2009 Fifth International Conference on Image and Graphics

Super-Resolution of Faces Using Texture Mapping on a Generic 3D Model

X. C. He, S. C. Yuk, K. P. Chow, K.-Y. K. Wong, R. H. Y. Chung
The Department of Computer Science, The University of Hong Kong,
Pokfulam Road, Hong Kong
{xche, scyuk, chow, kykwong, hychung}@cs.hku.hk

***Abstract*— This paper proposes a novel face texture mapping framework to transform faces with different poses into a unique texture map. Under this framework, texture mapping can be realized by utilizing a generic 3D face model, standard Haar-like feature based detector, active appearance model and pose estimation algorithm. By this texture map, correspondence of every pixel at the face across multiple distinct input images can then be established, which enables super-resolution algorithms to be applied directly on registered texture map to render high resolution faces. This paper details the proposed framework, and illustrates how the proposed super-resolution algorithm works with the help of weighted average and median filters. Convincing experimental results are also presented to validate the effectiveness of the proposed framework and super-resolution algorithm.**

Keywords-Super-resolution, face reconstruction; Pose estimation, Texture mapping; Face model

I. INTRODUCTION

Face recognition in visual surveillance are the key technologies for a number of application domains ranging from simple access authentications, to more complicated face recognition-based video retrieval applications. Recently, these algorithms have become very mature and reliable to the extent that they have already been put into practical applications for use in industrial and consumer electronics. However, the quality of the recorded images or videos is sometimes low, where objects of interest like human faces are usually captured at a relatively low resolution which is not sufficient for reliable recognition of individuals. The use of super-resolution (SR) techniques appears to be a natural solution to it and may be promising for improving the quality of such visual materials.

Several approaches of SR on faces have been reported in literature, which can be coarsely classified into reconstruction based [1-3] or learning based [4-8] approaches. In [1], a SR frame is computed using information from past and future low-resolution (LR) frames. It calculates the optical flow between interpolated versions of these frames and the initial SR frame. The updated SR frames is the average of the current SR frames and warped versions of neighboring interpolated LR frames. However, the algorithm could have poor performance when prominent movement of the face occurred between frames. In [2] the motion between frames is also estimated using optical flow. A probabilistic scheme is employed to determine whether the pixels in each input frame are visible in the SR frame or not, so that only the visible pixels are used to update the SR frame. In [3], after obtaining the optical flow by tracking feature points on the faces, epipolar geometry is utilized to reject outlying low vectors, which could improve the registration of the face over multiple frames. Capel and Zisserman [4], on the other hand, used eigenface from a training face database as model prior to constrain and super-resolve low-resolution face images. To further improve the performance, they divided human face into six unrelated parts and apply PCA [9] on them separately. A similar method was proposed by Baker and Kanade[5]. They established the prior based on a set of training face images by using Gaussian, Laplacian and feature pyramids. Freeman and Pasztor [6] tried to recover the lost high-frequency information from low-level image primitives, which were learnt from several general training images. They mapped the images and scenes into a Markov network, and learned the parameters of the network from the training data. A very similar image hallucination approach was also introduced in [7], in which the primal sketch is used as the prior to recover the smoothed high-frequency information. Jia and Gong [8] integrated the tasks of super-resolution and recognition by directly computing a maximum likelihood to identity parameter vector in high-resolution tensor space for recognition.

Our literature review reveals that learning based approaches appear to perform well only when high-resolution images were used in the training datasets as prior, but it may not perform equally well when high-resolution training images are not available. On the other hand, current reconstruction based approaches normally assume rigid planar objects or simplified scenes to ease the alignment of objects across multiple frames, which makes them less effective in handling objects or scenes that are not rigid nor planar, for instance, human faces. As a result, a more sophisticated registration method is required for a more accurate 3D human face reconstruction.

To overcome these hurdles, this paper proposes a novel face super-resolution approach by using texture mapping on a generic 3D model, which was derived from an automatic 3D face reconstruction framework from a single image. In this reconstruction framework, facial features of input image are detected automatically. Then, a generic 3D face model is adopted, where the transformation matrix to project 3D face onto input 2D image can be estimated by calculating the pseudo inverse matrix of the 3D coordinates of detected facial features. Texture of each patch in input image is extracted and interpolated to rebuild the texture map of the 3D face. For each LR frame, mapping from input image to the aforementioned texture map is functionally similar to the registration in the SR process. SR frame can then be obtained by fusing texture maps of each LR frame, or by getting the scattered points from each LR frame to produce a high resolution (HR) texture map by interpolation.

