



Eötvös Loránd University
Department of Informatics

Thesis:

Semi-supervised Adaptive Facial Tracking Method

Author:

Hy Truong Son

Supervisor:

Dr. Lőrincz András

Budapest, November 2014

Contents

Chapter I. Introduction	1
Chapter II. Methods	4
1. Supervised Descent Method	4
1.1. Newton's method	4
1.2. SDM	5
1.3. Pseudo-inverse solution of SDM	8
2. Extreme Learning Machine and OPIUM-Light	8
2.1. The standard ELM	8
2.2. Greville's method	10
2.3. OPIUM-Light	11
2.4. OPIUM-Dynamic	12
3. The adaptation method	13
3.1. Training process of supervised SDM	13
3.1.1. Initialization	13
3.1.2. Main program	14
3.2. Facial tracking with supervised SDM	16
3.3. Modified facial tracking	17
3.4. Adaptive facial tracking improved by unsupervised OPIUM-Light	18

Chapter III. Experiments and Results	21
1. Databases overview	21
1.1. The CMU Multi-PIE Face (Multi-PIE) Database	21
1.2. The Extended Cohn-Kanade (CK+) dataset	21
1.3. Binghamton University 3D Facial Expression (BU-3DFE) database	22
1.4. Binghamton–Pittsburgh 4D Spontaneous Expression (BP-4DSFE) Database	22
1.5. Multi-view gaze dataset	23
2. Results	23
2.1. Testing on BP-4DSFE database	23
2.2. Example of the original SDM	29
2.3. Example of the adaptive method	30
2.4. Testing on Multi-view gaze dataset	32
 Chapter IV. Future works and conclusion	 33
1. Future work	33
1.1. Higher order features	33
1.2. Facial tracking with larger angles of head pose	33
2. Conclusion	34
 Appendix A. Histogram of Oriented Gradients	 35
Appendix B. Procrustes Analysis	40
References.	42

Semi-supervised Adaptive Facial Tracking Method

Hy Truong Son
Eötvös Loránd University

Abstract

Facial tracking problem plays an important role in action unit detection and emotional classification. Previous approaches such as Active Appearance Model or Constrained Local Model have shown their limitation with unseen subjects and over-training problem. In this paper, we proposed the extension of Supervised Descent Method and the On-line Pseudo-Inverse Update Method OPIUM-Light in order to adapt and personalize unseen data with the absence of labeled ground truth from human interaction.

Chapter I. Introduction

Mathematical optimization methods for non-linear approximation becomes 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), first introduced by Xiong and De la Torre in the context facial alignment, is an optimization algorithm learning a sequence of descent directions that minimizes the mean of Non-linear Least Squares functions sampled at different points without computing the Jacobian nor the Hessian. Scale Invariant Feature

Transform (SIFT) descriptor was suggested. (*Reference to publication [1]*)

Extreme Learning Machine (ELM), since its invention by Huang and colleagues [2, 3], has become a widely used method for regression and classification of large and complex data. ELM is an emerging learning technique provides efficient unified solutions to generalized feed-forward net-works including single or multi-hidden-layer neural networks. The algorithm is based on a theory that hidden neurons are important but can be randomly generated and independent from applications. (*Reference to publication [4]*)

To simplify the incremental solution of the pseudo-inverse of a matrix developed by Edward Greville, André van Schaik and Jonathan Tapson introduced the On-line Pseudo-Inverse Update Method OPIUM-Light method which avoids the initial batch computation and update the pseudo-inverse after the appearance of a new instance with supervised manner for simplified ELM. (*Reference to publication [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 (eg. 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-Light 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

This summary of Newton's method is inherited from F. De la Torre's paper [1] and The Free Encyclopedia Wikipedia.

Newton's method attempts to construct a sequence x_n from an initial guess x_0 that converges towards x_* such that $f'(x_*) = 0$. This x_* is a stationary point of $f(\cdot)$. The second order Taylor expansion $f_T(x)$ of a function $f(\cdot)$ around x_n (where $\Delta x = x - x_n$) is:

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

and attains its extreme value when its derivative with respect Δx is equal to 0:

$$\lim_{\Delta x \rightarrow 0} \frac{f_T(x_n + \Delta x) - f(x_n)}{\Delta x} = 0 \Leftrightarrow f'(x_n) + f''(x_n)\Delta x = 0 \quad .$$

Thus, provided that $f(x)$ is a twice-differentiable function well approximated by its second order Taylor expansion and the initial guess x_0 is chosen close enough to x_* , the sequence (x_n) defined by:

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

will converge towards a root of f' , for example x_* for which $f'(x_*) = 0$.

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 $H_f(x)$:

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

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

The Newton's method has three main drawbacks:

- (1) The Newton steps might not be taken in the descent direction if the Hessian is not positive definite.
- (2) Newton's method requires the function to be twice differentiable. For instance, in many computer vision application, HOG descriptor is a non-differentiable image operator.
- (3) The computation of inverting the Hessian is expensive with the complexity $O(p^3)$ operations and $O(p^2)$ in space.

