ICIC Express Letters
Part B: Applications
Volume **7**, Number **10**, October **2016**

ICIC International ©2016 ISSN 2185-2766
pp. 2197–2203

ORB FEATURE BASED NEIGHBOR GRAPH CONSTRUCTION METHOD FOR GRAPH REGULARIZED NON-NEGATIVE MATRIX FACTORIZATION

Yuhao Liu

College of Information Engineering
Shanghai Maritime University
No. 1550, Haigang Ave., Shanghai 201306, P. R. China
liuyuhaos@126.com

Received April 2016; accepted July 2016

Abstract. *The ORB Feature is a fast binary descriptor based on BRIEF, which is rotation invariant and resistant to noise. In this paper, we introduce the ORB key point detector and descriptor to construct the neighbor graph for Graph regularized Non-negative Matrix Factorization. Due to the ORB Feature, distinguishable local feature can be extracted, which is helpful to extract more accurate neighbor information, expecting more reasonable neighbor graph. We have selected five candidate functions to compute the weight W_{ij} and chose the best function which achieved the best performance. Experiments show that ORB Feature based Graph regularized Non-negative Matrix Factorization achieves better performance. The Exponential and Linear functions performed better when the train number k is middle sized and the Logarithmic performed better when the train number k is small or large.*
Keywords: ORB Feature, Face recognition, Manifold, Non-negative Matrix Factorization (NMF)

1. **Introduction.** Face recognition is very challenging because of noise of image and rotation of face [1]. The rotation of face will lead to the same feature that two images cannot be the same place, so, local feature is needed to extract from the original images. There are many key point detector and descriptor, such as SIFT [2], SURF [3], BRIEF [4] and ORB [5].

The FAST keypoint detector [6] and BRIEF [4] descriptor are both well-known keypoint detectors. ORB Feature is built on FAST and BRIEF by introducing orient, which is named as Oriented FAST and Rotated BRIEF (ORB for short). ORB Feature is a newly proposed feature extraction method, which is rotation invariant and resistant to noise. ORB Feature extraction method is computationally-efficient, and experiments [5] show that ORB is at two orders of magnitude faster than SIFT while performing as well. So in this paper, we chose ORB Feature extraction method to our newly proposed Neighbor Graph Construction method.

Non-negative Matrix Factorization (NMF) is a widely-used method for low-rank approximation of a nonnegative matrix (matrix with only nonnegative entries), where non-negative constraints are imposed on factor matrices in the decomposition. NMF [8] was firstly proposed by Lee and Seung in Nature, and later they gave a detailed discussion in NIPS [9]. The NMF method was shown to be able to generate sparse representations of data. There are large bodies of past work on NMF [7], and most of them focuses on introducing an additional parameter that balances the reconstruction and other constraints. He and Niyogi [10] proposed the famous Locality Preserving Projections algorithm (LPP) which introduced a nonlinear manifold embedded into original data space. Cai et al. [11] proposed GNMF which introduced manifold into NMF, and GNMF constructed a weight matrix W to encode the geometrical information and sought a matrix factorization which respects the graph structure.

In this paper, we introduce the ORB key point detector and descriptor to construct the neighbor graph for Graph regularized Non-negative Matrix Factorization, and due to the fact that ORB Feature is rotation invariant, we can extract distinguishable local feature from face image with rotation, which is helpful to extract more accurate neighbor information, expecting more reasonable neighbor graph. We have selected five candidate functions to compute the weight W_{ij} and chose the best function which achieved the best performance, expecting a better recognition rate than GNMF. The remainder of this paper is organized as follows: in Section 2, we give a brief review of ORB key point detector and descriptor; in Section 3, ORB Feature based Neighbor Graph Construction method is introduced; comparison and experiment are presented in Section 4; finally, we will give some conclusions in Section 5.

