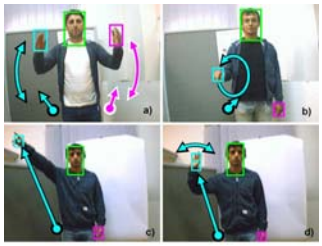
# An Efficient Methodology for 3D tracking and Pointing Localization for Robotic Guidance

Shankaranand Jagadeesan
Masters Thesis Defense

Advisor : Prof. Jezekiel Ben-Arie
Machine Vision Laboratory
ECE Department
University of Illinois at Chicago

## Gesture Recognition

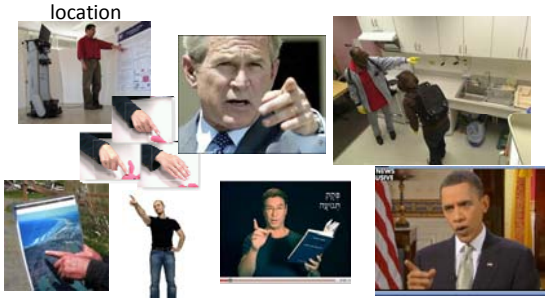
- Detect and recognize gestures performed by a human in an image.






## Pointing Gestures

- Normally when a person points to an object or location





## Find the pointed location?

- Location/object available in the camera



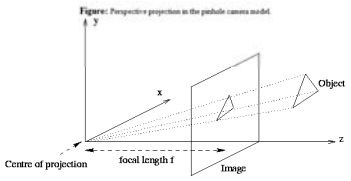
## Find the pointed location?

- Location/object not available in the camera



## Why is it hard ??

Perspective Projection in cameras – Only reason?



## Why it is hard ??

- View point variation
- Limited field of view
- Loss of Depth information due to perspective projection in 2D images
- Use a stereo, Depth camera – Still limits the field of view
- Occlusions



## We solve this problem

- Employ a multi-camera approach
- Calibrate the environment
- Detect Human in the image
- Detect regions of pointing information – Heads, Hands and Fingers.
- Estimate the direction of LOP – Line of Pointing
- Estimate the end of LOP, Objects in or near LOP



## Outline of the work

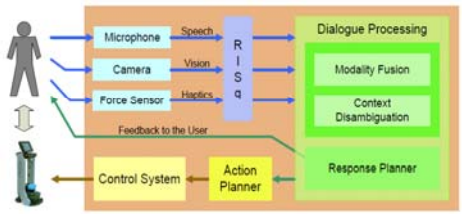
- Significance of Pointing Gestures
- Estimation of Human pointing localization
  - Self calibration of room
  - 3D Human tracking
  - Detection of Pointing Gestures
  - Obtaining the pointed location/object
- Results and Inference
- Summary and Future Work



## Significance of Pointing Gestures



**Robotic Assistants for Elderly: Integrating Speech, Vision and Haptics**

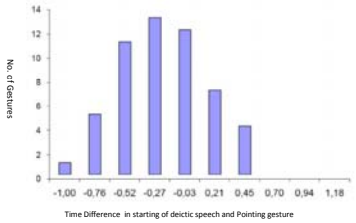



Proposed multi-modal communication interface



## Deictic Words

"Take that", "Put it here", "Give me this", "Go there"




**Time correlation between start of gesture and start of deictic words**

## Significance

- One of most used in Activities of Daily Living
- More visual clues than other gestures
- Involves multi-modal communication
- Effective contribution to Robotic Guidance
- 3D Tracking of the human in the calibrated environment



## Previous work

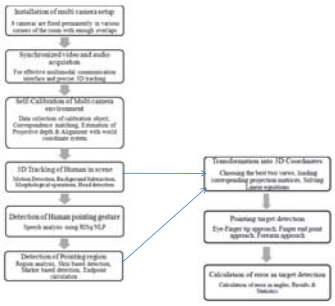
- Tracking of heads and hands
- Detection of Pointing gestures
- Pointing localization using stereo cameras, sensors and wearable interface.
- Comparison of Eye-Fingertip approach and forearm approach
- Multimodal communication interface for pointing gestures



## Estimation of Human Pointing Localization



## Overview of our method






## Calibration of the Cameras and 3D Reconstruction

- To find the internal camera parameters - focal length, image center, distortion
- To find the external camera parameters – Rotation, Translation
- Color Correction and Undistortion
- Estimation of fundamental matrices
- Stereo Rectification, Stereo correspondence
- Disparity calculation and 3D reconstruction



## Bad Multi camera system = Poor 3D Reconstruction

- Synchronization of multi-camera setup
- Quality of images from each camera

## Self-Calibration of Multi-camera Environment

- Inspired from "*Convenient multicamera self-calibration for virtual environments " by Tomas Svoboda et.al*
- No need of big calibration object
- Tele-presence of calibration object not needed
- Can tolerate occlusions in data collection of calibration
- Control of calibration process
- No restrictions on size of the environment
- Total time taken is less than a hour.

