Face Reconstruction and Camera Pose Using Multi-dimensional Descent

Varin Chouvatut, Suthep Madarasmi, and Mihran Tuceryan

***Abstract*—**This paper aims to propose a novel, robust, and simple method for obtaining a human 3D face model and camera pose (position and orientation) from a video sequence. Given a video sequence of a face recorded from an off-the-shelf digital camera, feature points used to define facial parts are tracked using the Active-Appearance Model (AAM). Then, the face's 3D structure and camera pose of each video frame can be simultaneously calculated from the obtained point correspondences. This proposed method is primarily based on the combined approaches of Gradient Descent and Powell's Multidimensional Minimization. Using this proposed method, temporarily occluded point including the case of self-occlusion does not pose a problem. As long as the point correspondences displayed in the video sequence have enough parallax, these missing points can still be reconstructed.

***Keywords*—**Camera Pose, Face Reconstruction, Gradient Descent, Powell's Multidimensional Minimization.

I. Introduction

ESTIMATION of point reconstruction and camera pose from multiple video frames can be done by many techniques. Some techniques for estimating camera pose require the placement of specific markers with known 3D world positions [3-7]. For certain environments, it may not be suitable or even possible to add markers to the scene. Even in cases where markers can be added to the scene, measuring their precise positions may still pose a problem. So, instead of using artificial markers in a scene, this paper uses natural feature-points that can be tracked by using the Active Appearance Model (AAM) [19], [20].

In order not to use any special markers of known world coordinates, auto- or self- calibration is proposed [8]. Auto-calibration methods estimates structure from motion based on a given number of images. Examples of methodologies include fundamental matrix for two views [8], trifocal tensor for three [9], quadrifocal tensor for four [10], and factorization for multiple views [11]. In the auto-calibration, there are three levels of reconstruction including projective, affine, and metric reconstruction. Considering the many steps involved and the details needed for reconstruction, noisy results from a lower level of reconstruction can propagate the error to a higher level. Thus, an optimization process is needed after each level of reconstruction. Furthermore, if points are missing because of temporary occlusions, special strategies must be applied to find correspondences prior to the 3D reconstruction process. In the case of self-occlusion, searching for correspondences is even harder. This proposed method is a simpler and more practical method where self-occlusions can be ignored while the robustness is still retained.

Gradient descent is a robust minimization method which has been used for various tasks such as training radial basis function (RBF) neural networks [12], predicting or searching motion of point in images [13], [15], and estimating 3D camera motion [14], [16]. Unfortunately, one main disadvantage of the original gradient descent is that the derivative of system of equations or the conjugate direction [13] must be calculated. This derivative can then be used to define the step size for parameter updating. An incorrect derivative-formula may cause the system to be prone to errors. With the method presented here, the derivative need not be calculated.

Another approach for finding the minimization and maximization of the system functions is Powell's line-minimization method [1]. Powell's minimization for multi-dimensional system can be seen in [1], [17], [18]. Line minimization used in Powell's multi-dimensional search can be illustratively explained as climbing down the valley shaped like a 3D-parabola graph. Although many variables make the multi-dimensional direction, only one variable or one line needs to be considered at a time. While considering a variable/line, the deepest point along that line may not be the correct result to obtain a global solution. Thus, the method proposed in this paper chooses not to climb down until the line minimization but, instead, climb with only one small step at a time in the currently correct direction.

For face reconstruction, Y. Zheng, J. Chang, Z. Zheng and Z. Wang [22] proposed 3D face reconstruction in 2007 using stereo images together with a reference 3D face model. Since they found that the traditional stereo methods based on intensity failed to provide good results in 3D face reconstruction, a reference 3D model of the human face is used to help in correspondence calculation. Although a reference 3D model is used, non-linear deformations and camera registration have to be solved in order to find correspondences in the stereo images. In 2008, S. W. Park, J.

V. Chouvatut is with the Computer Engineering Department, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand (e-mail: varin@cpe.kmutt.ac.th).

