Automatic Facial Pose Determination of 3D Range Data for Face Model and Expression Identification

Xiaozhou Wei, Peter Longo, and Lijun Yin

Department of Computer Science,
State University of New York at Binghamton, Binghamton, NY

Abstract. Many of the contemporary 3D facial recognition and facial expression recognition algorithms depend on locating primary facial features, such as the eyes, nose, or lips. Others are dependent on determining the pose of the face. We propose a novel method for limiting the search space needed to find these "interesting features." We then show that our algorithm can be used in conjunction with surface labeling to robustly determine the pose of a face. Our approach does not require any type of training. It is pose-invariant and can be applied to both manually cropped models and raw range data, which can include the neck, ears, shoulders, and other noise. We applied the proposed algorithm to our created 3D range model database, the experiments show the promising results to classify individual faces and individual facial expressions.

Keywords: Surface Normal Difference, Facial Pose Detection; 3D range model.

1 Introduction

The facial pose estimation is a first critical step towards developing a successful system for both face recognition [21] and facial expression recognition [15]. The majority of existing systems for facial expression recognition [21, 12] and face recognition [13, 14, and 18] operate in 2D space. Unfortunately, 2D data is unsatisfactory because it is inherently unable to handle faces with large head rotation, subtle skin movement, or lighting change with varying postures. With the recent advance of 3D imaging systems [23, 11], research on face and facial expression recognition using 3D data has been intensified [8, 9, 10, 16, 19, and 22]. However, almost all existing 3D-based recognition systems are based on *static* 3D facial data. We are interested in researching the face and facial expression recognition in a *dynamic* 3D space.

One of the prerequisites of dynamic 3D facial data analysis is to design an algorithm that can robustly determine a face's pose. Previously, a number of methods have been proposed for determining the pose of a face in 2D and 3D space. Most can be broadly categorized as being either feature-based [2] or appearance-based [6]. The feature-based methods attempt to relate facial pose to the spatial arrangements of significant facial features. The appearance-based methods consider the face in its entirety [5]. Recently, approaches have been developed that combine feature-based and appearance-based techniques [8, 9], the results are very encouraging. Some

S.-W. Lee and S.Z. Li (Eds.): ICB 2007, LNCS 4642, pp. 144–153, 2007.

Support Vector Regression based approaches [3] have shown impressive results, but they require the use of a training set.

In this paper, we describe a pose and expression invariant algorithm that can robustly determine a face's pose. We have tested our algorithm on preprocessed, manually cropped 3D facial models and on unprocessed raw 3D data coming directly from our dynamic 3D imaging system.

The general framework of our approach is outlined in Figure 1. The first step is to remove the image's boundary since there is no guarantee that it is smoothly cropped. Then, we apply our novel Surface Normal Difference (SND) algorithm, which produces groups of triangles. The groups containing the fewest triangles are ignored, and the triangles in the other groups are labeled as "potentially significant." The Principal Component Analysis (PCA) algorithm is run on the vertices of the "potentially significant" triangles in order to align the model in the Z direction and determine the location of the nose tip. Finally, we label the very concave "potentially significant" triangles as "significant," and use the resulting groups, as well as the symmetry property of the face, to find the nose bridge point. At final, we evaluate the proposed algorithms through the experiments on our developed systems of dynamic 3D face recognition and dynamic 3D facial expression recognition. Each part of our framework will be elaborated on in the following sections.