2. **An Introduction to ORB.** ORB Feature [5] is a newly proposed feature extraction method, which is rotation invariant and resistant to noise. ORB Feature extraction method makes use of the FAST keypoint orientation method to find the keypoint, and Rotation-Aware Brief method to create feature descriptors. Because ORB Feature is built on FAST and BRIEF while introducing orient, ORB Feature extraction method is computationally-efficient, and experiments [5] show that ORB is at two orders of magnitude faster than SIFT while performing as well.

The algorithm detects FAST (FAST-9) points in each image [6]. FAST algorithm does not consider the orientation, so a simple but effective measure of corner orientation (the intensity centroid [12]) is used. We can construct a vector from the corners center O to the centroid $\overrightarrow{OC}$, and the orientation of the patch can be computed as Equation (1):

$$\theta = a\tan 2(m_{10}, m_{01}) \tag{1}$$

where $a\tan 2$ stands for the quadrant-aware version of arctan.

After selecting the keypoints, Rotation of the BRIEF Operator is performed to create feature descriptors. As the algorithm is sensitive to the noise, smooth is performed. Consider a smoothed image patch P, and a binary test τ is defined by:

$$\tau(P; x, y) := \begin{cases} 1 & P(x) < P(y) \\ 0 & P(x) \geqslant P(y) \end{cases} \tag{2}$$

where $P(x)$ is the intensity of P at a point x. The feature is defined as a vector of n binary tests:

$$f_n(P) := \sum_{1 \leqslant i \leqslant n} 2^{i-1} \tau(P; x, y) \tag{3}$$

where Gausian distribution is used to select points around the center of the patch P, and the vector length $n = 256$, which means each descriptor is 256 bit (32 Bytes) length.

In order to allow BRIEF to be invariant to in-plane rotation, a more efficient method is to steer BRIEF according to the orientation of keypoints used. Figure 3(a) and Figure 3(b) show some matched keypoints for the examples of ORB Feature for two images. From Figure 3(a) and Figure 3(b) we can see that ORB Feature extraction method is rotation invariant, and most of the keypoints come from the eyes and nose, while few come from the mouth, which is close to the human being vision: we can recognize one face from eyes and nose easily, but we cannot recognize one face from mouth. We also need to notice that the shape of eyes and nose is stable, while the shape of mouth could be changed if we talk.

3. **ORB Feature Based Neighbor Graph Construction Method for GNMF.** In this section, we will introduce ORB Feature into the neighbor construction method of GNMF, and then we will give a new weight graph computing method.

GNMF is a well known Graph based NMF method which is introduced by Cai et al. [11]. Given a non-negative input matrix $X = [x_1, x_2, \cdots, x_m] \in \mathbb{R}_+^{n\times r}$, each column of X denotes an n-dimensional facial image, and $\mathbb{R}_+^{n\times m}$ stands for space of non-negative $n \times m$ matrices. The update rules can be deduced as Equation (4) (for detail of the GNMF, please refer to [11]):

$$\begin{cases} U_{ij} = U_{ij} \cdot \frac{\left(XV^T\right)_{ij}}{(UVV^T)_{ij}} \\ V_{ij} = V_{ij} \cdot \frac{\left(V^TX\right)_{ij} + \lambda(VW)_{ij}}{(V^TUV)_{ij} + \lambda(VD)_{ij}} \end{cases} \tag{4}$$

L is called graph Laplacian, where $L = D - W$. D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ij}$, and W is the Neighbor Graph weight matrix which encodes the geometrical information of the data X.

The key to GNMF is the weight matrix construction method, He and Niyogi [10] and Cai et al. [11] gave us two easy ways to construct the weight matrix without knowledge as Equation (5) and Equation (6):

$$W_{ij} = \begin{cases} 1 & \text{if } x_i \in N(x_j) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$W_{ij} = \begin{cases} \exp\left(-||X_i - X_j||^2/t\right) & \text{if } x_i \in N(x_j) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $x_i \in N(x_j)$ denotes the images x_i and x_j belong to the same class. If we do not own the classified information belonging to the database, we need to 'guess' the neighbor information of x_i. Usually, Euclidean distance is performed as $D = ||x_i - x_j||^2$, and then we need to choose p image which is the closest to x_i, ('close' means the Euclidean distance is small).

