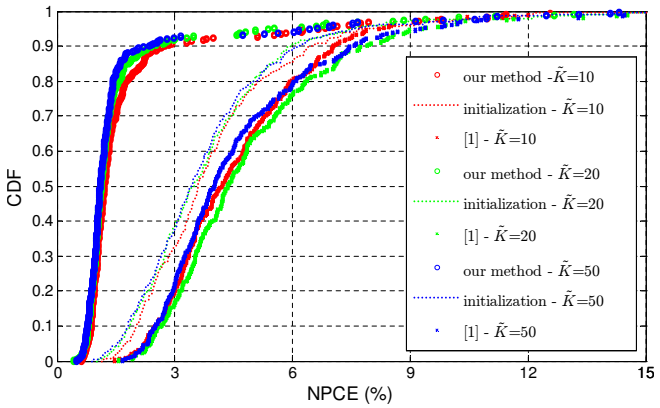

Fig. 4. Performances of the contours on two eyes using our algorithm, the baseline and initialization, when the number of labeled images $\tilde{K}$ is 10, 20, and 50.

The partial derivatives $\frac{\partial \mathbf{h}}{\partial \breve{x}_{u,v}}$, $\frac{\partial \mathbf{h}}{\partial \breve{y}_{u,v}}$, and $\frac{\partial \mathbf{h}}{\partial \theta_u}$ can be computed using the definition of derivative in the discrete case, i.e., $\frac{\partial \mathbf{h}}{\partial \breve{x}_{u,v}} = \mathbf{h}(\breve{x}_{u,v}, \breve{y}_{u,v}, \theta_u) - \mathbf{h}(\breve{x}_{u,v} - 1, \breve{y}_{u,v}, \theta_u)$. Similar ways of computing $\frac{\partial \mathbf{h}}{\partial x}$ and $\frac{\partial \mathbf{h}}{\partial y}$ have been used in [22].

For the case of open facial contour, such as nose and chin, we use the first term of Eqn. 17 as the objective function. Its solution is a simplified case of the above derivation, and hence will be omitted here.

5 Experimental Results

In this section, we will present the extensive experiments that demonstrate the capability of our proposed algorithm. For our experiments, we choose a subset of 350 images from the publicly-available PUT face database [24], which exhibits moderate variation in pose and facial expression (Fig. 1). The entire image set is partitioned into two sets: one with 300 images is used as the unlabeled image ensemble $\mathbf{I}$, the other with 50 images will be randomly chosen as the labeled image set $\tilde{\mathbf{I}}$. All images have manually labeled ground-truth on the facial contours of 7 components (Fig. 2). For example, the contour of an eye is labeled with 20 landmarks. There are 166 total landmarks labeled for all 7 contours. This ground-truth will not only provide the known curve parameter $\tilde{\mathbf{p}}_n$ for labeled image $\tilde{\mathbf{I}}_n$ (via Eqn. 12), but also be used in quantitative evaluation of the performance.

Since the very recent work of Tong *et al.* [1] is the most relevant to ours, it is chosen as the baseline approach for comparison. We have implemented both algorithms in Matlab and ensure they are tested under the same condition. Although PUT is a high quality face database, we downsample the face size to be around 70 pixels eye-to-eye, mostly based on the concern that the efficiency of the baseline algorithm largely depends on the average face size. For the labeled set, both algorithms have their $\tilde{\mathbf{p}}$ parameters computed from the manually labeled 166 landmarks per image. For the unlabeled set, both algorithms use the PittPatt face detector [25] to compute the initial $\mathbf{p}$, by placing an average set of landmarks/contours (see the top-right of Fig. 1), which is computed from the

