Nonrigid Surface Modelling and Fast Recovery

Zhu Jianke

Supervisor: Prof. Michael R. Lyu
Committee: Prof. Leo J. Jia and Prof. K. H. Wong

Department of Computer Science and Engineering
The Chinese University of Hong Kong
May 11, 2007

Department of Computer Science and Engineering,
Faculty of Engineering,
The Chinese University of Hong Kong

Outline

- Introduction
- Two-stage Scheme with Active Appearance Models for nonrigid surface recovery
- Progressive Finite Newton Approach to nonrigid surface detection
- Conclusion

Introduction

- Nonrigid surface modelling and detection are essentially the computer vision tasks in a variety of applications:
  – image alignment
  – medial imaging
  – augmented reality
  – human computer interaction
  – digital entertainment

Nonrigid Surface Modelling

- Data embedding: Principal Component Analysis
  - Active Appearance Models
  - Active Shape Models
  - 3D Morphable Models
- Physical model: Finite Element Model (FEM)
  - Active Contours (Snakes)
- Thin-Plate Spline
  - TPS-RPM
  - Shape Context
  - Gaussian Mixture Models based points set registration

Nonrigid Surface Recovery

- Appearance-based Method
  - Also known as direct method
  - Lucas-Kanade algorithm based methods [LK IJCAI'81]
  - Active Appearance Models [Cootes PAMI'01]
  - 3D Morphable Models [Vetter Siggraph'99, PAMI'01]
  - 2D+3D AAMs [Xiao CVPR'04]
- Feature-based Method
  - Shape context [Belongie PAMI'02]
  - TPS-RPM [Hui CVIU'03]
  - Kernel correlation [ECCV'04]
  - Gaussian mixture models [Jian ICCV'05]
  - Semi-implicit optimization scheme [Pilet CVPR'05, IJCV'07]

Part I

A Two-stage Scheme for Nonrigid Surface Recovery with Active Appearance Models

Part I: Structure

- Motivation
- Extended Active Appearance Models (AAMs) fitting
- Two stage scheme for nonrigid surface recovery
  - Offline construction of 3D shape model
  - Estimate 3D pose and non-rigid shape parameters
- Experiemental Results
- Summary of Part I

Motivation

- Nonrigid shape recovery for Augmented Reality
  - Rigid Object
    - L. Vacchetti et al. (PAMI‘04) proposed an efficient solution for 3D rigid object tracking
    - Two 2D AAMs approach for rigid object pose estimation
  - Non-rigid Object
    - V.Blandz: 3D Morphable Models
    - J.Ahlberg: 3D AAM with generic Model.
    - Jing X. (CVPR‘05) 2D+3D AAM

Active Appearance Models

- AAMs is defined by its shape and texture

$$\mathbf{s} = (x_1, \ldots, x_n, y_1, \ldots, y_n)^{\mathbf{T}}$$

- The 2D shape and texture are controlled by a statistical model. They can be represented as a base plus a linear combination of variations:

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{P_s b_s} \qquad \mathbf{s} = \bar{\mathbf{s}} + \mathbf{P_s W_s^{-1} P_{cs} c}$$

$$\mathbf{t} = \bar{\mathbf{t}} + \mathbf{P_t b_t} \qquad \mathbf{t} = \bar{\mathbf{t}} + \mathbf{P_t P_{ct} c}$$

- Direct Appearance Models $\mathbf{b_s} = \mathbf{P_{cs} P_{ct}^{+} b_t}$

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{Q_s b_t} \qquad \mathbf{Q_s} = \mathbf{P_s W_s^{-1} P_{cs} P_{ct}^{+}}$$

$$\mathbf{t} = \bar{\mathbf{t}} + \mathbf{P_t b_t}$$

Extended AAMs Fitting Algorithm

- Object:

$$|| I - Im||$$

- Idea: the proposed approach predicts shape directly from texture

$$\delta \mathbf{t} = \mathbf{R}_{\mathbf{t}} \mathbf{r} \ ,$$

$$\delta \mathbf{b}_{\mathbf{t}} = \mathbf{R}_{\mathbf{g}} \mathbf{r} \ .$$

where r = g*i* -g*m*, is the residual image

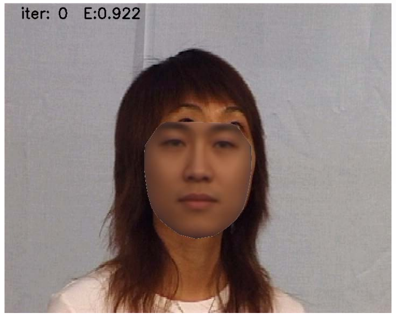
Extended AAM Fitting

The algorithm of AAM Matching

1. Generate texture vector $\mathbf{g}_m$ from model
2. Sample image below the model shape $\mathbf{g}_i$
3. Evaluate error vector $\mathbf{r} = \mathbf{g}_i - \mathbf{g}_m$ and error $\mathbf{E} = |\mathbf{r}|$
4. Compute displacements in pose $\delta\mathbf{t} = \mathbf{R_t r}$
5. Compute displacements in texture $\delta\mathbf{b_t} = \mathbf{R_g r}$
6. Update pose and texuture parameters with initial $k = 1$
7. Transform the shape by the estimated parameters
8. Repeat step 1-3 to form a new error $\mathbf{E}^{'}$
9. If $\mathbf{E}^{'} < \mathbf{E}$ accept the new estimate,
   otherwise goto step 6 to try other k=0.5, 0.25, ...

AAM Fitting Sample

The AAMs are built up with 140 still face image belonging to 20 individuals, 7 images for each. The fitting experiment is performed on an AAM with 14 shape parameters, 68 texture parameters, and 36335 color pixels.



Two-stage Scheme Nonrigid Surface Recovery

- Training and building offline basis

  Acquire the 2D shape of objects using the AAM fitting algorithm, then construct the 3D shape basis.

- Online tracking
  - Step 1. 2D AAM fitting and tracking
  - Step 2. estimate the 3D pose and shape parameters simultaneously via local bundle adjustment by building up the point correspondences between 2D and 3D.

