3D Face and Hand Tracking for American Sign Language Recognition

NSF-ITR (2004-2008)

D. Metaxas, A. Elgammal, V. Pavlovic (Rutgers Univ.)

C. Neidle (Boston Univ.)

C. Vogler (Gallaudet)

The need for automated Hand and Facial analysis

- Very tedious to perform manual annotation
- Necessary large scale statistics and the study of ASL as a language require computer-based analysis
- Allows the quantitative analysis and combined statistics for the head and face
- Facilitates the discovery of new knowledge

Our Approach and Goals

- Automatically track
  - Head
  - Hands
  - Use Linguistics information in the algorithms
  - Acquire important statistics in collaboration with ASL linguists
- Goals
  - Based on their kinematic analysis perform transcription as a first step
  - Prosodic information
  - Grammatical markers
  - Affect
  - Discovery of new knowledge through large scale analysis

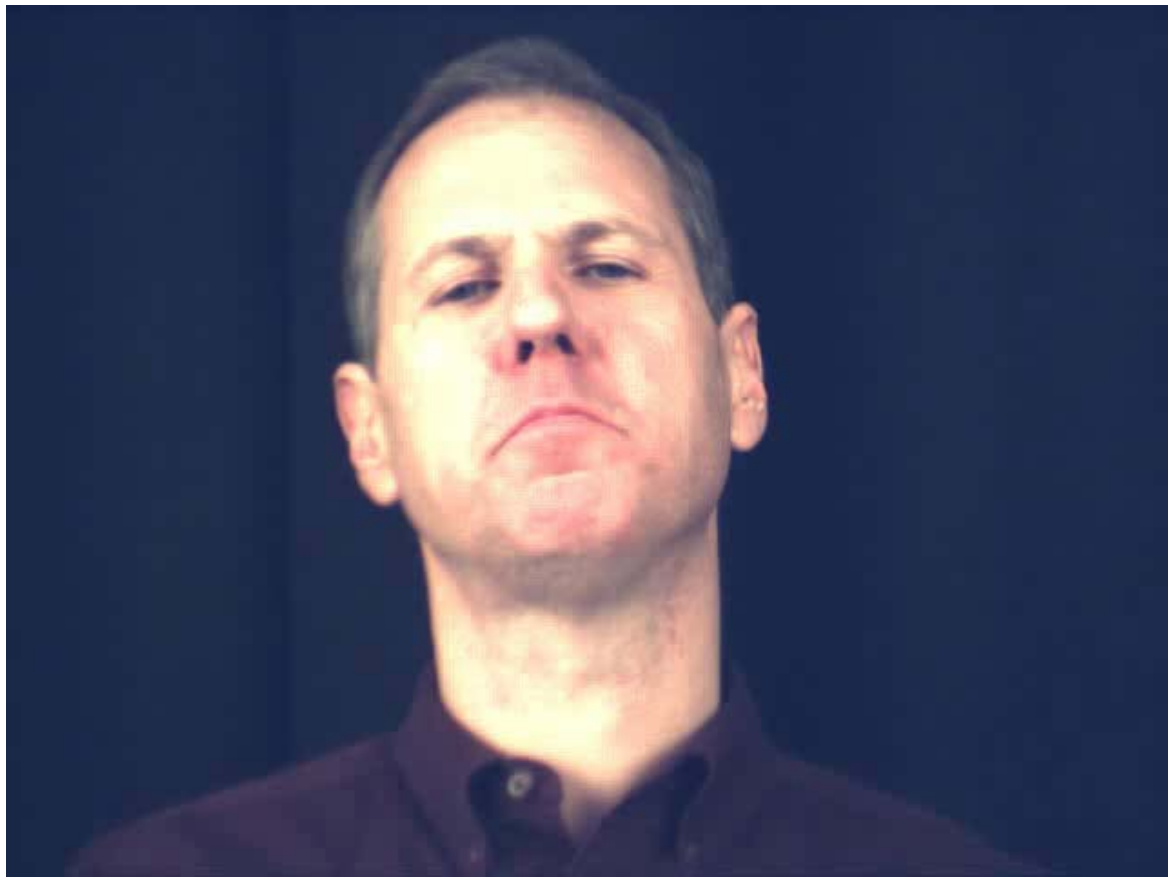
The need for facial analysis

- Lots of interesting information in head and facial movements
  - Prosodic information
  - Grammatical markers
  - Affect
- First steps:
  - Kinematic analysis
  - Detailed transcription of what is going on

Human annotations

- Humans have trouble with annotating data
- Time-consuming, boring
- Every annotation needs to be verified by experts
- Discrete vs continuous annotations

ASL video example



Human transcription

main gloss IX-1p 400 440 REMEMBER 620 740
PAST+ 940 1540 IX-1p 1660 1720 DRIVE 1820 2120

head pos: tilt fr/bk back 0 2120
head pos: turn start 660 800 right 820 1480 start 1500 1700 slightly left 1720 2120
head pos: tilt side start 100 280 right 300 1500 end 1520 1700

English translation I remember a while ago when I was driving.

