Virtual 3-D Blackboard:

Finger Tracking with a Single Camera

Andrew Wu

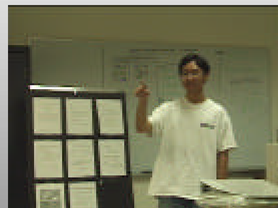
REU 1999

awu@uiuc.edu http://www.cs.ucf.edu/~vision
(go to REU99)

Project Goals

- Using computer vision, implement a virtual 3-D blackboard
- Program will parse 2-D image input, recording corresponding 3-D motion of a user's fingertip
- Motion of user's finger will be recognized as a certain type of 3-D gesture

Sample Picture

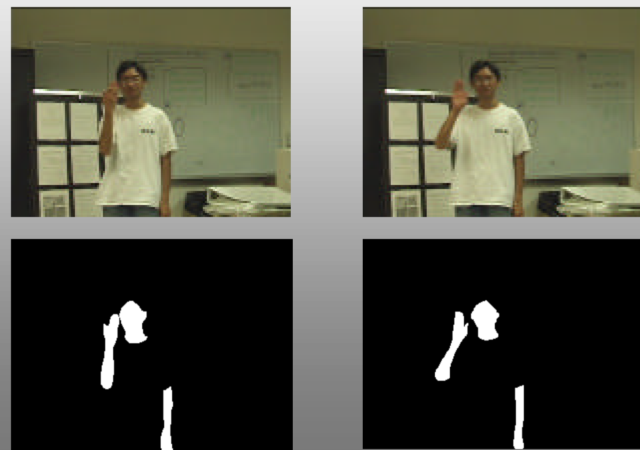


- Single color camera
- Static background
- One person in picture
- Consistent lighting

Skin Detection

- **Color Predicate**
  - **Skin-tones in RGB space marked by computer program**
  - **Trained on several color images with hand-drawn binary masks**
  - **Color Predicate data structure saved**

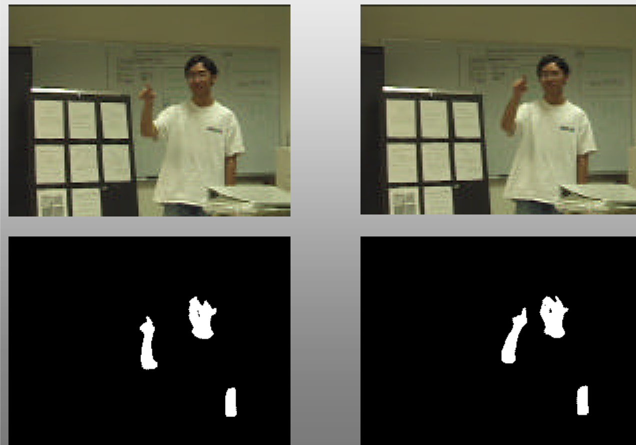
Example training images



Using the Color Predicate

- Check RGB values of every pixel in input image
- If RGB value satisfies Color Predicate, output as true in output binary image
- Median-filter binary output to remove noise and outliers

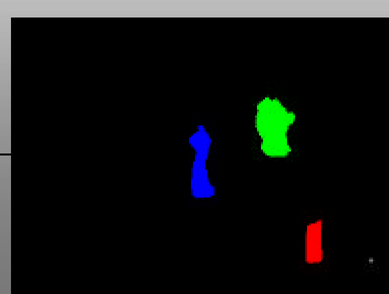
Results of skin detection



Separating regions

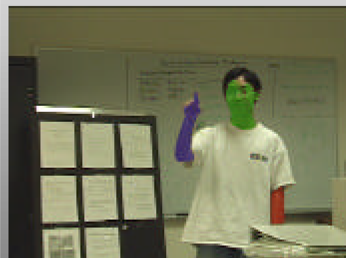
- Next step: demarcate connected skin regions
- Simple 8-connectivity algorithm that grows regions
- Cull three largest regions (presumably the head and two arms)