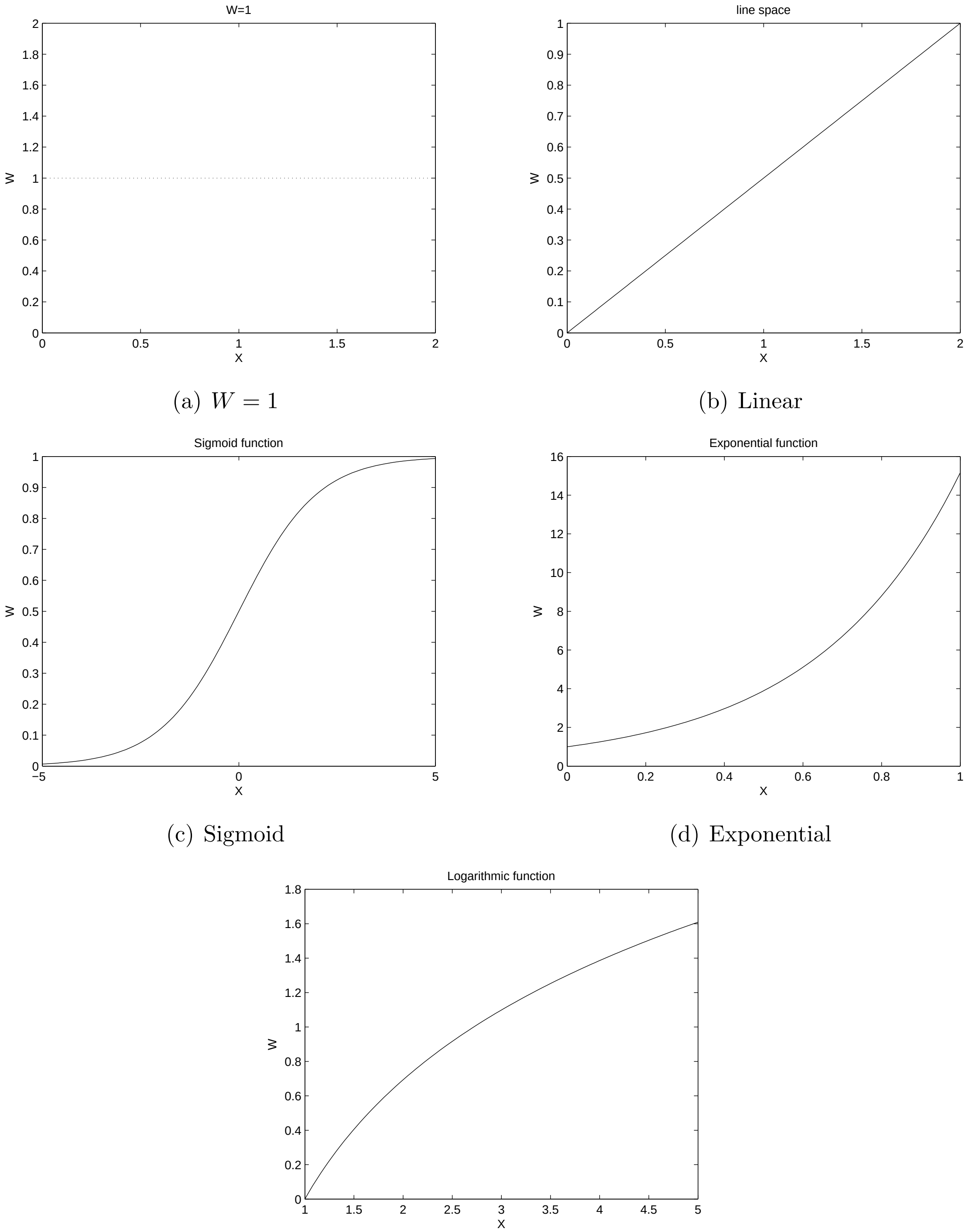
From Section 2, we can see that the ORB descriptor is a bit string description with its length as 256 bits (32 Bytes) for each keypoint. Different face images have different sizes of keypoints, so it is not easy to compute neighbor information of two descriptor sets because the corresponding keypoint for two images cannot be in the same place.

