A Head Pose Tracking System using RGB-D Camera

Songnan Li, King Ngi Ngan, Lu Sheng
The Chinese University of Hong Kong

Outline

- Introduction
  - Applications
- System overview
- Head movement prediction
  - K-means-like training of multiple predictors
- Conclusion

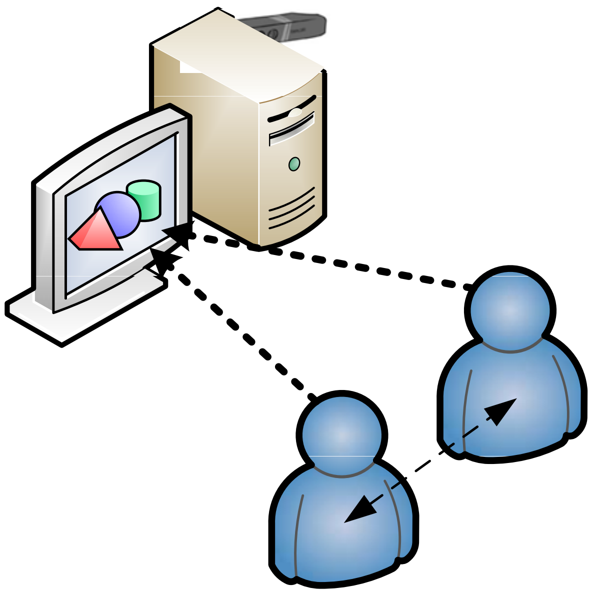
Introduction

- Using RGB-Depth camera (Kinect), we developed a head pose tracking system which can work in real-time (more than 30 fps without specific optimization).
- Demo: http://www.youtube.com/watch?v=UzqTuzqR6tE

Introduction

- Applications
  - *Free-viewpoint displaying.*

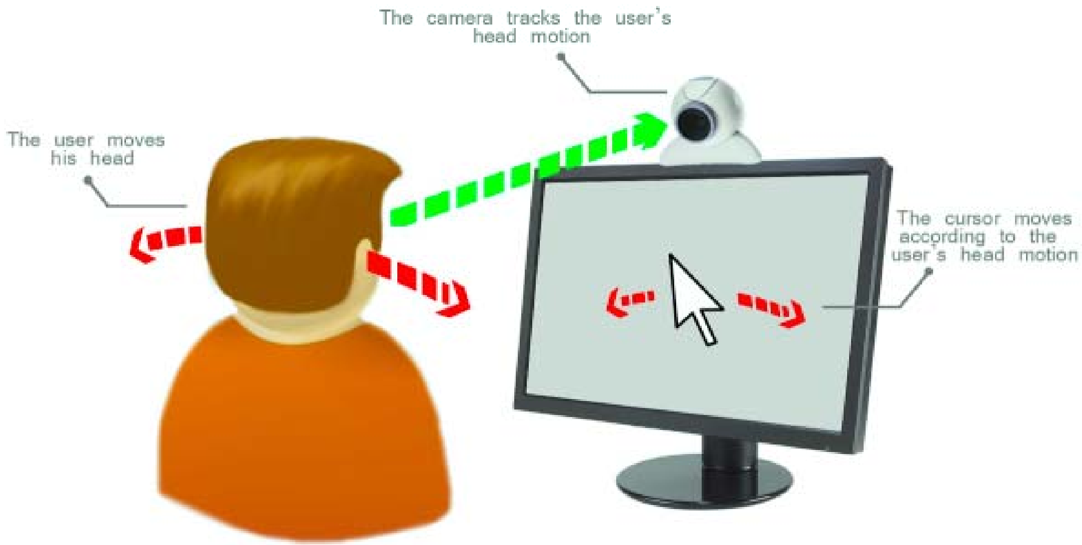
    Display video/graphics according to the user's viewing direction.



Introduction

- Applications (cont.)
  - *Human-computer interface*

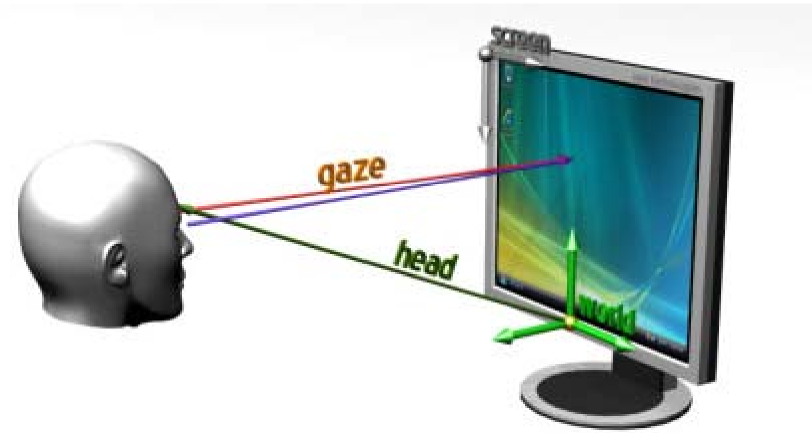
    Head movement can be used to control cursor.