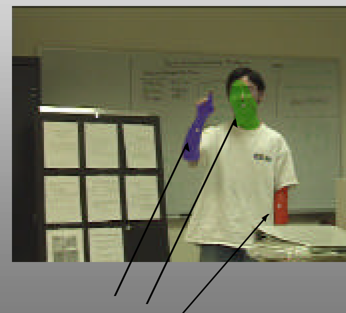
Three largest regions, pseudo-colored



Separating regions

•Find centroids for largest regions (regions alpha blended with original color image for effect)

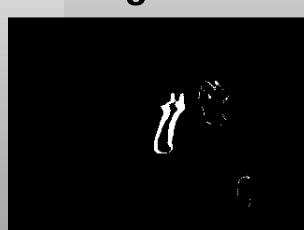




centroids

Finding the arm

- **Assume fastest moving centroid belongs to gesticulating arm**
- **Find largest delta between two skin frames**

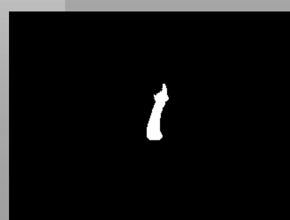


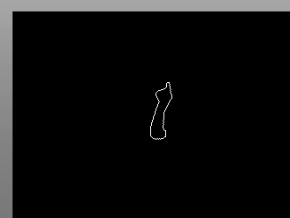
Difference picture between temporally proximal skin images

Difference picture for region of largest centroid delta, only

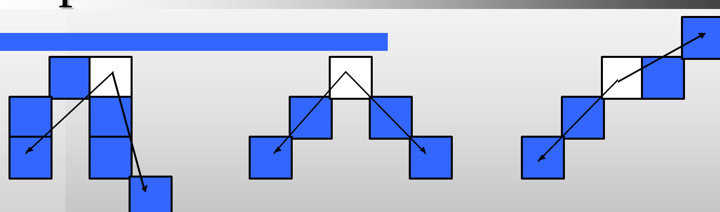
Outlining the arm

- Goal: find perimeter of isolated arm segment
- Assume contiguous region
- If pixel has 4-connectivity with black region, pixel is on the periphery





Dot product



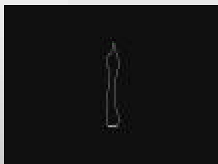
A dot B > 0 A dot B = 0 A dot B < 0

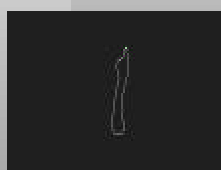
•Method: find two vectors between the current pixel and the pixels N steps away (N=3, 2, 2 above)

•Repeat procedure for all pixels in outline, searching for maximum dot product.

•Idea: largest dot product will be formed by two vectors extending in both directions from the fingertip

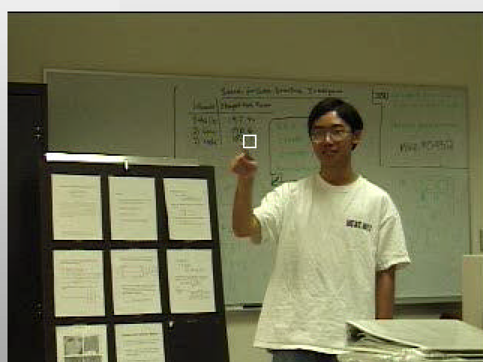
Results of dot product approach





- •Very good output
- •Found finger in all cases when given proper outline of arm
- •Perhaps can detect absence of finger from skin outline
- •For test data, found best value of N to be 3 pixels (N being number of pixels to step away for vector calculation)

Video



- •Program run on 7 continuous frames
- •In sum: finds skin from color, arm from centroid speed, then finger from dot product of pixel outline
- •Tracks finger fairly well