1.2. SDM

This session uses the mathematical notation system and some of the explanation which are credited to the F. De la Torre.'s paper [1].

Given an image $d \in \mathbb{R}^{m \times 1}$ 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 $x \in \mathbb{R}^{2p \times 1}$ 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 configuration 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 :

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

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:

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

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 :

$$x_{k+1}^i = x_k^i + R_{k+1} \phi_k^i - b_{k+1} \quad (10')$$

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

For mathematical 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, $\phi \in \mathbb{R}^{n+1}$ and $R_k \in \mathbb{R}^{2p \times (n+1)}$. The equation (10) can be rewritten as:

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

The new updating rule for x_k^i :

$$x_{k+1}^i = x_k^i + R_{k+1} \phi_k^i \quad (11')$$

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

1.3. Pseudo-inverse solution of SDM

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

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

where $Y_{k+1} = [\Delta x_{k+1}^1 \Delta x_{k+1}^2 \dots \Delta x_{k+1}^m] \in \Re^{2p \times m}$, $R_{k+1} \in \Re^{2p \times (n+1)}$ and $A_k = [\phi_k^1 \phi_k^2 \dots \phi_k^m] \in \Re^{(n+1) \times m}$ 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:

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

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-Light

2.1. The standard ELM

Some of the explanation in this session is inherited from A. van Schaik and J. Tapson.'s paper [4]. The mathematical notation system from the previous papers was changed and united in this thesis.

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

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

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, $g(\cdot)$ is the sigmoid function defined by

$g(x) = \frac{1}{1 + e^{-x}}$. Equation (14) can be rewritten in a matrix form:

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

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 :

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

From here, equation (15) becomes:

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

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

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

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

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

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:

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

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

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

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

Some of the explanation in this session is inherited from A. van Schaik and J. Tapson.'s paper [4]. The mathematical notation system from the previous papers was changed and united in this thesis.

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:

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

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

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

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

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

θ is the pseudo-inverse of the correlation matrix of the hidden layer activations. From (20)

and (21), we have:

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

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

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

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:

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

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 $C = I$:

$$\begin{aligned} \theta_t &= (\theta_{t-1}^{-1} + a^t I (a^t)^T)^+ = \theta_{t-1} - \frac{\theta_{t-1} a^t (a^t)^T \theta_{t-1}^T}{1 + (a^t)^T \theta_{t-1} a^t} \\ &\Leftrightarrow \theta_t = \theta_{t-1} - \theta_{t-1} a^t \frac{(a^t)^T \theta_{t-1}^T}{1 + (a^t)^T \theta_{t-1} a^t} \\ &\Leftrightarrow \theta_t = \theta_{t-1} - \theta_{t-1} a^t (b^t)^T \end{aligned} \quad (25)$$

To update the weights R_t :

$$\begin{aligned} R_t &= Y_t A_t^+ = [Y_{t-1} | y^t] \begin{bmatrix} A_{t-1}^+ (I - a^t (b^t)^T) \\ (b^t)^T \end{bmatrix} \\ &\Leftrightarrow R_t = Y_{t-1} A_{t-1}^+ (I - a^t (b^t)^T) + y^t (b^t)^T \\ &\Leftrightarrow R_t = Y_{t-1} A_{t-1}^+ - Y_{t-1} A_{t-1}^+ a^t (b^t)^T + y^t (b^t)^T \\ &\Leftrightarrow R_t = Y_{t-1} A_{t-1}^+ + (y^t - Y_{t-1} A_{t-1}^+ a^t) (b^t)^T \\ &\Leftrightarrow R_t = R_{t-1} + (y^t - R_{t-1} a^t) (b^t)^T \end{aligned} \quad (26)$$

2.3. OPIUM-Light

Some of the explanation in this session is inherited from A. van Schaik and J. Tapson.'s paper [4]. The mathematical notation system from the previous papers was changed and united in this thesis.

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 $c \in \mathfrak{R}$ 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} :

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

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

2.4. OPIUM-Dynamic

Some of the explanation in this session is inherited from A. van Schaik and J. Tapson.'s paper [4]. The mathematical notation system from the previous papers was changed and united in this thesis.

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 $A^+ = A^T (A A^T)^{-1}$ 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 $\alpha \in \mathfrak{R}$ 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, pre-processing 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 corresponding 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 $\{R_1, R_2, \dots, R_K\}$ that converges $x_0 \rightarrow x_*$ for each image of the training database is as follows:

For $k = 1 \rightarrow K$

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

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

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

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

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

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

landmark x_{k-1}^i

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

Compute the matrix $R_k = Y A^+ = Y A^T (A A^T)^{-1} \in \mathbb{R}^{2p \times (n+1)}$

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

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

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

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 $C = YA^T \in \mathbb{R}^{2p \times (n+1)}$ and $D = AA^T \in \mathbb{R}^{(n+1) \times (n+1)}$ 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 eyebrows (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 $k = 1 \rightarrow 2$ after the updating of new landmark positions. Let $\alpha^i = \text{angle}(x_k^i, x_0)$, $i = 1 \rightarrow m$ 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 $\{R_1, R_2, \dots, R_K\}$, 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 pre-processing mentioned in 3.1.1.