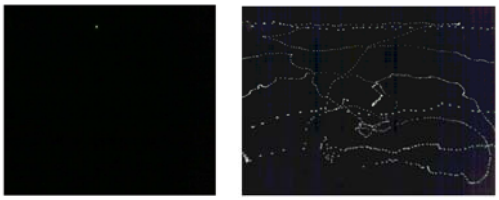


## Self Calibration – Data Acquisition and detecting corresponding points

- 6 HD USB Cameras
- Synchronized using MATLAB and I-Spy
- Wave the calibration point through the working volume in dark environment
- Record the videos of calibration and sample it to individual frames.
- Detect the points using image thresholding in particular color channel to sub pixel accuracy

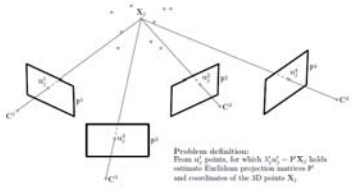


## Data acquisition (Continued)





## Self Calibration – Problem Defined



Problem definition:
From $\mathbf{u}_j^i$ points, for which $\lambda_j^i \mathbf{u}_j^i = \mathbf{P}^i \mathbf{X}_j$ holds estimate Euclidean projection matrices $\mathbf{P}^i$ and coordinates of the 3D points $\mathbf{X}_j$

$$\lambda_j^i \begin{bmatrix} u_j^i \\ v_j^i \\ 1 \end{bmatrix} = \lambda_j^i \mathbf{u}_j^i = \mathbf{P}^i \mathbf{X}_j, \qquad \lambda_j^i \in \mathbb{R}^+$$

- $j$ index points
- $i$ index camera
- $\lambda_j^i$ projective depths
- $\mathbf{u}_j^i$ point projections (we find them in images)
- $\mathbf{X}_j$ 3D points (we do not know the positions!)
- $\mathbf{P}^i$ camera matrices



## Self Calibration – Problem Defined

**Multicamera linear model**

$$\mathbf{W}_s = \begin{bmatrix} \lambda_1^1 \begin{bmatrix} u_1^1 \\ v_1^1 \\ 1 \end{bmatrix} & \cdots & \lambda_n^1 \begin{bmatrix} u_n^1 \\ v_n^1 \\ 1 \end{bmatrix} \\ \vdots & & \vdots \\ \lambda_1^m \begin{bmatrix} u_1^m \\ v_1^m \\ 1 \end{bmatrix} & \cdots & \lambda_n^m \begin{bmatrix} u_n^m \\ v_n^m \\ 1 \end{bmatrix} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{P}^1 \\ \vdots \\ \mathbf{P}^m \end{bmatrix}}_{\mathbf{P}\ 3m \times 4} \underbrace{[\mathbf{X}_1 \cdots \mathbf{X}_n]_{4 \times n}}_{\mathbf{X}}$$

Self-calibration (Euclidean stratification)

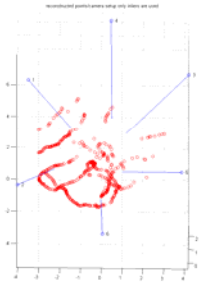
$$\mathbf{W}_s = \mathbf{P}\mathbf{X} = \mathbf{P}\mathbf{H}\,\mathbf{H}^{-1}\mathbf{X} = \hat{\mathbf{P}}\hat{\mathbf{X}}.$$



## Self-Calibration – 4 Step Process

1. Finds the projections $\mathbf{u}_j^i$ of the laser pointer in the images.
2. Discards misdetected points by pairwise RANSAC analysis.
3. Estimates projective depths $\lambda_j^i$ and fills the missing points to make scaled measurement matrix $W_s$ complete.
4. Performs the rank 4 factorization of the matrix $W_s$ to get projective shape and motion and upgrades them to Euclidean ones.