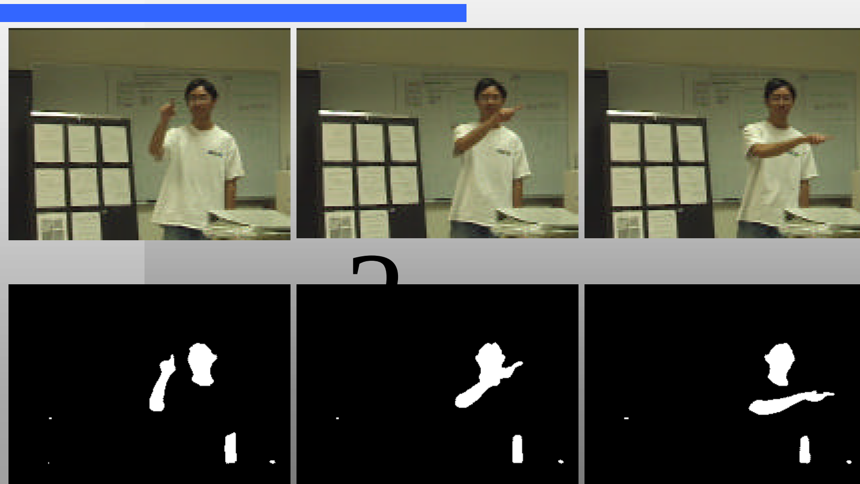
Next issue: Occlusion

•Comparatively easy to handle simple situations where hands and face are far apart (no occlusion)

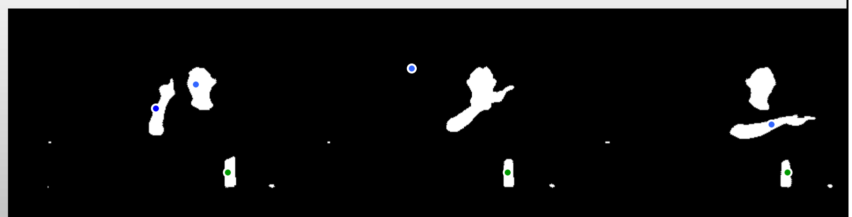
•Problems appear when one hand blocks face:

- -How to distinguish body parts?
- -Where are the centroids?
- -Can we find the finger?

Example images



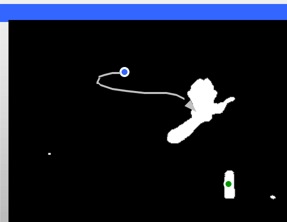
The problem



•Since the arm finding algorithm looks for region whose centroid has moved the most, the algorithm gets confused when centroids disappear and reappear

•Above, 3 centroids become 2 centroids, become 3

Current solution



•When centroids gets “lost”, missing centroid is assumed to have been assimilated by the largest contiguous region

•So, place “ghost” centroid marker on top of centroid of largest region

•When centroid reappears, each centroid accesses the next frame to find closest future centroid.

•Also, ensure a one-to-one mapping (two centroids in one frame should not map to only one centroid in next)

Video

Tracking through occlusion, a demo

http:../sony/public_html/move.html

..\public_html\move.html

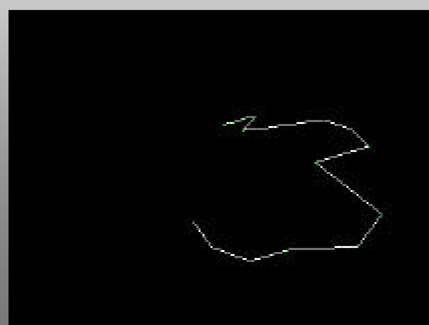
Comments on video

- Tracking through occlusion works much better than before, but still has some problem when centroid appears
- Perhaps more frame data (less delay between frames) would help?
- Finger not readily visible when finger occludes hand, but current tracker finds close match

(clip generated by running program on ~19 images)

Graphing the output

- Wrote graphing object that records finger's coordinates (for now only 2-D)
- 2D Graph output for previous clip:

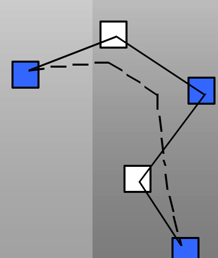


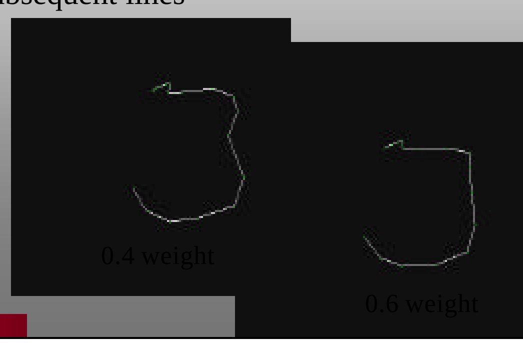
First discontinuity caused when hand occludes face