For $i = 1 \rightarrow m$

For $k = 1 \rightarrow K$

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

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

Update new landmark: $x_k^i = x_{k-1}^i + R_k \phi$

If $k = 1$ then $x_k^i = x_0 + R_k \phi$

End loop;

End loop;

Technical information: Remark that for $k = 1 \rightarrow 2$ 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:

$$x_0 \rightarrow R_1 \rightarrow x_1 \rightarrow R_2 \rightarrow \dots \rightarrow x_{K-1} \rightarrow R_K \rightarrow x_K$$

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 $i = 1 \rightarrow m$

For $j = 1 \rightarrow J$

For $k = 1 \rightarrow K$

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

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

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

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

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

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

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-Light

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 $\{R_1^{original}, \dots, R_K^{original}\}$ and the set of adaptive matrices $\{R_1^{adaptive}, \dots, R_K^{adaptive}\}$. 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 $\{R_3, \dots, R_K\}$ except 2 first matrices. The algorithm implementation has the below pseudo-code:

For $t = 1 \rightarrow \infty$

If $t \bmod \tau = 1$ 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 $\phi_{k-1,j} = [h(d^t(x_{K,j-1}^t)), -1]^T$

Update new landmark: $x_{k,j}^t = x_{k-1,j}^t + R_k^{adaptive} \phi_{k-1,j}$

If $k = 1$ and $j = 1$ then $x_{k,j}^t = x_0 + R_k^{adaptive} \phi_{k-1,j}$

If $k = 1$ and $j > 1$ then $x_{k,j}^t = x_{K,j-1}^t + R_k^{adaptive} \phi_{k-1,j}$

End loop;

End loop;

For $k = 3 \rightarrow K$

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

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

$R_k^{adaptive} = R_k^{adaptive} + (x_{K,J}^t - R_k^{adaptive} a) b^T$

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 dataset [13].

1.1. The CMU Multi-PIE Face (Multi-PIE) Database

The CMU Multi-PIE face database contains more than 750,000 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. In addition, high resolution frontal images were acquired as well. 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. (*Reference to publication [9]*)

1.2. The Extended Cohn-Kanade (CK+) dataset

The image data contains facial behavior of 210 adults. Participants were 18 to 50 years of age, 69% female, 81% European-American, 13% Afro-American, and 6% other groups. Participants were instructed by an experimenter to perform a series of 23 facial displays; these included single action units and combinations of action units. Each display began and ended in a neutral face with any exceptions noted. 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. (*Reference to publication [10]*)

1.3. Binghamton University 3D Facial Expression (BU-3DFE) database

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. Each subject performed seven expressions in front of the 3D face scanner. With the exception of the neutral expression, each of the six prototypic expressions (happiness, disgust, fear, angry, surprise and sadness) includes four levels of intensity. Therefore, there are 25 instant 3D expression models for each subject, resulting in a total of 2,500 3D facial expression models in the database. Associated with each expression shape model, is a corresponding facial texture image captured at two views (about +45 and -45 degree). As a result, the database consists of 2,500 two-view's texture images and 2,500 geometric shape models. (*Reference to publication [11]*)

1.4. Binghamton–Pittsburgh 4D Spontaneous Expression (BP-4DSFE) Database

The dataset 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. (*Reference to publication [12]*)

1.5. Multi-view gaze dataset

The dataset 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. (*Reference to publication [13]*)

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 average Root Mean Squares Error (RMSE) over the whole testing database of 41 subjects. RMSE between landmark $x \in \mathbb{R}^{2p \times 1}$ and the ground truth $x_* \in \mathbb{R}^{2p \times 1}$, where $p \in \{49, 14\}$ is the number of landmark points, is defined as follows:

$$RMSE(x, x_*) = \left\{ \frac{1}{p} \|x - x_*\|_2^2 \right\}^{1/2} = \left\{ \frac{1}{p} \sum_{i=1}^{2p} (x_i - x_{*i})^2 \right\}^{1/2}$$

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

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

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	<u>2.8463</u>	<u>2.8385</u>	<u>2.8591</u>
16	<u>3.4974</u>	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	<u>0.5830</u>	<u>0.5447</u>	<u>0.5508</u>	<u>0.5533</u>
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	<u>0.4638</u>	<u>0.4778</u>	<u>0.4780</u>
4	<u>0.4981</u>	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.