So we need to count the matched keypoints for two images intending the Euclidean distance, and then the weight matrix of W_{ij} is computed. The pseudocode of computing the weight matrix of W_{ij} based on matching keypoints of two descriptor sets is as follows.

Algorithm 1: compute the weight matrix of W_{ij}
Input: descriptors belonging to the face images C_1 and C_2, the threshold value γ.
Output: the number of the matching keypoints N_{match}.
Initial: $N_{ij} = 0$, $W_{ij} = 0$
Process: For each descriptor Q_i^{C1} from the face image $C1$ and for each descriptor Q_j^{C2} from the face image $C2$.
a. compute the Hamming distance between X_i^{C1} and X_j^{C2}.
b. choose the minimum Hamming distance $D_{\min}$, and the second minimum Hamming distance $D_{\min 2}$.
c. if $D_{ij}^{\min} < 0.8 \times D_{ij}^{\min 2}$ and $D_{ij}^{\min} < \gamma$, then $N_{ij} = N_{ij} + 1$.
d. compute the W_{ij} as $W_{ij} = f(N_{ij})$.

The Hamming distance is the number of positions in two strings of equal length for which the corresponding elements are different. The reason why we need to compare the minimum distance and the second minimum distance is that we assume each keypoint only has one corresponding keypoint from other images, so if the minimum distance and the second minimum distance are too close, it means that the keypoint has more than one corresponding keypoint, and the keypoints may be the common keypoints for different images, which will lead to missing match and is helpless to classify, so we omit that keypoints.

The function $f(N_{ij})$ will affect the recognition rate, so let us define the $f(N_{ij})$. Notice the measurement of the distance for two images is the sum of matching keypoint instead of the Euclidean distance, so we need to select new measurement to substitute the head kernel. So in this paper we select five candidate functions and we will perform several



(a) $W = 1$

(b) Linear

(c) Sigmoid

(d) Exponential

(e) Logarithmic

FIGURE 1. The example of the selected functions

experiments to show which is better. The image of the above five candidate functions is shown in Figure 1.

(1) $W_{ij} = \begin{cases} 1 & \text{if } x_i \in N(x_j) \\ 0 & \text{otherwise} \end{cases}$

This method is considered as set 1 for simple.

(2) $W_{ij} = \begin{cases} N_{match}/\max & \text{if } x_i \in N(x_j) \\ 0 & \text{otherwise} \end{cases}$

This method is considered as a Linear function, and max is the maximum in all N_{match}.

(3) $W_{ij} = \begin{cases} 1/[1+\exp(-(N_{match} - mid))] & \text{if } x_i \in N(x_j) \\ 0 & \text{otherwise} \end{cases}$

This method is considered as the Sigmoid function, which will separate the original matching number close to 0 (if smaller than middle) or 1 (if larger than middle), and mid is middle of all N_{match}.

(4) $W_{ij} = \begin{cases} \exp(N_{match}/\max) - 1 & \text{if } x_i \in N(x_j) \\ 0 & \text{otherwise} \end{cases}$

This method is considered as the Exponential function which will increase the middle zone of the original matching number.

(5) $W_{ij} = \begin{cases} \ln(e \times N_{match}/\max + 1) & \text{if } x_i \in N(x_j) \\ 0 & \text{otherwise} \end{cases}$

This method is considered as the Logarithmic function which will suppress the middle zone of the original matching number.

4. **Experiments.** In this section, we will perform a series of experiments to show the improvements of ORB Feature based Neighbor Graph Construction method. Our experiments were performed on the GT [13] database. The Nearest Neighbor (NN) classifier was used for all face recognition experiments, all the experiments were performed 10 times, and the mean recognition rates are recorded. In Georgia Tech (GT) face database [13], there are totally 50 people, and all people in the database are represented by 15 color images with cluttered background taken at resolution 640×480 pixels. In our experiments, original images are normalized such that the two eyes were aligned at the same position, then, the images are cropped, we will extract the ORB Feature and the Weight matrix from the cropped images, and we resize the cropped images into 31×23 in order to reduce the cost. Figure 2 shows some examples of the GT database.

