Eötvös Loránd University
Faculty of Informatics
Department of Software Technology and Methodology

Semi-supervised Adaptive Facial Tracking Method

Author: Hy Truong Son
Major: Computer Science BSc.

Supervisor: Dr. András Lőrincz
Position: Senior researcher

Budapest, 2016

Contents

Semi-supervised Adaptive Facial Tracking Method

Hy Truong Son

Eötvös Loránd University

Abstract

Facial tracking problem plays an important role in action units detection and emotions classification. Previous approaches such as Active Appearance Model or Constrained Local Model have shown their limitation with unseen subjects and over-fitting problem. In this paper, we proposed the extension of Supervised Descent Method and the Online Pseudo-Inverse Update Method in order to adapt and personalize unseen data with the absence of labeled ground truth from human interaction. Our semi-supervised algorithm not only performed highly accurate results with fast adaptation process for real time application, but also opened up new research directions in the field of facial analysis.

Chapter I. Introduction

Mathematical non-linear approximation optimization methods become popular techniques in solving many computer vision problems including facial tracking, image alignment, camera calibration and interpreting medical images. Some recent methods in the field are getting better and giving promoting results.

Supervised Descent Method (SDM) in the context of facial alignment was first introduced by Xiong and De la Torre [1]. The purpose of this optimization algorithm is to overcome the limitation of the previous 2nd order descent methods, which require heavy computation of the Jacobian and Hessian. The idea can be summarized as learning a sequence of descent directions for minimizing the mean

of Non-linear Least Squares functions in general and the sum of Euclidean distances between Scale Invariant Feature Transform (SIFT) descriptors sampled at critical points of face in specific [1].

Extreme Learning Machine (ELM) was invented by Huang and colleagues [2][3]. Basically, ELM is an emerging learning technique to regress and classify large and complex data. ELM aims to generalize feed-forward networks including single and multi-hidden-layer neural networks by providing an efficient unified solution. Fundamental basis of the algorithm is the theory that proposes: instead of finding weights for hidden neurons in MLP model by classical back propagation algorithm, these weights can be randomly generated and as a consequence, the weight mapping from hidden layer to output layer can be learned from solving Linear Least Squares in a closed form [4].

The original incremental solution of the pseudo-inverse of a matrix was developed and published by Edward Greville in [5][6]. In order to simplify the work of Greville, André van Schaik and Jonathan Tapson presented the Online Pseudo-Inverse Update Method (OPIUM) such as OPIUM-Light and OPIUM-Dynamic methods in [4] with supervised manner for ELM. The target of these new algorithms is to construct an iterative equation for pseudo-inverse, and avoids batch recomputation when a new instance appears [4].

The biggest problem of current approaches for facial tracking in specific and image alignment in general is the limitation of database and the unavailability of both human interaction and ground truth for real time adaptation. From the motivation of solving this challenge and getting higher precision as well as the necessity of an adaptive method for small databases (for eyes and gazes), we suggest the solution: OPIUM-Light with fast and precise unsupervised adaptation for the original SDM method.

In short, our idea can be summarized as follows. In the supervised training process, we used SDM as sequential linear regressions: from the initial guess of landmark positions, we move the current landmarks forward the ground truth based on Histogram of Oriented Gradients [8] (HOG) descriptors extracted around these points; after few steps, we can reach into the close neighborhood of the ground truth landmarks. In the unsupervised testing process, when a new image comes, we apply linear regressors trained before to converge the initial landmark to the best possibly estimated landmark; with this new landmark, we make a new training sample and adjust the linear regressors by OPIUM-Light algorithm.

Our paper is organized with 6 chapters:

- **Chapter I.** *Introduction:* Overview of the general idea.
- **Chapter II.** *Methods:* The mathematics behinds SDM, ELM, OPIUM and the adaptive facial tracking method.
- **Chapter III.** *Experiments and Results:* Discussion about databases, experiment methods, results, visualization and examples.
- **Chapter IV.** *Future work and conclusion:* New ideas and study directions for future researches.
- **Appendix A, B:** Introduction of HOG and 2D Procrustes algorithm.
- **References:** Related previous researches in the field.

Chapter II. Methods

1. Supervised Descent Method

1.1. Newton's method

In one dimension case, to find a stationary point $x_* \in \Re$ of a single variable function $f \in \Re \to \Re$ such that $f'(x_*) = 0$, starting from an initial guess x_0, Newton's method constructs a sequence (x_n) converging toward x_*. The second order Taylor expansion $f_T(x)$ of the function $f(x)$ in a neighbor of x_n is:

$$f_T(x_n + \Delta x) \approx f(x_n) + f'(x_n)\Delta x + \frac{1}{2}f''(x_n)\Delta x^2$$

where $\Delta x = x - x_n$. To attains its extreme value, the following derivation with respect to Δx has to be equal to 0:

$$\frac{\partial[f_T(x_n + \Delta x) - f(x_n)]}{\partial(\Delta x)} = 0 \Leftrightarrow f'(x_n) + f''(x_n)\Delta x = 0$$

Assuming that $f(x)$ is a twice differentiable function having second order Taylor expansion as a good approximation and the initial guess x_0 is selected close enough to stationary point x_*. The sequence (x_n) is defined by:

$$\Delta x = x - x_n = -\frac{f'(x_n)}{f''(x_n)} \Leftrightarrow x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}, n = 0, 1, ...$$

will converge towards a root of the first order derivative $f'(x)$, for instance x_* such that $f'(x_*) = 0$.

For higher dimensions, the above iterative algorithm can be generalized to several dimensions by replacing the derivative with the Jacobian matrix $J_f(x)$ and the reciprocal of the second derivative with the inverse of the Hessian matrix $H_f(x)$:

$$x_{n+1} = x_n - [H_f(x_n)]^{-1} J_f(x_n), n \geq 0 \qquad (1)$$

where the initial estimate $x_0 \in \Re^{p\times 1}$, and $H_f(x_n) \in \Re^{p\times p}$ and $J_f(x_n) \in \Re^{p\times 1}$ are the Hessian and Jacobian matrix evaluated at x_n.

Supervised Descent Method can solve the following limitations of Newton's

method:

- If the Hessian is not positive definite, Newton's method cannot guarantee that each step is being moved in the correct descent direction.
- In the area of computer vision, Histogram of Oriented Gradients descriptor is a non-differentiable image operator while Newton's method needs a twice differentiable function.
- The computation of inverting the Hessian is expensive with the upper bound of complexity proportional to cubic polynomial of number of dimensions $O(p^3)$ operations and memory space is approximately $O(p^2)$ floating point numbers.

1.2. SDM

This session reuses the mathematical notation system and refers to the explanation of SDM method in Xiong and F. De la Torre.'s paper [1].

Given an image $d \in \Re^{m \times 1}$ of m pixels, h is a non-linear feature extraction (for example HOG in this paper) and $h(d(x))$ is HOG feature extracted around the vector landmark $x \in \Re^{2p \times 1}$ where p is the number of landmarks. During training, we will assume that the correct p landmarks are known, and we will refer to them as x_*. We set the initial guess of the vector landmark x_0 as mean of x_* over the whole training database. Face alignment problem can be framed as minimizing the following function over Δx:

$$f(x_0 + \Delta x) = \| h(d(x_0 + \Delta x)) - \phi_* \|_2^2 \qquad (2)$$

where $\phi_* = h(d(x_*))$ represents the HOG values around the ground truth landmarks. In the training process, ϕ_* and x_* are known.

The goal of SDM is to learn a series of descent directions such that it produces a sequence of updates $x_{n+1} = x_n + \Delta x_n$ starting from x_0 and converging to x_* in the training data. Assuming that h is twice differentiable, we apply a second order Taylor expansion for equation (2):

$$f(x_0 + \Delta x) \approx f(x_0) + J_f(x_0)^T \Delta x + \frac{1}{2} \Delta x^T H_f(x_0) \Delta x \quad (3)$$

where $J_f(x_0)$ and $H_f(x_0)$ are the Jacobian and Hessian matrices of f evaluated at x_0. Differentiating (3) with respect to Δx and setting it to zero gives us the first update for x:

$$\Delta x_1 = -H_f^{-1}(x_0) J_f(x_0) = -2H_f^{-1}(x_0) J_h^T(x_0)(\phi_0 - \phi_*) \quad (4)$$

where $J_f(x_0) = 2J_h^T(x_0)(\phi_0 - \phi_*)$ - chain rule of derivation and $\phi_0 = h(d(x_0))$.