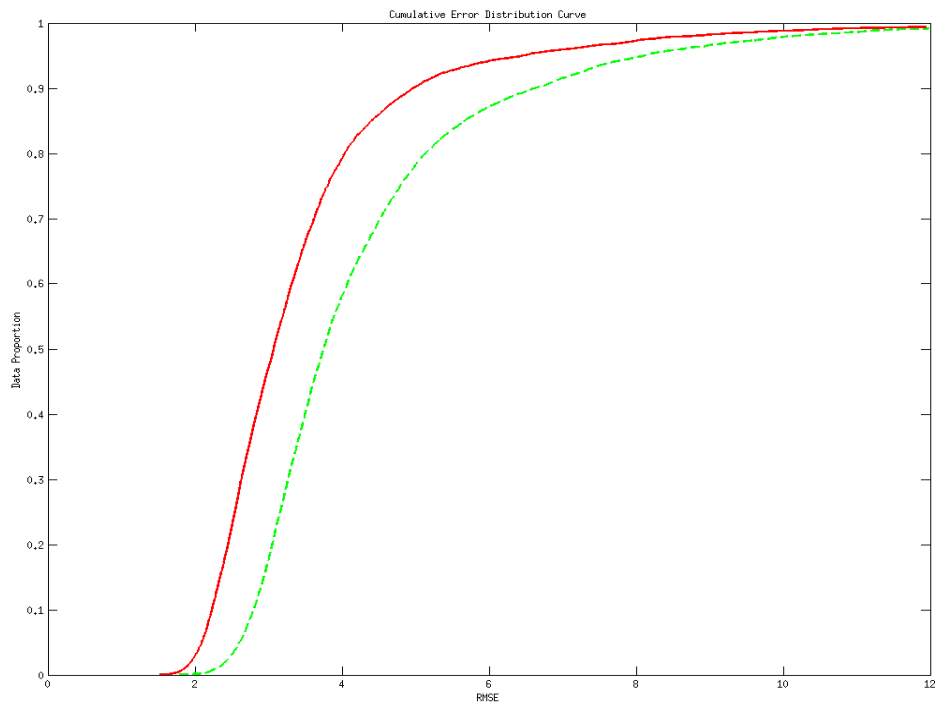


Figure 1. CED curve of 49 facial landmark points.

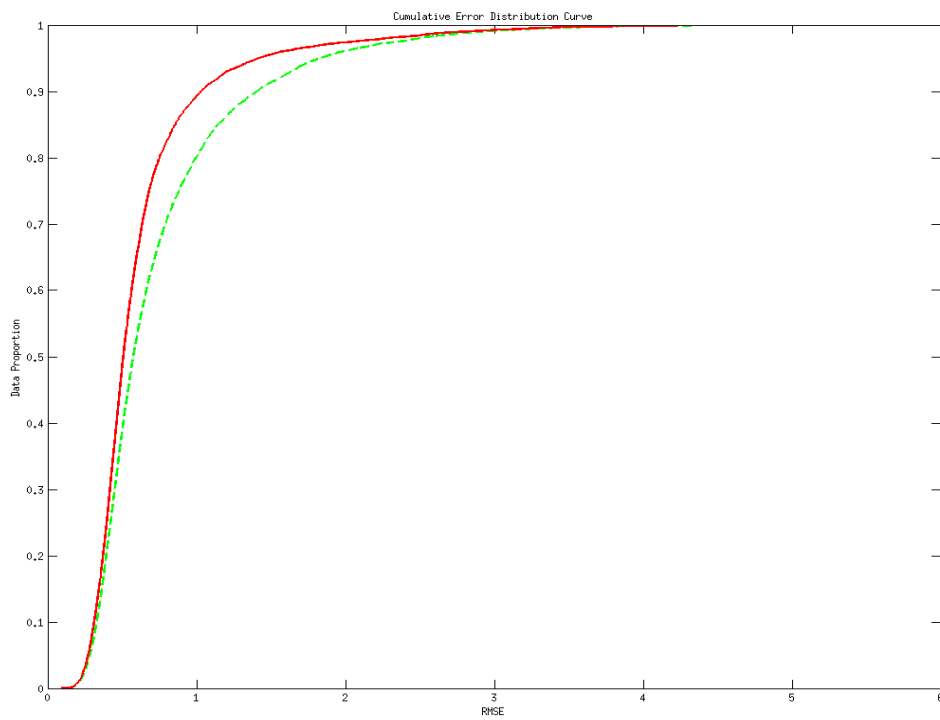


Figure 2. CED curve of 4 eye-corners landmark points.