Introduction

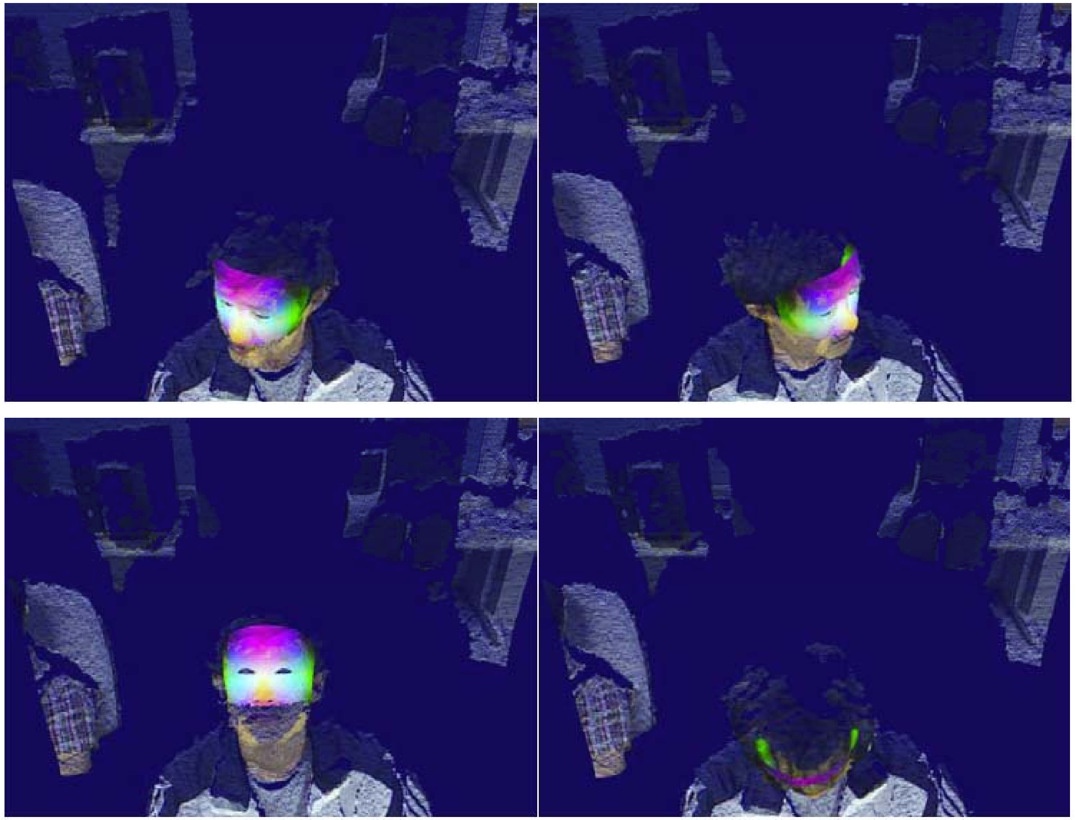
- Applications (cont.)
  - *Gaze estimation*

    A pre-processing step for head-pose-free gaze estimation.



System overview

- Basic idea: register a user-specific 3D face mask to point cloud captured by Kinect.