Next occurs when part of head mistaken for hand

Smoothing the graph

- First, assume that tracked points follow linear pattern
- As a simple smoothing mechanism, use a weighted average of the x, y position and the midpoint between previous and subsequent lines

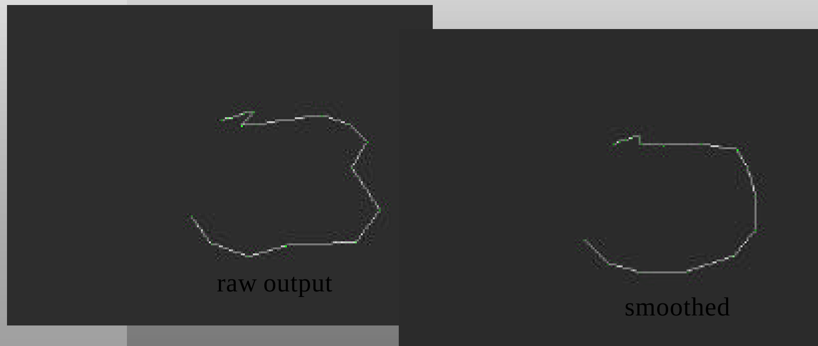




Updated images

http://www.cs.ucf.edu/~aw47967/move-fixed.html

..\public_html\move-fixed.html



Body Tracking

Why track the body?

- Helps us derive 3-D information
- Useful in correcting errors in finger tracking

Quo vadis?

("Where are you headed?", or Head Tracking)

Highest centroid (lowest Y) that does not belong to the arm should correspond to the head

•When centroids merge, do not track head, rather use previous data

•Simple approach, but works well

Approximating the shoulder

Need to guess shoulder location -- actual coordinates unimportant

•Relative distance to head should be consistent

•Assume shoulder is a certain distance from centroid of head

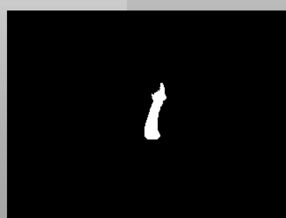
Procedure: Find approximate radius of head region

Shoulder <x,y> = Head Centroid <x,y> + <radius, radius * 1.618…>

(1.618… = (1 + sqrt(5))/2)

Finding the Elbow

For simplicity, assume elbow is part of skin image. If necessary, other techniques are applicable to images where this is not true

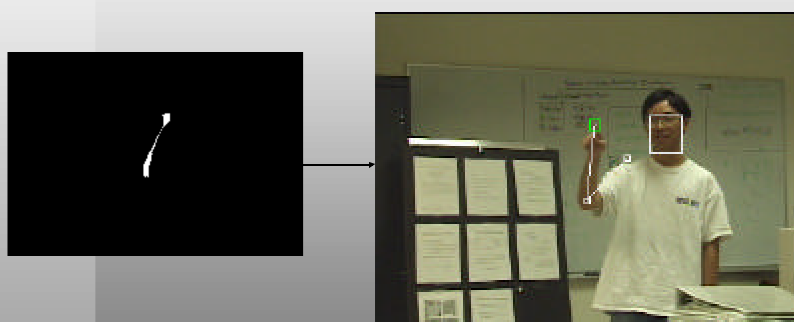


Erode arm region by several pixels

Using known finger location, find point on opposite side of arm

(that is, maximize distance between known finger point and unknown elbow position)

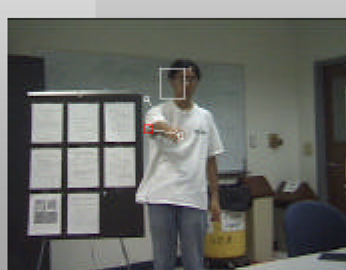
Results of Body Tracking



skeleton, superimposed

Improving the tracker

When finger occludes hand, tracker sometimes marks elbow as fingertip, because hand location has low curvature



So, if ‘bicep’ length is clearly greater than arm length, distance from shoulder to elbow is greater than distance from shoulder to finger, and unsure of finger location...