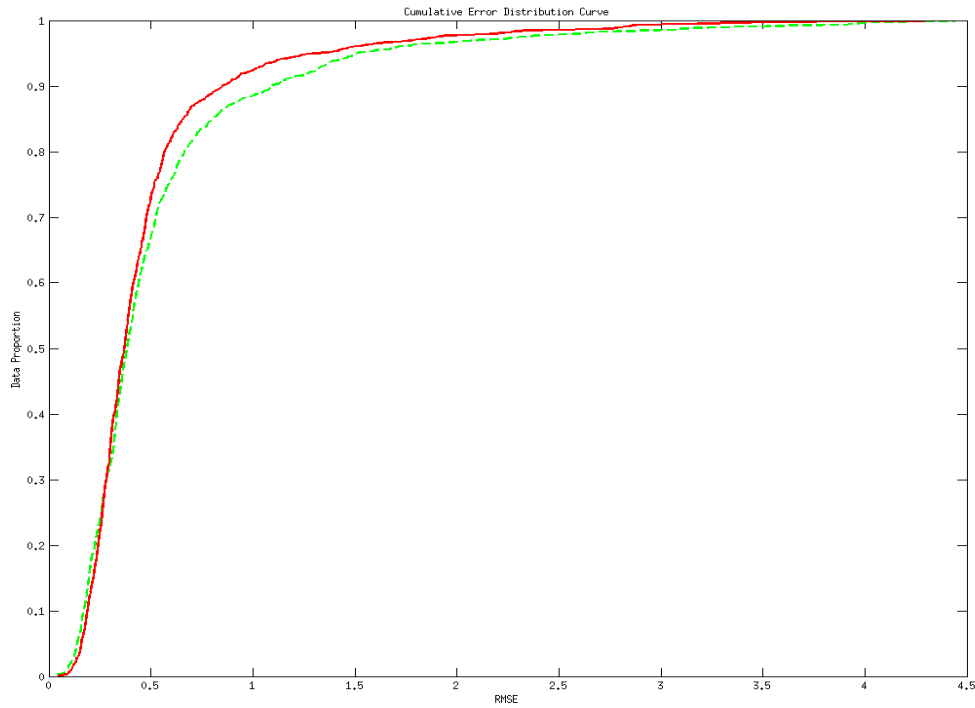


Figure 3. CED curve of 2 mouth-corners landmark points.

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

J	1	2	4	8
SDM	0.9528	1.0569	1.1078	1.1297
Adaptive	<u>0.8908</u>	0.9905	0.9836	1.0090

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

J	1	2	4	8
SDM	1.1279	0.9322	0.9596	0.9654
Adaptive	0.8658	0.7939	0.7753	<u>0.7718</u>

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

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 with one image from BP-4DSFE:

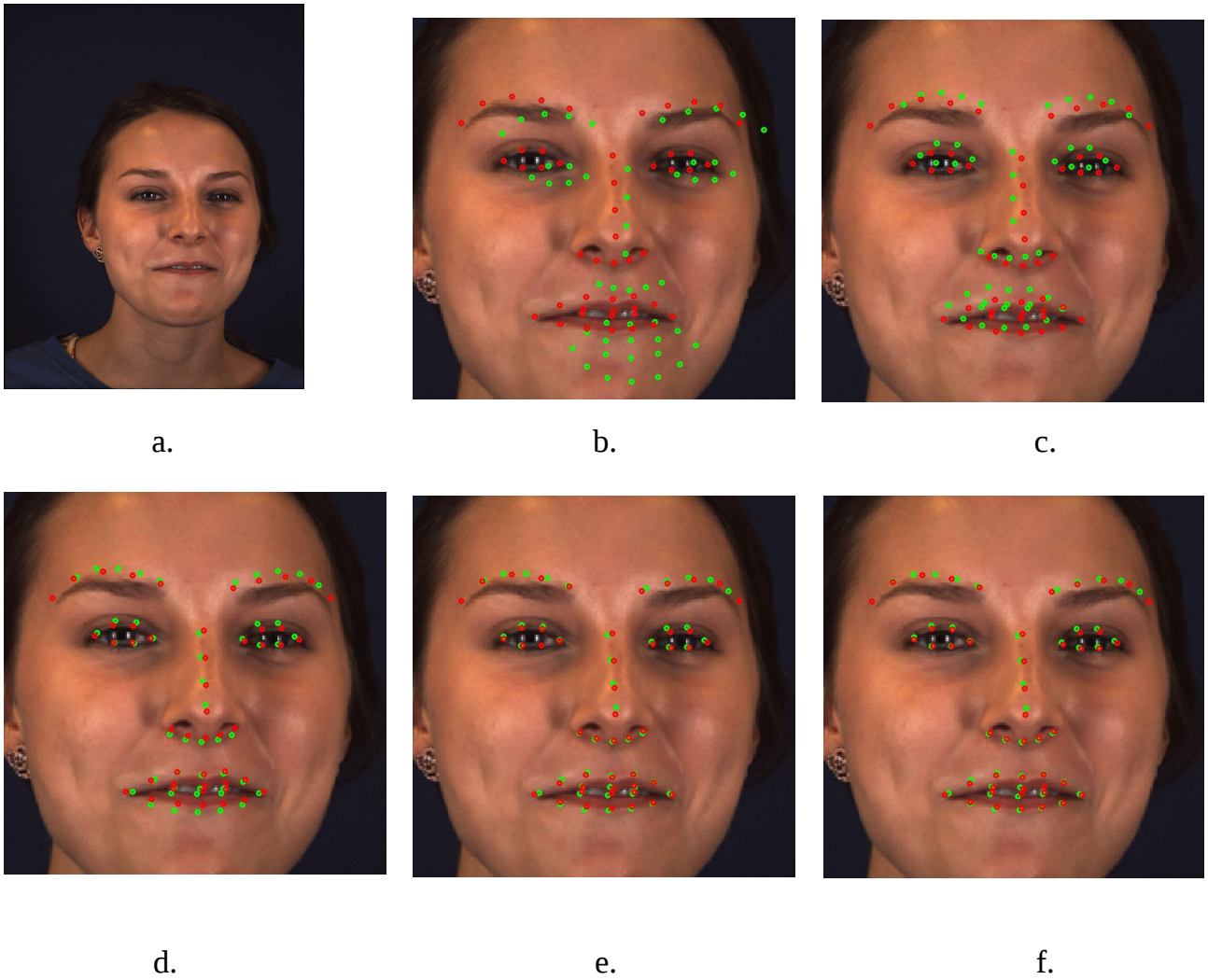


Figure 4.a. Input image

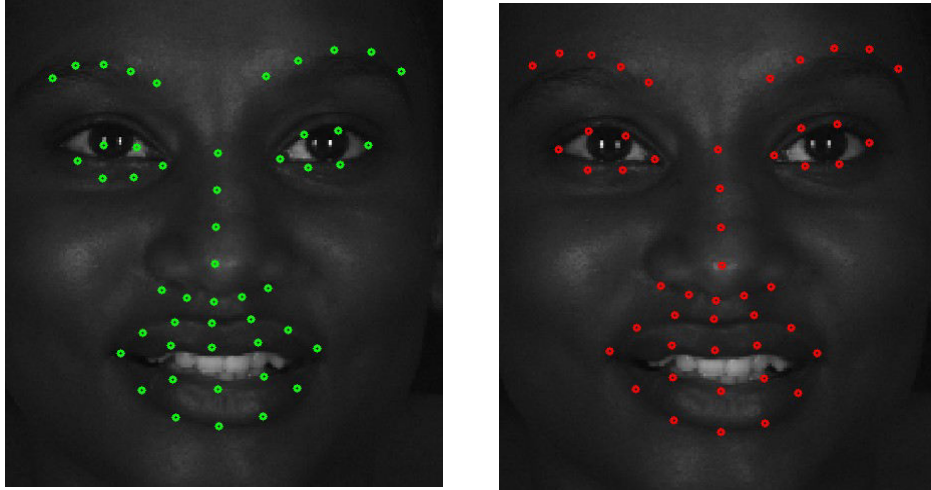
Figure 4.b. Mean shape x_0 (green) and the ground truth (red) in the canonical image

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

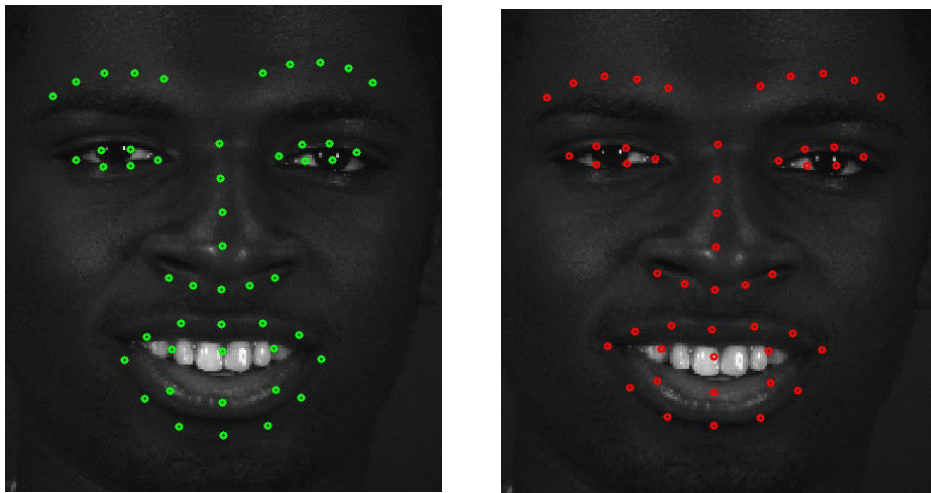
2.2. Examples of the adaptive method

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.



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

Figure 5 and Table 7. An 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

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

2.3. Testing on Multi-view gaze dataset

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 dataset. The green dots is the original SDM landmarks while the red dots represents the landmarks of adaptive method.