Building Offline 3D Model

- 3D shape

$$\mathbf{S} = \mathbf{S_0} + \sum_{i=1}^{m} p_i \mathbf{S_i} \qquad \mathbf{S}, \mathbf{S_i} \in R^{3\times n}, p_i \in R$$

- Weak-perspective projection

$$\begin{bmatrix} u_1 & u_2 & \cdots & u_n \\ v_1 & v_2 & \cdots & v_n \end{bmatrix} = \mathbf{R} \cdot (\sum_{i=0}^{m} p_i \mathbf{S_i}) + \mathbf{T}$$

Building Offline 3D Model

- Factorization method [Bregler CVPR'00, CVPR'01]

$$W = \begin{bmatrix} u_1^1 & u_2^1 & \cdots & u_n^1 \\ v_1^1 & v_2^1 & \cdots & v_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ u_1^N & u_2^N & \cdots & u_n^N \\ v_1^N & v_2^N & \cdots & v_n^N \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{R}_1 & p_1^1\mathbf{R}_1 & \cdots & p_m^1\mathbf{R}_1 \\ \mathbf{R}_2 & p_1^2\mathbf{R}_2 & \cdots & p_m^2\mathbf{R}_2 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{R}_N & p_1^N\mathbf{R}_N & \cdots & p_m^N\mathbf{R}_N \end{bmatrix}}_{\mathbf{M}} \cdot \underbrace{\begin{bmatrix} \mathbf{S_0} \\ \mathbf{S_1} \\ \vdots \\ \mathbf{S_m} \end{bmatrix}}_{\mathbf{B}}$$

$$\mathbf{M} = \tilde{\mathbf{M}} \cdot \mathbf{G}$$

$$\mathbf{B} = \mathbf{G}^{-1} \cdot \tilde{\mathbf{B}}$$

Online Algorithm

- The optimization problem can be derived as:

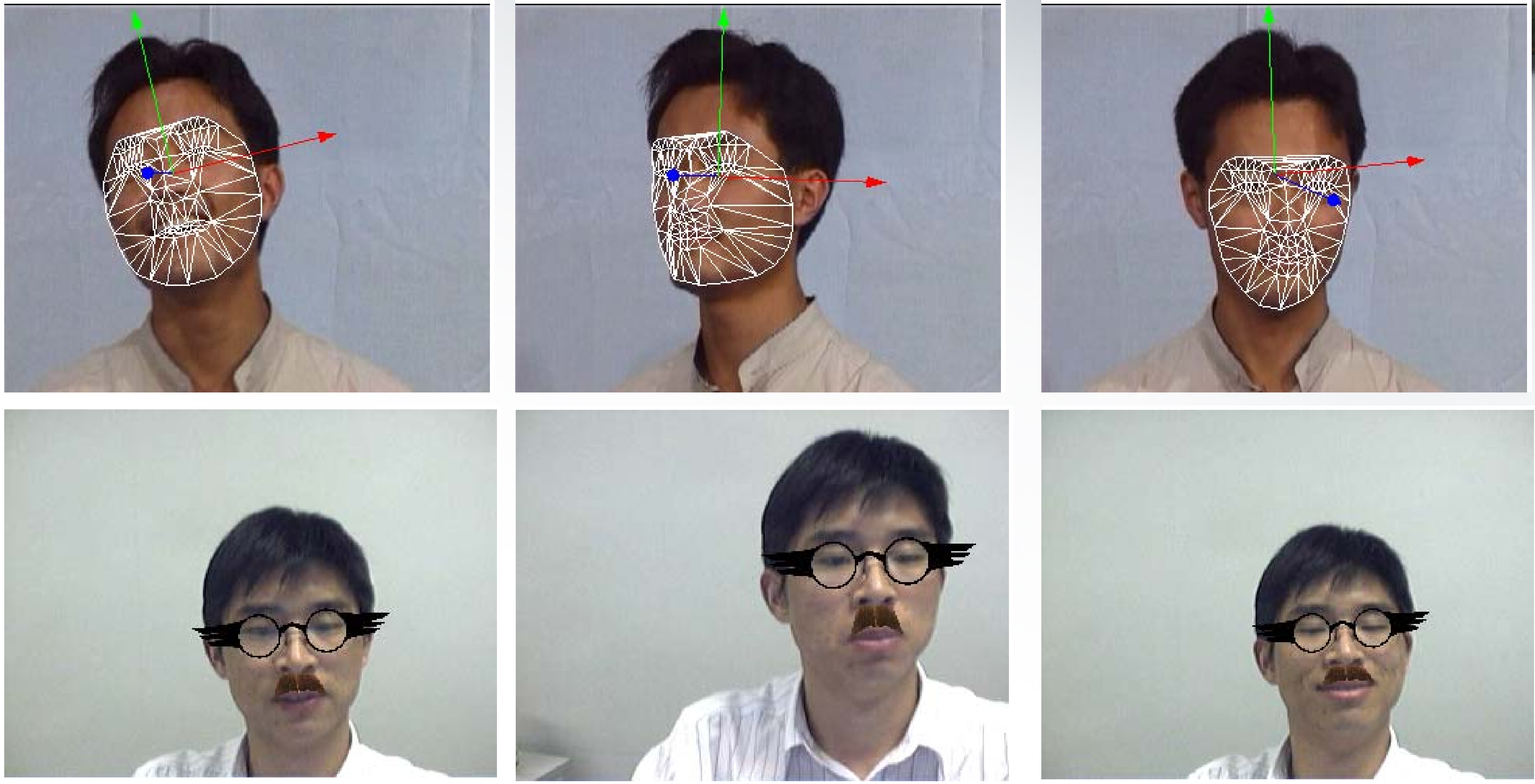
$$\min_{\mathbf{R},\mathbf{T},p} \rho\left(\mathbf{s}, \phi\left(\mathbf{A}[\mathbf{R}|\mathbf{T}], \mathbf{S_0} + \sum_{i=1}^{m} p_i \mathbf{S_i}\right)\right)$$

where

$$\rho(u) = \begin{cases} \frac{\alpha^2}{6}[1-(1-(\frac{u}{\alpha})^2)^3], & |u| \le \alpha \\ \frac{\alpha^2}{6}, & |u| > \alpha \end{cases}$$

$\phi\left(\mathbf{A}[\mathbf{R}|\mathbf{T}], \mathbf{S_0} + \sum_{i=1}^{m} p_i \mathbf{S_i}\right)$ denotes the projection of 3D shape given the parameters A, R and T.