Discrete annotations

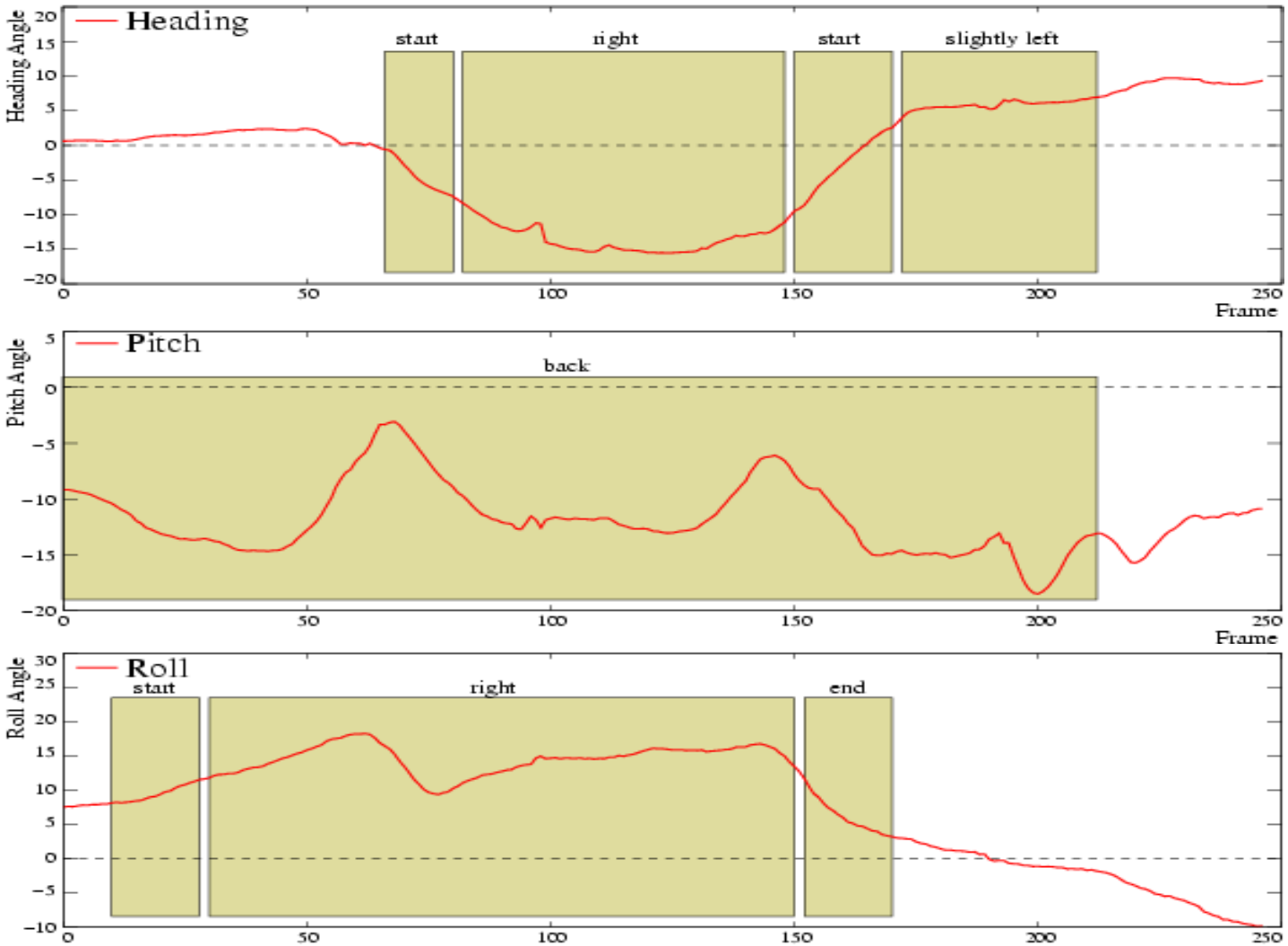
- This transcription is *discrete*
- Tells us if a certain feature is present or absent
- Does not tell us anything about varying degrees
  - Required for e.g. prosodic analysis
- Even worse ...

Kinematic analysis

- Human-made annotations are useless for kinematic analysis
- So far, the alternative was going through a video *frame by frame* and marking everything by hand
- This is where computer analysis can help ...

Tracked sequence

Computer annotation



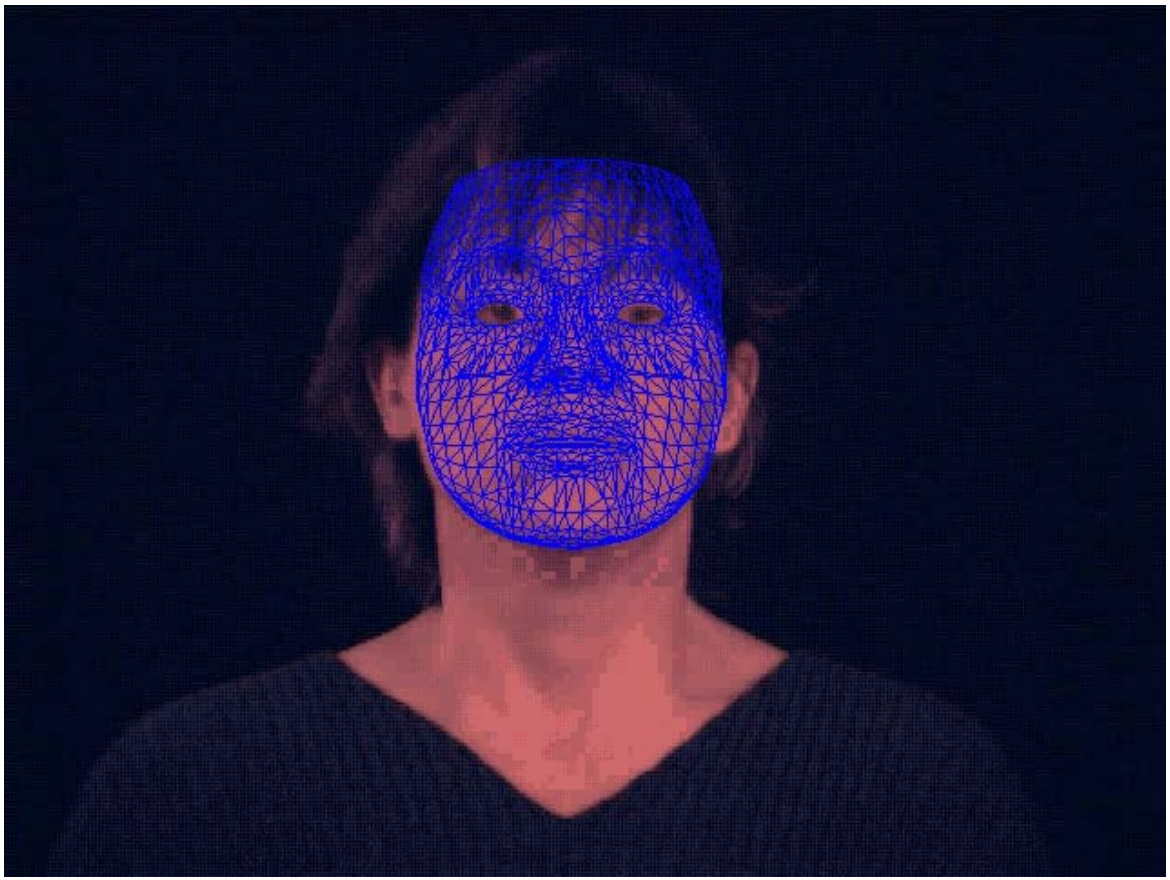
Continuous annotations

- In contrast to human annotations, the computer output contains continuous information
- Exact data signal over time, shows varying degrees of head tilt, etc.
- If the video image quality is *really good,* it is also possible to capture finer details of facial movements