Let matrix $R_1 = -2H_f^{-1}(x_0) J_h^T(x_0)$, equation (4) becomes:

$$\Delta x_1 = R_1(\phi_0 - \phi_*) = R_1\phi_0 - R_1\phi_* \quad (5)$$

During testing process, ϕ_* is absent, thus let column vector $b_1 = R_1\phi_*$ as a bias term, equation (5) becomes:

$$\Delta x_1 = R_1\phi_0 - b_1 \quad (6)$$

In the supervised manner, we estimate R_1 and b_1 by a linear regression between Δx_1 and ϕ_0. To deal with non-quadratic functions, the SDM algorithm will generate a sequence of descent directions:

$$x_k = x_{k-1} - 2H_f^{-1}(x_{k-1}) J_h^T(x_{k-1})(\phi_{k-1} - \phi_*) \quad (7)$$

Similarly as above, we have the set of $\{R_k\}$ and $\{b_k\}$, equation (7) becomes:

$$\Delta x_k = R_k\phi_{k-1} - b_k \quad (8)$$

such that the succession of x_k converges to x_* for all images in the training dataset.

Assuming that we are given a set of face images $\{d^i\}$ and their correspondent ground truth landmarks $\{x_*^i\}$. Over the whole training data, we need to find R_1 and b_1 which minimizes:

$$\sum_{d^i} \sum_{x_0^i} \| \Delta x_1^i - R_1\phi_0^i + b_1 \|_2^2 \quad (9)$$

where $\Delta x_1^i = x_*^i - x_0^i$ and $\phi_0^i = h(d^i(x_0^i))$. As the first step, we set the x_0^i as the mean shape of the set $\{x_*^i\}$. The subsequent R_k, b_k can be learned after R_{k-1}, b_{k-1} by minimizing:

$$\sum_{d^i} \sum_{x_k^i} \| \Delta x_{k+1}^i - R_{k+1} \phi_k^i + b_{k+1} \|_2^2 \qquad (10.a)$$

where $\Delta x_{k+1}^i = x_*^i - x_k^i$ and new feature vector $\phi_k^i = h(d^i(x_k^i))$. The updating rule for x_k^i:

$$x_{k+1}^i = x_k^i + R_{k+1} \phi_k^i - b_{k+1} \qquad (10.b)$$

Dimensions: $x_k^i \in \Re^{2p \times 1}$, $\phi_k^i \in \Re^n$ where n is the number of HOG features around the landmark x_k^i, $R_k \in \Re^{2p \times n}$ and $b_k \in \Re^{2p \times 1}$.

For mathematically convenience, we add one more row to all column feature vector ϕ such that $\phi = [\phi, -1]^T$ and we add one more column to all matrices R_k as the bias term b_k. From now on, $\phi \in \Re^{n+1}$ and $R_k \in \Re^{2p \times (n+1)}$. The equation (10.a) can be rewritten as:

$$\sum_{d^i} \sum_{x_k^i} \| \Delta x_{k+1}^i - R_{k+1} \phi_k^i \|_2^2 \qquad (11.a)$$

The new updating rule for x_k^i:

$$x_{k+1}^i = x_k^i + R_{k+1} \phi_k^i \qquad (11.b)$$

Note that: we keep the second sigma in formula (11.a) for the purpose of sampling x_k^i with Monte Carlo sampling.

1.3. Pseudo-inverse solution of SDM

From formula (11.a), we have the matrix version:

$$\| Y_{k+1} - R_{k+1} A_k \|_2^2 \qquad (12)$$

where $Y_{k+1} = [\Delta x_{k+1}^1 \Delta x_{k+1}^2 ... \Delta x_{k+1}^m] \in \Re^{2p \times m}$, $R_{k+1} \in \Re^{2p \times (n+1)}$ and $A_k = [\phi_k^1 \phi_k^2 ... \phi_k^m] \in \Re^{(n+1) \times m}$ where m is the number of training samples and $n+1$ is the number of HOG features extracted from landmark x_k^i. Each column i^{th} of Y_{k+1} and A_k makes a pair of $(\Delta x_{k+1}^i, \phi_k^i)$. The solution R_{k+1} from equation (12) can be given in a closed form:

$$R_{k+1} = Y_{k+1} A_k^T (A_k A_k^T)^{-1} = Y_{k+1} A_k^+ \qquad (13)$$

where $A_k^+ = A_k^T (A_k A_k^T)^{-1} \in \Re^{m \times (n+1)}$ is the pseudo-inverse of A_k.

2. Extreme Learning Machine and OPIUM

2.1. The standard ELM

This session refers to the overview introduction of ELM from A. van Schaik and J. Tapson.'s paper [4].

An ELM network with L input neurons, M hidden neurons and N output neurons can be modeled by:

$$y_n^t = \sum_{j=1}^{M} r_{nj}^{(2)} g(\sum_{i=1}^{L} r_{ji}^{(1)} x_i^t), n = 1, 2, .., N \qquad (14)$$

where $x^t \in \Re^{L\times 1}$ is an input data vector to input layer with t is a time or series index, $y^t \in \Re^{N\times 1}$ is the corresponding output vector from output layer, $r^{(1)} \in \Re^{M\times L}$ is the matrix weights mapping from input layer to hidden layer, $r^{(2)} \in \Re^{N\times M}$ is the matrix weights mapping from hidden layer to output layer, $g(.)$ is the sigmoid function defined by $g(x) = \frac{1}{1+e^{-x}}$. Equation (14) can be rewritten in a matrix form:

$$y^t = r^{(2)} g(r^{(1)} x^t) \qquad (15)$$

As in a standard ELM, the matrix weights $r^{(1)}$ are fixed to random values, usually with a normal distribution in some sensible range. Let a^t be the output of hidden layer at time t:

$$a_j^t = g(\sum_{i=1}^{L} r_{ji}^{(1)} x_i^t), j = 1, 2, .., M \Leftrightarrow a^t = g(r^{(1)} x^t)$$

From here, equation (15) becomes:

$$y^t = r^{(2)} a^t \qquad (16)$$

Let A be the matrix of hidden layer activations over a time series:

$$A_t = [a^1 a^2 ... a^t] \in \Re^{M\times t}$$

Let Y be the matrix of output layer activations over a time series:

$$Y_t = [y^1 y^2 ... y^t] \in \Re^{N\times t}$$

Let R be the matrix weights $r^{(2)}$ over a time series:

$$R_t = r^{(2)} \in \Re^{N\times M}$$

Equation (16) can be rewritten in a matrix form over a time series:

$$Y_t = R_t A_t \qquad (17)$$

With Moore-Penrose pseudo-inverse, we can find the weights R_t that minimize the error in (17):

$$R_t = Y_t A_t^+ \qquad (18)$$

From equation (13) and (18), we can see the relation between the pseudo-inverse solution of SDM and ELM when $r^{(1)}$ is replaced by a mapping function from landmark points to HOG features.

2.2. Greville's method

This session refers to the overview introduction of Greville's method from A. van Schaik and J. Tapson.'s paper [4].

Given a series of pairs (a^t, y^t), A_t and Y_t can be partitioned as follows:

$$A_t = [A_{t-1}|a^t] \text{ and } Y_t = [Y_{t-1}|y^t]$$

We can calculate R_t from R_{t-1}:

$$R_t = Y_t A_t^+ \text{ and } R_{t-1} = Y_{t-1} A_{t-1}^+$$

According to Greville's theorem [5][6], the pseudo-inverse A^+ is given by:

$$A_t^+ = [A_{t-1}|a^t]^+ = \begin{bmatrix} A_{t-1}^+ (I - a^t (b^t)^T) \\ (b^t)^T \end{bmatrix} \qquad (19)$$

where $b^t \in \Re^{M \times 1}$ is given by:

$$b^t = \frac{(A_{t-1}^+)^T A_{t-1}^+ a^t}{1 + (a^t)^T (A_{t-1}^+)^T A_{t-1}^+ a^t} \qquad (20)$$

To simplify the expression for b^t, we define a symmetric square matrix $\theta_{t-1} \in \Re^{M \times M}$:

$$\theta_{t-1} = (A_{t-1}^+)^T A_{t-1}^+ = (A_{t-1} A_{t-1}^T)^+ \qquad (21)$$

θ is the pseudo-inverse of the correlation matrix of the hidden layer activations. From (20) and (21), we have:

$$b^t = \frac{\theta_{t-1} a^t}{1 + (a^t)^T \theta_{t-1} a^t} \qquad (22)$$

To update θ_t when a new pair (a^t, y^t) is observed:

$$\theta_t = (A_t A_t^T)^+ = (A_{t-1} A_{t-1}^T + a^t (a^t)^T)^+ \qquad (23)$$

Kovanic [7] shows that the matrix inversion lemma may be applied to the pseudo-inverse of (23) to obtain an update rule for θ_t. The matrix inversion lemma states:

$$(D + BCB^T)^{-1} = D^{-1} - \frac{D^{-1}BB^TD^{-1}}{C^{-1} - B^TD^{-1}B} \tag{24}$$

where C, D are non-singular and BCB^T has the same dimensionality as D. Choosing $D = \theta_{t-1}^{-1}$, $B = a^t$ and $C = I$:

$$\theta_t = (\theta_{t-1}^{-1} + a^t I (a^t)^T)^+ = \theta_{t-1} - \frac{\theta_{t-1}a^t(a^t)^T\theta_{t-1}^T}{1 + (a^t)^T\theta_{t-1}a^t}$$