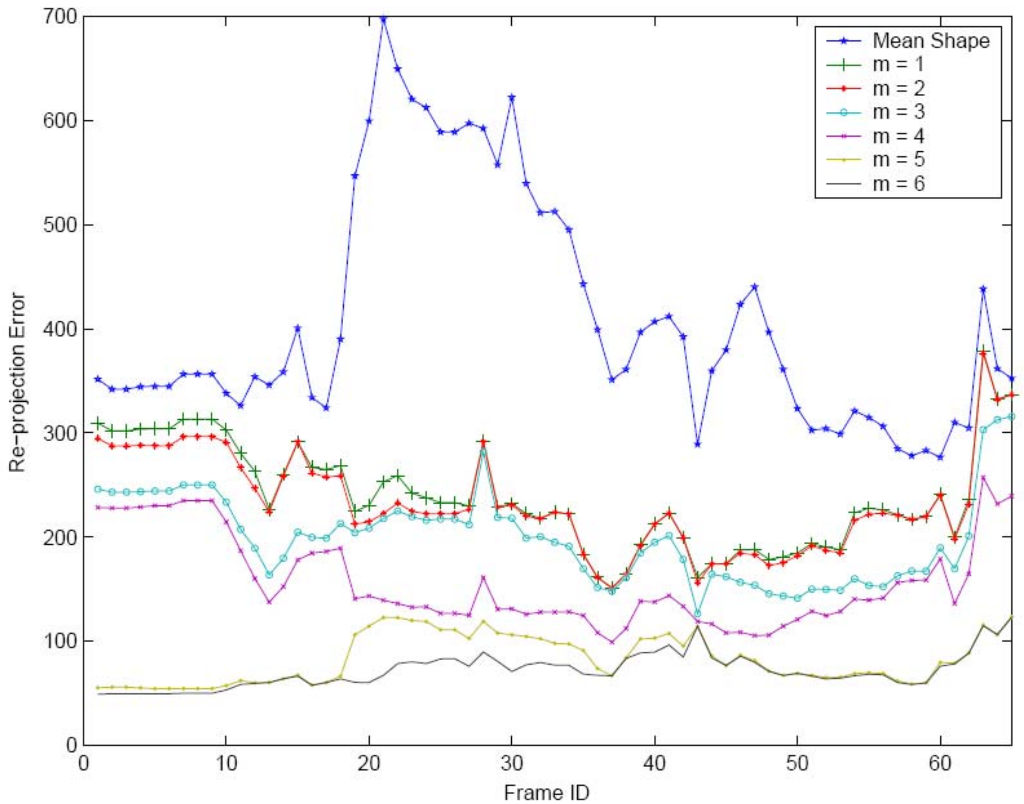
Experiemental Results I



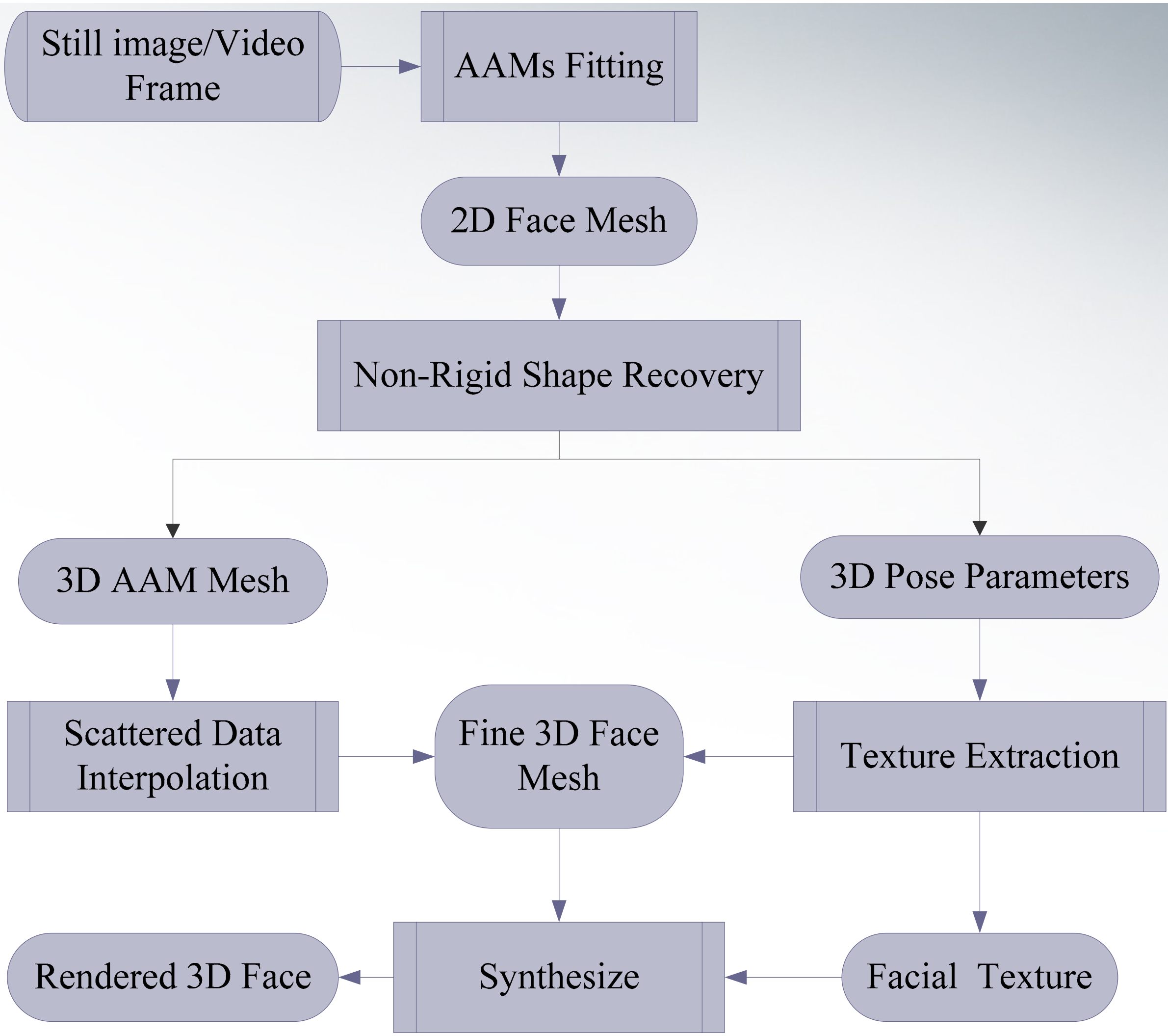
Pentium III 1GHz CPU, 200ms per image of size 352x288. AAM fitting takes 40ms and 3D recovery step takes 74ms. The AAM with 10 shape parameters, 52 texture parameters. 6 camera parameters and 6 3D shape parameters.