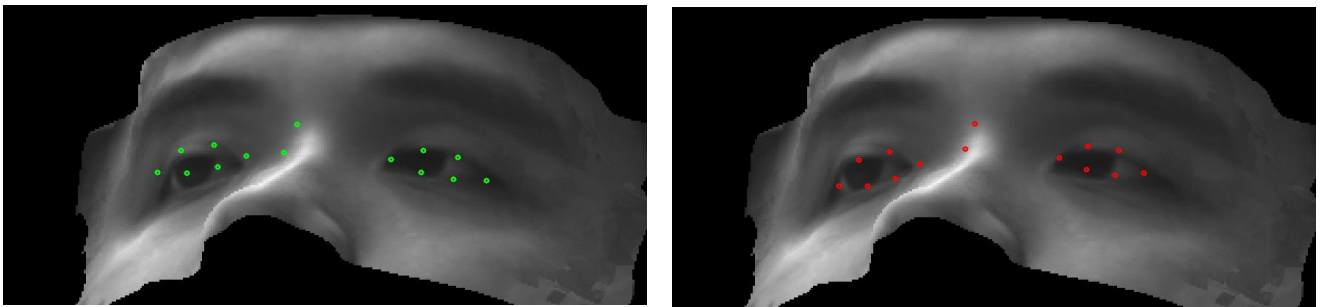


Figure 7. An example from Multi-view gaze dataset

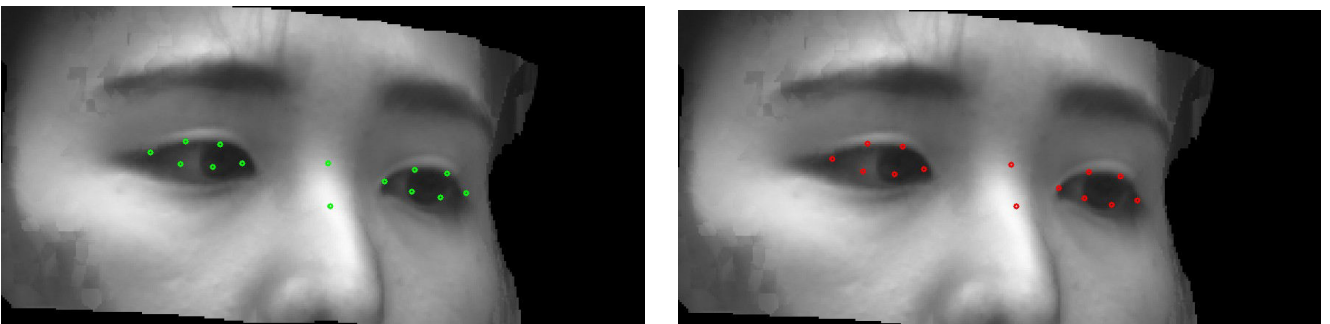


Figure 8. Another example from Multi-view gaze dataset

Chapter IV. Future work and conclusion

1. Future work

1.1. Higher order features

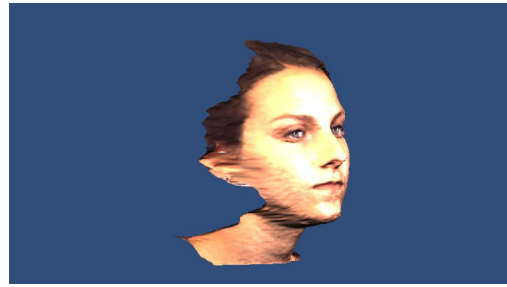
In this paper, we replaced the weights mapping from the input layer to hidden layer of a standard ELM by HOG descriptor and archived some promising results. To advance the ELM and OPIUM-Light, HOG features will be used as input signal for the input layer and a higher-order feature going through sigmoid 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.



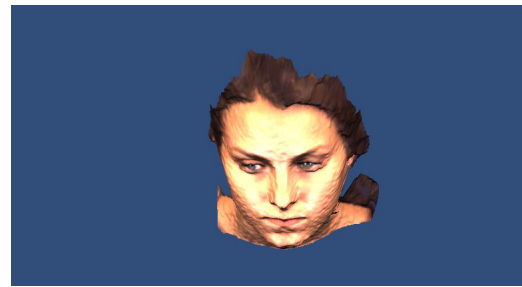
a. $yaw = 0^\circ$ and $pitch = 0^\circ$



b. $yaw = 45^\circ$ and $pitch = 0^\circ$



c. $yaw = 22.5^\circ$ and $pitch = 20^\circ$



d. $yaw = 0^\circ$ and $pitch = -30^\circ$

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

2. Conclusion

We extend 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 On-line Pseudo-Inverse Update Method OPIUM-Light algorithm and it opened up new ideas and directions for our future researches.

Appendix A. Histogram of Oriented Gradients

In this session, some of the content is related to Dalal and Triggs's paper [8] and the The Free Encyclopedia Wikipedia.

Histogram of Oriented Gradients (HOG) are feature descriptors used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image detection window, or region of interest (ROI). This method is similar to that of edge orientation histograms or scale-invariant feature transform (SIFT) descriptors, but differs in that it is computed on a dense grid or uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

The descriptor was first described by Dalal and Triggs in their June 2005 CVPR paper [8]. In this work they focused their algorithm on the problem of pedestrian detection in static images, although since then they expanded their results to include human detection in film and video, as well as to a variety of common animals and vehicles in static imagery.

The main idea behind the HOG descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The implementation of these descriptors can be achieved by dividing the image into small connected regions, called cells, and for each cell compiling a histogram of gradient directions or edge orientations for the pixels within the cell. The combination of these histograms then represents the descriptor. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination or shadowing.