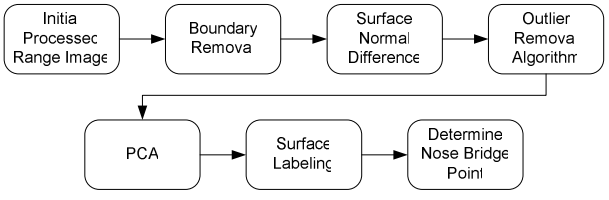

Fig. 1. Pipeline for determining the pose of a 3D facial range model

2 Surface Normal Difference (SND)

Let us define a spatial triangle as a tuple of three vertices. Assume a triangle t = (v1, v2, v3), which has a three dimensional normal vector, n_t, consisting of x, y, and z components. Assume a triangle s = (s1, s2, s3). We call s a "neighbor of" t if the sets {v1, v2, v3} and {s1, s2, s3} are not disjoint.

$$neighbor \quad (s,t) \longrightarrow \{v1, v2, v3\} \cap \{s1, s2, s3\} \neq 0$$

Assume a set of triangles, N, which contains all of t's neighboring triangles. For each triangle, u with normal n_u, in N, we determine the angle, θ_{tu}, between the normal vectors of t and u.

$$\theta_{tu} = \cos^{-1}((n_t \bullet n_u) / (|n_t| \times |n_u|))$$

We determine the maximum value of θ_{tu} and call it θ_{max}. If θ_{max} is greater than a specified angular tolerance, δ, we add t to set G. Otherwise, we add t to set L.

We repeat this procedure for all triangles in the facial mesh. Upon completion, we label the triangles in G as "potentially significant" and the triangles in L as "not significant."

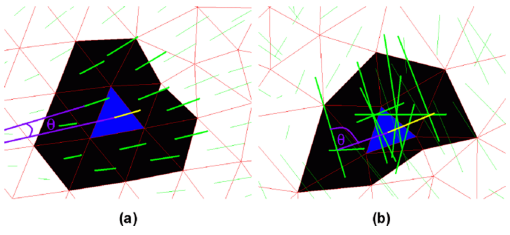

Fig. 2. Neighbor normal illustration: (a) Mesh comprising part of a cheek; (b) Mesh comprising part of an eye

Figure 2a shows a mesh comprising part of a cheek and Figure 2b shows a mesh comprising part of an eye. In both figures, t is the blue triangle. The triangles in N are colored black. n_t is represented by the thick yellow line protruding from the blue triangle. n_u, for each triangle in N, is represented by the thick green line protruding out of each black triangle.

θ_{max} is much less in Figure 2a than in Figure 2b. This is what we expect and what our approach is based on. We have found that many of the triangles in important facial regions, including the nose, eye, and lip regions, have much larger θ_{max} values than those in less important facial regions, such as the cheek and forehead regions.

3 Pose Determination

Our pose estimation approach consists of three key steps: region of interest (ROI) detection followed by nose tip and nose bridge determination.

Figure 3 illustrates a processed range image going through our pipeline, with Figure 3a showing a processed image from our database.

3.1 Determination of Region of Interest

The first step is to remove the 3D range image's boundary triangles since the boundary may be rough, and a rough boundary would negatively affect our approach's accuracy. Figure 3b shows the result of this initial step.

Next, the Surface Normal Difference (SND) algorithm is applied. The normal of every remaining triangle in the 3D mesh is determined, and the maximum angle that each triangle's normal makes with an adjacent triangle's normal is calculated. If this maximum angle is greater than an angular tolerance, δ, both triangles whose normal vectors made the angle are marked as "potentially significant." Otherwise, the corresponding triangle is marked as "not significant." This procedure is repeated, by incrementing δ, until the number of "potentially significant" triangles is less than α. We initially set δ to 10 degrees and have empirically set α to 3000. Figure 3c shows the "potentially significant" triangles after applying the SND algorithm.

Usually, the SND algorithm will keep a number of small, connected surfaces that are not part of a significant facial region. For example, it may keep a surface corresponding to a pimple on the forehead or a cut on the cheek. Most of the time, these outlying surfaces are composed of a very small number of triangles relative to

the largest remaining connected surface, which usually contains, minimally, the eye and nose regions. In order to filter out these outlying surfaces, the maximum number of triangles contained in any connected surface ρ is determined. Any surface composed of fewer than κ percent of ρ triangles is considered an outlying surface, and all of the triangles in these outlying surfaces are marked as "not significant." We have empirically set κ to 0.1.