Experimental Results II

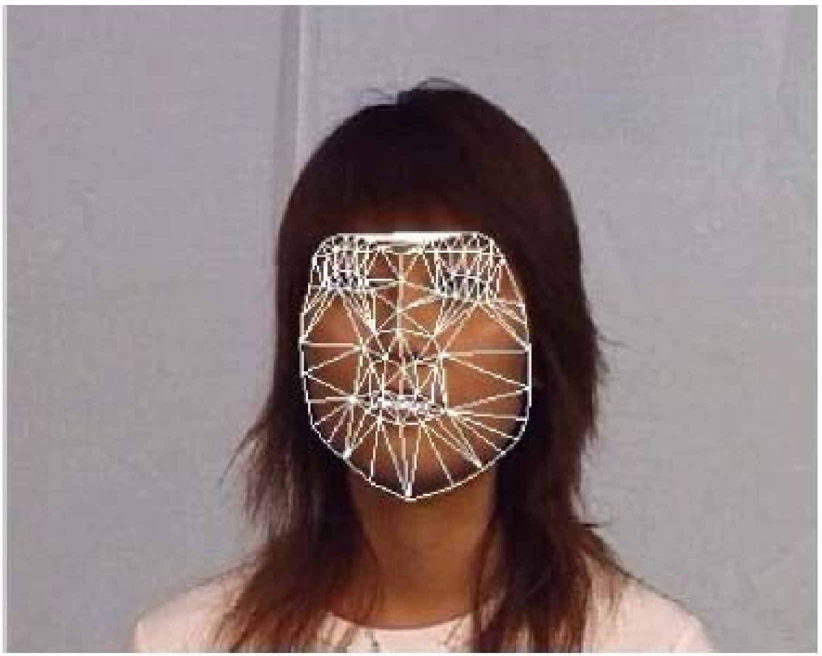
- Determine the number of 3D shape basis