System overview

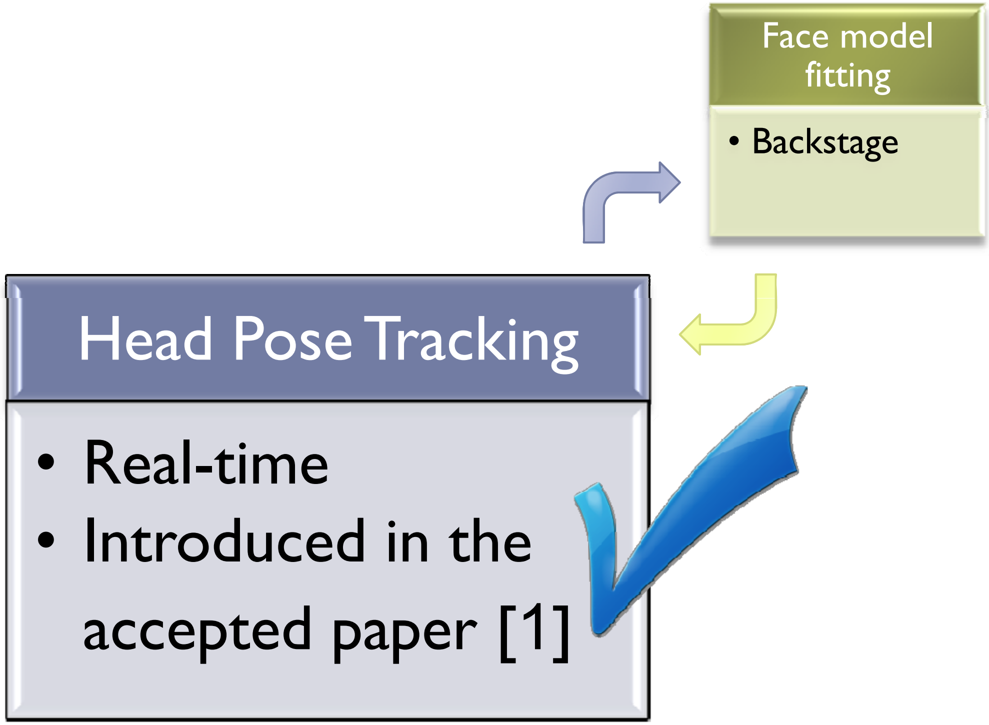
- Two technical components

Face model fitting
for arbitrary user

Head pose tracking
using user-specific face mask

System overview

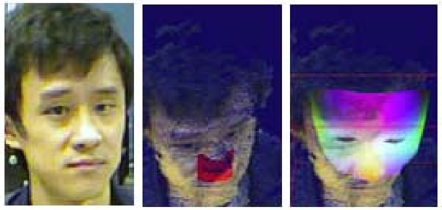
- Two technical components (cont.)



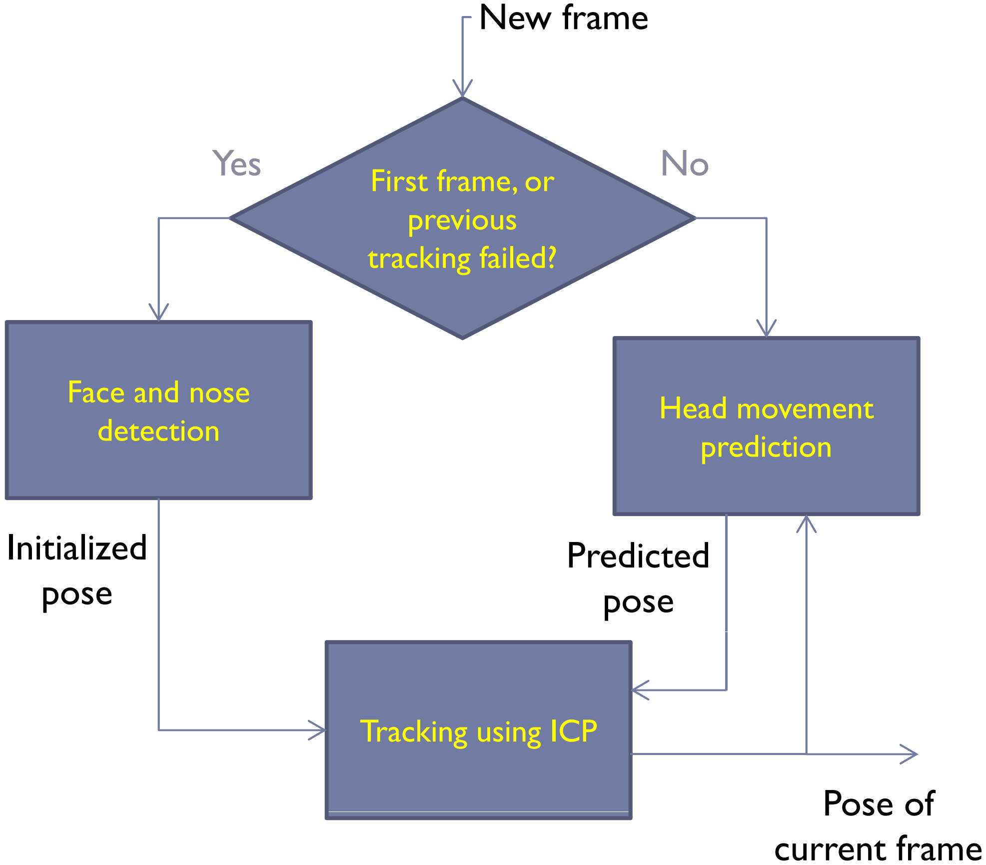
[1] Songnan Li, King Ngi Ngan, Lu Sheng, "A Head Pose Tracking System Using RGB-D Camera", The 9th International Conference on Computer Vision Systems (ICVS2013), St. Petersburg, Russia, 16-18, July 2013.