At this point, a vertex is labeled as "significant" if it is part of a "potentially significant" triangle. All other vertices are labeled as "not significant." We call the mesh comprised of the remaining "potentially significant" triangles a Sparse Feature Mesh (SFM). The SFM is shown in Figure 3d.

3.2 Determination of Nose Tip

The PCA algorithm is run on the SFM vertices in order to align the model in the Z direction and determine the location of the nose tip (NT). Figure 3e shows the result of this step.

$$NT \in V_{sig} \mid \max(z)$$

where V_{sig} denotes the set of "significant" vertices. Note that the principal depth direction (Z) of a model can be reliably estimated with the minimum eigen-value, given the SFM vertices, even though the X and Y components may not be aligned to the correct model orientation.

The shape index [1], which is a surface primitive based on surface curvatures, is calculated at each "significant" vertex. All triangles that have at least one very concave vertex (a shape index less than -.50) are marked as "significant." All other triangles are marked as "not significant." A number of discreet groups of triangles, usually numbering around 20, remain. These groups usually include the corners of the eyes and the sides of the nose and mouth. The "significant" triangles are shown in Figure 3f.

3.3 Determination of Nose Bridge

In order to locate the nose bridge point, all pairs of groups meeting certain general geometric criteria are iterated over, and the symmetry of the shape indices of the vertices near each line connecting a pair of candidate groups (L_{CG}) and near that line translated to the nose tip (L_{NT}) is calculated. The sum of these two symmetry values is minimized and the line perpendicular to the L_{CG} that passes through the nose tip (H_{NT}) is inspected.

All the "significant" vertices within an XY distance of γ_1 from a line connecting two candidate groups are the vertices that compose a L_{CG}.

$$L_{CG}(g1, g2) = \forall (v \in V_{sig} \mid \left| dist_{xy}(v, \overline{g_1 g_2}) \right| < \gamma_1)$$

where g1 and g2 are candidate groups and γ_1 is the 3D length of an arbitrary mesh triangle's side. The two points of maximum concavity on either side of the midpoint of the L_{CG}, at least a certain distance, γ_2, from the midpoint, are found, and the point between these two maximums with the greatest Z value is referred to as the PBMZ.

$$PBMZ(V) = v \in V \mid \max(z)$$

where V denotes the vertices in the region of interest. Let B be the set containing the L_{CG} vertices between these two maximums.

$$B = v \in V \mid v \geq (\max\ 1) \wedge v \leq (\max\ 2)$$

The symmetry of the shape indices of B about the PBMZ is determined by summing the mean squared differences of the shape indices of the 0.50*|B| vertices in B closest to the PBMZ. If there are not at least 0.25*|B| vertices between the PBMZ and either maximum, the L_{CG} is rejected because the nose bridge point is expected to be a point of close to maximum Z almost exactly in-between two maximum concavities (i.e. the eye corners). The L_{CG} is translated to the nose tip and the above procedure is repeated for determining the symmetry of the L_{NT}.

$$sym\ (B) = \sum_{i=1}^{.25*|B|} (b_{PBMZ\ (V)-i} - b_{PBMZ\ (V)+i})^2 / |B|$$

The optimal groups are found by using the symmetry minimization method, as defined below:

$$g1_{opt}, g2_{opt} \in G_{sig} \mid \min(sym(L_{CG}(g1, g2)) + sym(L_{NT}(g1, g2)))$$

If the sum is minimal, we inspect the corresponding H_{NT}, which is composed of the "significant" vertices within an XY distance of γ_1 from the line perpendicular to the L_{CG} that passes through the nose tip.

$$H_{NT}\ (g1, g2) = \forall (v \in V_{sig} \mid |dist_{xy}\ (v, \perp_{NT}\ (\overline{g1\ g2}))| < \gamma_1)$$

where $\perp_{NT}\ (\overline{g1\ g2})$ denotes the line perpendicular to $\overline{g1\ g2}$ that passes through the nose tip. The variances of shape indices of the 0.25*|H_{NT}| vertices closest to the nose tip on each side of the L_{NT} are calculated and compared. The side with the lesser variance is considered the nose bridge side. If either side has fewer than three vertices, the H_{NT} is rejected because the optimal H_{NT} is expected to have a large number of "significant" vertices.