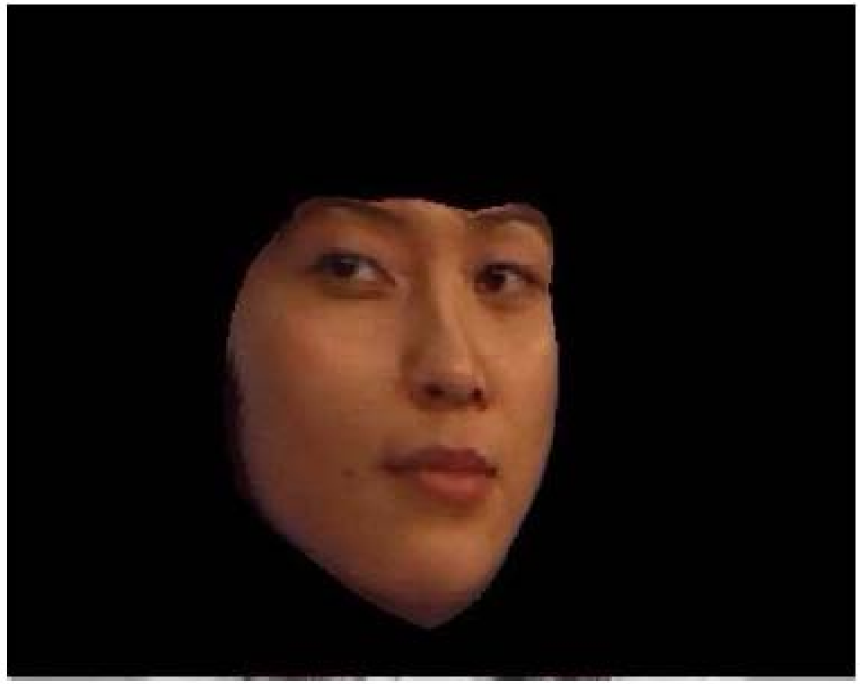


Automatic Face Modelling Scheme

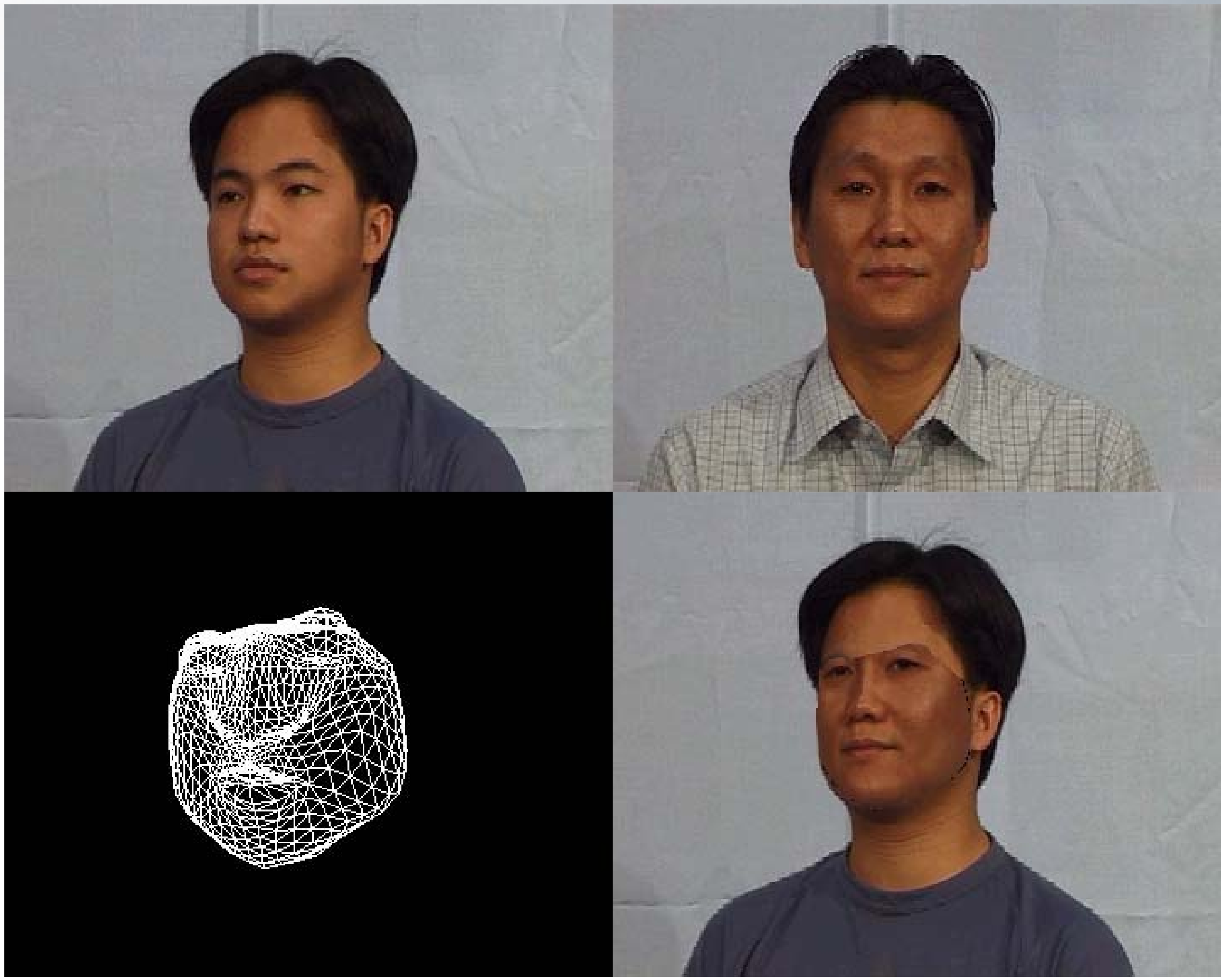


Rendering in Different View





Re-texturing face



Discussion

- Rigid vs. Non-Rigid
  - The prior model employed by L. Vacchetti et al. is only for rigid objects or deformable objects with small variations.
  - P. Mittrapiyanumic et al. do not take full advantage of AAM's deformation power
- Offline vs. Online
  - In contrast to the offline algorithms, our proposed method is able to work online by exploiting the 3D shape models that can be constructed offline effectively by AAM tracking
- Advantages for AR applications
  - Generic vs. person specific: handle large texture variations, fitting to different individuals
  - Combined 2D+3D AAM: weak-perspective model
- Disadvantages and Future Work
  - does not take full advantage of 3D information for speeding up AAM convergence.
  - Large rotation may be compensated by the 3D linear mode, therefore, the estimated pose is not so accurate.
  - Training the 3D AAM with the aligned 3D shapes instead of 2D shapes.

Summary of Part I

- A novel two-stage scheme for online non-rigid shape recovery toward Augmented Reality applications using AAMs.
- Obtain unbroken point correspondences across multiple frames to construct 3D shape models
- Provide 2D to 3D vertex correspondences in the online tracking.
- An efficient algorithm is proposed to estimate both 3D pose and non-rigid shape parameters via local bundle adjustment.

Part II

Progressive Finite Newton Approach to Nonrigid Surface Detection

Part II: Structure

- Motivation
- Progressive Finite Newton Approach to nonrigid surface detection
  - 2D nonrigid surface model
  - Feature-based nonrigid surface recovery
  - Finite Newton formulation
  - Optimization
- Experiemental Results
  - Computational efficiency and nonrigid surface detection
  - Augmented reality
  - Medical image registration
- Summary of Part II

Motivation

- **Nonrigid surface detection**: recovering the explicit surface with a few deformation parameters and finding out the correct correspondences from noisy data simultaneously.
- Unlike the rigid object, it is difficult to directly employ a robust estimator to remove the spurious matches for nonrigid surface detection.
- The iterative methods, such as TPS-RPM [CVIU’03] and Shape Contex [PAMI’02], are either sensitive to initial conditions and parameter choices, or involve too many iterations and a complex optimization procedure.
- Semi-implicit method: an automated approach, and can be applied for the real-time Augmented Reality. [Pilet et al CVPR’05, IJCV’07]