## Self-Calibration Output





## Alignment with World Coordinates

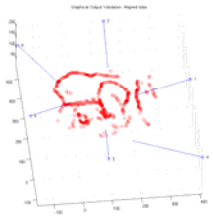
- Obtain the P matrices and Camera locations from the raw output
- Assign a world 3D origin in corner of the room
- Measure the camera coordinates in 3D world coordinates
- Obtain the similarity transformation using

$$B = T^{-1} A T$$

- Transform the projection matrices and 3D calibration points to the world coordinates
- Verify the aligned coordinate system



## Alignment with World Coordinates (Continued)





## Human Silhouette Extraction and 3D tracking

- Motion Detection in cameras
- Background Subtraction in YCbCr
- Morphological operations
- Region Analysis of Foreground Pixels
- Head detection using shape template matching
- Obtaining the 3D trajectory of the person from the 2D images

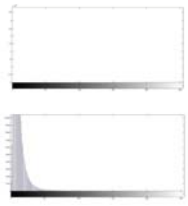


## Motion Detection in cameras

Absolute difference in Y, Cb and Cr channel between successive frames

Canny Edge detection and histogram of edged image

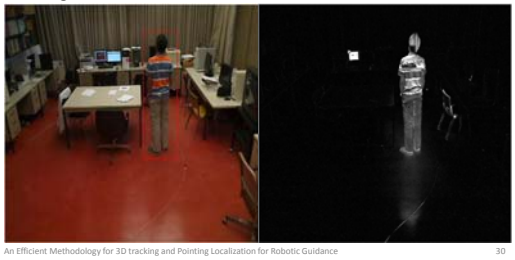
Highly useful in reducing processing time





## Background Subtraction

- Absolute difference in Y, Cb and Cr channel between the current frame and stored Background model
- Summation of each channel's difference to obtain difference image



## Morphological Operations

- Threshold the difference image
- Thickening and Closing (Dilation followed by erosion)



## Region Analysis of Foreground Pixels

- Inspect individual regions for its dimensions in pixels
- Select the region with largest area



## Head detection using shape template matching

- Head templates created at difference scales and stored

- Normalized correlation against the binary image with human silhouette.
- Template with maximum correlation coefficient returned.
- Verify the correlation coefficient value



## Head detection using shape template matching



## Reconstructing 3D from 2D images

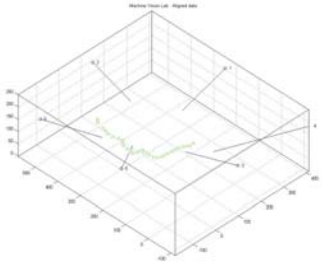
- Obtain the top-center 2D pixel of the head in two images
- Multiply these values with the corresponding projection matrices and solve for linear equations to obtain the 3D World Coordinates

$$\left.\begin{aligned}\tilde{\mathbf{x}}_i &= \mathbf{P}\tilde{\mathbf{X}}_i \\ \tilde{\mathbf{x}}'_i &= \mathbf{P}'\tilde{\mathbf{X}}_i\end{aligned}\right\} \Rightarrow \begin{bmatrix} \mathbf{P}_3 x_i - \mathbf{P}_1 \\ \mathbf{P}_3 y_i - \mathbf{P}_2 \\ \mathbf{P}'_3 x'_i - \mathbf{P}'_1 \\ \mathbf{P}'_3 y'_i - \mathbf{P}'_2 \end{bmatrix} \tilde{\mathbf{X}}_i = 0$$

- Find the corresponding pixels in each frame and plot the 3D value to obtain the trajectory of the moving person



## 3D Tracking of the Human

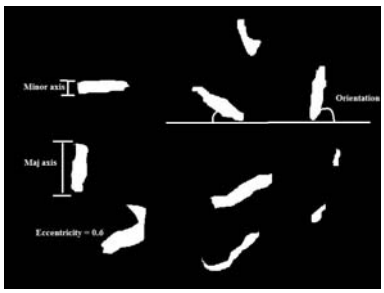


## Detecting Pointing Gesture

- Pointing frame detection using multi-modal approach
- Monitor Speech data for deictic words using RISq.
- High Correlation between pointing gesture and deictic words
- Obtain the video frame with deictic speech
- Verify Pointing gesture
- Saves lot of computation power and time
- Current Implementation - Manual Labeling