Finer details

More videos

More videos



Summary of Facial Analysis

- Lots of applications
  - Linguistic analysis of ASL, cued speech, other
  - Stress recognition
  - Kinematic analysis
  - Prosodic analysis
- Pie in the sky:
  - Combine face tracking with facial expression recognition to guide and correct students on proper articulation
  - Not yet practical

Hand Tracking in (ASL)

- Most signs in ASL and other signed languages are articulated: use of particular hand-shapes, orientations, locations of articulation relative to the body.
- To recognize ASL one should first be able to capture the arm movements and hand articulations => 3D hand tracking, I.e., first perform transcription

Steps to ASL Hand Movement Analysis

Useful Constraints

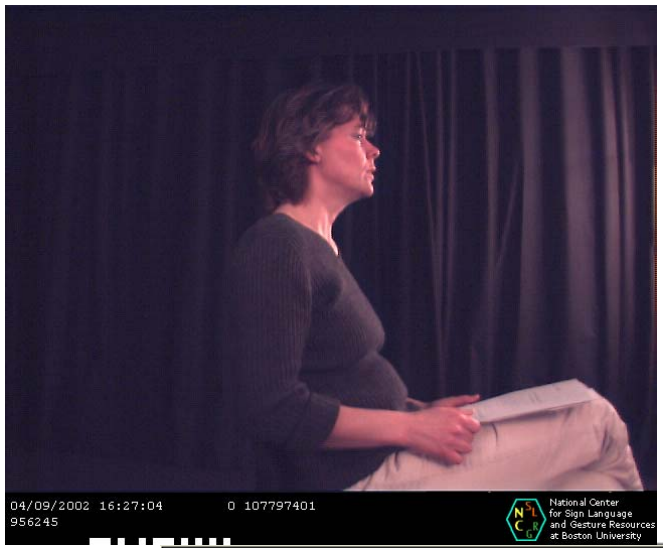
Fingerspelling vs. Continuous signs

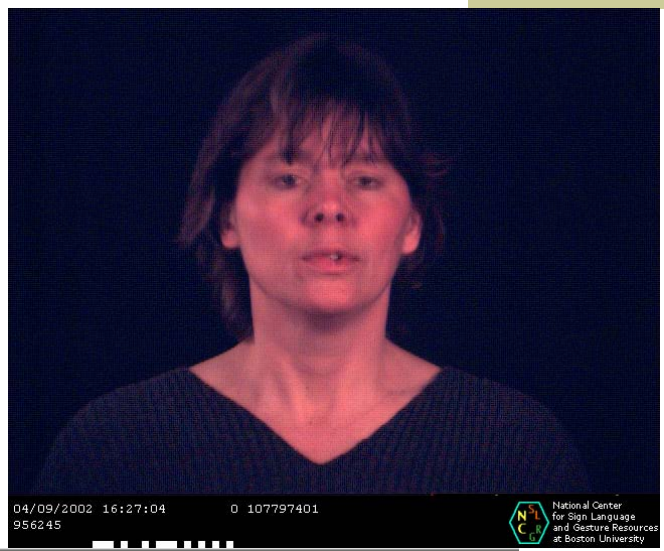
- Fingerspelling
  - 26 letters of the alphabet (for names etc.)
  - Hand moving from left to right with faster/higher finger articulations
- Continuous signs
  - Usually smoother finger articulations
  - Larger global hand displacements

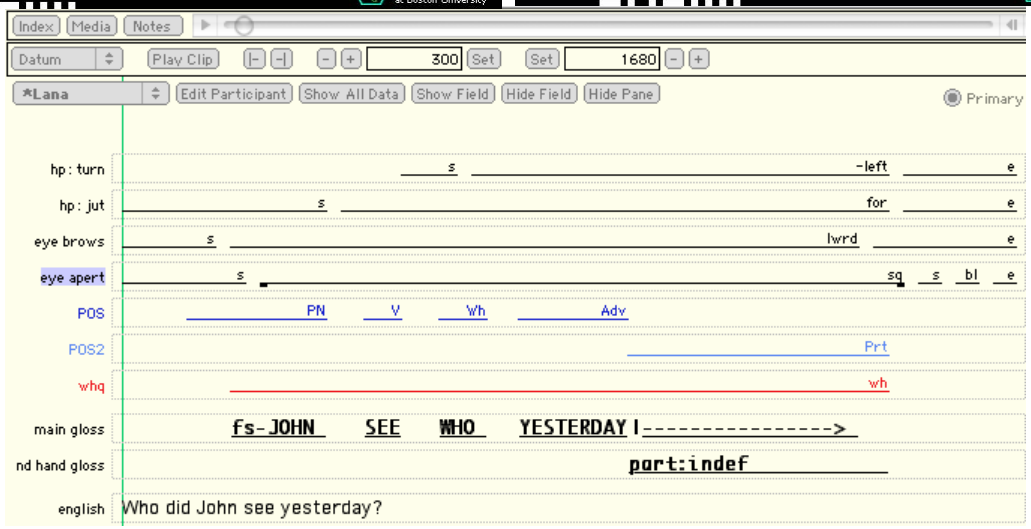
Useful Constraints (cont.)

- Two handed signs
  - Shape:
    - Both hands having the same shape
    - Different shapes:
      - Dominant/non-dominant hand
  - Movement
    - Symmetric
    - Non-symmetric
- Given a beginning hand shape, there is a limited number of possible ending shapes

Example


04/09/2002 16:27:04
956245
0 107797401
National Center for Sign Language and Gesture Resources at Boston University


04/09/2002 16:27:04
956245
0 107797401
National Center for Sign Language and Gesture Resources at Boston University


Index
Media
Notes
Datum
Play Clip
300
Set
Set
1680
*Lana
Edit Participant
Show All Data
Show Field
Hide Field
Hide Pane
Primary
hp : turn
hp : jut
eye brows
eye apert
POS
POS2
whq
main gloss
nd hand gloss
english
-left
for
lwrd
sq
bl
PN
V
Wh
Adv
Prt
wh
fs-JOHN
SEE
WHO
YESTERDAY
part:indef
Who did John see yesterday?