S. Madarasmi is with the Computer Engineering Department, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand (e-mail: suthep@kmutt.ac.th).

M. Tuceryan is with the Computer and Information Science Department, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202-5132 USA (e-mail: tuceryan@cs.iupui.edu).

Heo, and M. Savvides [21] used a single image of a human face for 3D reconstruction. An average 3D face model is created from 3D training images. If the given image is not a frontal image of the face, pose correction of feature points must be done to re-create points for the frontal view before mapping the re-created points to the 3D average model via Delaunay triangulation. Reconstructing 3D structure from one single image may appear to provide a big advantage, but a 3D model of the underlying object needs to be prepared prior to using the algorithm. In other words, one must know what kind of object is being reconstructed. Also, the optimal solution of 3D structure for the unseen feature points may not be obtained because an assumption that the reconstructed object is symmetric is needed.

Y. Zheng and Z. Wang [23] reconstructed a 3D face model from a single frontal face image. They used a learning based approach for the reconstruction. That is, a database indicating the relationship between feature point and point depth must be prepared in advance. Since only one image is needed, point-depth estimation must be solved accurately. To estimate the point depth from just one image, a learning step of mapping between texture on a face and depth determination cannot be avoided. In other words, a database with depth of facial features must be available. Furthermore, the learning process to convert from the pattern map to the depth map from the prepared database is a high dimensional, nonlinear problem.

Combination concepts of the line minimization used in Powell's multi-dimensional search and the simple gradient descent optimization are adapted in the work proposed in this paper. The main concept of this proposed method is based on the fact that a feature point's 2D image points seen in several images will be back-projected to the same point in three dimensions [10]. Given 2D correspondences of N 3D-points seen in M frames provides more equations than unknowns. This constraint is sufficient to solve for all unknowns of the system of equations. Using the assumption that each contiguous frame from a video sequence has a small difference in image motion, camera motion can be easily achieved. The case missing feature points due to temporary occlusion or self-occlusion does not pose a problem in our proposed method as long as the points are seen in a sufficient number of frames to provide the above mentioned constraint. If there is sufficient parallax among the points, each point's 3D structure can be obtained by this proposed method.

After obtaining the face's 3D structure and camera motion, one may use them directly for applications such as Augmented Reality (AR), security, person identification, etc., or even use its re-projection to improve on searching correspondences of the missing points in AAM or other tracking systems.

II. Methodology

To reconstruct a human's face, 68 feature points are defined for the important parts of the face such as eyebrows, eyes, nose, mouth, and chin. The feature points defining facial parts on a human face are determined as shown in Fig. 1:

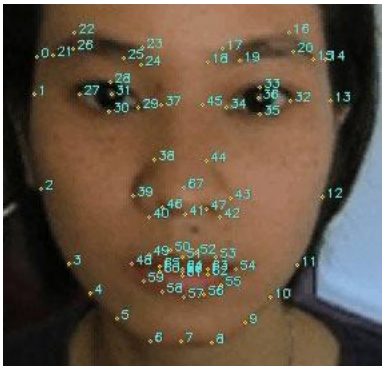


Fig. 1 Feature points used to define facial parts

Some video frames may have some invisible points due to self-occlusion of the face as shown in Fig. 2:

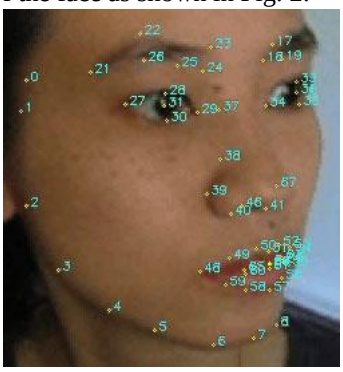


Fig. 2 An example of point missing due to self-occlusion

Let a transformation matrix composed of rotation and translation parameters be

$$M_{4\times4} = \begin{bmatrix} R_{3\times3} & \boldsymbol{T}_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \quad (1)$$