978-0-7695-3883-9/09 $26.00 © 2009 IEEE
DOI 10.1109/ICIG.2009.74

IEEE computer society

In Section 2, we introduce the concept of our proposed texture mapping framework, and the details of texture map registration are described. With the registered texture map, different SR schemes, which will be detailed in Section 3, could be adopted. Experimental results are provided in Section 4 to evaluate the effectiveness of the proposed algorithm.

II. Face Reconstruction

Basically, the proposed face reconstruction framework includes several parts, which are facial feature detection, 3D modeling, pose estimation, and texture map rebuilding. Firstly, Haar-like feature based object detection [10] was adopted to detect frontal or profile human face from input image, followed by active appearance model (AAM) [11] which locates detailed facial features. After that, a generic 3D face model including 3D mesh, texture-map and texture coordinates was used to render the reconstructed 3D face. Next, based on the 2D image coordinates of detected facial features and their corresponding 3D coordinates in the 3D mesh, the head pose of the face in the input image and the associated projection matrix of this pose can then be estimated. Finally, texture patches were extracted from input image and mapped accordingly onto the texture-map. After some post-processing on this texture-map, it was then used to render the reconstructed 3D face. Details of each step will be discussed in the following subsections:

A. Facial Feature Detection

The very first step of this framework is to detect and classify the input human face as frontal, half-profile or profile face. A popular face detector [10], a cascade of boosted classifiers working with Haar-like features, is adopted. It should be noted that we need two detectors here, one for frontal face, and the other for profile face. Typically profile face detector can only detect face towards one direction, and face to another direction can be detected by simply flipping input image horizontally. Without loss of generality, we assume all the profile faces are oriented towards the left side.

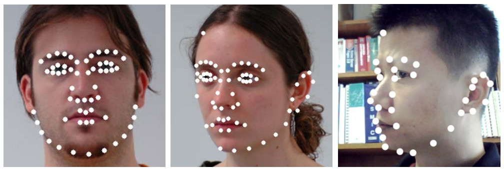

Fig.1. Examples from the training sets for AAM.

Then, by using the AAM algorithm [11] we can match the pre-trained face appearance model to the input image and locate the detailed facial features. Similar to the method in [12], we use three distinct appearance models, which are frontal, half-profile and profile. The training set consists of labeled images, where key landmark points are manually marked on each example face. Figure 1 shows examples of the labeled images used in our training set.

The idea of AAM search is to minimize the difference between an input image and the one synthesized by the appearance model. The difference could be defined as the magnitude of difference vector:

$$\Delta \mathbf{I} = \left| \mathbf{I}_i - \mathbf{I}_m \right|^2 \qquad (1)$$

where $\mathbf{I}_i$ is the vector of grey-level values in the image, and $\mathbf{I}_m$ is the vector of grey-level values for the current model parameters. Individual models are then applied to match the input face image and search for the best fit. The one with minimum value of $\Delta \mathbf{I}$ is adopted for locating the facial features. AAM algorithm includes an initialization procedure and a search procedure. Since Haar-feature face detector is faster than AAM initialization, it will be more efficient to use result of Haar-feature face detector as input of AAM algorithm to reduce the dynamic range of the subsequent search space.

B. Three-dimension face modeling

According to some existing works [13-15], a 3D face was normally represented by a shape-vector S and a texture-vector T, where shape-vector contains 3D coordinates and texture-vector contains the R, G, B color values of all the vertices. In these approaches, large number of vertices (approximately 70,000 vertices in [13]) are usually required to achieve a reasonably good high resolution appearance due to the limitation that there is only one color value per vertex.

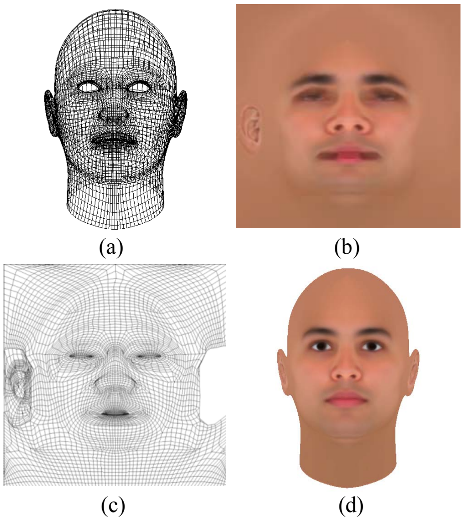

Fig. 2. 3D face model. (a) 3D mesh, (b) Skin texture map, (c) Texture coordinates map, (d) Texture rendered.