NB_1 is a set containing all of the H_{NT} vertices on one side of the L_{NT} and NB_2 is a set containing all of the H_{NT} vertices on the other side of the L_{NT}. Of these two sets, the one containing the H_{NT} vertices on the nose bridge side of the L_{NT} is renamed $NB_{candidates}$ and the other is renamed $NB_{non\text{-}candidates}$.

$$NB_1 = \forall (v \in H_{NT}\ (g1, g2) \mid dist_{xy}\ (v, L_{NT}\ (g1, g2)) < 0)$$
$$NB_2 = \forall (v \in H_{NT}\ (g1, g2) \mid v \notin NB_1)$$
$$\text{var}'(V, L) = \text{var}(v \in V \mid dist_{xy}(v, L) <= dist_{xy}(middle\ (V), L))$$
$$NB_{candidates} = \text{var}'\ (NB_1, L_{NT}) <= \text{var}'\ (NB_2, L_{NT})\ ?\ NB_1 : NB_2$$

If a set of nose bridge candidates was found, all the vertices on the "nose bridge side" of the H_{NT} are iterated over. The point with the greatest absolute difference in maximum and minimum curvature that is not the nose tip is concluded to be the nose bridge point (NB).

$$NB \in NB_{candidates} \mid \max(|k_{\max}(NB) - k_{\min}(NB)|) \wedge NB \neq NT$$

where k_{min} is the minimum surface curvature and k_{max} is the maximum surface curvature

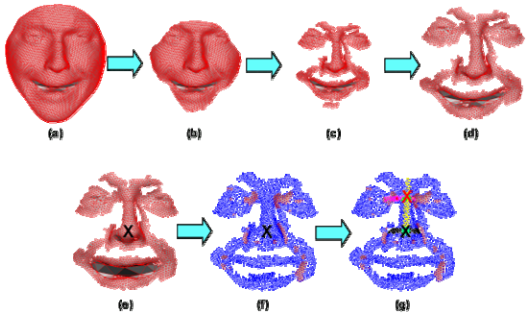

Fig. 3. Finding the pose of a 3D range image. (a) The initial range model; (b) The range model after initial boundary removal; (c) The range image after application of our SND algorithm; (d) The range image after running our outlier removal algorithm; (e) The range image after running the PCA algorithm (the nose tip is marked with a large black "X"); (f) "significant" vertices in blue and "significant" triangles in red; (g) The nose bridge is marked with a large red "X".

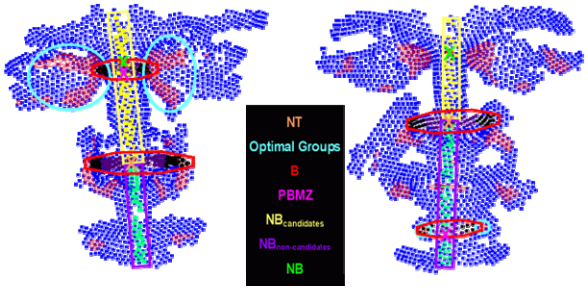

Fig. 4. Two models wit0068 the NT, Optimal Groups, B, PBMZ, $NB_{candidates}$, $NB_{non\text{-}candidates}$, and NB labeled

Figure 4 shows the key groups involved in pinpointing the nose bridge point. The result of this procedure is shown in Figure 3g, where the nose bridge point is marked with a large red "X." The optimal L_{CG} is represented by the green and pink line passing through the nose bridge and the optimal L_{NT} is represented by the black and white line passing through the nose tip. The optimal H_{NT}, on which the nose bridge point lies, is the yellow line.