$$\Leftrightarrow \theta_t = \theta_{t-1} - \theta_{t-1}a^t\frac{(a^t)^T\theta_{t-1}^T}{1 + (a^t)^T\theta_{t-1}a^t}$$

$$\Leftrightarrow \theta_t = \theta_{t-1} - \theta_{t-1}a^t(b^t)^T \tag{25}$$

To update the weights R_t:

$$R_t = Y_tA_t^+ = [Y_{t-1}|y^t]\begin{bmatrix} A_{t-1}^+(I - a^t(b^t)^T) \\ (b^t)^T \end{bmatrix}$$

$$\Leftrightarrow R_t = Y_{t-1}A_{t-1}^+(I - a^t(b^t)^T) + y^t(b^t)^T$$

$$\Leftrightarrow R_t = Y_{t-1}A_{t-1}^+ - Y_{t-1}A_{t-1}^+a^t(b^t)^T + y^t(b^t)^T$$

$$\Leftrightarrow R_t = Y_{t-1}A_{t-1}^+ + (y^t - Y_{t-1}A_{t-1}^+a^t)(b^t)^T$$

$$\Leftrightarrow R_t = R_{t-1} + (y^t - R_{t-1}a^t)(b^t)^T \tag{26}$$

2.3. OPIUM-Light

This session presents the idea of OPIUM-Light algorithm which is invented by A. van Schaik and J. Tapson. and refers to [4].

From the experiments, it can be observed that off-diagonal elements of θ_t are very small and the diagonal elements have similar values. The OPIUM-Light algorithm is based on an assumption that $\theta_t = c.I$ where $c \in \Re$ is a constant. We can make more simplification for updating rules of b^t and R_t which are now independent from θ_{t-1}:

$$b^t = \frac{\theta_{t-1}a^t}{1 + (a^t)^T\theta_{t-1}a^t} = \frac{a^t}{1/c + (a^t)^Ta^t},$$

$$R_t = R_{t-1} + (y^t - R_{t-1}a^t)(b^t)^T \qquad (27)$$

2.4. OPIUM-Dynamic

This session presents the idea of OPIUM-Dynamic algorithm which is invented by A. van Schaik and J. Tapson. and refers to [4].

From equation (18), we can have more derivations:

$$R_t = Y_t A_t^+ = Y_t A_t^T (A_t A_t^T)^{-1} = \psi_t \theta_t$$

where $\psi_t = Y_t A_t^T$ and $\theta_t = (A_t A_t^T)^{-1}$, ψ is the cross correlation matrix between output vectors and hidden layer activations, θ is the inverse of the auto correlation of the hidden layer activations. The pseudo-inverse identity $A^+ = A^T(AA^T)^{-1}$ is valid if the rows of A are linearly independent. The updating rule for ψ_t:

$\psi_t = Y_t A_t^T = [Y_{t-1}|y^t][A_{t-1}|a^t]^T = Y_{t-1}A_{t-1}^T + y^t(a^t)^T = \psi_{t-1} + y^t(a^t)^T$

We introduce a new constant $\alpha \in \Re$ to weight the training samples as follows:

$$\psi_t = (2-\alpha)\psi_{t-1} + \alpha y^t (a^t)^T,$$
$$\theta_t = ((2-\alpha)A_{t-1}A_{t-1}^T + \alpha a^t (a^t)^T)^+$$

The new updating rules will be:

$$\theta_t = \frac{1}{2-\alpha}(\theta_{t-1} - \theta_{t-1}a^t(b^t)^T),$$
$$R_t = R_{t-1} + (y^t - R_{t-1}a^t)(b^t)^T$$

with

$$b^t = \frac{\theta_{t-1}a^t}{\frac{2-\alpha}{\alpha} + (a^t)^T\theta_{t-1}a^t}.$$

If $\alpha = 1$: the original version of OPIUM, but if $\alpha > 1$: more weight is given to the more recent training examples.

3. The adaptation method

3.1. Training process of supervised SDM

3.1.1. Initialization

In this part, in order to make the training process in a uniform basis,

preprocessing can be described as follows:

For each i^{th} image of the database

Find the Viola Jones (VJ detector) rectangles containing the face,

Extract the part of image inside VJ rectangle,

Normalize the facial image to a fixed size $\rightarrow$ canonical image,

Translate and normalize the ground truth landmark x_*^i orresponding to the canonical image.

End loop;

Besides, we need to compute the starting landmark x_0 as the mean shape of the ground truth landmarks over the whole training database:

$$x_0 = \frac{1}{m}\sum_{i=1}^{m} x_*^i$$

where m is the number of training samples.

3.1.2. Main program

The pseudo-code of SDM finding linear regression set of matrices $\{R_1, R_2, .., R_K\}$ that converges $x_0 \rightarrow x_*$ for each image of the training database is as follows:

For $k = 1 \rightarrow K$

Create matrix $Y = [\Delta x_k^1 \Delta x_k^2 .. \Delta x_k^m] \in \Re^{2p\times m}$ where

$\Delta x_k^i = x_*^i - x_{k-1}^i \in \Re^{2p\times 1}, i = 1 \rightarrow m$

If $k = 1$then $\Delta x_k^i = x_*^i - x_0$.

Create matrix $A = [\phi_{k-1}^1 \phi_{k-1}^2 .. \phi_{k-1}^m] \in \Re^{(n+1)\times m}$ where

$\phi_{k-1}^i = [h(d^i(x_{k-1}^i)), -1]^T \in \Re^{(n+1)\times 1}, i = 1 \rightarrow m$ and

$h(d^i(x_{k-1}^i))$ is HOG feature extracted from the i^{th} image around the landmark x_{k-1}^i

If $k = 1$ then $\phi_{k-1}^i = [h(d^i(x_0)), -1]^T$

Compute the matrix $R_k = YA^+ = YA^T(AA^T)^{-1} \in \Re^{2p\times(n+1)}$

Version with regularization parameter λ: $R_k = YA^T(AA^T + \lambda.I)^{-1}$

Update the current landmarks: $x_k^i = x_{k-1}^i + R_k\phi_{k-1}^i, i = 1 \to m$

If $k = 1$ then $x_k^i = x_0 + R_k\phi_{k-1}^i$.

End loop;

Technical information:

a. In practice, when the training database is huge as m is a big number, it is not necessary to store both matrices Y and A at once. We can separate Y and A into many batches and keep updating two matrices $C = YA^T \in \Re^{2p\times(n+1)}$ and $D = AA^T \in \Re^{(n+1)\times(n+1)}$ by summing up technique. Matrix $R_k = C(D + \lambda.I)^{-1}$ where C and D are not so big matrices.

b. In practice, to balance between computational complexity and accuracy, the number of matrices R is chosen $K = 4$. The set of R: $\{R_1, R_2, R_3, R_4\}$.

c. The number of landmark points of each face is $p = 49$ which includes 10 points of eye-brows (both left and right), 9 points of nose, 12 points of eyes (both left and right) and 18 points of mouth. A vector landmark is in that form: $x = [x_1y_1x_2y_2..x_py_p]^T$ where (x_i, y_i) is a coordinate of a landmark point.

d. The regularization parameter λ is chosen by heuristics as $\lambda = 0.1 \times m$ where m is number of training samples.

e. HOG descriptor has patch size 24 by 24 pixels and 9 spatial bins with degree from 0 to 360. Patch is separated into 4 by 4 partitions, each partition is a 6 by 6 matrix. A block contains 2 by 2 partitions which has 12 by 12 pixels. A patch consists of 3 by 3 overlapping blocks. From each block, the voting process is

applied to choose 9 main features from 9 spatial bins. Totally, each patch has 81 HOG features. In this paper, with 49 landmark points, we extract HOG features from 49 patches centered at the landmark points. Therefore, the number of HOG features from 1 vector landmark is $n = 3969$. For the case of 14 landmark points including 12 points of eyes (6 for the left and 6 for the right) and 2 points of upper nose, the number of HOG features from 1 vector landmark is $n = 1134$.
Introduction of HOG descriptor is mentioned in the Appendix A.

f. The 2D Procrustes process can be applied inside the main loop for k = 1, 2 after the updating of new landmark positions. Let $\alpha^i = angle(x_k^i, x_0), i = 1 \rightarrow m$ be the angle between new landmark x_k^i with the reference x_0 which can be calculated from Procrustes algorithm. After that, rotate the canonical image d^i as well as x_k^i by α^i-degree angle. The purpose is to rotate the face into the frontal pose. Procrustes is a heavily computational and because of that reason, it is only applied for 2 first iterations, later the poses are almost frontal.
More detail of Procrustes algorithm is mentioned in the Appendix B.

g. We can apply the Principal Component Analysis (PCA) for HOG descriptors over the whole training database to extract the main features and reduce the feature dimensions.