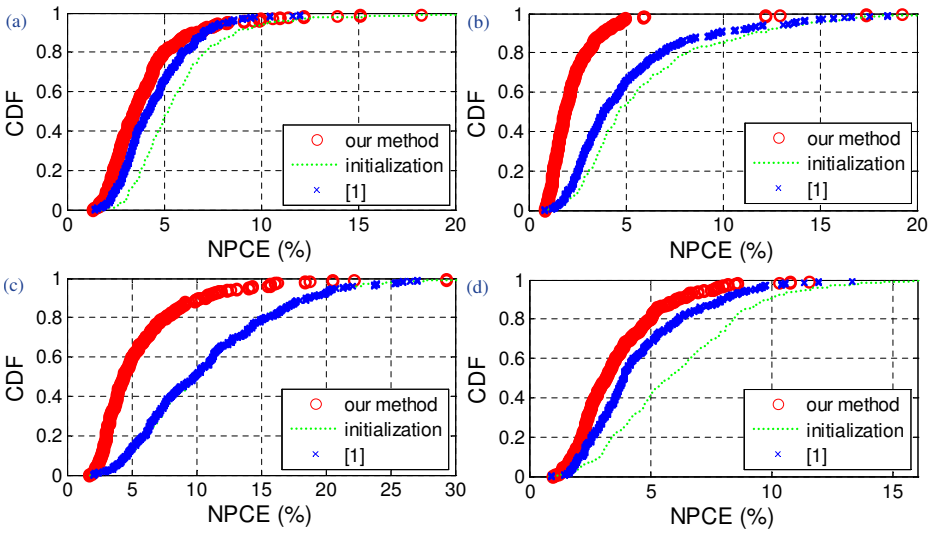

Fig. 5. Comparison of our algorithm, the baseline and initialization ($\tilde{K} = 10$) for (a) two eyebrows, (b) mouth, (c) chin, (d) nose.

small set of labeled images, within the detected face rectangle. This is a valid initialization since face detection is almost a commodity.

For our algorithm, once the estimation of curve parameters is completed, we compute the average of the distances between each ground-truth landmark and the estimated curve, and then divide it by the distance between the two eye centers. This quantitative evaluation is called *Normalized Point to Curve Error* (NPCE), and is expressed as a percentage. We also compute NPCE for the baseline because a curve function can be fitted to the estimated landmarks via the baseline algorithm.

5.1 Performance Comparison

We will first present the performance comparison between our algorithm and the baseline approach. For each unlabeled image in our algorithm, once the average contour is placed within the face detection rectangle, we have the initial curve parameters for all seven components. Then the contour congealing is conducted on each component independently. Note that we only use the very basic face detection functionality and no additional landmark detection, such as eye and nose, is conducted using the PittPatt SDK. Hence, it is obvious that face detection can have quite a large variation on localization, especially for smaller components. However, our algorithm does surprisingly well in handling this real-world challenging initialization and congealing all components independently. Of course, one potential future work is to use the better-congealed components, and global spatial distribution of components learned from labeled data, to produce a better initialization for other components.

For both algorithms, we perform three sets of experiments, each with a different number of labeled images, $\tilde{K}$=10, 20, and 50. For all components in our

algorithm, we use 4_{th}-order polynomials ($n=4$) in the curve function, and the 2×2 cell 8-bin HOG feature for the curve descriptor, where $V \in [3,5]$ and $r \in [1,2]$ for various components. We fix $\alpha = 0.5$ and $\beta = 50$ throughout the experiments.

To better understand the system performance, we plot the comparison for two eye components in Fig. 4. The cumulative distribution function (CDF) of NPCE is plotted for the results of our algorithm, the baseline, and the initialization via face detection. It is clear that our algorithm improves the initialization with a large margin, while the baseline performs slightly worse than the initialization. We attribute this worse performance of the baseline to the pose variation in the data, which makes the image warping and progressive affine approximation less likely to work well. Note that for our algorithm, more than 90% of the unlabeled images have the final eye localization error less than 2% of eye-to-eye distance. For the right eye, it takes our algorithm $13-15$ iterations (about 3.5 hours on the conventional PC) to converge for the entire test set when $\tilde{K}$ is 10 or 20.