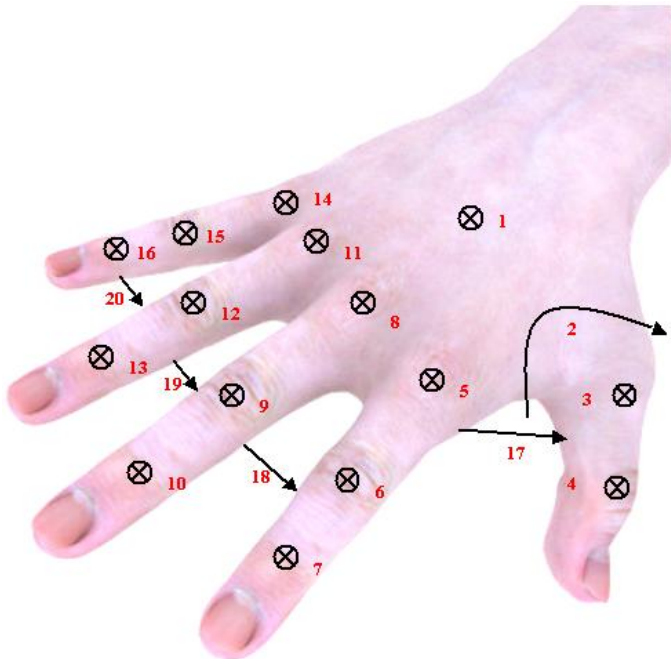
What is 3D Hand Tracking?

- Object (3D) tracking: estimate object's (3D) shape and position over time
- Hands: articulated objects
  - Position defined by the position of the wrist or the center of the palm
  - Configuration: vector containing all 3D joint angles
- Thus:

  3D Hand Tracking =

  estimate the position and the

  3D joint angle vector over time



Difficulties in Tracking

- Why is tracking a difficult problem?
  - 3D tracking is in general a difficult task (depth estimation)
  - Hands' high DoFs increase the complexity
  - Fast movements difficult to be estimated from frame to frame (motion estimation constraints)
  - Fast hand articulations
  - Occlusions
  - Hands segmentation from complicated and moving background (individual's head and torso)
  - Lighting conditions
  - Hand resolution
  - Signs are usually performed fast and with variations from the "dictionary"

3D vs. 2D Hand Tracking

- So far ASL recognition is done using primarily 2D features (2D hand shape and edges)
- 2D information is extracted efficiently but cannot describe the hand configuration explicitly
- Explicit hand configuration estimation = accuracy in recognition
- 3D+2D information is the ideal solution

3D Hand Tracking

- Continuous (temporal) tracking:

  From previous configuration(s) and motion (temporal) information, estimate current configuration

  - Fast and accurate
  - Hard to recover from error = error accumulation over time = need for model re-initializations
- Discrete tracking:

  Handle each frame separately, as a still image

  - Hand configurations database for shape retrieval
  - No error accumulation
  - Limited accuracy depending on the database size
  - Increased complexity

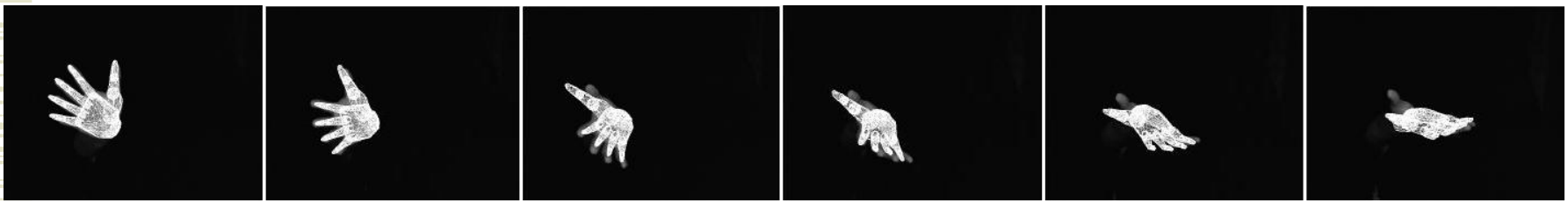
The Optimal Solution

- Use primarily continuous tracking
- When continuous tracking fails, obtain re-initialization from discrete tracking
- Efficient tracking error indication
- Optimize the discrete tracking complexity

Continuous 3D Hand Tracking

- Model-based
- 2D features used
  - 2*D* edge-driven forces
  - optical flow
  - shading
- 2D => 3D: use of a perspective camera model
  - velocity
  - acceleration
  - new position of the hand
  - model shape refinement based on the error from the cue constraints

