

Estimation of Gaze Direction in images

YANG Xin

School of Computing

National University of Singapore

April 24, 2009

Introduction

- Being able to tell the gaze direction of a person in an image can help in a lot of areas, like the human computer interaction
- How to tell?



Outline

- Problem Definition
- Algorithm
- Result and Discussion
- Conclusion

Assumption

- The anthropomorphic model of a generic human model is given
- We use the face orientation to approximately estimate the gaze direction

3D Face Model



Inputs

- The 2D image I containing an human face
- The 3D face mesh model M that represents the generic human face

Output

- The orientation of the face in 3D coordinate system given by $\theta = (\theta_x, \theta_y, \theta_z)$

Problem Formulation

- The problem for this project could be divided into two sub-problems
 - Detect the eyeballs and nose from the 2D image, as well as the face region(that is, detect the approximate face area from the image)



- Intensity based registration between the 2D input image and the projected image of the 3D face model and find the rigid transformation on the 3D model that gives the best fitting result, e.g. the intensity difference is minimized

Problem formulation-Detect (face, nose, eyeballs)

Notations:

I : denotes an image

$M(x, \theta)$: denote the 2D model of human face where $x=(x, y)$ means any location in the image

$\theta = (R, T, S, D, I)^T$, contains all the parameters for the face model.

R: the rotation of the model

T: the translation of the model

S: The scaling for the model,

D: the deformation for the model

I: the intensity of the model

$G(I)$: denotes the set containing all the coordinates and the size for each of the faces in the image I .

For each P in model $M(x, \theta)$, the corresponding point in the image is P' , then ideally, we should have $J(P) = J(P')$, where $J(P)$ is the function which returns the intensity for the point P, so the problem formulation is as follows,

Given an image I , determine the parameter θ for the model $M(x, \theta)$ that minimizes E :

$$E(\theta) = \sum_{x \in M} (J(M(x, \theta)) - J(I(x)))^2 \quad (1)$$

5/5/2009 When $\min(E(\theta)) < \mu$, μ is the user defined threshold value, the part of image $I(x)$, $x \in M$ is detected as a human face, and the coordinate for the face x and the size s is determined.

2D-3D registration

Notations:

$M = \{a_i\}$ denotes the feature points we selected from the generic human face model,

$S = \{p_i\}$ denotes the feature points we get from the 2D image, that is the eyeballs, nose.

$I(x_i)$ returns the intensity value of points x_i

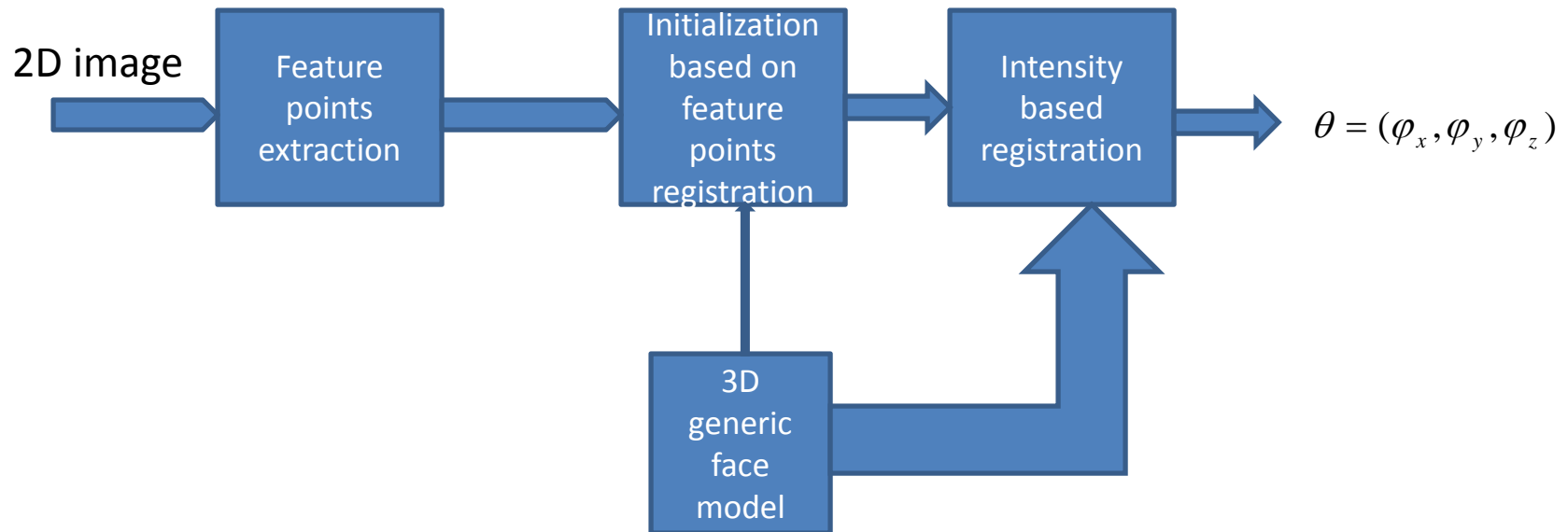
Given M and S , determine rigid transformation T that minimizes the error E :