In comparing the results with various $\tilde{K}$, we can see that our approach at $\tilde{K}=10$ is almost as good as when $\tilde{K}=50$. This is a very important property since it means our approach can be used with a very small set of labeled images. The similar property is also observed in the comparison of other components. Hence, due to limited space, we show the results of other components only when $\tilde{K}=10$ in Fig. 5. Again, for all the remaining components, our algorithm performs substantially better than the baseline, which also improves over the initialization except the chin contour. We attribute the improvement of our algorithm to three reasons: 1) the partition scheme and using a set of affine warps to approximate non-rigid deformation of [1] pose limitation on the accuracy; 2) the feature extracted along the curve better describes the appearance information than the feature in the partitioned rectangle of [1]; 3) the HOG feature is more suitable for localization than the intensity feature in [1]. We also illustrate the congealing results of our approach on various components in Fig. 6.

5.2 Labeling Confidence

Knowing when an algorithm does not converge is often as important as overall algorithm performance. This is especially true for semi-supervised algorithms. Hence, a confidence score is desirable for practical applications in order to evaluate the quality of labeling without ground truth. For this we use $\varepsilon_i(\mathbf{p}_i^1, \mathbf{p}_i^2)$ in Eqn. (17). A smaller ε_i indicates a higher-confidence in labeling. By partitioning the 300 confidence scores into 5 bins, Fig. 7 shows labeled left eye component from the lowest 20% to the highest 20% confidence scores, in our 300-image ensemble ($\tilde{K}=10$). Fig. 8 also illustrates the distribution of the estimated ε_i versus the actual labeling error represented by the NPCE for the left eye component. With the increase of the ε_i, the landmark labeling error increases significantly. Hence, it is clear that this confidence score is indicative of labeling performance. The linear correlation between ε_i and NPCE can also be shown by the computed Pearson correlation coefficient, which is 0.715. Similar phenomena have been observed for experiments on other facial components. In practice, after labeling, one can use this confidence score to select only high-confident samples for a