The 3D world coordinates of the feature points in the real-world can be defined as $\boldsymbol{W} = [X_w \quad Y_w \quad Z_w \quad 1]^T$. The 3D transformation between the world coordinate and the camera coordinates, $\boldsymbol{C} = [X_c \quad Y_c \quad Z_c \quad 1]^T$, can be obtained by

$$\boldsymbol{C} = M^{w\to c}\boldsymbol{W} \quad (2)$$

From (1) and (2), the transformation matrix can be re-written as:

$$\boldsymbol{C} = \left(R_x^{-1}R_y^{-1}R_z^{-1}T^{-1}\right)\boldsymbol{W} \quad (3)$$

where the rotation and translation matrices in (3) are defined as:

$$R_x^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x & 0 \\ 0 & -\sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$R_y^{-1} = \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z^{-1} = \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 & 0 \\ -\sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

After transforming a 3D-point from its world coordinate to a camera one, the camera coordinate can then be projected into

the image captured from this camera by using equation (5). The obtained 2D image coordinate point, $\boldsymbol{S} = [u \quad v \quad 1]^T$, is the feature-point that serves as the input.

$$\boldsymbol{S} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ 1 \end{bmatrix} \quad (5)$$

Considering the system equations, the unknown variables include camera's position (t_x, t_y, t_z) and orientation (θ_x, θ_y, θ_z) for each video frame, a single focal length (f), and 3D reconstructed points (X_w, Y_w, Z_w) for each feature point representing facial parts. So with M frames and N feature points, there will be $6M + 3N + 1$ variables to be estimated. As input of the system, each 2D feature-point has two components (u, v) so N feature-points seen in M frames will provide $2MN$ system-equations in total. The number of equations must be more than the number of unknown variables.

The steps in the proposed method are briefly explained as follows:

1. Feature points are tracked with the precision of sub-pixels by the AAM method. Note that, in the method proposed here, feature points need not be visible in all the frames.
2. Initialize all variables and step sizes (each variable has its own step size) used for updating variables; Z_w to a negative value such as -1, f to a sensibly positive (one may choose the range of 300-1,000), step-size to a positive small value such as 1, and the other variables can be initialized as 0's.
3. Repeat steps 4 and 5 for all variables until the convergence to a global energy solution calculated by average distance of point re-projection or until a maximum number of iterations.
4. Add the current variable with its step size. Note that both positive and negative value of the step size must be tried.
5. Decrease step-size of the current variable by half or by a ratio smaller than one until local energy of the variable is reduced or until the step size is too small, e.g., 1×10^{-8}.

Since for video recordings consecutive frames generally have a small difference in camera motion, the updated camera-pose of the current frame should not be too different from the previous frame. If a big difference is observed, the current camera's pose is reset to the previous one. Also, since all 3D-points should be in front of the camera only (i.e. all Z_w's should be negative), a swap in signs of Z_w must be done whenever Z_w becomes a positive number. Furthermore, the focal length must be positive and non-zero, so an absolute value is needed, to avoid negative focal length values. Also, if the focal length becomes zero, it is re-initialized to a positive value such as 1 instead.

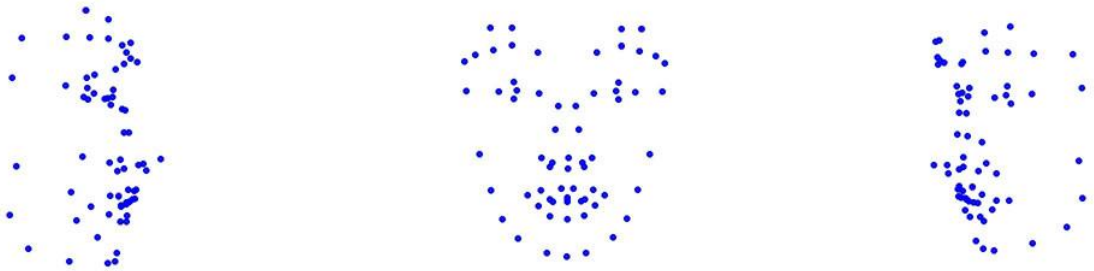
III. Experimental Results

A. Synthetic Environment

Experimentation on synthetic data can be used to test the accuracy both in two and three dimensions because the exact 3D information is known. For two dimensions, pixel-errors can be calculated from the displacement of point's re-projection. The accuracy in three dimensions can be demonstrated from both the reconstructed 3D-structure of a synthetic face and the camera pose of all video frames.