System overview

- Head pose tracking
  - Framework



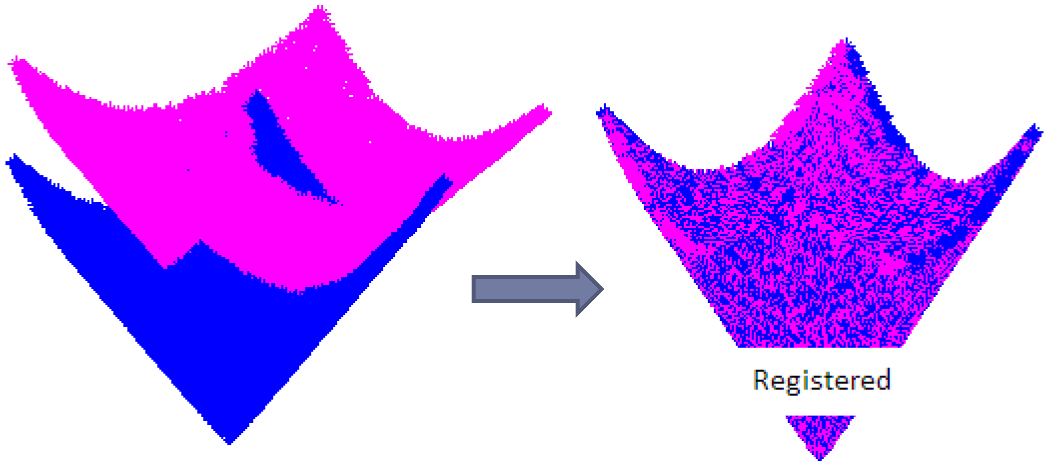
P.Viola, M. J. Jones, "Robust Real-time Face Detection", International Journal of Computer Vision, 57(2), 137–154, 2004.



System overview

- Tracking using ICP
  - ICP - Iterative Closest Point algorithm

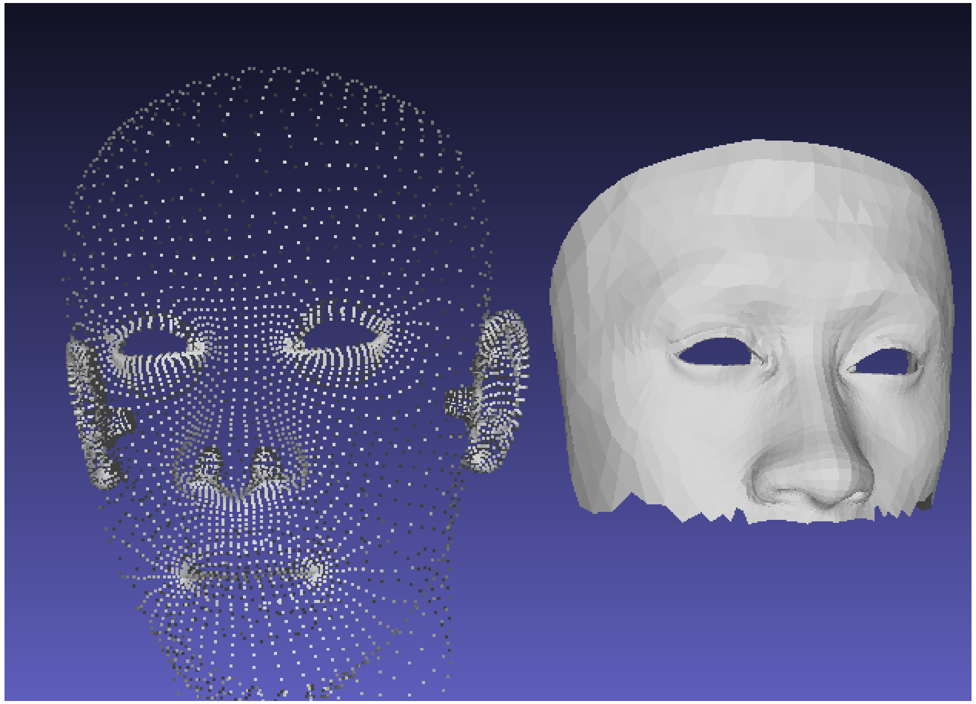
    e.g., P. J. Basl, N. D. McKay, "A method for registration of 3-D shapes", TPAMI, 14(2), 239-256, 1992. (8359 citations, June 2013)



System overview

- Tracking using ICP
  - ICP - Iterative Closest Point algorithm

    e.g., P. J. Basl, N. D. McKay, "A method for registration of 3-D shapes", TPAMI, 14(2), 239-256, 1992. (8359 citations, June 2013)