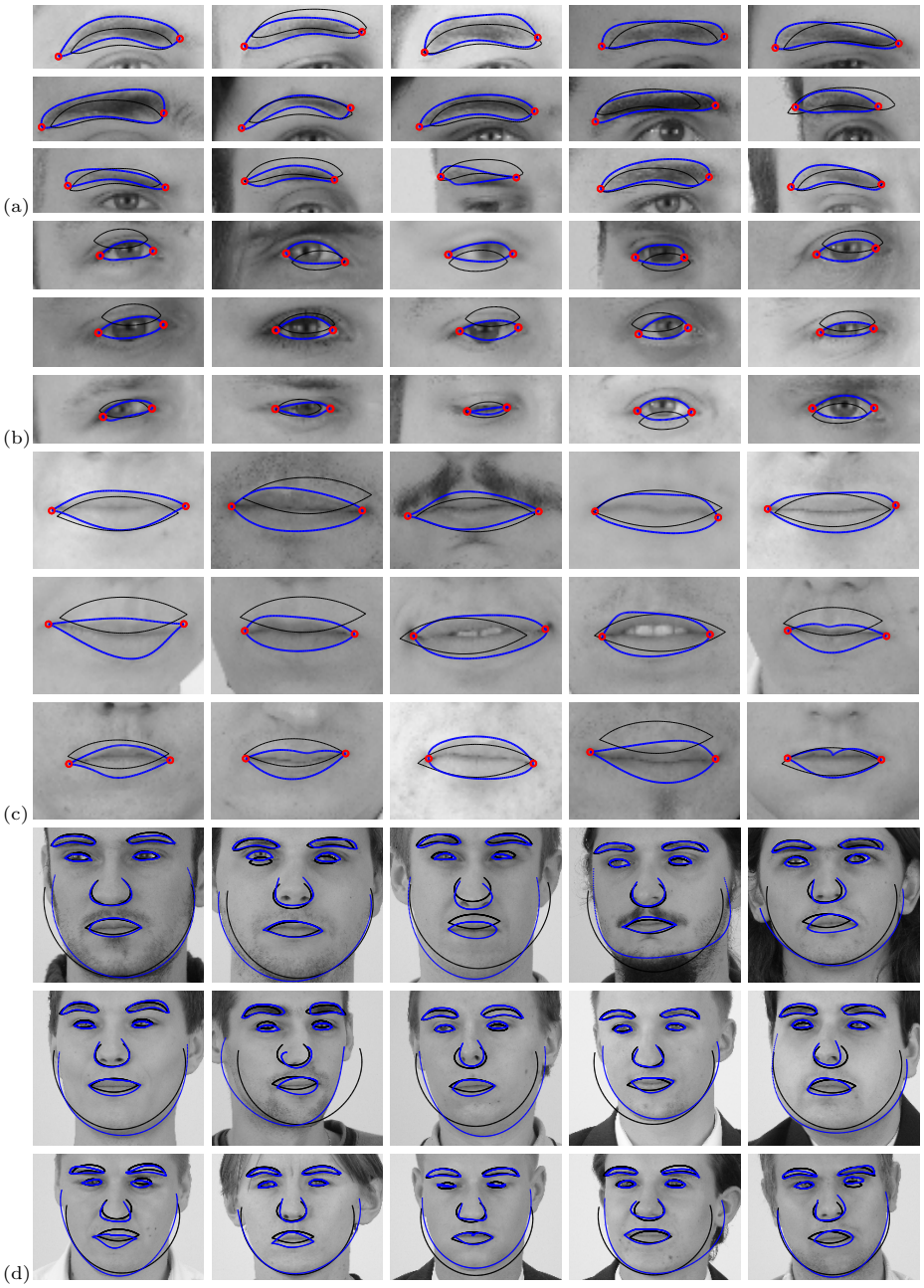

Fig. 6. The initialization and results of our algorithm on various facial components: (a) left eyebrow ($\tilde{K}=20$), (b) left eye ($\tilde{K}=50$), (c) mouth ($\tilde{K}=10$), and (d) whole face ($\tilde{K}=10$). It can be seen that some of the images, especially those with background displayed, are of faces with noticeable pose variation. Notice the large amount of shape variation exhibited in the data that can be handled by our algorithm.

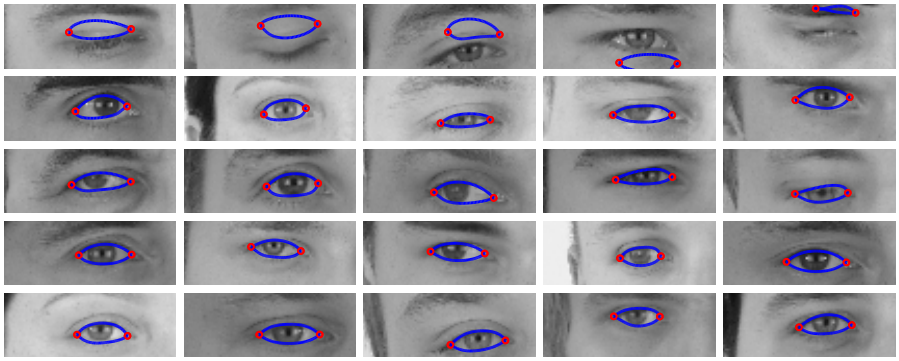

Fig. 7. The confidence of labeling the eye component increases from the top row to bottom row. We observe that almost all failed cases can be found in the category with lowest confidence score.

training set, or to select low-confident samples for other appropriate additional processing.

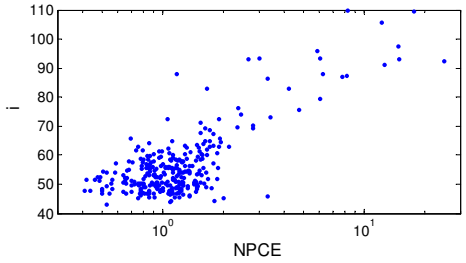

Fig. 8. The correlation between the labeling confidence (ε_i) and actual labeling error (NPCE). The Pearson correlation coefficient between these two variables is 0.715.