## Region Properties Explained






## Skin color based hand detection

- Skin segmentation of Human Silhouette region
- Region analysis of the Skin pixels
- Verify
- Area > 100 pixels
- Eccentricity > 0.75
- Absolute value of Orientation < 55 deg
- 11 > Major axis length/Minor axis length > 2
- Region with its centroid farthest from center of human silhouette

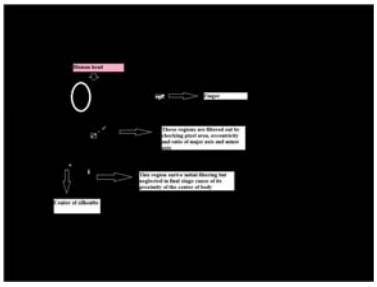


## Marker based finger detection

- Color segmentation of Human Silhouette region with stored color model
- Region analysis of the marker pixels
- Verify
- Area > 7 pixels
- Eccentricity > 0.50
- Absolute value of Orientation < 60 deg
- 8 > Major axis length/Minor axis length > 3
- Region with its centroid farthest from center of human silhouette



## Marker based finger detection (Continued)





## Finding the Pointed location/Object

- Accurate Pointing Vs Approximate Pointing
- Analysis of data gathered from ADL experiments in Rush
- Different types of pointing gestures
- Disadvantage of Eye-Fingertip pointing
- Advantage of Forearm pointing and finger pointing technique

## Disadvantage of Eye-Fingertip pointing

Figure 44: Misdetection of pointing location by Eye-Fingertip approach



## Detection of End points in Forearm/Finger

- Concept of extreme points
- Obtain left top and right top extreme point
- Obtain centroid point of the region
- Select Points with maximum Euclidean distance from human silhouette center
- Shift the end-points to top portion of the hand/finger to get the estimated LOP



## Detection of End points in Forearm/Finger

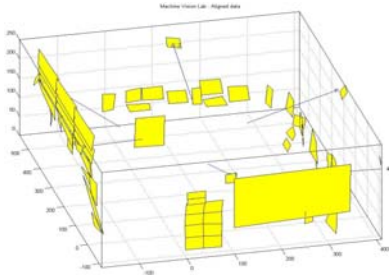


## Estimation of pointed location

- Reconstruct the LOP in 3D coordinates
- Extend the LOP in the direction of pointing
- Verify
- Object planes intersected by LOP
- LOP ends at camera planes, i.e. end of room
- Return the objects name and distance from the user
- List Closest objects if LOP doesn't intersect any object



## Object Planes and Database





## Object Planes and Database

| Objects | | Ltop | | | Rtop | | | Lbottom | |
|---|---|---|---|---|---|---|---|---|---|
| Printer 1 | -101.79 | 43.3246 | 164.205 | -101.79 | 4.98713 | 164.205 | -101.79 | 43.3246 | |
| Box 1 | -98.581 | 44.6152 | 146.986 | -101.79 | 6.27777 | 146.986 | -98.581 | 44.6152 | |
| White cabinet 1 | -93.677 | 43.5636 | 130.61 | -93.677 | 5.22618 | 130.61 | -87.256 | 42.5287 | |
| Camera 6 | -143.04 | 97.3041 | 225.077 | -145.69 | 75.5879 | 225.077 | -147.36 | 96.6285 | |
| Camera 4 | -95.292 | 412.082 | 206.576 | -152.86 | 402.127 | 208.505 | -104.51 | 417.774 | |
| Brown Box 2 | -102.12 | 308.634 | 94.4954 | -146.87 | 309.794 | 95.383 | -88.946 | 299.838 | |
| Motherboard Box | 2.19983 | 351.821 | 84.4086 | -34.959 | 319.045 | 83.0905 | 9.83002 | 343.697 | |
| WS Speaker | 18.1831 | 371.081 | 100.629 | 3.26901 | 355.989 | 103.64 | 14.5352 | 374.312 | |
| PC Chair 1 | 73.5589 | 296.165 | 87.577 | 23.1319 | 285.481 | 89.9149 | 73.5589 | 296.165 | |
| WS Monitor | 90.1602 | 357.711 | 127.293 | 42.1235 | 354.811 | 128.62 | 88.2555 | 357.895 | |