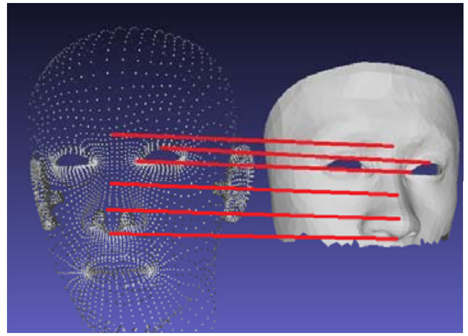


System overview

- Tracking using ICP
  - ICP algorithm generally consists of two essential steps

Repeat *until converge:*

1. *Matching points* $\{d_i, f_i\}, i \in [1,2,\ldots,N]$
   - *Closest neighbor*
   - *Projective data association*
   - *Normal shooting*
   - …
2. *Calculating the transformation matrix T by minimizing an error metric*
   - *Point-to-point* → $T = \arg\min_T \sum_i \| T \times f_i - d_i \|^2$
   - *Point-to-plane* → $T = \arg\min_T \sum_i \| q_i \cdot (T \times f_i - d_i) \|^2$
   - *Plane-to-plane*
   - *Generalized-ICP*
   - …



d_i — f_i

Head movement prediction

- Why use head movement prediction
  - Given the previous tracking results, predict the head pose of the current frame.

$$\{T_{k-1}, T_{k-2}, \ldots\} \Rightarrow T_k$$

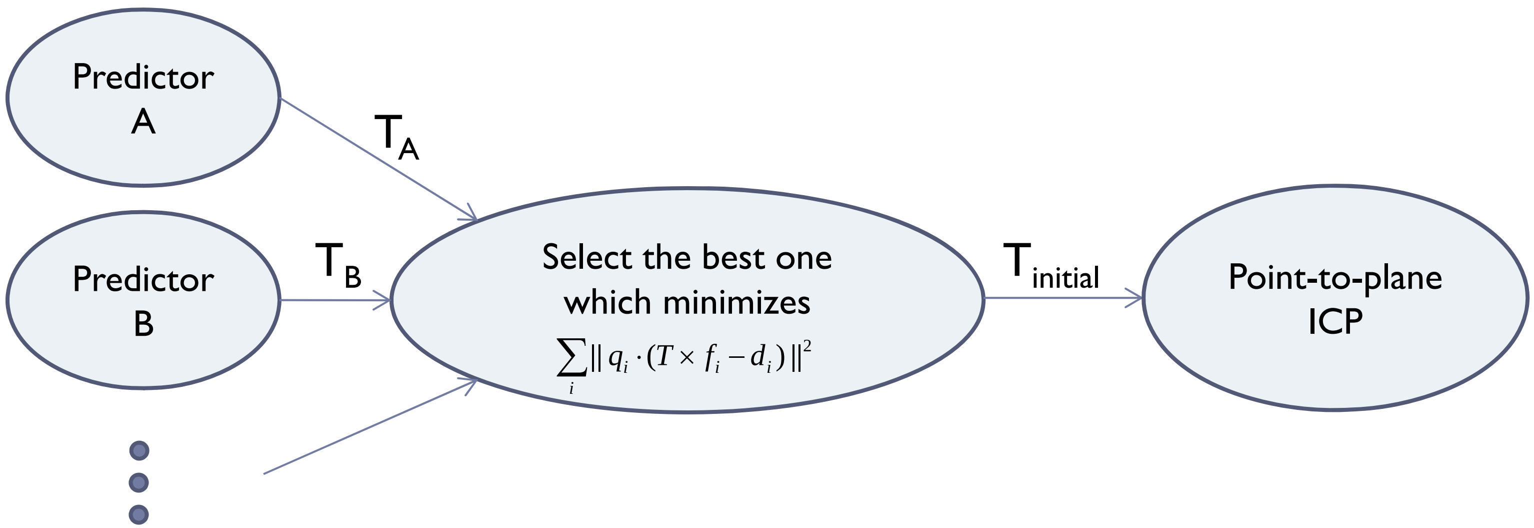
  - Head movement prediction provides a good initial transformation matrix for ICP, so that ICP can converge robustly even when head motion is large.
  - Furthermore, it can help ICP converge faster, therefore, improve the system's efficiency. For example, the processing speed of our system is increased from 25 fps to 33 fps approximately by using the proposed prediction method.

Head movement prediction

- Why propose a new method
  - To predict movements, Kalman filter is typically used. But Kalman filter assumes linearity of the system. Head movement, on the other hand, is nonlinear.
  - For nonlinear movements, particle filter can be applied. But particle filter needs a large number of samples to estimate a probability distribution, which makes it time consuming.
  - We propose a new method that uses only three samples and can provide quite accurate prediction results.

Head movement prediction

- Motivation
  - In our system, several predictors can be used together. Each predictor provides a transformation matrix. The one that minimizes ICP error metric will be selected as the initial transformation matrix.