$$E = \sum_i \|I(s_i) - I(t_i)\|^2 + \alpha \sum_1^4 \|P(S_i) - P(T_i)\| \quad (2)$$

Algorithm Overview

- Basic idea: two basic task
 - Detecting the eyeballs positions and nose position in the 2D image as well as the face and locate the face area approximately
 - Do the 2D-3D registration between the segmented out face image and 3D generic human face model points to recover the face orientation in the 3D world

Algorithm-con't



Step1: Detecting the face, eyeballs and nose, mouth

- We perform the search in the 2D image using the classifier that has been trained using lots of face images, including the different eye positions.
- We then try to find the nose point using the training method as well
- The detected nose and mouth is used to help eliminate all those falsely detected face and also used in the following step for the registration part (feature points must be corresponding to the 3D feature points as well, which is an constraint we put in the next step to before performing intensity based registration.

Step2: 3D-2D intensity based registration

- Using the extracted face region from previous step as the reference frame from which we want to recover the 3D orientation
- Do the intensity based registration, that is determine the orientation of the 3D face model and the translation parameters that gives the best fitting between the projected image and the reference image. We could use PCA to reduce the dimensions here. And the intensity difference is defined as follows,

$$E = \sum_i \|I(s_i) - I(t_i)\|^2 + \alpha \sum_1^4 \|P(S_i) - P(T_i)\|$$

Initialization for the first step

- Determine the rigid transformation that roughly transform the 3D feature points to be close to the reference feature point
- Without this initialization step, the algorithm could end up in an local minimum easily and the orientation doesn't make sense at all