## Results and Inference



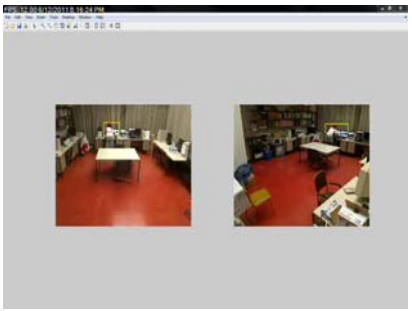
## Forearm detection



## Finger Detection

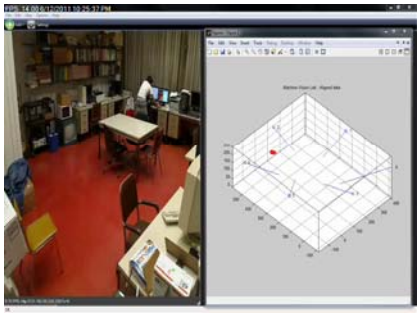


## Head detection in multiple cameras



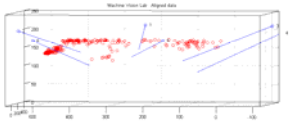


## 3D Trajectory of Human in the room





## 3D Trajectory (continued)

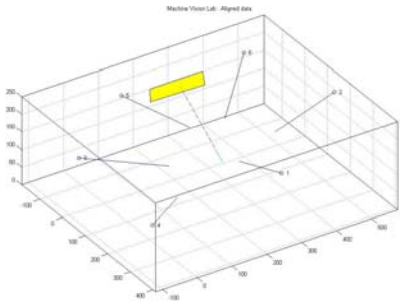


## Information to the elderly human

- Images of pointed object with red bounding box
- Speech output of the pointed object (TTS)

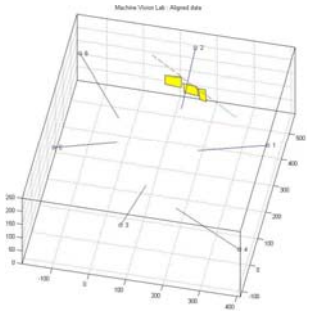



## Pointed Object Identification



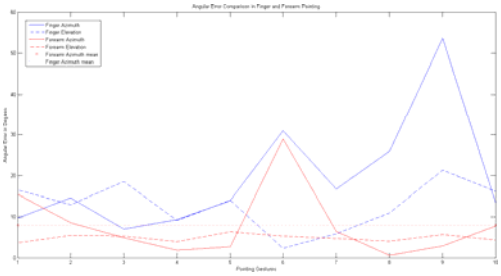


## Vacant Pointing – Closest Objects Listed



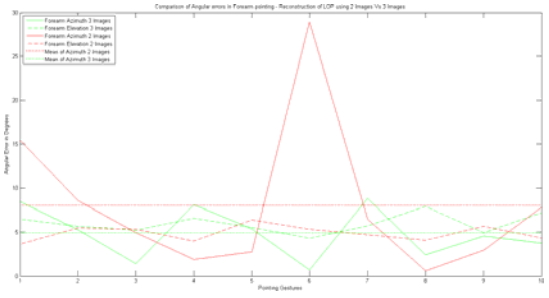


## Forearm Pointing vs Finger Pointing





## Forearm Pointing – Reconstruction from 2 Images Vs 3 Images





## Statistics and Inference

| Pointing vs. Angular errors | Mean | | Standard Deviation | |
|---|---|---|---|---|
| | Azimuth θ° | Elevation φ° | Azimuth θ° | Elevation φ° |
| Finger Pointing | 19.4980 | 12.7720 | 14.1094 | 5.8165 |
| Forearm Pointing – 2 Images | 8.0373 | 4.8850 | 8.4933 | 0.8672 |
| Forearm Pointing – 3 Images | 4.8915 | 5.9539 | 2.9127 | 1.0825 |

- Forearm pointing performs better than finger pointing
- Detection rate, Forearm – 91 % Finger – 32 %
- Accuracy can be improved using more number of views
- Error in Elevation can be rectified

•Advantages
- No marker or glove used
- Works well with different types of pointing
- Longer LOP. More Tolerant with error in correspondence and endpoint pixel detection

## Summary of Contributions

- Efficient multi-camera setup to record and monitor multi-modal communication
- Successful Calibration of Rush ADL room and MVL Lab
- Head Detection and 3D Human tracking in the room
- Proposed multi-model method to detect pointing gestures
- Detection of Pointed locations and Objects
- Comparison of Finger pointing and Forearm Pointing



## Future Work

- Employ 3D tracking of human, based on image features with probabilistic techniques
- Automated Pointing Location Finder – Integrating with Speech
- Integrating Head Orientation, Speech with Pointing



## Thank you!!