As a result, the model Z direction, nose tip and nose bridge points uniquely determine the facial pose.

4 Experimental Results

We tested our approach on our created 3D facial expression database [7], which contains 2,500 models. Each model has two types of data: a face data which contains pure face region and a raw data which includes the face region and shoulder. The test subjects cover a wide variety of ethnicities and ages. The subjects expressed seven prototypic facial expressions (neutral, anger, disgust, fear, happiness, sadness, and surprise).

After comparing the nose tip and nose bridge found by our algorithm with a manually labeled nose tip and nose bridge, we found the angular difference between the lines connecting the former two points and the latter two points. The angular difference was less than 10 degrees for 98% of the pure face models and 87% of the raw data. The maximum distance between an expected nose tip and an actual nose tip, on a successful test (one that resulted in an angular difference of less than 15 degrees), was less than 7 units.

Figure 5 shows four examples of the processed pure face models and Figure 6 shows examples of raw models. In each figure, the first row shows the initial texture-mapped 3D range image. The second row shows the same images with the normal of each mesh triangle indicated by a thin green line protruding from the triangle. The third row shows the image after passing it through our pipeline. The "significant" vertices are represented by blue dots and the "significant" triangles are represented by red triangles. The nose tip is marked with a large black "X" and the nose bridge is marked with a large red "X." The fourth row shows the nose tip and the nose bridge marked on the original texture-mapped range images.

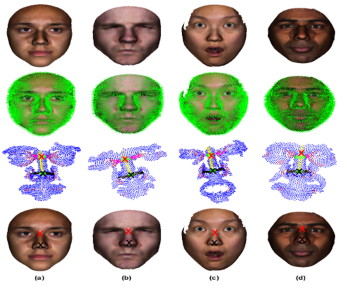

Fig. 5. Four examples of trimmed 3D range models with different expressions. The second row shows normals of the models.

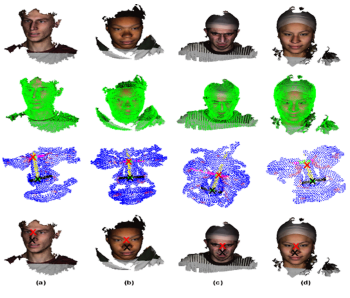

Fig. 6. Four examples of raw 3D range models with different head poses

5 Applications of Pose Estimation for Identification of Range Models and Their Expressions

We applied our pose estimation algorithm to the 3D model sequences, which are created from our facial expression database [7] using our structure-lighting based capture system. There are 600 model sequences with 100 subjects. Examples are shown in Figure 7.

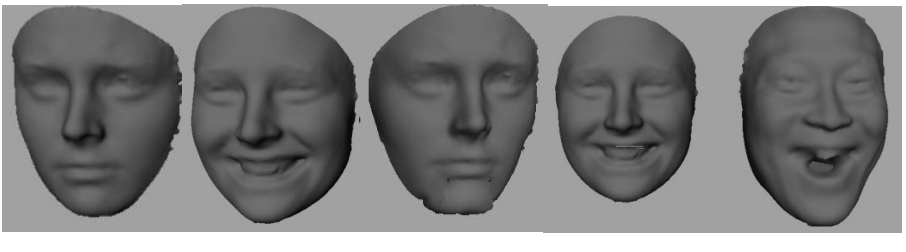

Fig. 7. Example frames of model sequences of two subjects with different poses and different expressions