…swap finger and elbow points

Judging tracker accuracy

Based on magnitude and sign of dot product, we can assign a “confidence” value to each data set recorded

Graphically, these confidence values are represented as:

- green -- high confidence
- dark green -- lesser confidence
- red -- low degree of confidence

Movie

2-D Body tracking with graphical tracker confidence display, the movie

..\public_html\tracker-tracking.html

Leaving Flatland

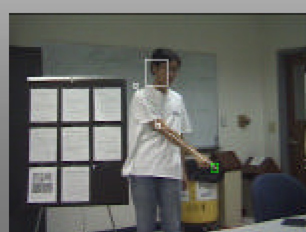
Armed with body data, we can begin to derive 3-D information

•First, assume orthographic projection

•Need to record “true” pixel length of Humerus, Radius (upper arm and lower arm)

•Assume that at some point in footage, arm is fully extended

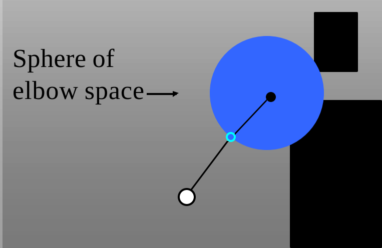
•Store largest lengths



Spherical kinematics

Humanoid bones have constant length

Thus, for example, since distance from elbow to shoulder is fixed, elbow can only move tangent to a certain sphere (sphere centered on shoulder)



Collapsing the hemisphere

Ignoring the half of the sphere that lies behind the plane of the body, we are left with the surface of one hemisphere

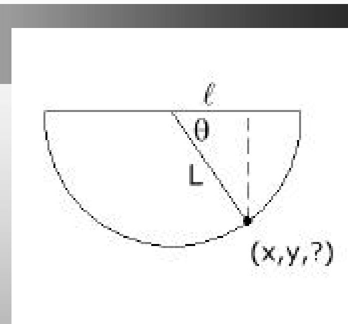
If we look at the hemisphere along the polar axis, we can see the entire surface of the hemisphere

That is, along the polar axis, we can collapse the hemisphere from 3-D to 2-D without loss of information, as well as recover the hemisphere from its 2-D projection.

Math

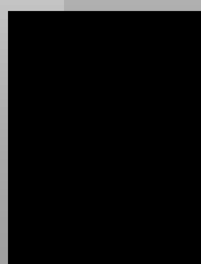
L = r = radius of sphere

l = length of projection of 3-D line onto 2-D circle

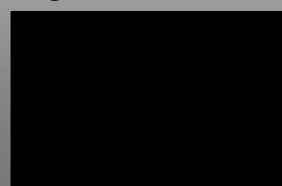


(View from above)

Geometrical:



Algebraic:



cos(theta) = l / L

theta = acos(l/L)

z = L sin(theta)

z = L sin acos(l/L)

x^2 + y^2 + z^2 = r^2

z^2 = r^2 - (x^2 + y^2)

z = sqrt(r^2 - l^2)

Relative Z

From shoulder point and elbow point, we can calculate relative Z from shoulder to elbow

Using a similar line of reasoning, we can deduce the relative Z coordinate of finger compared to Z of elbow

Setting Z coordinate of shoulder to be 0,

elbow.Z = 0 + relative Z from shoulder to elbow

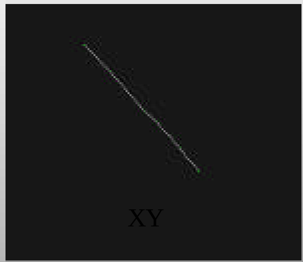
finger.Z = elbow.Z + relative Z from elbow to finger