A video sequence is generated from 3DS Max with 225 frames in length, frame rate is 15 frames per second, and image frame size is 640×480 pixels. Some of the video frames and pixel-errors calculated from point re-projection are shown in Fig. 3.

(a) Frame 1, 122, and 225 respectively of the video sequence



(b) Corresponding displacement from point re-projection

Fig. 3 Input video frames and their corresponding pixel-errors calculated from re-projection

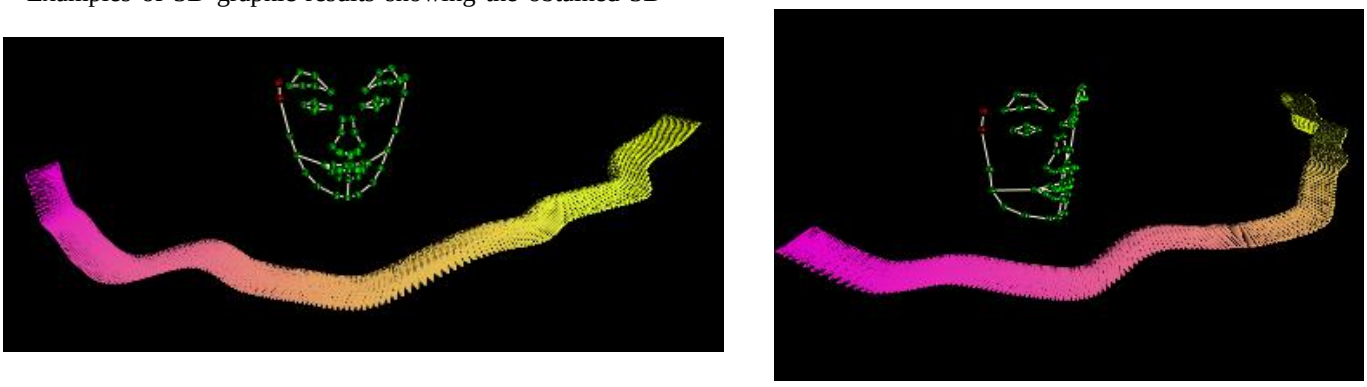
From Fig. 3(b), the pixel-errors are very low so that the observed (input) points and the re-projected ones are seen as overlaid in the same positions. The average pixel-error of this experiment is calculated from the Root Mean Square (RMS)

error as shown in Table I.

TABLE I
AVERAGE PIXEL-ERRORS FOR SYNTHETIC FACE

#Equations	#Variables	Iterations	Time(sec.)	Error(pixels)
30,600	1,549	3,500	407	2.54e-4

Examples of 3D graphic-results showing the obtained 3D-face and camera motion are displayed in the OpenGL environment as shown in Fig. 4. Since, in this experiment, a camera pans around a synthetic face from one side to the other, the camera motion of all video frames represented by several colored cones looks like a long ribbon in Fig. 4.