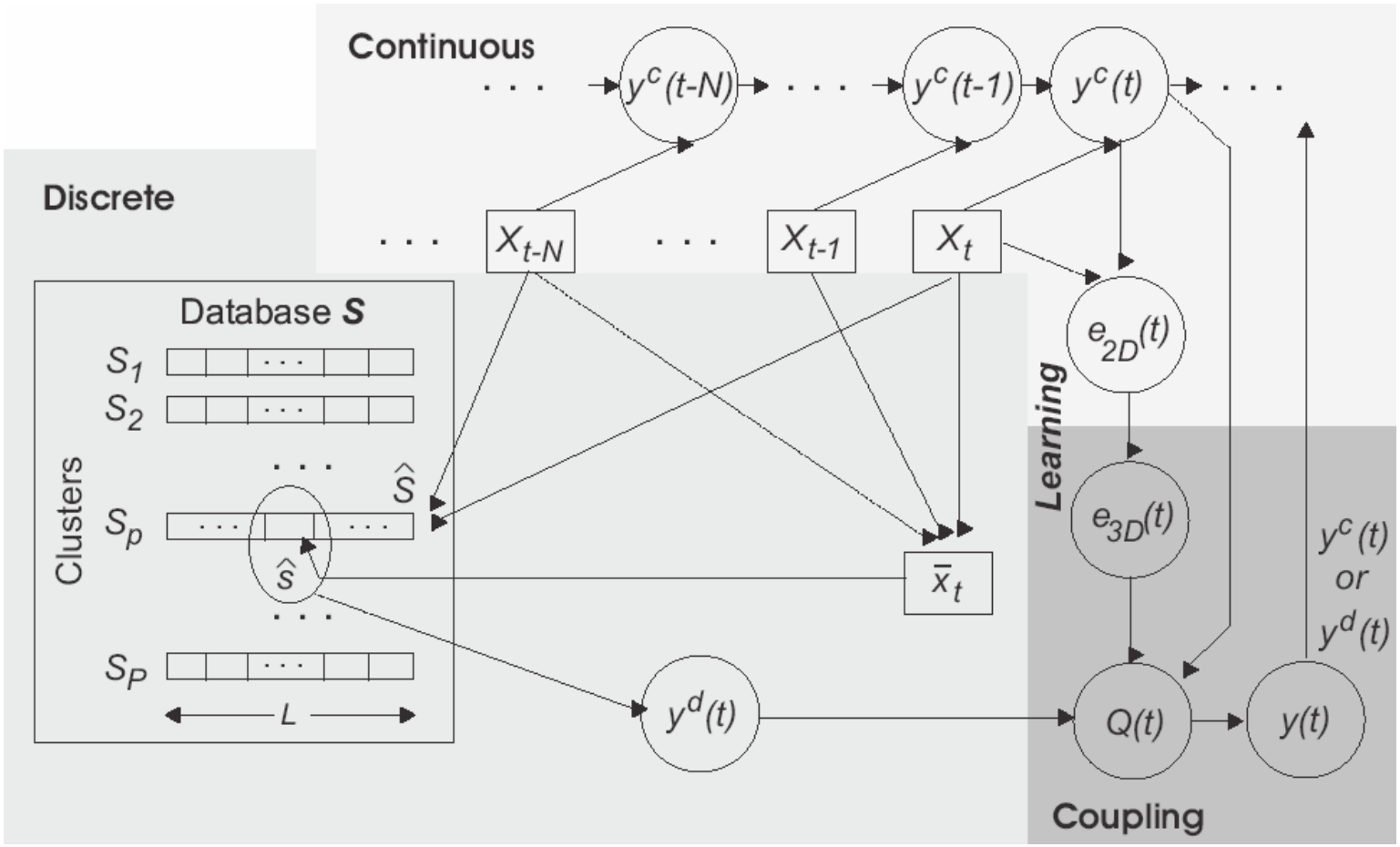
Continuous Tracking Error



Need for model re-initialization

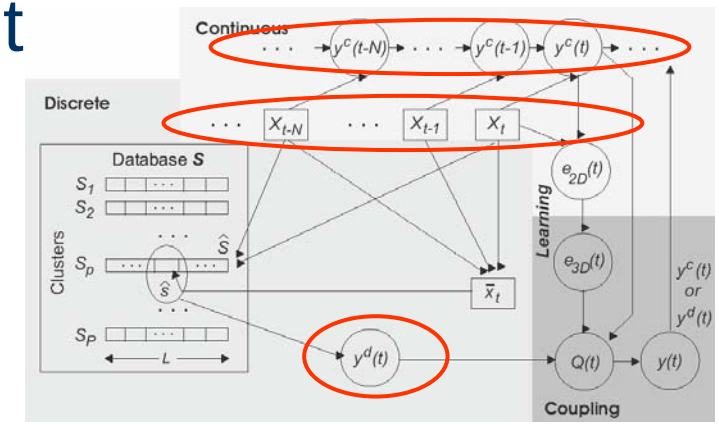
Coupling Continuous with Discrete

- Overall scheme



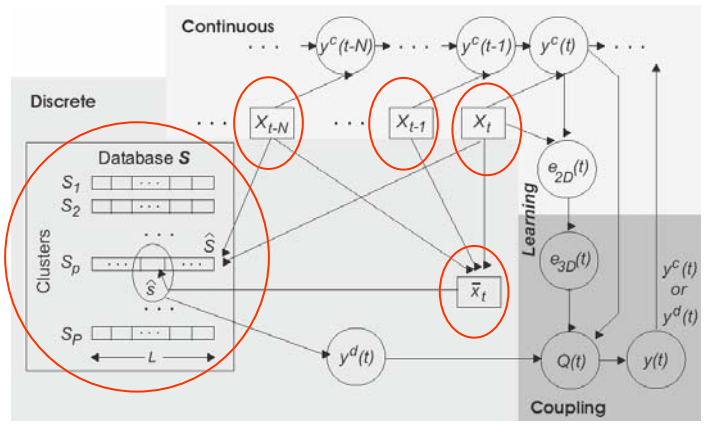
Coupling Continuous with Discrete (cont.)

- Both trackers run in parallel
- $y_c(t)$: continuous tracking result
- $y_d(t)$: discrete tracking result
- X_t: 2D observation vector
  - curvature
  - edge orientation histogram
  - Number of visible fingers
  - Hand view (palm/knuckles/side)



Coupling Continuous with Discrete (cont.)

- For the discrete tracking use configuration sequences instead of single configurations
- Database of configuration sequences
- Database clustering based on the first and last observation vectors
- Integrate the observation vector for a number of input frames (Isomap embedding)
- At each instance, locate the best database cluster to search in
- Search in the database cluster using the embedded descriptors



Coupling Continuous with Discrete (cont.)