3.2. Facial tracking with supervised SDM

With the set of trained matrices $\{R_1, R_2, .., R_K\}$, the starting vector landmark x_0, the testing database of the set of new images $\{d^i\}$can be tracked as follows:
Assuming that every d^i is normalized VJ rectangle – canonical image as preprocessing mentioned in 3.1.1.

For $i = 1 \rightarrow m$

For $k = 1 \rightarrow K$

Feature extraction: $\phi = [h(d^i(x^i_{k-1})), -1]^T \in \Re^{(n+1)\times 1}$

If k = 1 then $\phi = [h(d^i(x_0)), -1]^T$

Update new landmark: $x^i_k = x^i_{k-1} + R_k\phi$

If $k = 1$ then $x^i_k = x_0 + R_k\phi$

End loop;

End loop;

Technical information: Remark that for $k = 1 \to 2$ after the updating new landmark, the 2D Procrustes process can be applied as mentioned above.

3.3. Modified facial tracking

The pipeline of tracking process for one canonical image is as following:

$$x_0 \to R_1 \to x_1 \to R_2 \to ... \to x_{K-1} \to R_K \to x_K$$

where x_0 is the starting vector landmark and x_K is the final result vector landmark. We have discovered that the result can be improved if we take previous x_K and set it to x_0 and then repeat the tracking process again, get a new x_K and repeat it few times. The tracking algorithm in 3.2 can be modified as following with J is the number of repetition.

For $i = 1 \to m$

For $j = 1 \to J$

For $k = 1 \to K$

Feature extraction: $\phi = [h(d^i(x^i_{k-1,j})), -1]^T \in \Re^{(n+1)\times 1}$

If $k = 1$ and $j = 1$ then $\phi = [h(d^i(x_0)), -1]^T$

If $k = 1$ and $j > 1$ then $\phi = [h(d^i(x^i_{K,j-1})), -1]^T$

Update new landmark: $x^i_{k,j} = x^i_{k-1,j} + R_k\phi$

If $k = 1$ and $j = 1$ then $x^i_{k,j} = x_0 + R_k\phi$

If $k = 1$ and $j > 1$ then $x^i_{k,j} = x^i_{K,j-1} + R_k\phi$

End loop;

End loop;

End loop;

Technical information: In our experiment, J is in the range of $[1, 8]$.

3.4. Adaptive facial tracking improved by unsupervised OPIUM

We have discovered that adaptation can be tempted on-line when the ground truth or expert annotation is not available. We adjust the pseudo-inverse by generating new training samples from the best available estimation. We introduce the set of original matrices $\{R_1^{original}, ..., R_K^{original}\}$ and the set of adaptive matrices $\{R_1^{adaptive}, ..., R_K^{adaptive}\}$. At first, $R_k^{adaptive} := R_k^{original}, k = 1 \rightarrow K$. We utilized the OPIUM-Light algorithm which is precise and very fast (equation 27). Assuming that our algorithm is running with infinite time series $t = 1 \rightarrow \infty$, after a period of τ frames, we need to reset the set of adaptive matrices to the original ones. In our experiment, we found that the OPIUM-Light adaptation can work only with the set of matrices $\{R_3, ..., R_K\}$ except 2 first matrices. The algorithm implementation has the below pseudo-code:

For $t = 1 \rightarrow \infty$

If $t \ mod \ \tau = 1$ then $R_k^{adaptive} := R_k^{original}, k = 1 \rightarrow K$

Get a new image d^t, normalize it to the VJ rectangle.

For $j = 1 \rightarrow J$

For $k = 1 \rightarrow K$

Feature extraction:

$\phi_{k-1,j} = [h(d^t(x_{k-1,j}^t)), -1]^T \in \Re^{(n+1)\times 1}$

If $k = 1$ and $j = 1$ then:

$\phi_{k-1,j} = [h(d^t(x_0)), -1]^T$

If $k = 1$ and $j > 1$ then:

$$\phi_{k-1,j} = [h(d^t(x^t_{K,j-1})), -1]^T$$

Update new landmark:

$$x^t_{k,j} = x^t_{k-1,j} + R^{adaptive}_k \phi_{k-1,j}$$

If $k = 1$ and $j = 1$ then:

$$x^t_{k,j} = x_0 + R^{adaptive}_k \phi_{k-1,j}$$

If $k = 1$ and $j > 1$ then :

$$x^t_{k,j} = x^t_{K,j-1} + R^{adaptive}_k \phi_{k-1,j}$$

End loop;

End loop;

For $k = 3 \rightarrow K$

$$a = \phi_{k-1,J} \in \Re^{(n+1)\times 1}$$

$$b = \frac{a}{1/c + a^T a} \in \Re^{(n+1)\times 1}$$

$$R^{adaptive}_k = R^{adaptive}_k + (x^t_{K,J} - R^{adaptive}_k a)b^T$$

End loop;

End loop;

Technical information:

a. The constant c is set to be 1 in our experiment.

b. The finding of optimal values for τ and J will be discussed in Chapter III. Experiments and Results.

Chapter III. Experiments and Results

1. Databases overview

Our supervised SDM algorithm was trained with Multi-PIE [9], CK+ [10] and BU-3DFE [11] databases. Both unsupervised adaptive facial tracking with OPIUM-Light algorithm and supervised SDM were tested on BP-4DSFE [12] database. We also tested the performance of the adaptive tracking with 14 landmark points (12 points of eyes and 2 points of upper nose) in the upper face over Multi-view gaze database [13].

1.1. The CMU Multi-PIE Face Database

The full description of the database can be found at [9].

Short name: Multi-PIE.

The Carnegie Mellon University Multi-PIE face database contains more than 750,000 high resolution images of 337 people recorded in up to four sessions under 15 view points and 19 illumination conditions while displaying a range of facial expressions [9]. The Multi-PIE database has addressed several shortcomings from the previous PIE database: a limited number of subjects, a single recording session and only few expressions captured [9].

1.2. The Extended Cohn-Kanade Database

The full description of the database can be found at [10].

Short name: CK+.

The image data contains facial behavior of 210 adult participants ranging from 18 to 50 years of age, 69% female, 81% European-American, 13% Afro-American, and 6% other groups [10]. Participants were instructed by an experimenter to perform a series of 23 facial displays which include single action units and combinations of action units [10]. Image sequences for frontal views and 30-degree views were digitized into either 640 x 490 or 640 x 480 pixel arrays with 8-bit gray scale or 24-bit color values [10].

1.3. Binghamton University 3D Facial Expression Database

The full description of the database can be found at [11].

Short name: BU-3DFE.

The database presently contains 100 subjects (56% female and 44% male), ranging age from 18 years to 70 years old, with a variety of ethnic/racial ancestries, including White, Black, East-Asian, Middle-east Asian, Indian, and Hispanic Latino [11]. Each subject performed seven expressions (neutral, happiness, disgust, fear, angry, surprise and sadness) with four levels of intensity in front of the 3D face scanner [11]. There are 25 instant 3D expression models for each subject, resulting in a total of 2,500 3D facial expression models in the database [11]. Associated with each expression shape model, there is a corresponding facial texture image captured at two views (about +45 and -45 degree) [11]. As a result, the database consists of 2,500 two-view's texture images and 2,500 geometric shape models [11].

1.4. Binghamton–Pittsburgh 4D Spontaneous Expression Database

The full description of the database can be found at [12].

Short name: BP-4DSFE.

The database consists of over 368,000 frame models from 41 subjects (56% female, 48.7% European-American, average age 20.2 years) of ethnically and racially diverse (European-American, African-American, East-Asian, Middle-Eastern, Asian, Indian, and Hispanic Latino) backgrounds. Subjects were imaged individually using the Di3D (Dimensional Imaging) dynamic face capturing system while responding to a varied series of 8 emotion inductions that elicited spontaneous expressions of amusement, surprise, fear, anxiety, embarrassment, pain, anger, and disgust. The 3D models range in resolution between 30,000 and 50,000 vertices. For each sequence, manual FACS coding by highly experienced and reliable certified coders was obtained.

1.5. Multi-view gaze Database

The full description of the database can be found at [13].

The database consists of a total of 50 (15 female and 35 male) people ranging in age approximately from 20 to 40 years old. A chin rest was used to stabilize the head position located at 60 cm apart from the monitor. During recording sessions, participants were instructed to look at a visual target displayed on the monitor. The screen was divided into a 16 x 10 regular grid, and the visual target moved to the center of each grid in a random order. The white circle shrank after the target stops at each position, and cameras were triggered at the time the circle disappeared. As a result, G = 160 (gaze directions) x 8 (cameras) images were acquired from each participant at SXGA resolution, together with the 3D positions of the visual targets. The gaze directions spanned approximately ± 25 degrees horizontally and ± 15 degrees vertically, and this covered the range of natural gaze directions. Images are recorded by a fully calibrated 8 multi-camera system, and the 3D reconstruction of eye regions was done by using a patch-based multi-view stereo algorithm.