5.1 Face Model Classification

We applied a generic-model based tracking approach to estimate the motion of facial surfaces. The poses of range models in each frame are estimated. Since the poses of all the frames of a model sequence are estimated, we normalize them to the front view and applied the model adaptation algorithm [20] to fit a low-resolution generic model to the high-resolution range models. The characteristics of the facial surface are captured by the adapted tracking model. Since a model sequence contains various facial expressions, we warp them to a standard shape which exhibits a neutral expression. Then, the entire warped frame models of the sequence are averaged, resulting in the individual's representative model, based on which the subsequent face recognition is carried out. We used 600 3D model sequences to test our face recognition algorithm. Based on our previous approach used in [17], we conduct the optimal feature selection, and compare the individual's representative models to the database models. The recognition is based on the feature correlation criterion. The average correct recognition rate is 90.7%. About 94 percent of facial poses of the entire database have been correctly estimated.

5.2 Face Model Expression Classification

We applied our developed approach as in [20] to analyze facial model expressions. The approach is briefly described as follows: (1) Estimate model poses in each frame; (2) Fit a tracking model to the 3D facial scan sequence; (3) Label the facial surface model using curvature information and generate a label histogram map; (4) Construct a facial expression descriptor using the tracked motion vectors and the facial surface feature label map.

Given the face scan's pose and the positions of its eyes and nose from the previous stage, we can rigidly adapt the tracking model to the face scan via the affine transformation. Then a fine adaptation procedure can be conducted in order to deform the tracking model into a non-rigid facial surface area. This procedure is realized by using the energy minimization method based on the dissimilarity (error function) between the tracking model and the face scan [20]. As a result, the 3D motion trajectories are estimated by vectors from the tracked points of the current frame to the corresponding points of the first frame with a neutral expression. Each motion trajectory is represented by a three-tuple vector v_i. Sixty-four feature points on the facial regions (e.g., eyes, nose, mouth, eyebrows, and chin) are selected to construct a facial expression motion vector $\mathbf{t} = [t_1,...,t_{64}]$. It represents the *temporal* facial expression information. In addition, the *spatial* facial expression information, so called facial expression label map (FELM), is created using small scale curvature based labeling approach. The different label distributions show the different facial expression characteristics. For each expression, a FELM vector $\mathbf{s} = [s_1,...,,s_n]$ is generated after the model is labeled. Each element e_i of a FELM is a ratio of the number of vertices with a specific label type to the number of vertices in the whole facial region. n denotes the 12 label types (detail in [20]). Given the same facial expression from different subjects, the FELMs of expressive regions exhibit the similar characteristics of histograms.

To this end, a spatio-temporal facial expression descriptor $\mathbf{E_d=[t, s]}$ is constructed for each expression model. We conducted person-independent facial expression

recognition experiments using our dynamic 3D facial expression database, and applied linear discriminant analysis (LDA) to classify the six prototypic facial expressions. The data from 70 subjects are used for training. The remaining 30 subjects are for test. The average correct recognition rate is 84.7%.

6 Limitations and Conclusion

Automatically locating the nose tip and the orientation of a face is crucial for face and facial expression recognition. We have developed an algorithm that decreases the search space needed to find the primary features of a face. This is the first step towards developing an automatic facial and facial expression recognition system. Our approach could be used as a preprocessing step in pose-variant systems to determine the pose of the face and make these systems pose-invariant. Note that the PCA based approach could be potentially used for the pose estimation. However, it may not achieve the satisfactory results since it requires the 'clean' data with a rigid symmetric property, which is not always the case for many of our models.

After using our SND algorithm to eliminate a large number of triangles, we use a curvature-based approach to further decrease the search space. Future improvements could be obtained by applying more accurate curvature estimation methods [4, 6].

Note that in some cases, the nose tip was incorrectly labeled. Since our algorithm is dependent on correctly locating the nose tip, it is not surprising that it found the wrong nose direction. Most of these images contained a large percentage of extraneous data, such as the shoulders and neck, which was not removed by our boundary removal algorithm. Our future work will investigate a method to have our boundary removal algorithm automatically adjust itself depending on the perceived noise of a range model using approaches based on machine learning in order to improve the robustness of the algorithm.