From Figure 2 we can see that there are many rotated faces, which is unsuitable for Euclidean distance to construct the Weight matrix, because Euclidean distance is a pixel-level method, so it is sensitive to the rotated face because slight rotation will lead to that the pixel moved another point, while the ORB method is feature based and is rotated-insensitive. Figure 3 gives an example of two images belonging to the same class matching with ORB method. From Figure 3 we can see that, in Subfigures 3(a) and 3(b), both images come from one face, and we can see that there are much more matched keypoints. In Subfigure 3(a), there are two much different in the Euclidean distance because the left

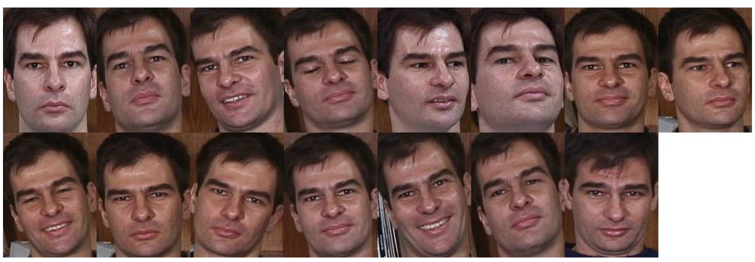
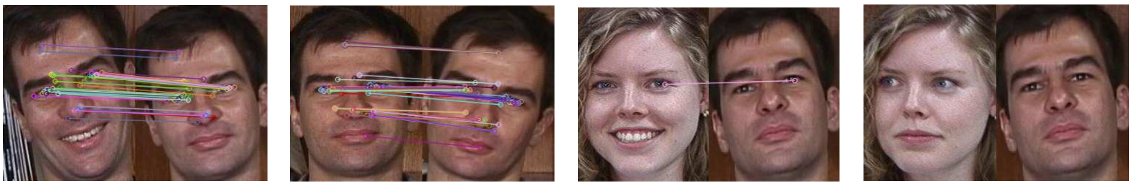


FIGURE 2. The examples of the GT database



(a) (b) (c) (d)

Figure 3. The example of the matched keypoints in two images

image laughs while the right image is a neutral, but we can see that with the help of the ORB method, we can extract many keypoints from two images. The similar situation goes for Subfigure 3(b). From Subfigure 3(c) and Subfigure 3(d), we can see that if two images come from different people, there are few matched keypoints (only one in Subfigure 3(c)), even no matched keypoints (such as in Subfigure 3(d)).

In this experiment, we select GNMF under the Neighbor Graph Construction method with ORB along with traditional GNMF algorithms. We have introduced five candidate weight functions mentioned in Section 3, so we will perform all the methods to select the best. There are three ways to construct the neighbor graph for traditional GNMF, (set $W_{ij} = 1$ with knowledge, set $W_{ij} = 1$ without knowledge, and heart kernel function). Generally speaking, GNMF with knowledge should be the best one, because it has the most accurate information. Table 1 shows the mean recognition rate of different methods on the GT database.

Table 1. The mean recognition rate (%) of different methods on the GT database (k is the train number for each class)

method / k	ORBGNMF -set1	ORBGNMF -Linear	ORBGNMF -Sigmoid	ORBGNMF -Exp	ORBGNMF -Ln	GNMF -set1	GNMF-heartKernal	GNMF-with Knowledge
5	44.8	63	58.6	63	**64**	57.4	57.6	55.4
7	53.8	64	60	**67.3**	59.5	50.3	62.8	62
9	53.3	**69.7**	66.3	69.3	66.3	55.7	61.7	63.7
11	49.5	69.5	61.5	**70**	68	56.5	63.5	69.5
13	49	65	66	65	**69**	53	63	69