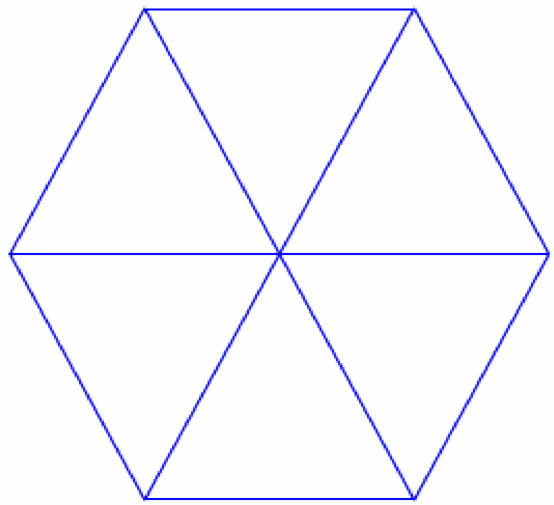
Robust Methods

- RANSAC
- PROSAC
- M-estimator
- Hough transform

2D Nonrigid Surface Model

$$\mathbf{s} = \begin{bmatrix} \mathbf{x} & \mathbf{y} \end{bmatrix}^\top = \begin{bmatrix} x_1 & x_2 & \dots & x_N & y_1 & y_2 & \dots & y_N \end{bmatrix}^\top$$

$$T_{\mathbf{s}}(\mathbf{m}) = \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \end{bmatrix} \begin{bmatrix} \xi_1 & \xi_2 & \xi_3 \end{bmatrix}^\top$$



Where (ξ_1, ξ_2, ξ_3) are the barycentric coordinates for the point m.

Department of Computer Science and Engineering,
Faculty of Engineering,
The Chinese University of Hong Kong

Nonrigid Surface Recovery

- Energy function

$$E(\mathbf{s}) = E_e(\mathbf{s}) + \lambda_r E_r(\mathbf{s})$$

  - Ee(s) is the sum of the weighted square error residuals for the matched points.

$$E_e(\mathbf{s}) = \sum_{\mathbf{m} \in M} \omega_{\mathbf{m}} \mathcal{V}(\delta, \sigma)$$

  - *Er*(s) is the regularization term that represents the surface deformation energy
  - λ_r is a regularization coefficient.

Regularization Term

- Er(s), also known as internal force in Snakes, is composed of the sum of the squared second-order derivatives of the mesh vertex coordinates.

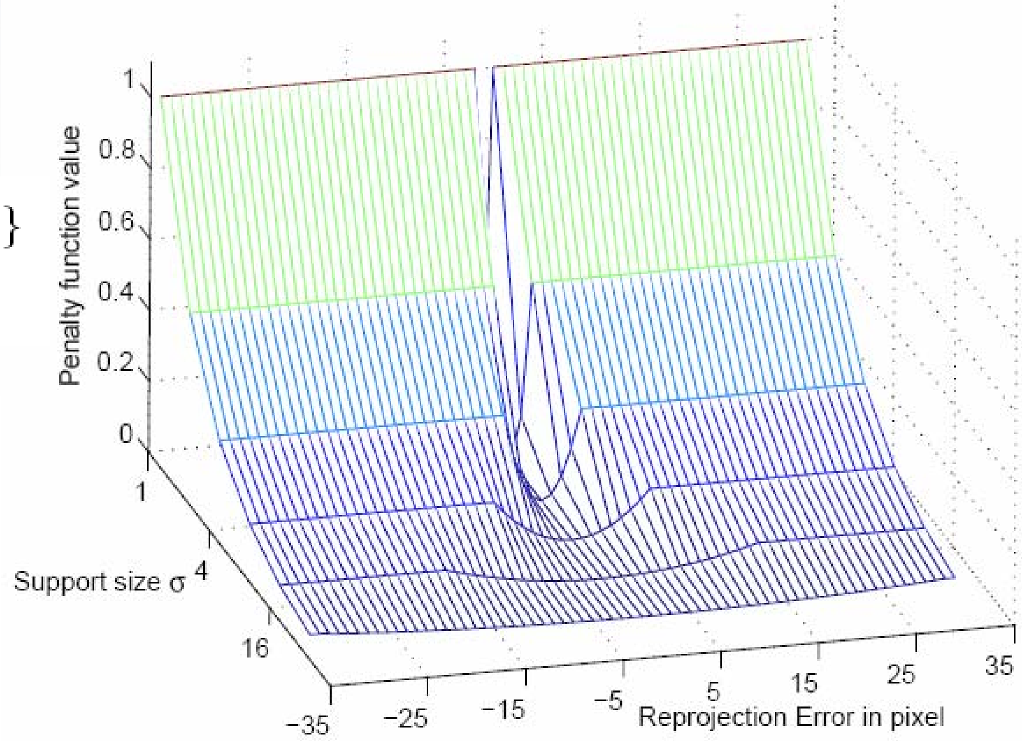
$$E_r(\mathbf{s}) = \mathbf{s}^\top \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \mathbf{s}$$

where K is a sparse and banded matrix which is determined by the structure of the mesh model.

Robust Estimator

- Define the residual error $\delta = \mathbf{m}_1 - T_\mathbf{s}(\mathbf{m}_0)$
- The robust estimator

$$\mathcal{V}(\delta,\sigma) = \begin{cases} \frac{\|\delta\|}{\sigma^n}, & M_1 = \{\mathbf{m} \mid \ \|\delta\| \le \sigma^2\} \\ \sigma^{2-n}, & M_2 = \overline{M_1} \end{cases}$$



*M*1 contains the inlier matches
*M*2 is the set of the outliers
The order n determines the scale of the residual.

Finite Newton Formulation

- The robust estimator function is not convex
- The associated penalty function approximation problem becomes a hard combinational optimization problem
- Tackle this problem under the finite Newton optimization framework

Finite Newton Formulation

- An augmented vector: $\mathbf{t}_i = \xi_1 \quad \mathbf{t}_j = \xi_2 \quad \mathbf{t}_k = \xi_3$
- Then,

$$\begin{aligned} ||\delta|| &= (u - \mathbf{t}^\top \mathbf{x})^2 + (v - \mathbf{t}^\top \mathbf{y})^2 \\ &= u^2 + v^2 - 2(u\mathbf{t}^\top \mathbf{x} + v\mathbf{t}^\top \mathbf{y}) + \mathbf{x}^\top \mathbf{t}\mathbf{t}^\top \mathbf{x} + \mathbf{y}^\top \mathbf{t}\mathbf{t}^\top \mathbf{y} \end{aligned}$$

where (*u, v*) are the coordinates of m1.