Optimization

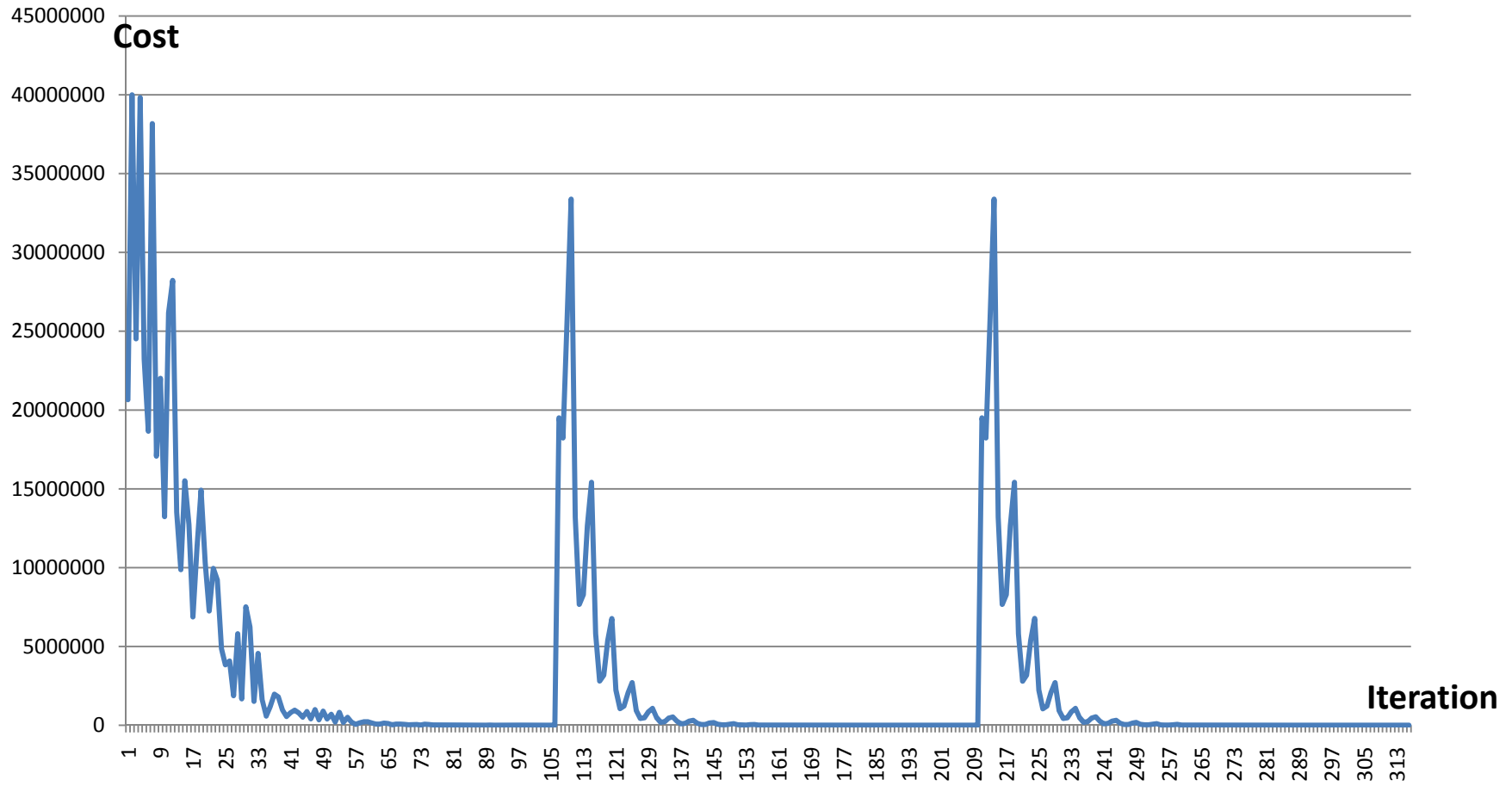
- Define the six degree of freedom

$$(\theta_x, \theta_y, \theta_z, x, y, z)$$

- Using the gradient decent method to search in the parameter space
- The cost function is defined as follows,

$$E = \sum_i \|I(s_i) - I(t_i)\|^2 + \alpha \sum_1^4 \|P(S_i) - P(T_i)\|$$

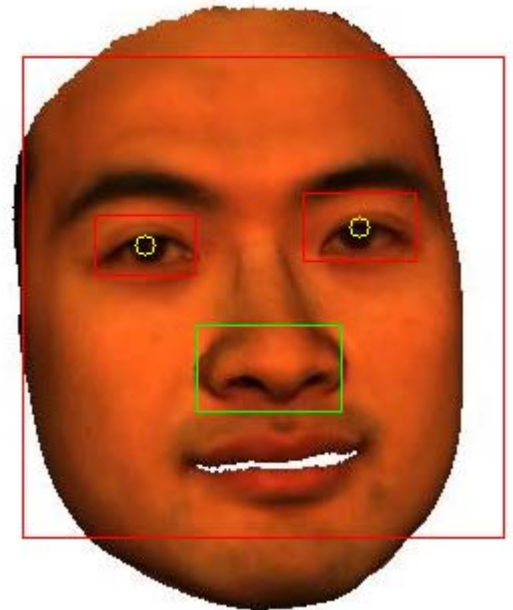
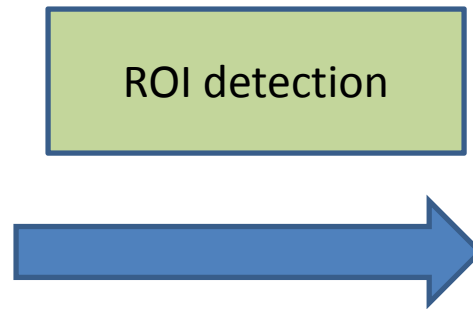
Optimization