2. Results

2.1. Testing on BP-4DSFE Database

After the training of the set of matrices

$$\{R_1^{original}, R_2^{original}, R_3^{original}, R_4^{original}\}$$

over Multi-PIE, CK+ and BU-3DFE databases, we estimated the performance of the unsupervised adaptive tracking and supervised SDM on BP-4DSFE database. The personalization and adaptation of our method was tested with 41 subjects in the database. Each subject contains 8 different tasks with different emotional expressions, each task is a series of images framed from a video. From each task, we have selected randomly 32 images. Each subject has 256 randomly chosen images.

To check the improvement of the adaptive method for each person, we adapted

$\tau \in \{0, 1, 2, 4, 8, 16, 32, 64\}$ random images out of 256 images to find the set of matrices $\{R_1^{adaptive}, R_2^{adaptive}, R_3^{adaptive}, R_4^{adaptive}\}$ and use these matrices to track the rest of images without any more adaptation. In the case of $\tau = 0$, no adaptation and the set of R matrices is still the same as the original one. To check the improvement after each big iteration, we tried with $J \in \{1, 2, 4, 8\}$ times of big iterations.

To measure the accuracy compared to the ground truth landmarks, we calculated the averge Root Mean Squares Error (RMSE) over the whole testing database of 41 subjects. RMSE between landmark $x \in \Re^{2p \times 1}$ and the ground truth $x_* \in \Re^{2p \times 1}$, where $p \in \{49, 14\}$ is the number of landmark points, is defined as follows:

$$RMSE(x, x_*) = \{\tfrac{1}{p} \parallel x - x_* \parallel_2^2\}^{1/2} = \{\tfrac{1}{p} \sum_{i=1}^{2p} (x_i - x_{*i})^2\}^{1/2}$$

The average RMSEs over the testing databases with m samples can be defined as:

$$\tfrac{1}{m} \sum_{i=1}^{m} RMSE(x^i, x_*^i)$$

We measured the RMSEs for all 49 landmark points of face, 4 landmark points of eye corners and 2 landmark points of mouth corners. Here are the result tables for face, eye corners and mouth corners. The rows are τ and the columns are J. The highlighted and underlined numbers are the best average RMSEs in each column – after each J big iterations.

τ/J	1	2	4	8
0	4.3363	3.1580	3.1839	3.1957
1	4.1406	3.0392	3.0791	3.0878
2	4.0100	2.9660	3.0016	3.0046
4	3.8289	2.9010	2.9150	2.9192
8	3.6574	**2.8463**	**2.8385**	**2.8591**

16	**3.4974**	2.8841	2.8951	2.9209
32	3.6167	3.2403	3.2196	3.2221
64	4.6297	4.1705	4.1673	4.1607

Table 1. Average RMSEs of 49 facial landmark points

τ/J	1	2	4	8
0	0.6695	0.5742	0.5841	0.5889
1	0.6288	0.5563	0.5663	0.5687
2	0.6005	0.5450	0.5538	0.5576
4	**0.5830**	**0.5447**	**0.5508**	**0.5533**
8	0.5936	0.5560	0.5573	0.5659
16	0.6362	0.5957	0.5941	0.5999
32	0.7148	0.6804	0.6883	0.6847
64	0.8969	0.7971	0.8059	0.7992

Table 2. Average RMSEs of 4 eye-corners landmark points

τ/J	1	2	4	8
0	0.5247	0.4875	0.4934	0.4962
1	0.5056	0.4694	0.4839	0.4848
2	0.4998	**0.4638**	**0.4778**	**0.4780**
4	**0.4981**	0.4694	0.4793	0.4791
8	0.5202	0.4917	0.4836	0.4884
16	0.5247	0.4830	0.4737	0.4808
32	0.5837	0.5334	0.5239	0.5337
64	0.5951	0.5417	0.5327	0.5320

Table 3. Average RMSEs of 2 mouth-corners landmark points

Here are the Cumulative Error Distribution (CED) curves of RMSEs after the first big iteration ($J = 1$) with 49 facial landmark points ($\tau = 16$), 4 eye-

corners landmark points ($\tau = 4$), 2 mouth-corners landmark points ($\tau = 4$). The horizontal axis the RMSE value and the vertical axis is portion of testing data, ranges from 0 to 1. The red line represents the unsupervised adaptive tracking, while the dotted green line represents the original SDM tracking.

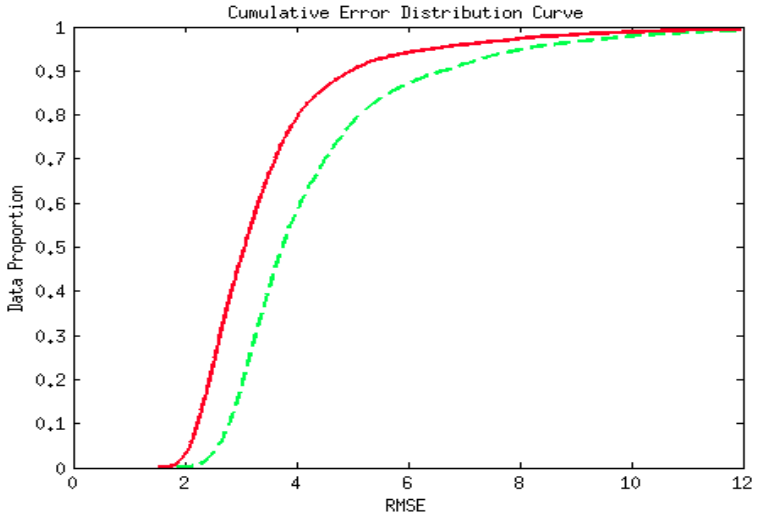


Figure 1. CED curve of 49 facial landmark points.

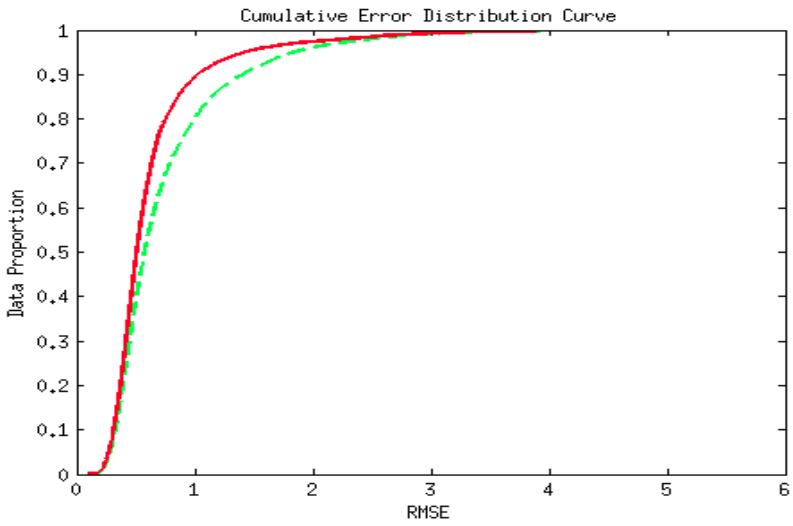


Figure 2. CED curve of 4 eye-corners landmark points.

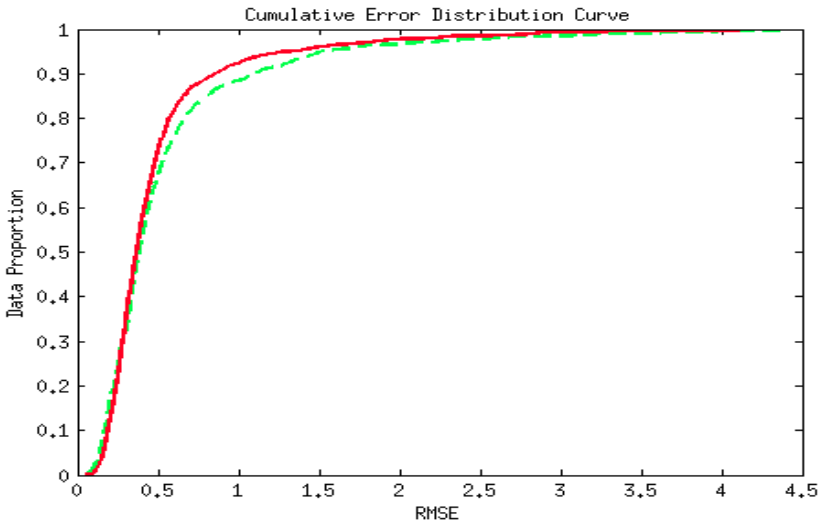


Figure 3. CED curve of 2 mouth-corners landmark points.

From our experiment, the adaptive method can improve significantly with African and Indian people having dark skin, when the original SDM tracking could not work properly. From the BP-4DSFE database, we separated a portion including 5 African females, 1 Indian female and 2 African males and tested the adaptive method with this group of people. We have chosen $\tau = 16$ for this experiment. The following table shows the average RMSEs of 49 facial landmark points with $J \in \{1, 2, 4, 8\}$ between the original SDM and the adaptive method.