Re-write the error term as

$$E_e = \sum_{\mathbf{m} \in M_1} \frac{\omega_{\mathbf{m}}}{\sigma^n} \left(u^2 + v^2 - 2 \begin{bmatrix} u\mathbf{t} \\ v\mathbf{t} \end{bmatrix}^\top \mathbf{s} + \mathbf{s}^\top \begin{bmatrix} \mathbf{t}\mathbf{t}^\top & 0 \\ 0 & \mathbf{t}\mathbf{t}^\top \end{bmatrix} \mathbf{s} \right) + q\sigma^{2-n}$$

where q is the number of outliers

Finite Newton Formulation

- Define vector b as:

$$\mathrm{b} = \begin{bmatrix} \mathrm{b}_x \\ \mathrm{b}_y \end{bmatrix} = \sum_{\mathrm{m} \in M_1} \frac{\omega_{\mathrm{m}}}{\sigma^n} \begin{bmatrix} u\mathrm{t} \\ v\mathrm{t} \end{bmatrix}$$

- Matrix A (N by N):

$$A = \sum_{\mathrm{m} \in M_1} \frac{\omega_{\mathrm{m}}}{\sigma^n} \mathrm{t}\mathrm{t}^\top$$

- Energy function

$$E = \mathrm{s}^\top \begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \mathrm{s} - 2\mathrm{b}^\top \mathrm{s} + q\sigma^{2-n} + \sum_{\mathrm{m} \in M_1} \frac{\omega_{\mathrm{m}}}{\sigma^n}(u^2 + v^2)$$

Finite Newton Formulation

- The finite gradient of the energy function *E* with respect to s

$$\nabla = 2\left(\begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \mathrm{s} - \begin{bmatrix} \mathrm{b}_x \\ \mathrm{b}_y \end{bmatrix}\right)$$

- Hessian

$$H = 2\begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix}$$

- Rewrite gradient $\nabla = H\mathrm{s} - 2\mathrm{b}$

Finite Newton Formulation

- Update equation: $\mathbf{s} \leftarrow \mathbf{s} - \gamma H^{-1}\nabla$
- Set r = 1, and obtain the linear equation

$$H\mathbf{s} = \mathbf{b}$$

$$\begin{bmatrix} \lambda_r K + A & 0 \\ 0 & \lambda_r K + A \end{bmatrix} \mathbf{s} = \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \end{bmatrix}$$

$$\mathbf{x} = (\lambda_r K + A)^{-1}\mathbf{b}_x$$

$$\mathbf{y} = (\lambda_r K + A)^{-1}\mathbf{b}_y$$

Optimization

- Coarse-to-fine scheme to deal with large outliers
  - Support σ of robust estimator is progressively decayed at a constant rate
  - λ_r is kept constant
  - Result is used as the initial state for next minimization
  - Stop when σ is close to the expected precision
  - Report a successful detection when the number of inlier matches is above a given threshold.

Optimization

- Start from a sufficiently large support σ
  - To avoid getting stuck at local minima
  - Needs a few iterations to compensate for the errors due to pose variations
- Modified RANSAC
  - Closed-form solution
  - Draw from progressively larger sets of top-ranked correspondences
  - Sample size

Experimental Setup

- In order to register the mesh model conveniently, a model image is acquired when the nonrigid surface contains no deformation.
- A random-trees based method [Lepetit PAMI’06] is used to build the correspondences between the model image and input image.
- A set of synthetic data is used to select the parameters
- The best regularization coefficient is found by grid searching.
- n = 4, decay rate is 0.5
- Pentium-4 3.0GHz PC with 1GB RAM
- A DV camera size of 720x576

Computational Efficiency

- The complexity of the proposed method is mainly determined by the number of vertices N
- Another factor is the number of inlier matches
- Proposed method: 8 iterations, around 18 frames per second
- Semi-implicit method [CVPR'05, IJCV'07]: 40 iterations, about 9 frames per second on Coffee mat video

Table 3.1. Computational time of proposed method at each step.

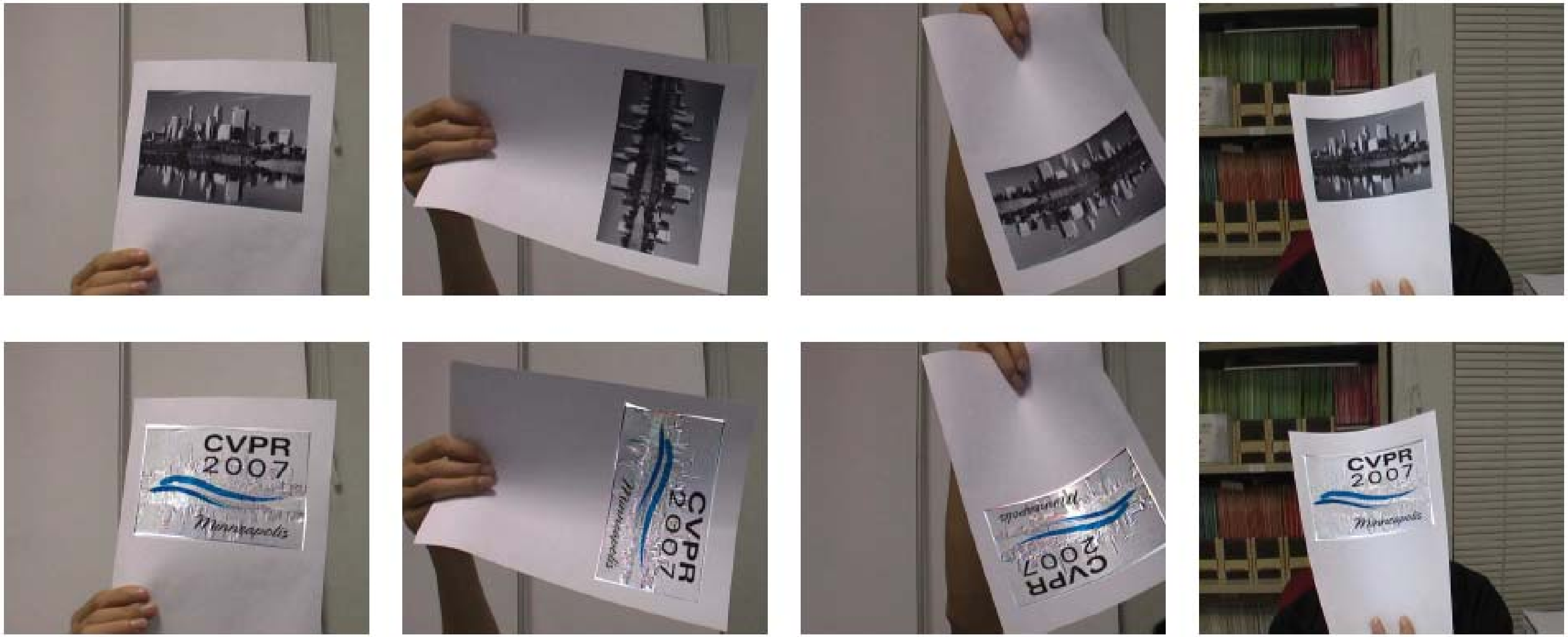
Total	Match	Optimization	Iteration	Other
57ms	27ms	14ms	$\sim$ 1.9ms	16ms