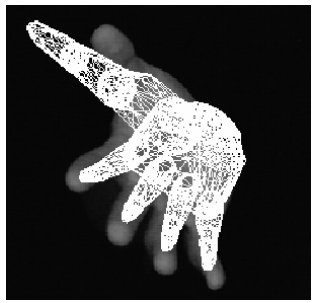
Embedded curvatures

Undirected Chamfer Distances

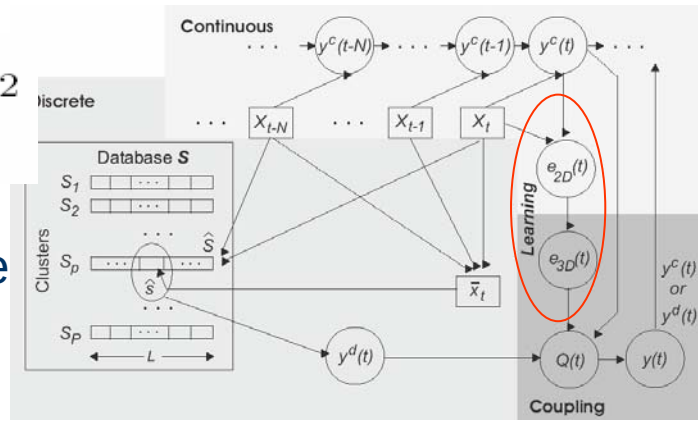
$< case >$	(a)	(b)	(c)
(b)	10.33		
(c)	7.87	6.93	
(d)	8.54	6.31	1.21

Coupling Continuous with Discrete (cont.)

- Tracking error:
  - 2D error: difference between the hand and the hand model projection on the image plane $e_{2D}(t) = d(E_b^f(t), E_b^m(t))$



  - Not always reliable: large configuration errors may correspond to small 2D errors
  - 3D error: $e_{3D}(t) = \sum_{i=1}^{\varphi} \|y_i(t) - y_i^c(t)\|^2$
  - Off-line learning 2D <-> 3D error:
    - Run continuous tracking in the database
    - Support Vector Regression



Coupling Continuous with Discrete (cont.)

- Q: decision of which solution to be used

$$P(Q(t)|e_{2D}(t)) =$$

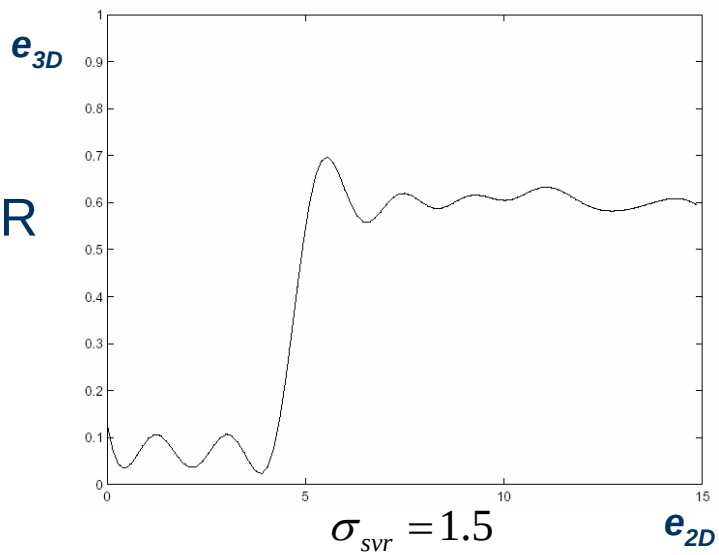
$$= P(Q(t)|e_{3D}(t)) \cdot P(e_{3D}(t)|e_{2D}(t))$$

Run continuous tracking over M database samples
and mark the failures (no probability density estimation)

$$P(Q(t) \ |e_{3D}(t)) = \sum_{i=1}^{M} p(Q(i) \ |e_{3D}(i))$$

Probability density estimation with SVR

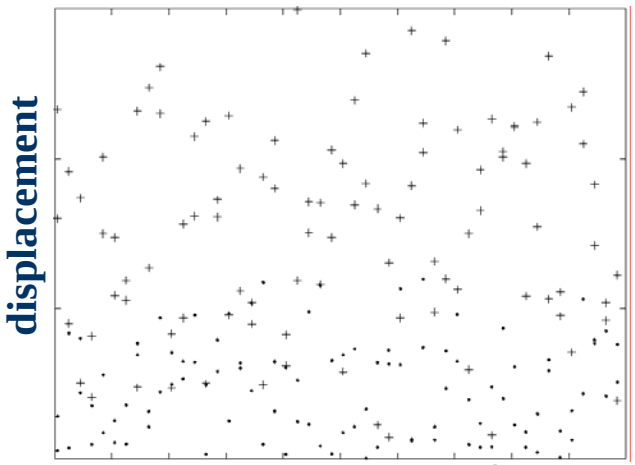
$$e_{3D} = f(e_{2D}, \sigma_{svr}, \mathbf{a}, b)$$



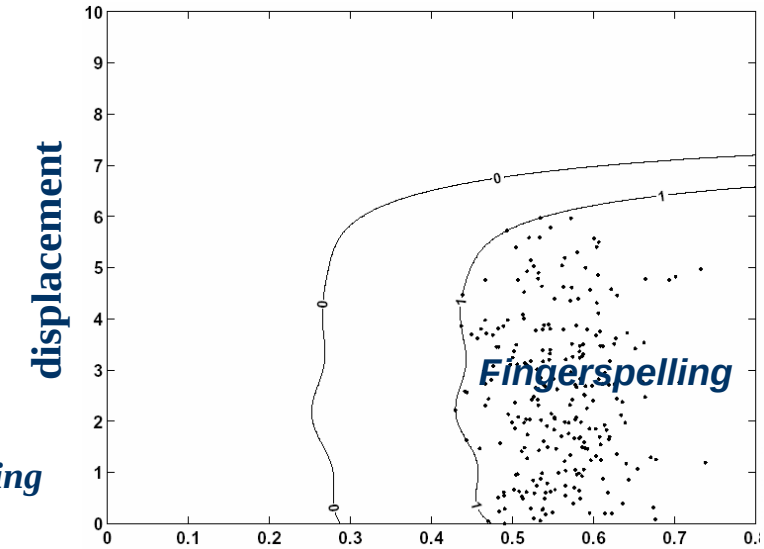
JOHN-SEE-WHO-YESTERDAY

Fingerspelling vs. Continuous Signs

- Criteria:
  - Fingers articulations (fast in fingerspelling)
  - General hand position (large displacements in continuous signing)
- Support Vector Machine Classification



Crosses: continuous signs, dots: fingerspelling



$$D_{\tilde{y}} = \frac{1}{\varphi} \sum_{i=1}^{\varphi} d\tilde{y}_i / dt$$

Fingerspelling vs. Continuous Signs (cont.)