The result - eyeball and nose detection

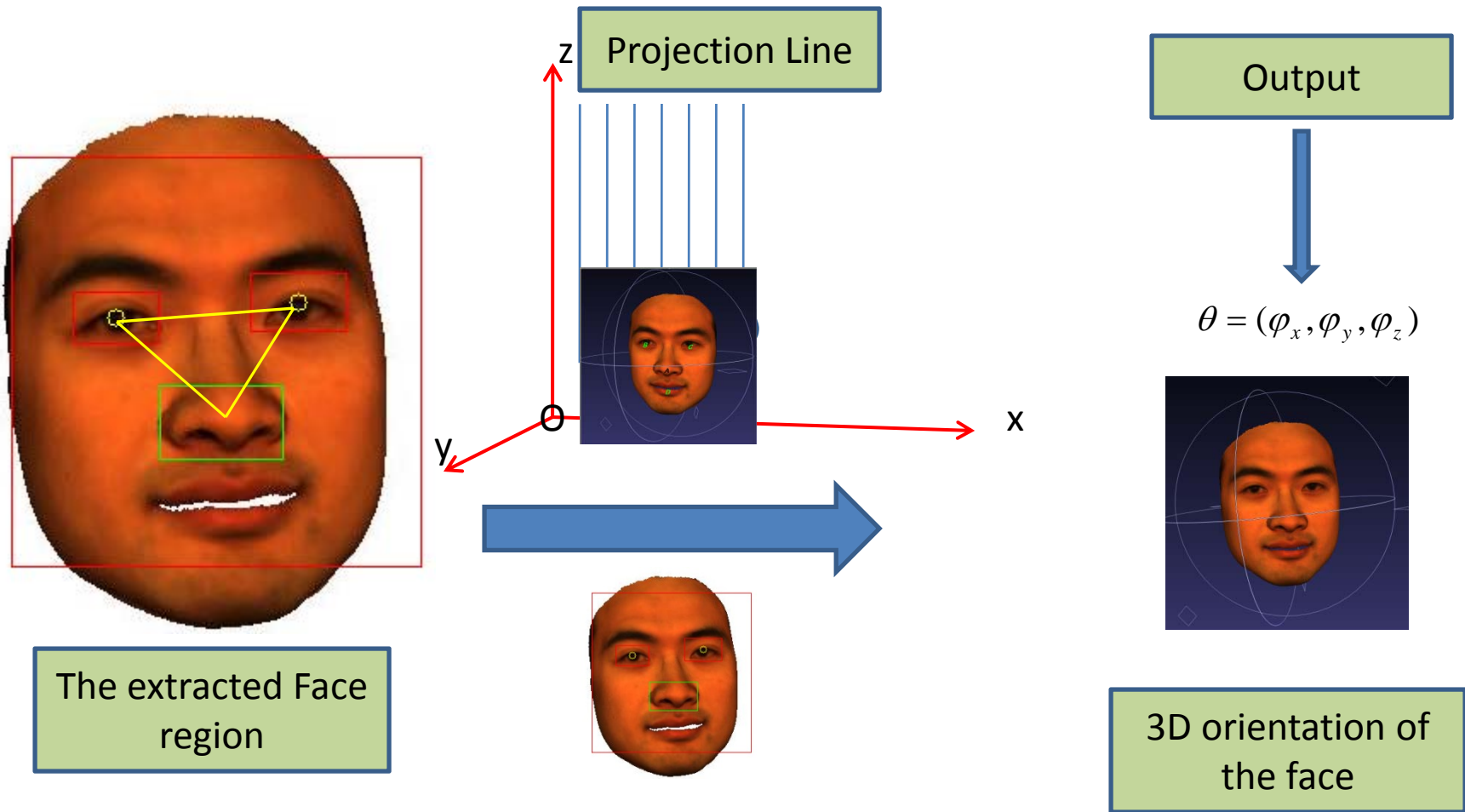


2D input face image

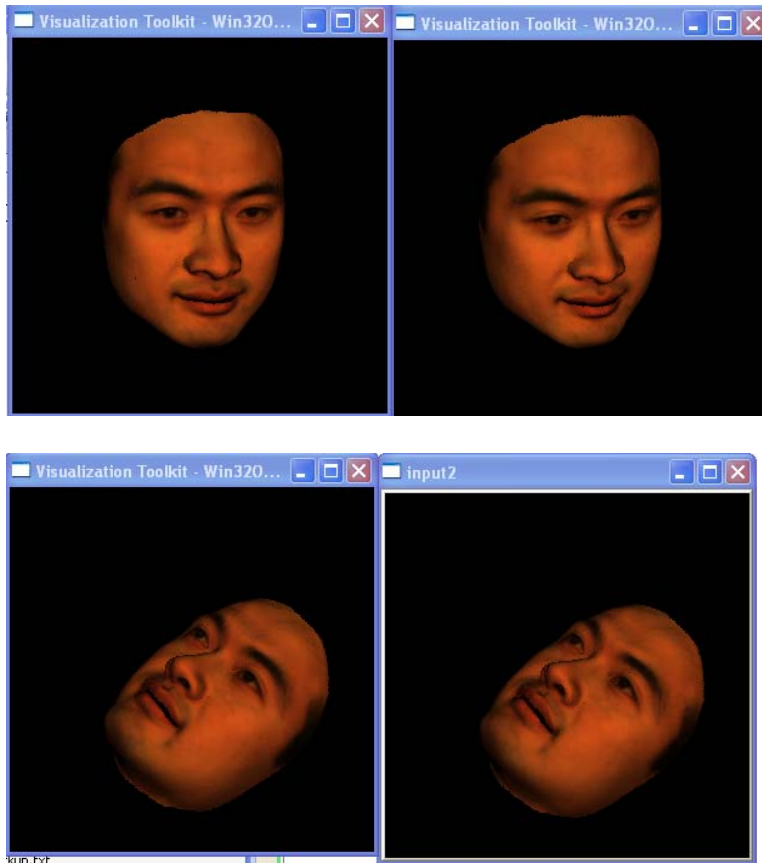


ROI detected

The result: 2D-3D Intensity based registration

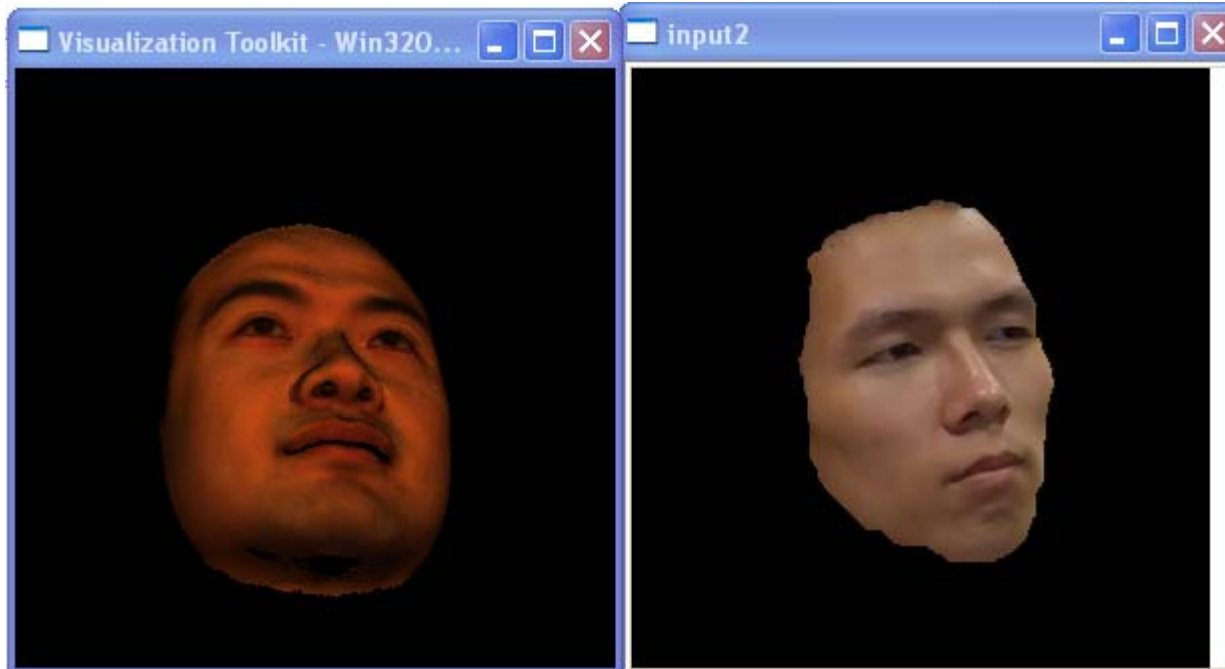


Result

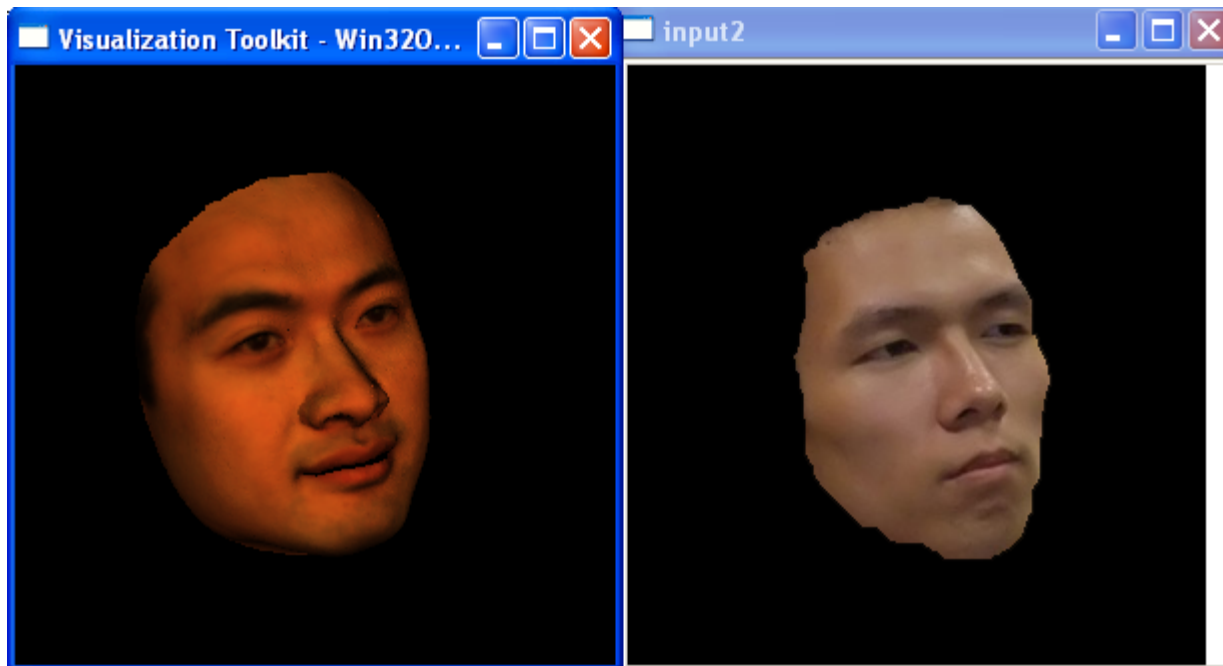


Video Demo

Initialization



Fine Matching



Possible solution for one model different skin color or illumination

- When new image is provided, we first do skin color segmentation to locate the face area, then a Laplacian pyramid is built for the input image, so that appearance variations caused by illumination can be reduced.
- We then search for the nose-tip in a coarse-to-fine manner using our tensor model and estimate the head pose with the most probable nose-tip location.

Discussion and Limitations

- The algorithm can automatically initialize the general orientation based on the feature points correspondence
- Given the
- If the feature points are falsely detected, then the whole result will be bad
- Intensity based feature of the algorithm makes it not quite suitable for real time video gaze detection, which is the future work
- We need a good model and develop a more robust algorithm for performing the reference face region detection and feature points localization
- Limitations
 - Highly depending the accuracy of the eyeball and nose detection (precondition)
 - Need a good way to limit the eliminate the falsely detected eyes, nose, or even

Conclusion

- The algorithm gives us good result when we are dealing with a face image that is more close to the model in terms of shape, illumination, skin color, etc.
- But for the face that is quite different from the face model, the estimated orientation result accuracy will be affected, and the solution tend to end up in the local minimum more frequently

Thank you