Coffee mat [Video demo]

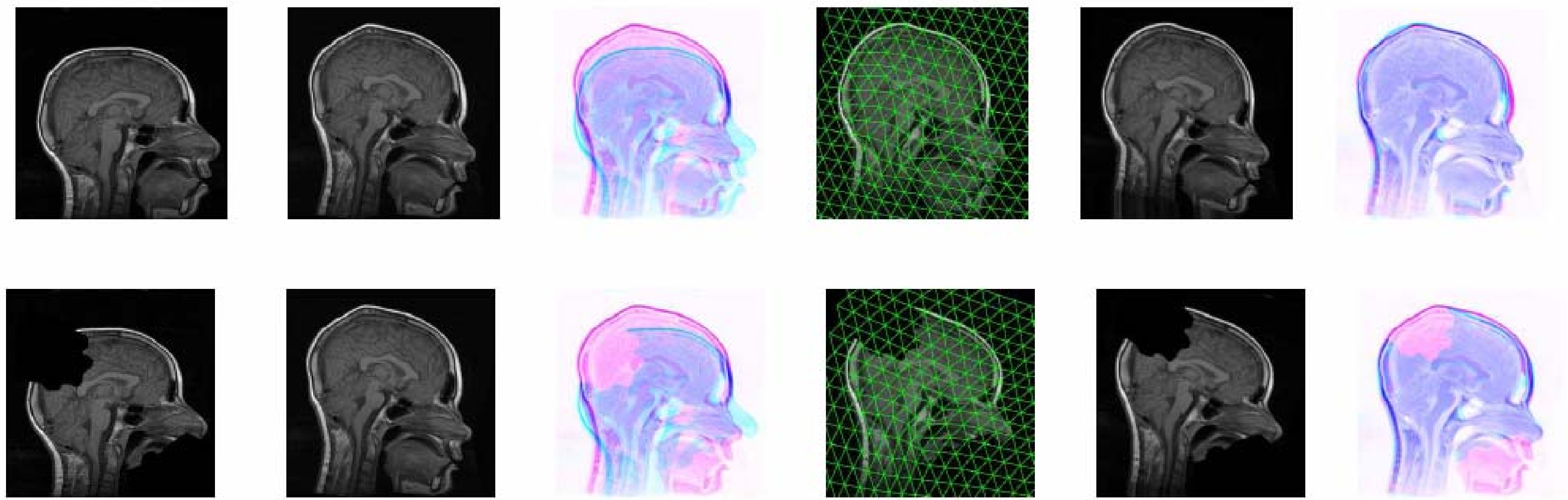
(a) Model image (b) Result (c) Result (e) Plastic cup

Re-texturing a T-shirt [Video demo]

Paper [Video demo]



Medical image



Summary

- A novel progressive scheme to solve the non-rigid surface detection problem
- In contrast to the previous semi-explicit method, we directly solve the unconstrained quadratic optimization problem by an efficient factorization
- Modified RANSAC scheme can handle high-dimensional spaces with noisy data.
- We have conducted extensive experimental evaluations on diverse objects with different materials.
- The proposed method is very fast and robust, and can handle large deformations and illumination changes.

Conclusion

- Two different approaches to nonrigid surface recovery have been investigated
- Appearance-based method:
  - Using more information, may be more accurate
  - Offering image coding capability
  - Tend to be computational expensive
  - Easy to stuck at the local optima
- Feature-based method
  - Automatic solution
  - Essentially fast
  - Taking advantage of the advances in feature matching and object recognition
  - Errors occur in the region lacking texture

Future work

- Fusion approach
- Explore new regularization method
- Find fast and accurate feature matching algorithm
- Including more features, such as edges and silhouette

The End

Thank you

References

- T. Cootes, G. Edwards, and C. Taylo. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6), June 2001.
- J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 535–542, 2004.
- L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 26(6), Oct. 2004.
- J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration, and realistic augmentation. *Int'l J. Computer Vision*, 2007.
- J. Zhu, S. C. Hoi, E. Yau, and M. R. Lyu. Automatic 3d face modeling using 2d active appearance models. In *Proc.13th Pacific Conf. Computer Graphics and Applications*, pages 133–135, 2005.
- J. Zhu, S. C. Hoi, and M. R. Lyu. Real-time non-rigid shape recovery via active appearance models for augmented reality. In *Proc. European Conf. Computer Vision*, pages 186–197, 2006.
- J. Zhu and M. R. Lyu. Progressive finite newton approach to real-time nonrigid surface detection. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2007.
- J. Zhu, S. C. Hoi, and M. R. Lyu. A multi-scale tikhonov regularization scheme for implicit surface modelling. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2007.