J	1	2	4	8
SDM	5.9639	5.2144	5.4403	5.4758
Adaptive	5.0474	**4.6274**	4.7083	4.7938

Table 4. Average RMSEs of 49 facial points between SDM and the adaptive method with dark skin people

J	1	2	4	8
SDM	0.9528	1.0569	1.1078	1.1297
Adaptive	**0.8908**	0.9905	0.9836	1.0090

Table 5. Average RMSEs of 4 eye corners between SDM and the adaptive method with dark skin people

J	1	2	4	8
SDM	1.1279	0.9322	0.9596	0.9654
Adaptive	0.8658	0.7939	0.7753	**0.7718**

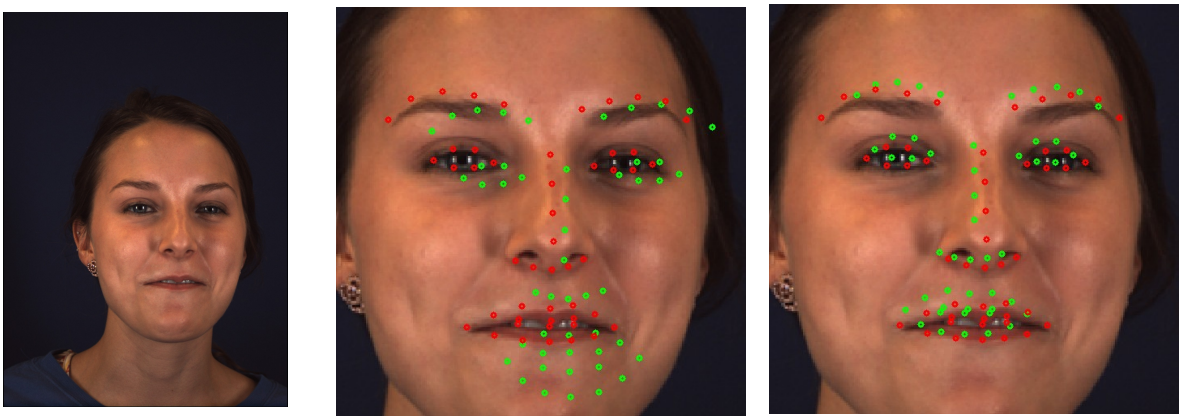
Table 6. Average RMSEs of 2 mouth corners between SDM and the adaptive method with dark skin people

From the testing results, for the value of τ and J should be in the range of $[1, 32]$ and $[1, 8]$ to ensure that the tracking system can converge. It can be observed that big number of τ can make the system diverge. In practice, we avoid to adapt with so long history. In real time running, when two consecutive frames are similar with speed of camera 16 fps, the adaptive method can converge successfully during a period of 7 – 8 seconds, after that we need to reset the adaptive matrices to original ones.

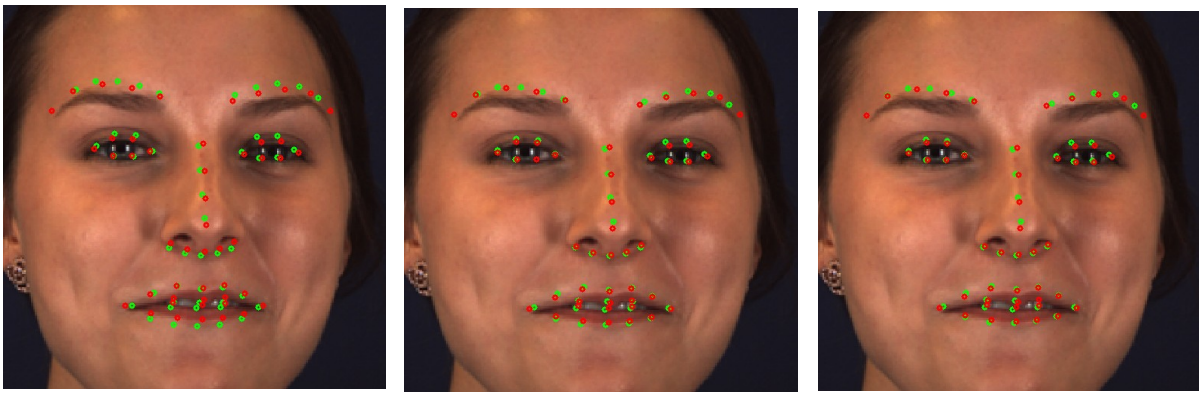
In summary, the adaptive method utilizing OPIUM-Light algorithm is a very fast and efficient solution to improve and personalize the original SDM facial tracking.

2.2. Example of the original SDM

This is an example of the SDM tracking through 4 matrices with one image of BP-4DSFE database.



a. b. c.



d. e. f.

Figure 4.a. Input image

Figure 4.b. Mean shape x_0 (green) and the ground truth (red) in the canonical image

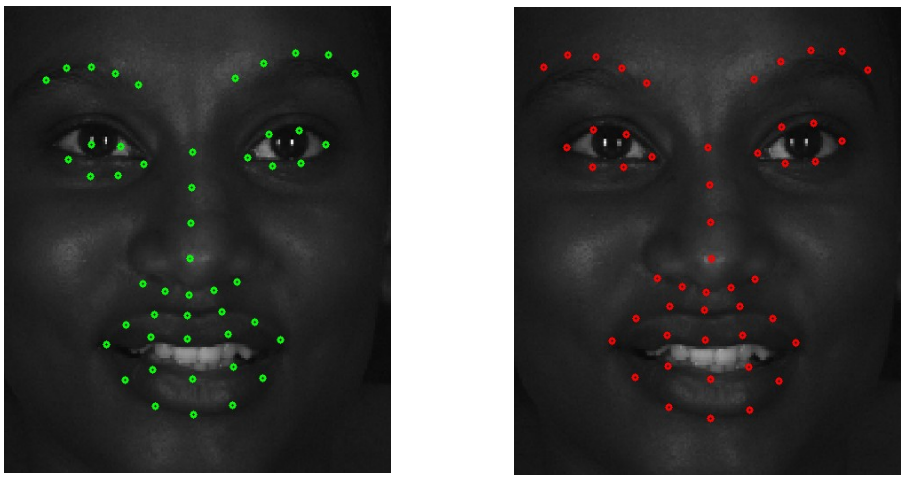
Figure 4.c, d, e, f. Landmark x_1, x_2, x_3, x_4 (after R_1, R_2, R_3, R_4 - green) and the ground truth

RMSE of 49 landmark points is 3.6223

2.2. Examples of the adaptive method

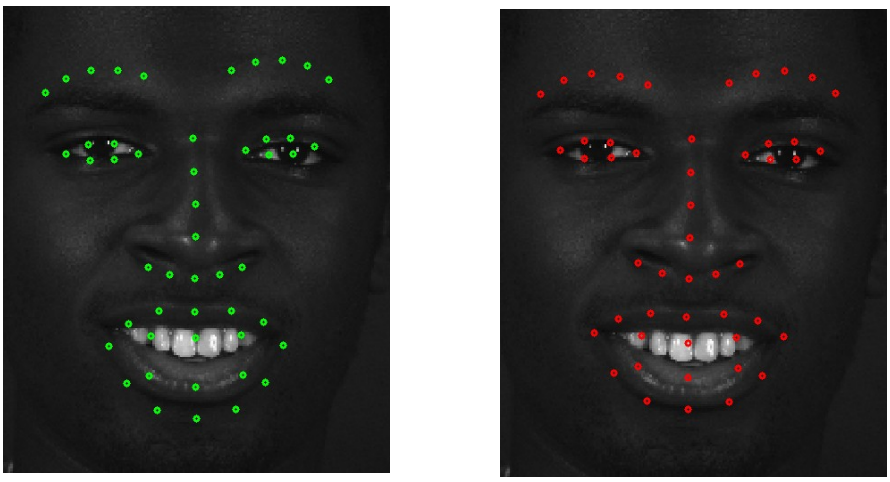
Here are two examples comparing the original SDM and the adaptive method with dark skin people from BP-4DSFE database. The green dots are original SDM landmarks, the red dots are the adaptive landmarks.

Figure 5 and Table 7. An example from BP-4DSFE database



RMSEs	Face	Eye corners	Mouth corners
SDM	5.4193	1.4488	0.6497
Adaptive	4.4806	0.7266	0.3907

Figure 6 and Table 8. Another example from BP-4DSFE database



RMSEs	Face	Eye corners	Mouth corners
SDM	7.5251	1.0476	1.6018
Adaptive	4.8759	0.3805	0.4137

2.3. Testing on Multi-view gaze Database

In order to apply the adaptive method for gaze direction detection, we trained a special SDM with 14 landmark points on only the upper half of a face including 12 landmark points for eyes (6 for the left one and 6 for the right one) and 2 landmark points for the upper part of nose. The adaptation could give a better estimation of landmark positions and we hope it would give a better result for gaze direction detection in the near future. Here are some examples from the Multi-view gaze database. The green dots is the original SDM landmarks while the red dots represents the landmarks of adaptive method.