Discovery of Informative Unlabeled Data for Improved Learning

Motivation

- The cost of acquiring labeled data is high
- However, unlabeled data are conveniently available
  - How to utilize the unlabeled data?

- Can the unlabeled data help improve the classifier?
  - Just adding the “sure” data does not help.

Previous Work: Co-Training

Two assumptions:

- Two redundant but not completely correlated feature sets
- Each feature set would be sufficient for learning if enough data were available

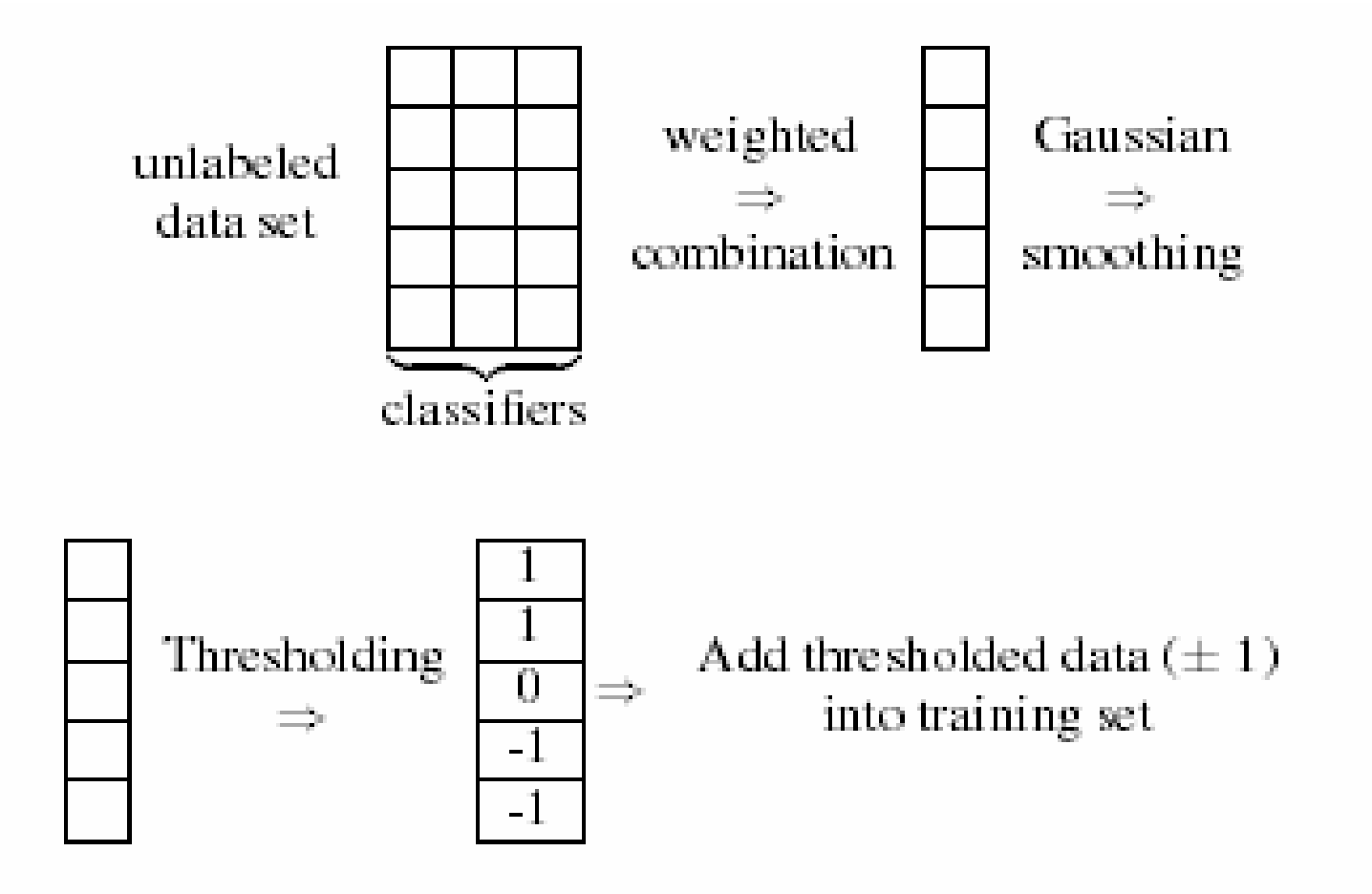
Idea: the predictions of one classifier on new unlabeled examples are expected to generate informative examples to enrich the training set of the other

However…

- The Co-Training assumptions may not hold in many computer vision applications.
- And we may have more than 2 different feature sets.

- **Idea 1:** Combine the predictions from multiple classifiers like boosting.
- **Idea 2:** Utilize the spatio-temporal pattern among the unlabeled data (informative unlabeled data can learn their labels through their neighbors)