(a) Front view (b) Perspective view

Fig. 4 Example graphic-results showing the face model and camera motion in OpenGL's 3D environment

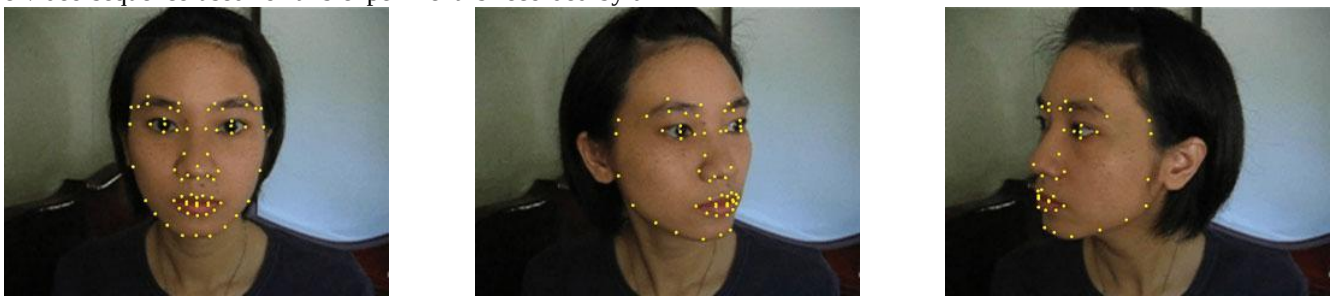
Numerical errors in three dimensions are shown in Table II. Note that, for calculating the 3D numerical-errors, the position and orientation of the first *calculated* camera must be transformed to be the same as that of the first *input* camera. The real size of the face model can be recovered by re-scaling it with a ratio between sizes of the input face and the calculated face.

TABLE II
AVERAGE 3D-ERRORS FOR SYNTHETIC FACE

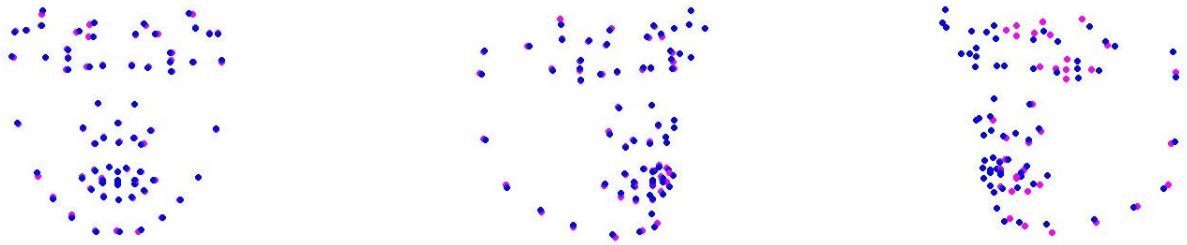
Orientation (degrees)			**Position (specified unit)**			**Face Model (specified unit)**		
r_x	r_y	r_z	t_x	t_y	t_z	X_w	Y_w	Z_w
0.015105	0.039641	0.031228	0.030606	0.003449	0.010833	0.002064	0.002856	0.003208

B. Real-World Environment

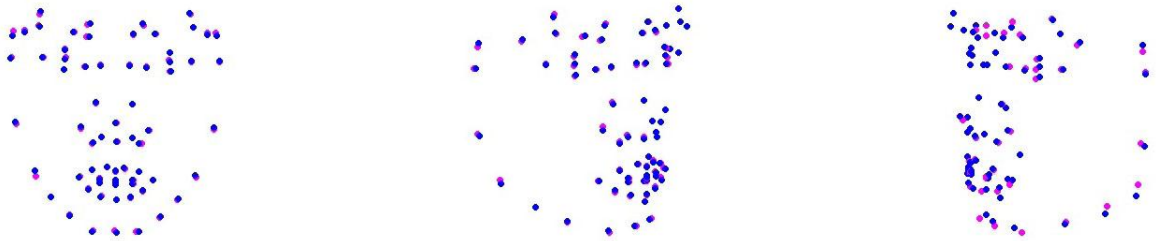
Since the precise distance between any two contiguous feature-points on a real person's face cannot be measured, only 2D errors calculated from displacement of point's re-projection are shown. In this experiment, the camera motion can be explained as that the camera pans around a human face; starts from the front of the person, then moves to the right of the face, and finally moves back until to the left of the face. The video sequence used for this experiment is recorded by a hand-held digital-camera with 128 frames in length, a frame rate of 15 frames per second, and a frame size of 320×240 pixels. Some examples of video frames and pixel-errors calculated from the intermediate and the final results are shown in Fig. 5. From Fig. 5, the two sets of colored points representing the observed and the re-projected points become closer after a greater numbers of iterations.