In contrast, in this proposed framework, we employ texture mapping technology instead, where detailed texture could be rendered on a 3D shape with substantially fewer vertices. To illustrate the effectiveness of the framework, the face model we employed in this paper comes from a 3D mesh with less than one-tenth of the vertexes (6292 vertices and 6152 facets) of the mesh employed in [13]. The face model, as well as the texture maps and texture coordinates are depicted in Figure 2(a)-(c). In essence, texture map is a 2D image which contains all the texture of a human face and texture coordinates determine how the texture is mapped from the 2D texture map onto the final 3D mesh shown in Figure 2(d). It should be noted that this face model is derived from FaceGen (a software product of Singular Inversions Inc.), where the 3D mesh contains several objects including skin, eyes, sock, teeth and tongue, which is useful for morphing 3D face into a variety of face expressions. Texture map of skin is shown in Figure 2(b), and other texture maps are omitted here. In our work, the texture maps of skin and eyes will be rebuilt according to the input image, while default texture maps will be employed for the rest.

It is well known that 3D face reconstruction includes shape reconstruction and texture reconstruction. This paper focused on the latter, and therefore a single mean shape was adopted in this work. However, 3D face geometry reconstruction algorithm in [15] could be utilized to get a personalized 3D shape if required. Detected facial features in Section 2.1 could be utilized for 2D face alignment then.

C. Pose estimation

In order to extract texture of input image for filling patches in the 3D mesh, coordinates of each mesh points has to be projected onto the input image. Therefore we need to estimate the pose of the 3D face to align the mesh points properly with the 2D face in the input image. Fortunately, by using 2D coordinates of facial features detected by AAM in the input image and their corresponding 3D coordinates in 3D mesh as hints, the transformation matrix for this projection can be deduced as follows:

Let $\mathbf{Q}$=(X_Q, Y_Q, Z_Q) be an arbitrary point in 3D, let $\mathbf{q}$=(x_q, y_q) be the corresponding 2D image coordinates of $\mathbf{Q}$. A forward mapping function, $\mathbf{\Phi}$, which defines the transform function from a point in 3D coordinates to a point in the 2D image coordinates is given as, $\mathbf{q} = \Phi\{\mathbf{Q}\}$.

By perspective transformation, we have

$$\left[x_q \quad y_q \quad z_q \quad 1\right] = \left[X_Q \quad Y_Q \quad Z_Q \quad 1\right] \cdot \mathbf{T} \,, \qquad (2)$$

where $\mathbf{T}$ is the transformation matrix. Then we assume that sub-matrix $\mathbf{T}_1$ is the first two column of $\mathbf{T}$, we have

$$\left[x_q \quad y_q\right] = \left[X_Q \quad Y_Q \quad Z_Q \quad 1\right] \cdot \mathbf{T}_1 \,. \qquad (3)$$

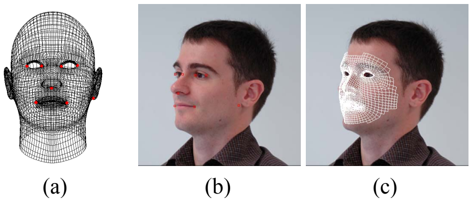

Fig. 3. (a) Feature points on 3D mesh, (b) Feature points in 2D input image, (c) Fit 3D mesh into 2D image.

Among all the detected facial features, we pick some for pose estimation, e.g. eye corners, mouth corners, ear tips and nose tip, as shown in Figure 3 (a)-(b). Note that our framework is less tightly coupled to the 3D model employed, in the sense that virtually any 3D face model can be employed as long as there exists a direct correspondence between key facial feature points and those in the mesh. Assume m feature points are utilized, we have

$$\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ . & . \\ x_m & y_m \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ . & . & . & 1 \\ X_m & Y_m & Z_m & 1 \end{bmatrix} \cdot \mathbf{T}_1 \qquad (4)$$

Then $\mathbf{T}_1$ can be calculated as

$$\mathbf{T}_1 = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ . & . & . & 1 \\ X_m & Y_m & Z_m & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ . & . \\ x_m & y_m \end{bmatrix} \qquad (5)$$

where $[\cdot]^{-1}$ represent Moore-Penrose pseudo inverse. The matrix thus obtained is actually a least square solution of the mesh fitting problem. With this matrix, projection of every vertex of 3D mesh onto the input image could be calculated by Eqt.(3), which enables us to rebuild the texture map.