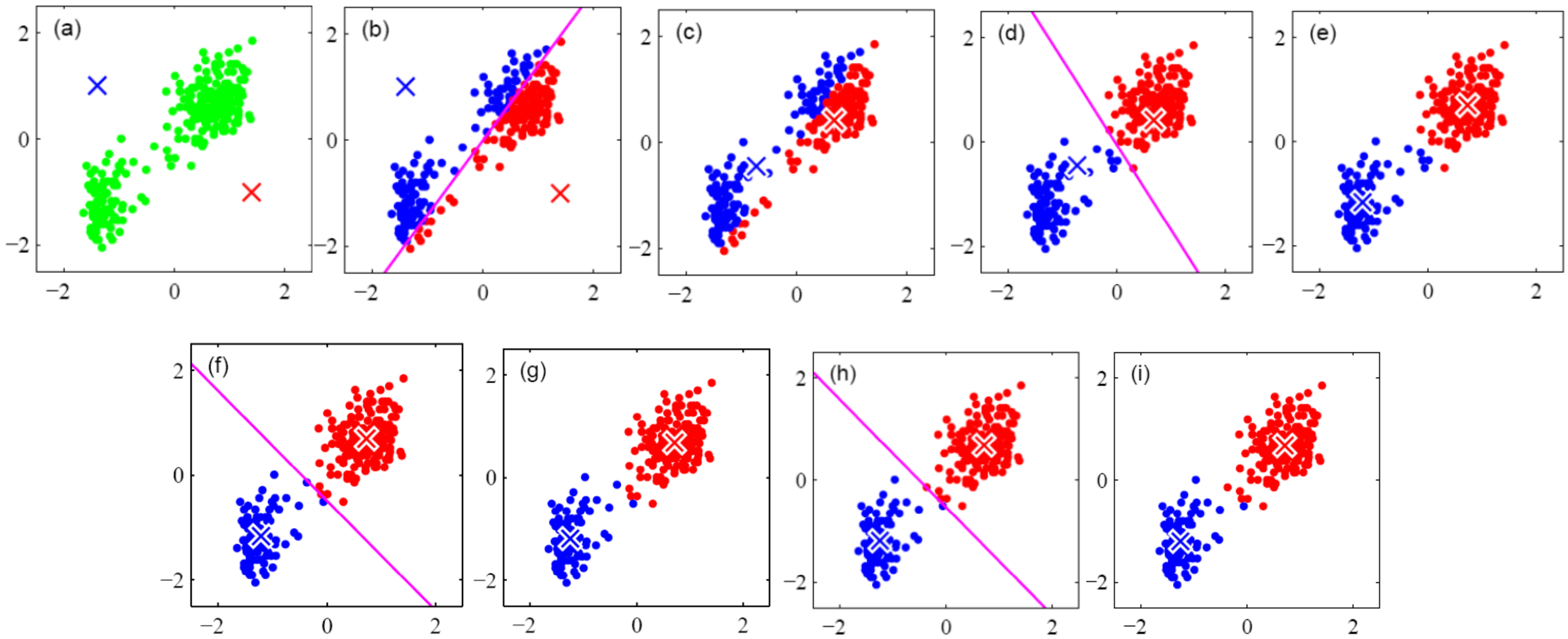


Head movement prediction

- Motivation (cont.)
  - The predictors can be trained using real-world head movement data.
  - However, if all predictors (with similar prediction capability) are trained on the same set of training data, they will perform similarly. For example, in a extreme case, all predictors will output the same transformation matrix. Therefore, the benefit of using multiple predictors vanishes.
  - So we propose to use a "K-means-like" method to automatically cluster the training data into groups. Each group is used to train a specific predictor.

Head movement prediction

- K-means clustering



Head movement prediction

- K-means-like training

1. *Choose K predictors, and tune their parameters using all the training data.*
2. *Repeat until converge:*
   - 2.1 *Compare prediction results, and accordingly divide the training data into K sets, in a way that all training data in set j are best predicted by predictor j.*
   - 2.2 *Re-train each predictor j with its specific training set j.*

Head movement prediction

- Choose predictors
  - The predictor should be able to determine its parameters automatically given the training data.
  - Since multiple predictors work together, complex predictors are not mandatory. For real-time processing, each predictor should be simple.
  - There are many possible choices. We choose to use the following prediction model

$$T_k = a_0 1 + a_{-1} T_{k-1} + a_{-2} T_{k-2} + \ldots + a_{-L+1} T_{k-L+1}$$

where $T_k = \{\nabla x_k, \nabla y_k, \nabla z_k, \nabla \phi_k, \nabla \theta_k, \nabla \omega_k\}$ corresponding to transitions and rotations along the XYZ axes, respectively. *L* is the filter length.