(a) Frame 1, 64, and 128 respectively of the video sequence



(b) Displacement from point re-projection after 500 iterations



(c) Displacement from point re-projection after 5,000 iterations

Fig. 5 Input video frames and their corresponding pixel-errors calculated from re-projection

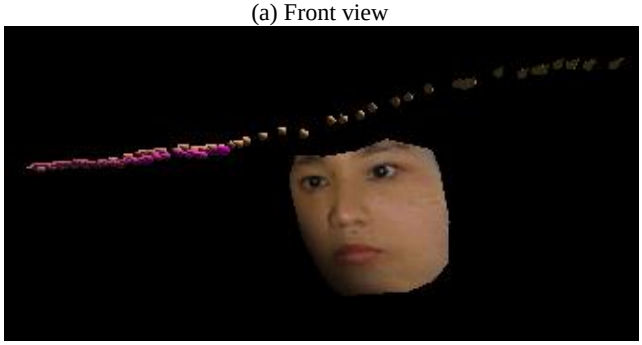
The average pixel-errors of this experiment are calculated after various numbers of iterations and shown in Table III.

TABLE III
AVERAGE PIXEL-ERRORS FOR REAL HUMAN'S FACE

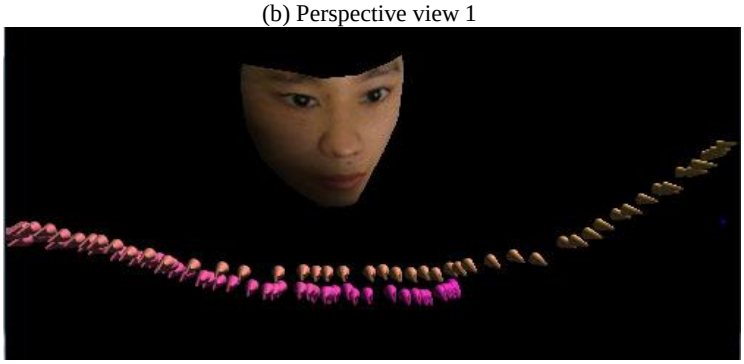
#Equations	#Variables	Iterations	Time(sec.)	Error(pixels)
14,836	967	500	32	3.59e-2
14,836	967	1,000	64	3.48e-2
14,836	967	3,500	221	2.60e-2
14,836	967	4,500	284	2.43e-2
14,836	967	5,000	315	2.41e-2

After the feature points of the facial parts are reconstructed, the Delaunay triangulation method introduced in [1], [2] is used to create a 3D mesh followed by texture-mapping and 3D-lighting using OpenGL libraries. The final 3D graphics-result showing the reconstructed 3D-face and the camera motion is displayed for different viewpoints in Fig. 6.

(a) Front view



(b) Perspective view 1



(c) Perspective view 2

Fig. 4 Example graphic-results showing the face model and camera motion in OpenGL's 3D environment

IV. CONCLUSION

From the experiments on the synthetic face and real human face, one may see that the missing image-points do not pose a problem in our proposed method for estimating 3D-face reconstruction and camera pose computation. The proposed approach is simple, practical, but yet robust.

REFERENCES

[1] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical Recipes – The Art of Scientific Computing," 3rd ed., Cambridge, 2007.

[2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," ACM Transactions on Mathematical Software, vol. 22, no. 4, pp. 469–483, Dec 1996.