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 HOG descriptor maintains a few key advantages over other descriptor methods. Since the HOG descriptor operates on localized cells, the method upholds invariance to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions. Moreover, as Dalal and Triggs discovered, coarse spatial

sampling, fine orientation sampling, and strong local photometric normalization permits the individual body movement of pedestrians to be ignored so long as they maintain a roughly upright position. The HOG descriptor is thus particularly suited for human detection in images.

The first step of calculation in many feature detectors in image pre-processing is to ensure normalized color and gamma values. However, this step can be omitted in HOG descriptor computation, as the descriptor normalization essentially achieves the same result. Image pre-processing thus provides little impact on performance. Instead, the first step of calculation is the computation of the gradient values. The most common method is to simply apply the 1-D centered, point discrete derivative mask. This means filtering the intensity data of the image with the following filter kernels:

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

Other, more complex masks were also tested, but they gave poorer performance in human image detection experiments. The Gaussian smoothing before filtering was also considered, but it was found that omission of any smoothing performed better in practice.

The second step of calculation involves creating the cell histograms. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. The cells themselves can either be rectangular or radial in shape, and the histogram channels are evenly spread over 0 to 360 degrees or 0 to 180 degrees, depending on whether the gradient is unsigned or signed. Dalal and Triggs found that unsigned gradients used in conjunction with 9 histogram channels performed best in their human detection experiments. As for the vote weight, pixel contribution is usually the gradient magnitude itself.

In order to account for changes in illumination and contrast, the gradient strengths must be locally normalized, which requires grouping the cells together into larger, spatially connected blocks. The HOG descriptor is then the vector of the components of the normalized cell histograms from all of the block regions. These blocks typically overlap, meaning that each cell contributes more than once to the final descriptor. Two main block geometries exist: rectangular R-HOG blocks and circular C-HOG blocks. R-HOG blocks are generally square grids, represented by three parameters: the number of cells per block, the number of pixels per cell, and the number of channels per cell histogram.

C-HOG blocks can be found in two variants: those with a single, central cell and those with an angularly divided central cell. In addition, these C-HOG blocks can be described with four parameters: the number of angular and radial bins, the radius of the center bin, and the expansion factor for the radius of additional radial bins.

It was shown that block normalization gives very significant improvement over the non-normalized data. Dalal and Triggs explored four different methods for block normalization. The normalization is as follows. 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 (the exact value, hopefully, is unimportant). Then the normalization factor can be one of the following:

$$L2\text{-norm: } f = \frac{v}{(\|v\|_2^2 + e^2)^{1/2}},$$

$L2\text{-hys}$: $L2$ -norm followed by clipping (limiting the maximum values of v to 0.2) and re-normalizing,

$$L1\text{-norm: } f = \frac{v}{\|v\|_1 + e},$$

$$L1\text{-sqrt: } f = \sqrt{\frac{v}{\|v\|_1 + e}}.$$

To sum up, implementation of the HOG descriptor algorithm is as follows:

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 angular bins according to the gradient orientation.
3. Each cell's pixel contributes weighted gradient to its corresponding angular bin.
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

In this session, some English phrases are inherited from The Free Encyclopedia Wikipedia.

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 $\{(x_1, y_1), (x_2, y_2), \dots, (x_p, y_p)\}$.

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:

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

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:

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

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 $\{(x_1, y_1), (x_2, y_2), \dots, (x_p, y_p)\}$ and $\{(u_1, v_1), (u_2, v_2), \dots, (u_p, v_p)\}$. 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.

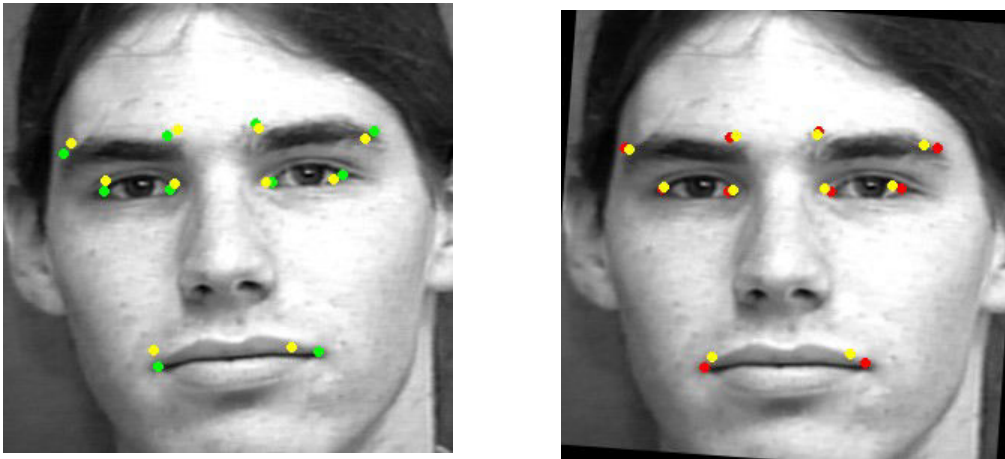


Figure 11. An example of 2D Procrustes rotation

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.