Head movement prediction

- Training data
  - We captured a head movement sequence with N=1829 frames, and used exhaustive search to get the ground truth.
- Determine predictor parameters
  - The predictor parameters can be obtained by optimizing this least square problem

$$\min_{[a_0,a_{-1},\ldots,a_{-L+1}]^t} \left\| \begin{bmatrix} \mathbf{T}_N \\ \mathbf{T}_{N-1} \\ \vdots \\ \mathbf{T}_L \end{bmatrix} - \begin{bmatrix} 1 & \mathbf{T}_{N-1} & \mathbf{T}_{N-2} & \ldots & \mathbf{T}_{N-L+1} \\ 1 & \mathbf{T}_{N-2} & \mathbf{T}_{N-3} & \ldots & \mathbf{T}_{N-L} \\ & \vdots & & \ddots & \vdots \\ 1 & \mathbf{T}_{L-1} & \mathbf{T}_{L-2} & \ldots & \mathbf{T}_1 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_{-1} \\ \vdots \\ a_{-L+1} \end{bmatrix} \right\|^2$$

  - or $\min_{\mathbf{V}} \|\mathbf{P} - \mathbf{Q} \times \mathbf{V}\|^2$ which has a closed form solution

$$\mathbf{V} = (\mathbf{Q}^t \times \mathbf{Q})^{-1} \times \mathbf{Q}^t \times \mathbf{P}$$

Head movement prediction

- Three predictors

Table 1. Parameters initialization using all training data

	a_0	a_{-1}	a_{-2}
Predictor A	0.009	0	0
Predictor B	0.004	0.585	0
Predictor C	0.005	0.469	0.198

Note: a_{-1} and a_{-2} of predictor A are forced to be 0. a_{-2} of predictor B is forced to be 0.

Table 2. Parameters after K-mean-like training

	a_0	a_{-1}	a_{-2}
Predictor A’	-0.054	0.041	0.039
Predictor B’	-0.008	0.927	0.003
Predictor C’	0.088	0.352	0.888