From Table 1 we can see that the recognition rate for set $W_{ij} = 1$ method with ORB performed the worst, and the reason is obvious: from Figure 3 we can see that both images belonging to the same class (Subfigures 3(a) and 3(b)) have much more matched keypoints, while in Subfigure 3(c), there is only one matched keypoint. If we set $W_{ij} = 1$, the more matched image pairs have the same weight with the few matched image pairs, which will introduce much more confusion. We can also see that the recognition rate under Exponential and Linear functions play better under different sizes of k. The recognition rate under Logarithmic function is sensitive to the parameter k, it performed best of all when $k = 5$ and $k = 13$, but it performed worse than Linear and Exponential functions under the middle size of k. The reason we guess is that the middle size of the matched numbers plays an important role for classification, and the Logarithmic function suppresses the middle size number which will lead to worse performance.

The GNMF with ORB under Sigmoid function performed better than set $W_{ij} = 1$ method, and that under Logarithmic function did worse than Linear and Exponential functions. We also see that the GNMF with knowledge does not always perform the best, and the reason we guess is that the set $W_{ij} = 1$ method would enlarge the far distance images belonging to the same class, which will lead to slight over fitting. From the five candidate functions we can see that the Exponential and Linear functions should be chosen if we want to get the best performance.

5. **Conclusions.** In this paper, we introduce the ORB key point detector and descriptor to construct the neighbor graph for Graph regularized Non-negative Matrix Factorization. Due to the ORB Feature, distinguishable local feature can be extracted from face images even there is rotation, which is helpful to Neighbor Graph Construction. We have selected five candidate functions to compute the weight W_{ij} and chose the best one. Experiments show that ORB Feature based Graph regularized Non-negative Matrix Factorization achieves better performance. The Exponential and Linear functions performed better when the train number k is middle sized and the Logarithmic performed better when the train number k is small or large. As the ORB Feature is rotation invariant, our future work will try to introduce some illumination insensitive method into ORB based GNMF, expecting illumination insensitivity.

Acknowledgment. This work is partially supported by the Youth Teacher Training Program of Shanghai University; Natural Science Foundation of China (Grant No. 61303099); Shanghai Municipal Natural Science Foundation under Grant No. 13ZR1455600. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] W. Zhao, R. Chellappa, P. Phillips and A. Rosenfeld, Face recognition: A literature survey, *ACM Computing Surveys*, vol.35, no.4, pp.399-458, 2003.
[2] D. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, vol.60, no.4, pp.91-110, 2004.
[3] H. Bay, T. Tuytelaars and L. Gool, Surf: Speeded up robust features, *Computer Vision & Image Understanding*, vol.110, no.3, pp.404-417, 2006.
[4] M. Calonder, V. Lepetit, C. Strecha and P. Fua, Brief: Binary robust independent elementary features, *Computer Vision*, pp.778-792, 2010.
[5] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, ORB: An efficient alternative to SIFT or SURF, *Computer Vision*, pp.2564-2571, 2011.
[6] E. Rosten, R. Porter and T. Drummond, Faster and better: A machine learning approach to corner detection, *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol.32, no.1, pp.105-119, 2010.
[7] Y. Wang and Y. Zhang, Non-negative matrix factorization: A comprehensive review, *IEEE Trans. Knowledge and Data Engineering*, vol.25, no.6, pp.1336-1353, 2013.
[8] D. Lee and H. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, vol.401, no.6755, pp.788-791, 1999.
[9] D. Lee and H. Seung, Algorithms for nonnegative matrix factorization, *NIPS*, pp.556-562, 2001.
[10] X. He and P. Niyogi, Locality preserving projections, *NIPS*, vol.16, 2004.
[11] D. Cai, X. He, X. Wu and J. Han, Non-negative matrix factorization on manifold, *Proc. of IEEE Int. Conf. Data*, pp.63-72, 2008.
[12] P. Rosin, Measuring corner properties, *Computer Vision and Image Understanding*, vol.73, no.2, pp.291-307, 1999.
[13] *Georgia Tech Face Database*, http://www.anefian.com/research/face_reco.htm, 2012.