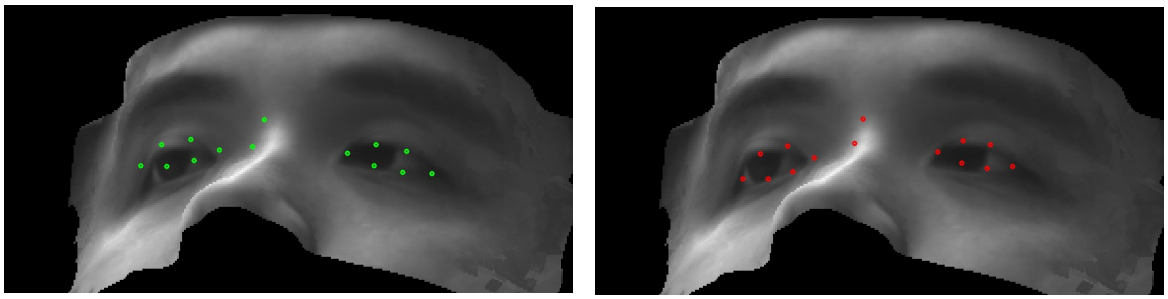


Figure 7. An example from Multi-view gaze database

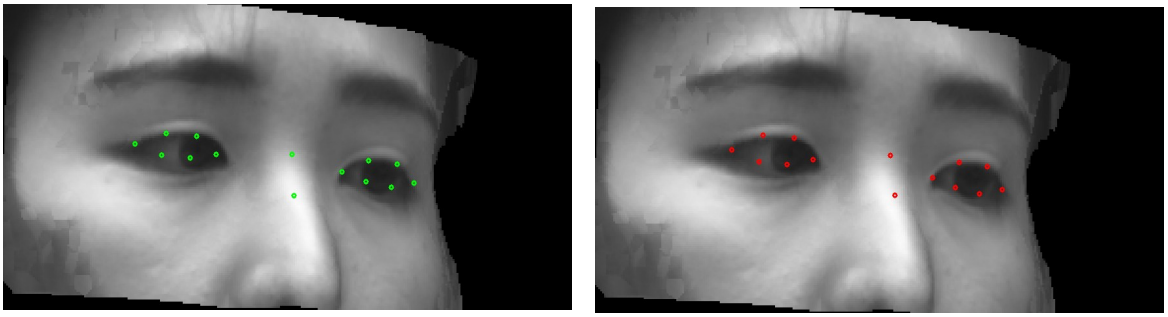


Figure 8. Another example from Multi-view gaze database

Chapter IV. Future work and conclusion

1. Future work

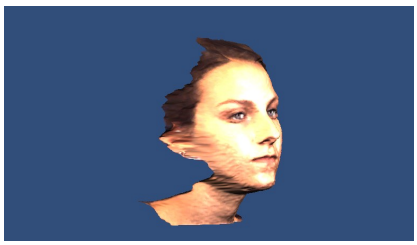
1.1. Higher order features

In this paper, we replaced the weights mapping from the input layer to hidden layer of a standard ELM by HOG descriptor and archived some promising results. To advance the ELM and OPIUM-Light, HOG features will be used as input signal for the input layer and a higher-order feature going through sigmoid function from the hidden layer will be learned for linear regression of $\{R_1, R_2, R_3, R_4\}$. Besides, OPIUM-Dynamic with more flexible in adapting recent samples could reach a better performance for the adaptive method. The estimation of gaze direction and action units as well as emotions can be also improved with more precise adaptive method.

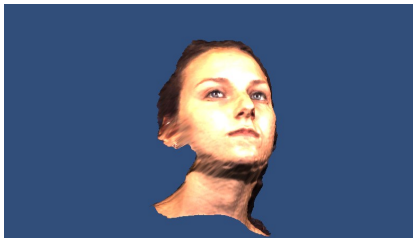
1.2. Facial tracking with larger angles of head pose

To address the limitation of current databases in small and discrete head pose, from 3D meshes and textures of BP-4DSFE database, we have created a software by Unity technology which can rotate 3D model and the corresponding landmarks in many head poses and project it again into 2D image. We selected randomly 13,120 3D models from BP-4DSFE and generated more than 200,000 2D images with $yaw \in \{0, \pm 22.5, \pm 45, \pm 80, \pm 105\}$ degree and $pitch \in \{0, \pm 20, \pm 30\}$ degree. The next step of this paper will be the adaptive facial tracking method with angle dependences which is expected to increase the accuracy of landmark estimation for special and unseen head poses.

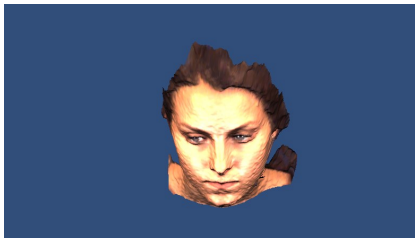
a. $yaw = 0°$ and $pitch = 0°$



b. $yaw = 45°$ and $pitch = 0°$



c. $yaw = 22.5°$ and $pitch = 20°$



d. $yaw = 0°$ and $pitch = -30°$

Figure 9. Some examples from our generated 3D database.

2. Conclusion

We extended the Supervised Descent Method facial tracking with bootstrap adaptation for individually users in the absence of labeled training samples. We achieved considerable improvements in precision for facial landmarks with fast and precise unsupervised Online Pseudo-Inverse Update Method OPIUM-Light algorithm which can applied for real time application and it opened up new ideas and directions for our future researches.

Appendix A. Histogram of Oriented Gradients

In the area of computer vision and image processing, Histogram of Oriented Gradients (HOG) is one of the *state of the art* feature descriptor popularly used for object detection and image alignment. In general, the idea of HOG is to analyze the occurrences of gradient orientation in specific localized portions or regions of interest in an image. In practice, HOG has shown a fast, efficient and high precision performance compared to some other similar feature extraction methods such as Edge Orientation Histograms or Scale Invariant Feature Transform (SIFT).

The descriptor was first proposed by Dalal and Triggs in their June 2005 CVPR paper [8] in the context of pedestrian detection in static images. To prove the effectiveness of HOG descriptor as well as generalize their results, Dalal and Triggs expanded the algorithm for human detection in video, animals and and vehicles in static images.

Dalal and Triggs stated that key features of shape and object appearances can be selected from analyzing intensity gradient distribution and voting to find significant edge directions [8]. First, the algorithm divides image into cells which are groups of connected pixels. For each cell, the magnitude (or intensity) and angle of gradients of every pixel are computed. To improve the accuracy, the authors suggested contrast-normalization by measuring the intensities across multiple of cells, which is block, then normalizing all cells inside the block. This improvement has performed its advantage in different kinds of illumination and shadowing.

In more detail, to compute the gradient value for each pixel, the widely used method is simple application of the 1-D centered filter kernel (in another name, point discrete derivative mask):

$$[-1\ 0\ 1] \text{ and } [-1\ 0\ 1]^T$$

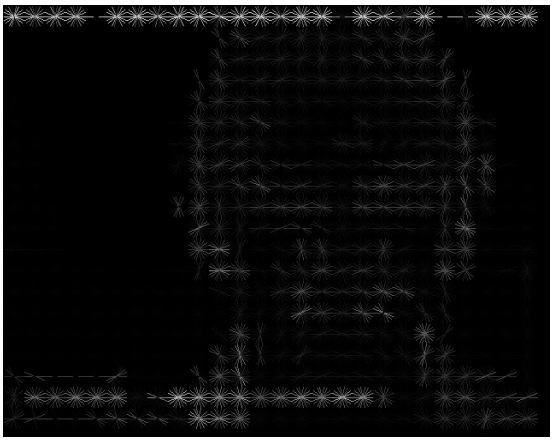
Let dx be the horizontal edge (result of applying the first filter) and dy be the vertical edge (result of applying the second filter). The magnitude m and angle α of the gradient in that pixel can be given by:

$$m = \sqrt{dx^2 + dy^2} \text{ and } \alpha = tan^{-1}(dy/dx)$$

Because image preprocessing causes little impact on performance when achieving the same results, the authors of HOG omitted that step before gradient computation. Other, more complex masks and Gaussian smoothing before filtering were also considered but the experiments gave poorer performance in human image detection.

a.

b.



c.