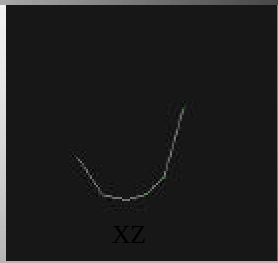
Movie

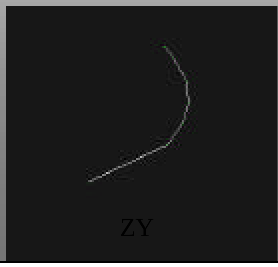
3-D finger tracking of a semi-circle

..\public_html\montage-semi.html

Graph output







Graphs of semi-circle movement, from varying viewpoints

Movie

3-D finger tracking of circle motion

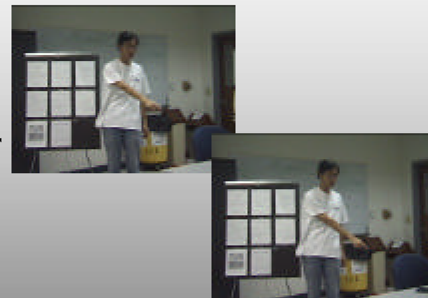
montage-circle.html

..\public_html\montage-circle.html

Tracker improvements

Problem:

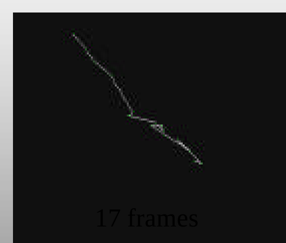
Arm not identified properly when little or no motion in frame



Approach:

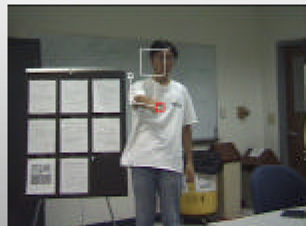
Set arbitrary threshold for minimum amount of movement. If threshold not reached, use previously found arm centroid

Comparison



- Same data set, from semi-circle motion
- Extra data shows how local tracking errors can be quite significant

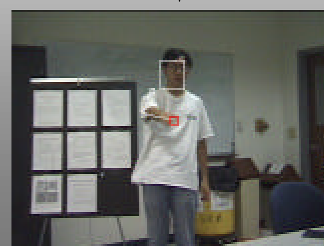
Increasing accuracy



before

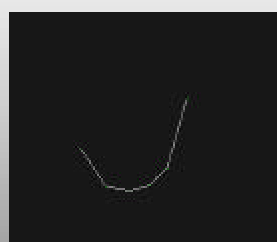
Check if shoulder point in background.

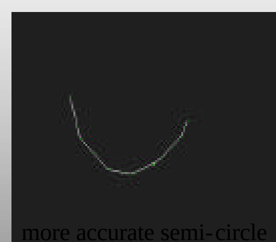
If not, move diagonally until body reached



after

Comparison of accuracy





6 frames,

raw, uneducated shoulder guess

17 frames,

adjusted shoulder position

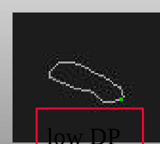
Model of Error

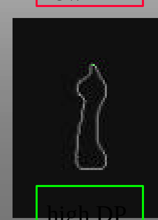
- Previously did not use "confidence" recorded
- If we can estimate error, we can use estimate to adjust level of smoothing
- High Confidence means Low Error

Error and Confidence

- Estimated Confidence = Linear Combination of 4 Factors
- First Factor:
  Dot Product

Higher dot product (DP) means Higher confidence



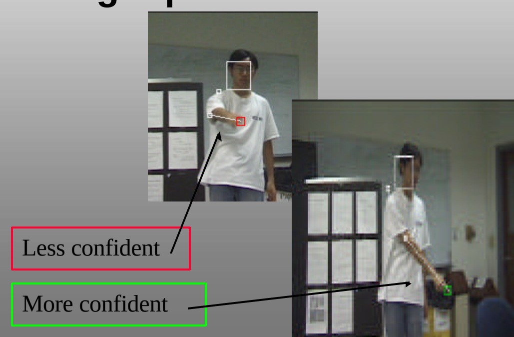


Confidence from 3D

- We know that certain 3D positions of the finger are harder to detect
- Worst case: when finger points at camera