Head movement prediction

- Comparison between before and after K-means-like training

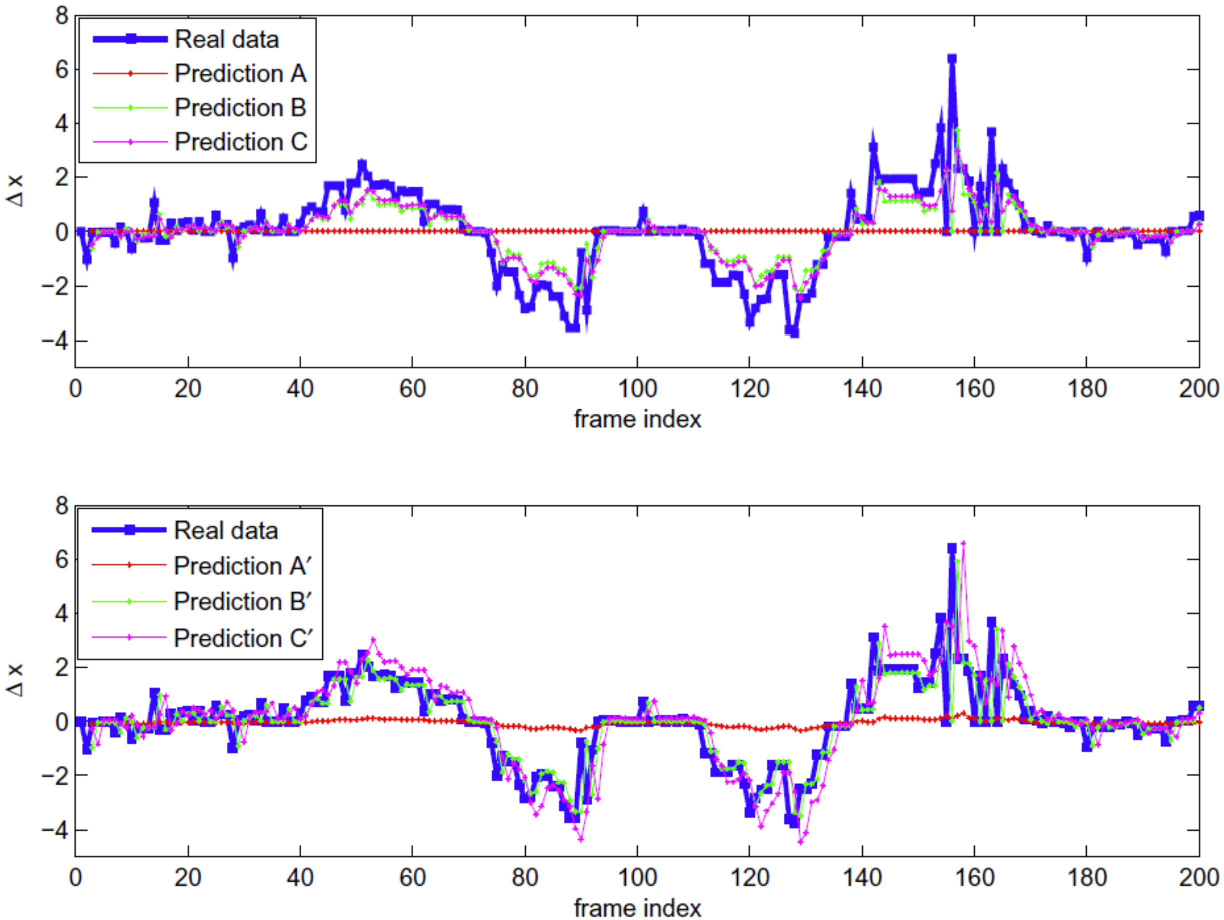


Figure 1. 200 frames of predictions on ∇x. Upper subfigure illustrates prediction results using predictors A, B, and C as given in Table 1. Bottom subfigure illustrates prediction results using predictors A', B', and C' as given in Table 2.

Head movement prediction

- Prediction error (ground truth **available**)
  - When the ground truth is available, the prediction error can be measured by MSE

$$MSE = \frac{\sum_{k=L}^{N} \|\mathbf{T}_k - \hat{\mathbf{T}}_k\|^2}{6(N-L+1)}$$

where $\mathbf{T}_k$ is the ground truth, and $\hat{\mathbf{T}}_k$ is the predicted one.

Table 3. Prediction errors of five predictors on the training sequence and a test sequence.

		A	B	C	Min(A,B,C)	Min(A',B',C')
MSE	Training	240.3	138.7	127.7	102.8	80.3
	Test	391.0	168.9	153.3	135.7	93.1

Head movement prediction

- Prediction error (ground truth **unavailable**)
  - When ground truth is unavailable, the ICP error metric can be used to evaluate prediction error

$$\sum_i \| q_i \cdot (T \times f_i - d_i) \|^2$$

  - Notice in the right figure, we use "identity" (T_k=0, i.e., static head) to replace A and A' (whose training results are close to "identity").

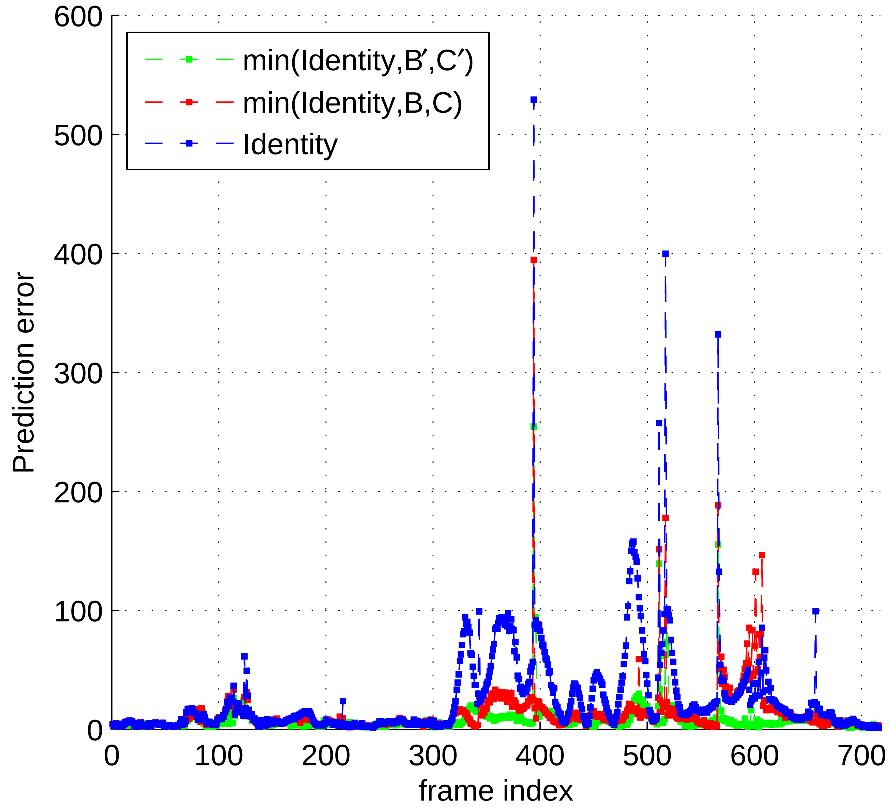


Fig. 2. Prediction errors measured by ICP error metric on the test sequence.

Head movement prediction

- Future work
  - Use tracking devices to get more accurate training data.
  - Try other simple predictors, like the Kalman filters.
  - Compare with prior studies, e.g., particle filters.
  - Code optimization.

Conclusion

- Introduced a Kinect-based real-time head tracking system. Discussed its possible applications.
- Introduced the iterative closest point algorithm (ICP) briefly.
- Presented in detail the K-means-like training method which can train several simple predictors together, enhancing the overall prediction accuracy.

Q & A