6 Conclusions

Real-world objects can exhibit a large amount of shape deformation on 2D images due to intra-object variability, object motion, and camera viewpoint. Rather than the conventional landmark-based representation, we propose to use curve functions to describe the facial contour. We demonstrate a complete system that is able to simultaneously align facial contour for a large set of unlabeled images with face detection results, given a few labeled images. Extensive experiments demonstrate that our system has achieved much more accurate labeling results compared to the previous state-of-the-art approach on face images with moderate changes in pose and expression.

References

1. Tong, Y., Liu, X., Wheeler, F.W., Tu, P.: Automatic facial landmark labeling with minimal supervision. In: CVPR. (2009)

2. Miller, E., Matsakis, N., Viola, P.: Learning from one example through shared densities on transforms. In: CVPR. Volume 1. (2000) 464–471
3. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE T-PAMI **23** (2001) 681–685
4. Matthews, I., Baker, S.: Active appearance models revisited. IJCV **60** (2004) 135–164
5. Liu, X.: Discriminative face alignment. IEEE T-PAMI **31** (2009) 1941–1954
6. Dalal, N., Triggs, W.: Histograms of oriented gradients for human detection. In: CVPR. Volume 1. (2005) 886–893
7. Vetter, T., Jones, M.J., Poggio, T.: A bootstrapping algorithm for learning linear models of object classes. In: CVPR. (1997) 40–46
8. Learned-Miller, E.: Data driven image models through continuous joint alignment. IEEE T-PAMI **28** (2006) 236–250
9. Cox, M., Sridharan, S., Lucey, S., Cohn, J.: Least squares congealing for unsupervised alignment of images. In: CVPR. (2008)
10. Balci, S., Golland, P., Shenton, M., Wells, W.: Free-form B-spline deformation model for groupwise registration. In: MICCAI. (2007) 23–30
11. Baker, S., Matthews, I., Schneider, J.: Automatic construction of active appearance models as an image coding problem. IEEE T-PAMI **26** (2004) 1380–1384
12. Kokkinos, I., Yuille, A.: Unsupervised learning of object deformation models. In: ICCV. (2007)
13. Cootes, T., Twining, C., Petrovic, V., Schestowitz, R., Taylor, C.: Groupwise construction of appearance models using piece-wise affine deformations. In: BMVC. Volume 2. (2005) 879–888
14. Cristinacce, D., Cootes, T.: Facial motion analysis using clustered shortest path tree registration. In: Proc. of the 1st Int. Workshop on Machine Learning for Vision-based Motion Analysis with ECCV. (2008)
15. Torre, F., Nguyen, M.: Parameterized kernel principal component analysis: Theory and applications to supervised and unsupervised image alignment. In: CVPR. (2008)
16. Langs, G., Donner, R., Peloschek, P., Horst, B.: Robust autonomous model learning from 2D and 3D data sets. In: MICCAI. Volume 1. (2007) 968–976
17. Saragih, J., Goecke, R.: A nonlinear discriminative approach to AAM fitting. In: ICCV. (2007)
18. Sidorov, K., Richmond, S., Marshall, D.: An efficient stochastic approach to groupwise non-rigid image registration. In: CVPR. (2009)
19. Meltzer, J., Soatto, S.: Edge descriptors for robust wide-baseline correspondence. In: CVPR. (2008)
20. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. IJCV **56** (2004) 221–255
21. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models — their training and application. CVIU **61** (1995) 38–59
22. Liu, X., Yu, T., Sebastian, T., Tu, P.: Boosted deformable model for human body alignment. In: CVPR. (2008)
23. Liu, X., Tong, Y., Wheeler, F.W.: Simultaneous alignment and clustering for an image ensemble. In: ICCV. (2009)
24. Kasinski, A., Florek, A., Schmidt, A.: The PUT face database. Technical report, Poznan University of Technology, Poznan, Poland (2009)
25. Schneiderman, H.: Learning a restricted Bayesian network for object detection. In: CVPR. (2004)