[3] V. Chouvatut and S. Madarasmi, "A Comparison of Two Camera Pose Methods for Augmented Reality," 7th IASTED International Conference on Signal and Image Processing (SIP), pp. 554-559, 15-17 Aug 2005.
[4] V. Chouvatut and S. Madarasmi, "Estimation of Camera Pose for Use in Augmented Reality System," 20th International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC), Vol. 3, pp. 979-980, 4-7 Jul 2005.
[5] R.Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Camera and Lenses," IEEE Journal of Robotics and Automation, Vol. RA-3, Issue 4, pp. 323-344, Aug 1987.
[6] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System," Proceeding 2nd IEEE and ACM International Workshop on Augmented Reality, pp. 85-94, Oct 1999.
[7] T. Okuma, K. Sakaue, H. Takemura, and N. Yokoya, "Real-Time Camera Parameter Estimation from images for a Mixed Reality System," IEEE Proceeding 15th International Conference on Pattern Recognition, Vol. 4, pp. 482-486, 3-7 Sep 2000.
[8] R. I. Hartley, "Projective Reconstruction and Invariants from Multiple Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, Issue 10, pp. 1036-1041, Oct 1994.
[9] S. Avidan and A. Shashua, "Novel View Synthesis by Cascading Trilinear Tensors," IEEE Transactions on Visualization and Computer Graphics, Vol. 4, Issue 4, pp. 293-306, Oct-Dec 1998.
[10] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," 2nd ed., Cambridge, 2006.
[11] J. Li and R. Chellappa, "A Factorization Method for Structure from Planar Motion", IEEE Workshop on Motion and Video Computing (WACV/MOTIONS), Vol. 2, pp. 154-159, Jan 2005.
[12] N. B. Karayiannis, "Reformulated Radial Basis Neural Networks Trained by Gradient Descent", IEEE Transactions on Neural Networks, Vol. 10, Issue 3, pp. 657-671, May 2000.
[13] O.T.-C. Chen, "Motion Estimation Using a One-Dimensional Gradient Descent Search", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, Issue 4, pp. 608-616, Jun 2000.
[14] J.J. Guerrero and C. Sagues, "Estimating the Motion Direction from Brightness Gradient on Lines", IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 31, Issue 3, pp. 419-426, Aug 2001.
[15] L.M. Po, K.H. Ng, K.W. Cheung, K.M. Wong, Y. Uddin, and C.W. Ting, "Novel Directional Gradient Descent Searches for Fast Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19, Issue 8, pp. 1189-1195, Aug 2009.
[16] A. Smolic, "Robust Generation of 360-Degree Panoramic Views from Consumer Video Sequences", 4th EURASIP-IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications (VIPromCom), pp. 431-435, 16-19 Jun 2002.
[17] A.M. Sasson, "Combined Use of the Powell and Fletcher – Powell Nonlinear Programming Methods for Optimal Load Flows", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-88, Issue 10, pp. 1530-1537, Oct 1969.
[18] X. Xu and R.D. Dony, "Differential Evolution with Powell's Direction Set Method in Medical Image Registration", IEEE International Symposium on Biomedical Imaging: Nano to Micro, Vol. 1, pp. 732-735, 15-18 Apr 2004.
[19] G.J. Edwards, C.J. Taylor, and T.F. Cootes, "Interpreting Face Images using Active Appearance Models", 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pp. 300-305, 14-16 Apr 1998.
[20] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Models", International Proceedings European Conference on Computer Vision, Vol. 2, pp. 484-498, 1998.
[21] S. W. Park, J. Heo, and M. Savvides, "3D Face Reconstruction from a Single 2D Face Image," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR), pp. 1-8, 23-28 Jun 2008.
[22] Y. Zheng, J. Chang, Z. Zheng, and Z. Wang, "3D Face Reconstruction from Stereo: A Model Based Approach," IEEE International Conference on Image Processing (ICIP), Vol. 3, pp. III-65 - III-68, 16 Sep 2007 – 19 Oct 2007.
[23] Y. Zheng and Z. Wang, 2008, "Robust Depth Estimation for Efficient 3D Face Reconstruction," 15th IEEE International Conference on Image Processing, pp. 1516-1519, 12-15 Oct 2008.