D. Texture map rebuilding

After pose estimation, texture map could be rebuilt by filling each texture element with those extracted from the input image. It should be noted that only the central region of texture map, which includes every feature of human face, is needed to be rebuilt. Coordinates of mesh in central face projected on the input image are calculated and marked as shown in Figure 3(c), where every patch has a correspondence in the texture coordinates map. For an individual patch of mesh, we just map the texture of this patch in the input image into the corresponding patch in the texture map. Since this mapping is from 2D image to 2D image, affine transform could be utilized for interpolation. Affine transform is determined by the coordinates of all vertices of this patch in the two images, and the transformation and the filling procedure are illustrated in Figure 4(a).

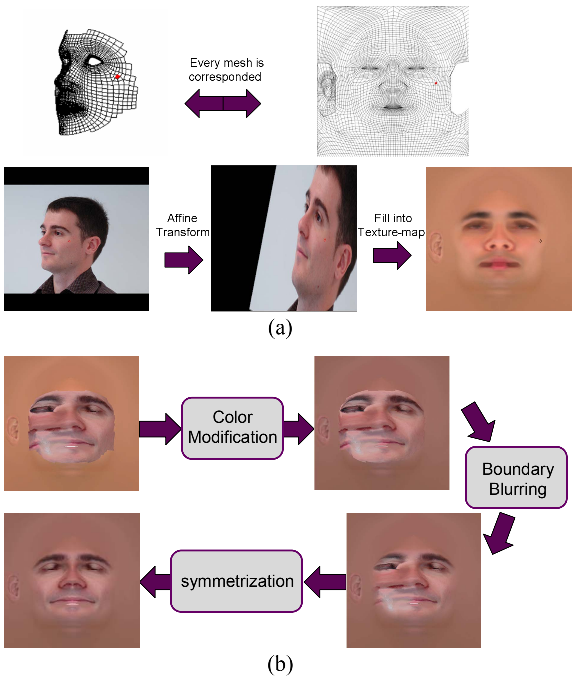

Fig. 4. Diagram of rebuilding skin texture map. (a) Transform and filling. (b) Post processing.

Central region of texture map (called face region) is filled with the texture that comes from the input image, while the other parts of it (called non-face regions) are filled with default texture. A post processing is performed to smooth out the transition values between these two regions for better 3D face reconstruction results. First, color of non-face region is modified to that of the color tune of the face region. Then boundary of two regions is blurred to smooth out the color transition from one region to the other. On the other hand, for non-frontal face, the texture of the occluded face region cannot be reconstructed well as they are not visible in the 2D input image. Therefore, the post processing step also includes a symmetrization process to fill visible texture into occluded region and to make the texture map laterally symmetric. These post processing steps are illustrated in Figure 4(b).

It was mentioned before that our face model includes several objects and texture maps. The above process illustrates how skin texture map can be rebuilt. It should be noted that eyes texture map could also be rebuilt in a similar fashion.

III. Super-Resolution Scheme

The above face texture mapping process transformed input human face images, which could have various poses, into unified texture maps. And by this texture map, every pixel at the face on distinct input images was corresponded. Upon closer inspection, this reconstruction process is actually functionally similar to the registration in SR processing. Suppose that we have several LR image frames of same person. By texture mapping, we can rebuild LR texture map for each of them. HR texture map can be achieved by fusing LR texture maps of each LR frame. Average filter (AF), a simple end efficient fusion method which is also utilized by many SR algorithms [1, 16], is one of the strategy we adopted here. By using this HR texture map, we can therefore render SR face images with various poses.

Another SR strategy is to perform fusion process before rebuilding texture map. Revisiting how pixels of input image were filled into texture map as described in section 2.4: Firstly, each pixel of input image was mapped into texture map with sub-pixel level. Then scattered data interpolation method was applied to get grid data, nearest neighbor (NN) and triangulation based linear interpolation (TBLI) were two possible options.