Acknowledgments. This material is based upon work supported by the National Science Foundation under grants IIS-0541044 and IIS 0414029, and the NYSTAR's James D. Watson Investigator Program.

References

[1] Dorai, C., Jain, A.: COSMO-A representation scheme for 3D free-form objects. IEEE Trans. on PAMI 19(10), 1115–1130 (1997)
[2] Hattori, K., Sato, Y.: Estimating pose of human face based on symmetry plane using range and intensity images. In: ICPR 1998 (1998)
[3] Rajwade, A., Levine, M.D.: Facial Pose from 3D Data. Journal of Image and Vision Computing 2007 (to appear)
[4] Razdan, A., Bae, M.: Curvature estimation scheme for triangle meshes using biquadratic Bezier patches. Computer-Aided Design 37(14) (2000)
[5] Srinivasan, S., Boyer, K.L.: Head pose estimation using view based eigenspaces. In: ICPR'02 (2002)

[6] Tanaka, H.T., Ikeda, M.: Curvature-based face surface recognition using spherical correlation-principal directions for curved object recognition. In: ICPR'96, pp. 25–29 (1996)
[7] Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M.: A 3D facial expression database for facial behavior research. In: IEEE FGR 2006, Southampton, UK, pp. 211–216. IEEE Computer Society Press, Los Alamitos (2006)
[8] Bowyer, K., Chang, K., Flynn, P.: A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. CVIU 101(1), 1–15 (2006)
[9] Bronstein, A., Bronstein, M., Kimmel, R.: Three dimensional face recognition. IJCV 5(30) (2005)
[10] Blanz, V., Vetter, T.: Face Recognition Based on Fitting a 3D Morphable Model. IEEE Trans. on PAMI 25(9) (2003)
[11] Chang, Y., Vieira, M., Turk, M., Velho, L.: Automatic 3D facial expression analysis in videos. In: IEEE ICCV05 Workshop on Analysis and Modeling of Faces and Gestures. IEEE Computer Society Press, Los Alamitos (2005)
[12] Cohen, I., Sebe, N., Garg, A., Chen, L., Huang, T.: Facial expression recognition from video sequences: temporal and static modeling. CVIU 91(1) (2003)
[13] Gross, R., et al.: Quo vadis face recognition? In: Workshop on empirical evaluation methods in computer vision (2001)
[14] Li, S., Jain, A.: Handbook of face recognition. Springer, New York (2004)
[15] Pantic, M., et al.: Automatic analysis of facial expressions: the state of the art. IEEE Trans. PAMI 22(12) (2000)
[16] Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: IEEE CVPR. IEEE Computer Society Press, Los Alamitos (2005)
[17] Sun, Y., Yin, L.: 3d face recognition using two views face modeling and labeling. In: CVPR05 Workshop on A3DISS.
[18] Tang, X., Li, Z.: Video based face recognition using multiple classifiers. In: IEEE FGR04. IEEE Computer Society Press, Los Alamitos (2004)
[19] Lu, X., Jain, A., et al.: Matching 2.5D face scans to 3D models. IEEE Trans. PAMI 28(1), 31–43 (2006)
[20] Yin, L., Wei, X., Longo, P., Bhuvanesh, A.: Analyzing facial expressions using intensity-variant 3d data for human computer interaction. In: ICPR 2006, Hong Kong.
[21] Zhao, W., Chellappa, R., Phillips, P., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys 35(4) (December 2003)
[22] Kittler, J., Hilton, A., et al.: 3D Assisted Face Recognition: A Survey of 3D Imaging, Modelling and Recognition Approaches. In: CVPR 2005 Workshops on A3DISS.
[23] Wang, Y., Samaras, D., Metaxas, D., Elgammal, A., et al.: High resolution acquisition, learning, transfer of dynamic 3D face expression. In: Eurographics (2004)