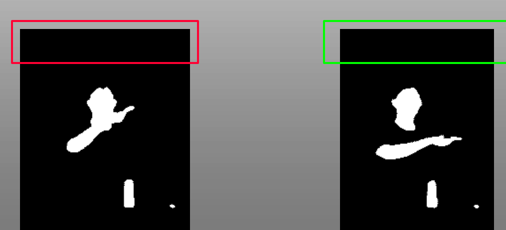
Generalization:

The length of the lower arm is proportional to our confidence in tracking



Next Factor: Occlusion

- When there is occlusion, we are less sure of the tracking
- Thus, the number of centroids is proportional to confidence



Movement Disparity

- **We expect that the finger will move as the (centroid of the) arm moves**

Note how both vectors (white arrows) are very similar

Less disparity means greater confidence

(Disparity = magnitude of the difference of both vectors)

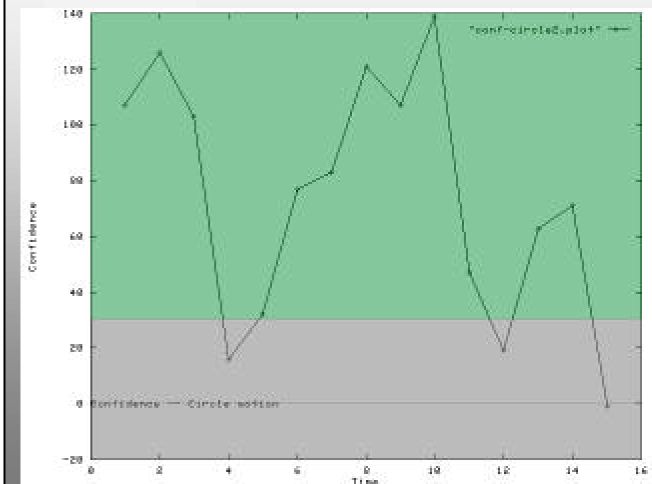
Linear Combination

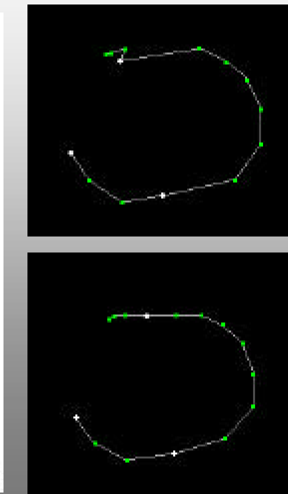
- **Confidence =**
  C_1 * dot product +
  C_2 * length of lower arm +
  C_3 * number of centroids +
  C_4 * movement disparity

$C_1 = 8$, $C_2 = 1$, $C_3 = 6$, $C_4 = -2$

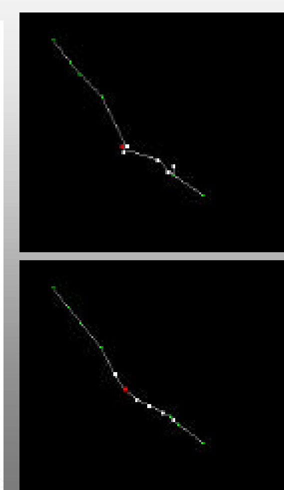
Constants chosen by hand to match error determined by (human) visual inspection

Graph of Confidence - Circle





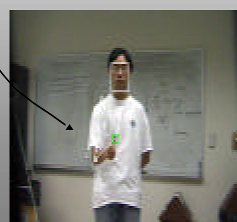
Confidence - Semicircle



Improving the User

•Low-tech improvement: have user stick out thumb while gesturing

•When index finger occluded, thumb is not...

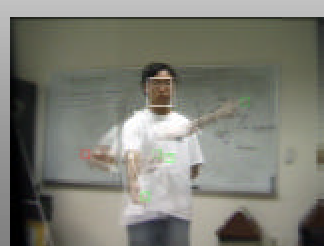


user points toward camera

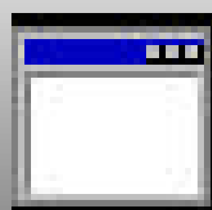
…but index finger still has higher curvature than thumb:

Saddle Point movie

..\public_html\move-sad.html



..\public_html\move-spire.html



Fin