With multiple input images, severalfold scattered points exist on texture map, which naturally could be utilized to rebuild a HR texture map. In our research, a weighted average of nearest neighbors with median filter (WANNM) was adopted instead of NN method. For each grid pixel P_0 in texture map, find the nearest five scattered sub-pixel points of it, and then eliminate two points whose color are the farthest away from the mean color of those five points, and finally calculate color of it by weighted average color of remained three points (P_1 to P_3) as following:

$$C_0 = \begin{cases} \dfrac{C_1/d_1 + C_2/d_2 + C_3/d_3}{1/d_1 + 1/d_2 + 1/d_3} & \text{When } d_i \neq 0 \\ Ci & \text{When } d_i = 0 \end{cases} \qquad (6)$$

where, C_0 is color of grid pixel in texture map, C_i is color of scattered sub-pixel points (i=1,2,3), and d_i is distance from scattered sub-pixel point P_i to the grid pixel P_0. Performance of AF, NN, TBLI and WANNM will be discussed in experiment section.

IV. Experiments and Discussion

GTAV face database [17] is adopted in our experiments to serve as a common reference for performance evaluation. Furthermore, a USB web camera is utilized for capturing more face images for testing the robustness of the framework in reconstructing faces from a complex background. Input images were all normalized to a size of 512×512, and the typical processing time are 1~2 seconds, which varies according to the size of the face in the image. Among all the several steps of the proposed framework, facial feature detection consumes most of time.

Some single frame reconstruction results are demonstrated in Figure 5, where the reconstructed frontal view results are shown in Figure 5(b), and Figure 5(c) shows synthetic faces with various poses. When the face in the input image is a non-frontal view (first three rows in Figure 5), symmetrization processing will be performed, which will be determined automatically according to which active appearance model is adopted.

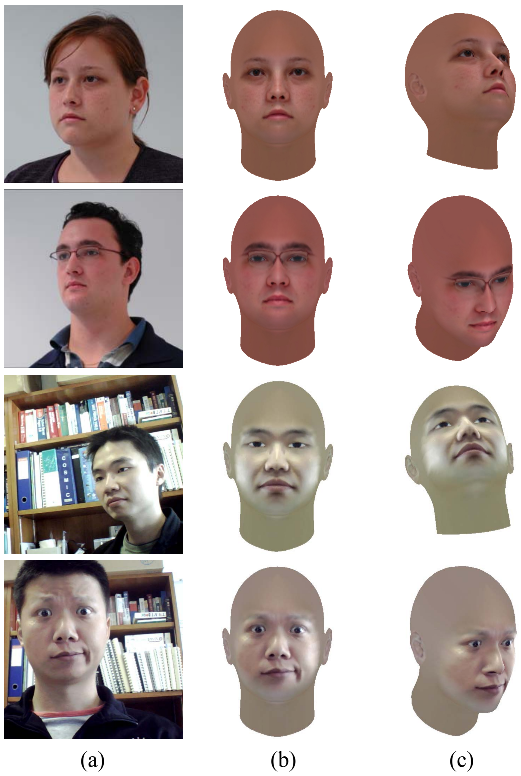
(a) (b) (c)

Fig. 5. Reconstruction results. (a) Original image, (b) Synthetic results of frontal view, (c) Synthetic results of various poses.

The SR experiment results were illustrated in Figure 6, and area of mouth and nose was enlarged for a more clear illustration. Four reconstructed LR faces were shown in Figure 6(a)-(d), and the SR results produced by average filter, TBLI and WANNM were shown in Figure 6(e)-(g) respectively. Texture in Figure 6(a)-(d) looks quite coarse, due to few scattered points exist in LR texture map, and pixels in a small block have same color since they have same nearest neighbor. By average fusing all four LR texture map, results in Figure 6(e) looks smooth and there is no obvious block in it. However, similar to Gaussian smooth, result of average fusion looks a little bit blur on edges and loses some high frequency information. It should be noted that although all face on LR frame was coming from same person, they were captured at different time instants. Therefore environment color and luminance of them could be different, and the colors of scattered points mapped into texture map, which belong to distinct LR frames, could be inconsistent. If we use SR algorithm like NN or TBLI directly, there will be stripes on rendered HR results, which is illustrated in Figure 6(f). By using median filter to eliminated some erratic points, and averaging color of neighbor points with reciprocal of distance as weight, proposed WANNM algorithm produce HR results with smooth surface as well as sharp edges as shown in Figure 6(g).