Figure 10.a. A sample image from CK+ database

Figure 10.b, c. HOG feature of patch size 8 by 8 and 24 by 24 with 9 spatial bins

The next step of algorithm is creating the cell histograms. Overall, the cell can be rectangular or circular in shape and the histogram channels can be unsigned (from 0 to 180 degree) or signed (from 0 to 360 degree). Histogram channels or spatial bins can be defined as groups of gradients having similar angles. For instance, Dalal and Triggs discovered that unsigned gradients performed with 9 histogram channels the best in their human detection experiments. In normal words, they divided 180 degrees into 9 separate parts: $0^\circ \rightarrow 20^\circ$, $20^\circ \rightarrow 40^\circ$, and so on to $160^\circ \rightarrow 180^\circ$. In our experiments in the context of facial tracking problem, we found that the sign of gradients play a very important role in changing movement descents and 360-degree-histogram-channel HOG gets better accuracy. Inside each spatial bin, the voting process is implemented to select the most significant magnitude, in other words, to determine the maximum magnitude in each of 9 direction angles.

As mentioned above, to overcome the challenges of illumination and shadowing, the local normalization of gradient strengths must be necessary, in which we group the cells together into larger, connected blocks. The authors recommended blocks overlapping, that means each cell contributes more than once in the final descriptor. Rectangular R-HOG blocks and circular C-HOG blocks are recently used geometric block shapes. In our facial tracking experiments, we applied R-HOG blocks.

For block normalization, Dalal and Triggs introduced four different methods as following. Let v be the non-normalized vector containing all histograms in a given block, $\| v \|_k$ be its k-norm for $k = 1, 2$ and e be some small constant. Then the normalization factor can be one of the following:

- L_2-norm: $f = \frac{v}{(\| v \|_2^2 + e^2)^{1/2}}$
- L_2-hys: Limit the maximum values of v to 0.2 and then apply L_2-norm
- L_1-norm: $f = \frac{v}{\| v \|_1 + e}$
- L_1-sqrt: $f = \sqrt{\frac{v}{\| v \|_1 + e}}$

In our facial tracking experiment, L_2-norm is used and gets slightly better result compared to L_1-norm.

In summary, implementation of the HOG descriptor algorithm can be described as 5 main steps:

1. Divide the image into small connected regions called cells, and for each cell compute a histogram of gradient directions or edge orientations for the pixels within the cell.

2. Discrete each cell into spatial bins (histogram channels) according to the gradient orientation.

3. Each cell's pixel contributes gradient magnitude to its corresponding spatial bin (histogram channel).

4. Groups of adjacent cells are considered as spatial regions called blocks. The grouping of cells into a block is the basis for grouping and normalization of histograms.

5. Normalized group of histograms represents the block histogram. The set of these block histograms represents the descriptor.

Appendix B. Procrustes Analysis

Procrustes analysis is a form of statistical shape analysis used to analyze the distribution of a set of shapes. Before comparing the shape of two or more objects, the Procrustes superimposition must be performed to all objects by optimally translating, rotating and uniformly scaling.

Suppose we have the set of p landmark points $\{(x_1, y_1), (x_2, y_2), ..., (x_p, y_p)\}$.

1. Translation:

The mean of these points is (x_{mean}, y_{mean}) where

$$x_{mean} = \frac{1}{p}\sum_{i=1}^{p} x_i \text{ and } y_{mean} = \frac{1}{p}\sum_{i=1}^{p} y_i$$

Translate these points so that their mean is the origin:

$$(x_i, y_i) \to (x_i - x_{mean}, y_i - y_{mean}), \forall i = 1 \to p$$

2. Uniform scaling

Scale the object so that the root mean square distance (RMSD) from the points to the translated origin is 1. RMSD is defined as:

$$s = \left(\frac{1}{p}\sum_{i=1}^{p}(x_i - x_{mean})^2 + (y_i - y_{mean})^2\right)^{1/2}$$

The scale becomes 1 when the point coordinates are divided by the object's initial scale:

$$(x_i, y_i) \to \Big((x_i - x_{mean})/s, (y_i - y_{mean})/s\Big), \forall i = 1 \to p$$

3. Rotation

Consider two objects composed of the same number of points with scale and translation removed, one of these objects can be used to provide a reference orientation. Let the points be $\{(x_1, y_1), (x_2, y_2), ..., (x_p, y_p)\}$ and $\{(u_1, v_1), (u_2, v_2), ..., (u_p, v_p)\}$. Fix the reference object and rotate the another around the origin, we need to find an optimum angle of rotation θ such that the

sum of the squared distances (SSD) between the corresponding points is minimized. Formula of θ is given by:

$$\theta = tan^{-1}\left(\frac{\sum_{i=1}^{p}(u_i y_i - v_i x_i)}{\sum_{i=1}^{p}(u_i x_i + v_i y_i)}\right)$$

Let $\{(x_i, y_i)\}$ be the reference set. The rotation of $\{(u_i, v_i)\}$ can be given by:

$$(u_i, v_i) \leftarrow (cos\theta u_i - sin\theta v_i, sin\theta u_i + cos\theta v_i), \forall i = 1 \rightarrow p$$

The Procrustes distance is defined as:

$$d = \left(\sum_{i=1}^{p}(u_i - x_i)^2 + (v_i - y_i)^2\right)^{1/2}$$

Here is an example of Procrustes rotation with a sample image of CK+ database. We take 10 landmark points including 4 eye corners, 4 eye-brow corners and 2 mouth corners. The yellow dot is the mean shape, the green dot is the before-rotation landmark and the red dot is the after-rotation landmark. We use the mean shape as a reference of rotation. After rotation, the face becomes frontal.

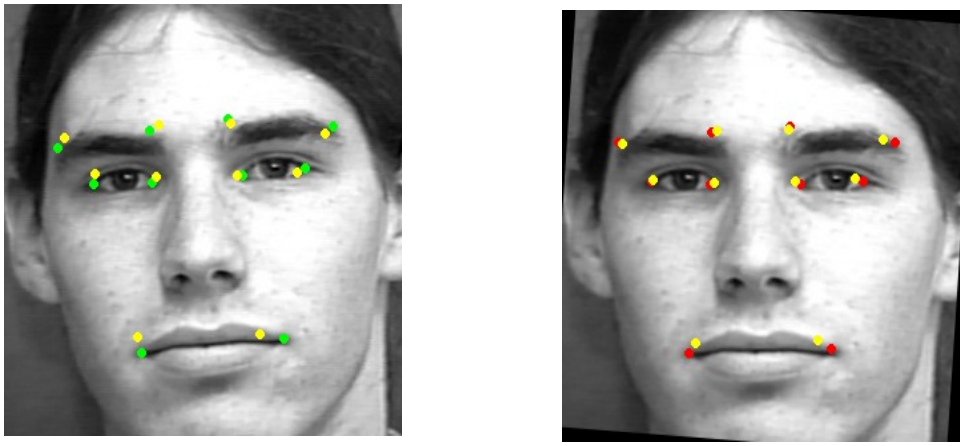


Figure 11. An example of 2D Procrustes rotation

References

[1] X. Xiong and F. De la Torre. *Supervised descent method and its applications to face alignment*. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 532–539, June 2013.

[2] G.B. Huang, Q.Y. Zhu, and C.K. Siew. *Extreme learning machine: Theory and applications*. Neurocomputing, vol. 70, no. 1–3, pp. 489–501, Dec. 2006.

[3] N.Y. Liang, G.B. Huang, P. Saratchandran, and N. Sundararajan. *A fast and accurate on-line sequential learning algorithm for feed-forward networks*. IEEE Trans, Neural Networks, vol. 17, no. 6, pp. 1411–23, Nov. 2006.

[4] A. van Schaik and J. Tapson. *On-line and adaptive pseudo-inverse solutions for ELM weights*. arXiv preprint arXiv:1405.7777, 2014.

[5] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory And Applications, Second*. New York: Springer, 2003.

[6] T. Greville. *Some applications of the pseudo-inverse of a matrix*. SIAM Rev., vol. 2, no. 1, pp. 15–22, 1960.

[7] P. Kovanic. *On the Pseudoinverse of a Sum of Symmetric Matrices with Applications to Estimation*. Kybernetika, vol. 15, no. 5, pp. 341–348, 1979.

[8] N. Dalal and B. Triggs. *Histograms of oriented gradients for human detection*. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 886893.

[9] Gross, R., Matthews, I., Cohn, J. F., Kanade, T., & Baker, S. *Multi-PIE*. Proceedings of the Eighth IEEE International Conference on Automatic Face and Gesture Recognition, 2008.

[10] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. *The Extended Cohn-Kande Dataset (CK+): A complete facial expression dataset for action unit and emotion-specified expression*. The Third IEEE Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010).

[11] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew Rosato. *A 3D Facial Expression Database For Facial Behavior Research*. The 7th International Conference on Automatic Face and Gesture Recognition (2006). IEEE Computer Society TC PAMI, April 10-12 2006, p211-216.

[12] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard. *BP4D-spontaneous: a high-resolution spontaneous 3D dynamic facial expression database*. Image and Vision Computing, 32 (10): 692 – 706, 2014. Best of Automatic Face and Gesture Recognition 2013.

[13] Y. Sugano, Y. Matsushita, and Y. Sato. *Learning-by-synthesis for appearance-based 3D gaze estimation*. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference, pages 1821–1828, June 2014.