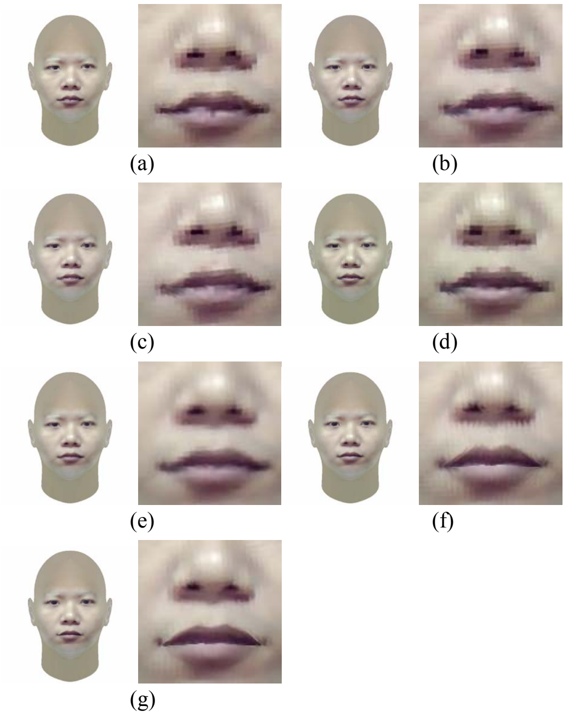

Fig. 6. Super-Resolution results. (a)-(d) Recon-structed results from LR frames, (e) SR results by averaging filter, (f) SR results by TBLI, (g) SR results by WANNM algorithm.

In conclusion, experimental results demonstrate that input faces with distinct poses can be synthesized into a unified texture map for the rendering of a realistic frontal face. SR algorithms could be applied on registered texture map directly, and produce convincible HR results. In our future works, we expect to apply the above proposed framework on video, to reconstruct HR 3D faces from multiple video frames, and to evaluate how frontal face reconstruction and SR algorithms could help to improve the accuracy in face recognition.

Acknowledgment

This research was sponsored by the Innovation and Technology Commission of the Government of the Hong Kong Special Administrative Region, under the Grant ITS/174/08.

References

[1] [1] S. Baker and T. Kanade, "Super-resolution optical flow," CMU-RI-TR-99-32, Robotics Institute, Carnegie Mellon University, 1999.

[2] [2] R. Fransens et al., "A Probabil-istic Approach to Optical Flow based Super-Resolution," *presented at Workshop of Conference on Computer Vision and Pattern Recognition*, pp. 191, 2004.

[3] [3] R. Den Hollander, et al., "super-resolution of faces using the epipolar constraint," *presented at British Machine Vision Conference*, paper 263, 2007.

[4] [4] D. Capel and A. Zisserman, "Super-resolution from multiple views using learnt image models," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 627-634, 2001.

[5] [5] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol.2, pp. 372-379 2000.

[6] [6] W. T. Freeman and E. C. Pasztor, "Learning low-level vision," *Proceedings of the IEEE International Confer-ence on Computer Vision*, vol.2, pp. 1182-1189, 1999.

[7] [7] S. Jian et al., "Image hallucination with primal sketch priors," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp 729-36, 2003.

[8] [8] K. Jia and G. Shaogang, "Multi-modal tensor face for simultaneous super-resolution and recognition," *Proceedings of IEEE International Conference on Computer Vision*, Vol. 2, pp. 1683-1690 2005.

[9] [9] M. A. Turk and A. Pentland, "Face recognition using eigenfaces," *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition*, pp 586-591, 1991.

[10] [10]R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection", *Proc. of IEEE ICIP*, vol. 1, pp 900-903, 2002.

[11] [11]T.F. Cootes et al., "Active appearance models", *IEEE Trans. on PAMI*, vol. 23, pp. 681-685, 2001.

[12] [12]T.F. Cootes et al. "View-based active appearance models", *Proc. of IEEE AFGR*, pp 227-232, 2000.

[13] [13]V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model", *IEEE Trans. on PAMI*, vol. 25, pp. 1063-1074, 2003.

[14] [14]S. W. Park et al., "3D face reconstruction from a single 2D face image", *presented at IEEE Conf. on CVPR Workshops*, pp 1-8, 2008.

[15] [15]Z. Zhang et al., "Minimum variance estimation of 3D face shape from multi-view", *Proc. of IEEE Conf. on AFGR*, pp 547-552, 2006.

[16] [16]L. C. Pickup et al., "Optimizing and Learning for Super-resolution," *presented at British Machine Vision Conference*, Edinburgh, UK, 2006.

[17]F. Tarres and A. Rama, "GTAV Face Database," available at http://gps-tsc.upc.es/GTAV/ResearchAreas/UPCFaceDatabase/GTAVFaceDatabase.htm