Learning framework



Pseudo-Code

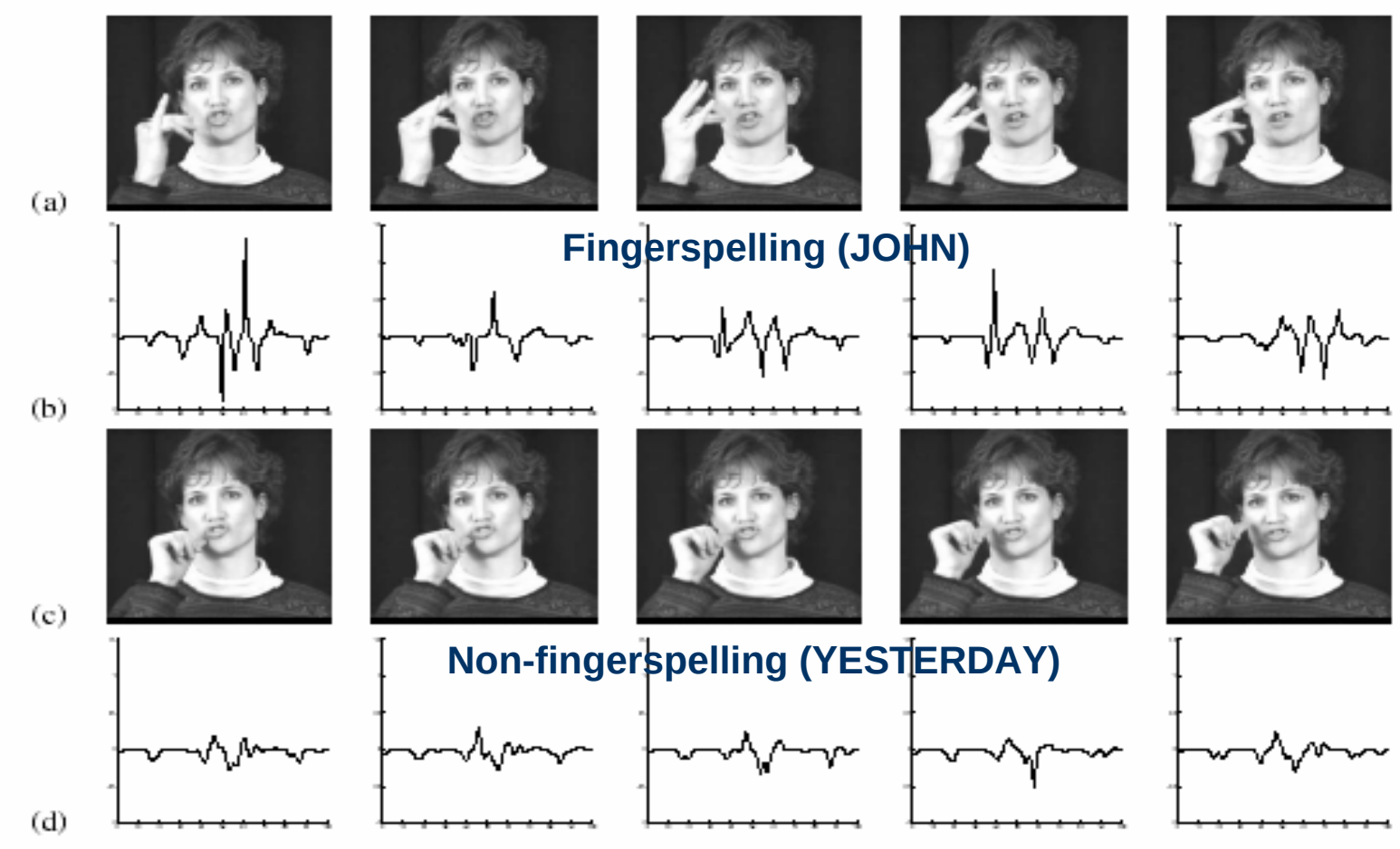
1. Input parameters: thresholds τ_1 and τ_2, standard deviation σ of discrete Gaussian kernel
2. Train Classifier $i, i = 1, \ldots, n$ on feature set i
3. ϵ_i is the prediction error on validation data.
4. $\alpha_i = \frac{1}{2} \ln(\frac{1-\epsilon_i}{\epsilon_i})/C$
5. For each unlabeled set
6. Predict: $L = \mathcal{L}(x)$
7. Weighted combination: $W = \mathcal{W}(L)$
8. Gaussian smoothing $S = W * G_\sigma$
9. Thresholding $T = \mathcal{T}(S)$
10. Add unlabeled data ($T = \pm 1$) to training set
11. Re-train Classifier $i, i = 1, \ldots, n$
12. Update ϵ_i and α_i
13. EndFor
14. Output: Classifier i, ϵ_i and $\alpha_i, i = 1, \ldots, n$

Feature Sets

- 5 consecutive frames as a group to decide the classification of the middle frame.
- Curvature of the hand contour (the middle frame)
- Changes of curvature of the hand contour

Support Vector Machines (SVM) are used as the base classifiers on each feature set (polynomial kernel with degree=3)

Fingerspelling Segmentation - Curvature



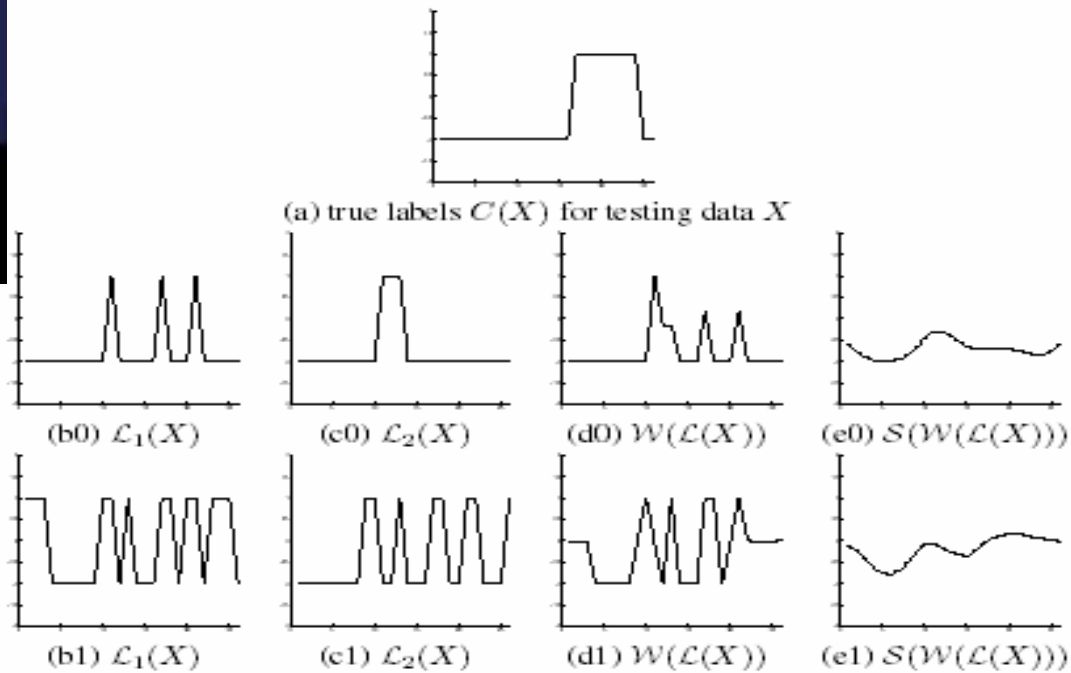
Results

"MARY"

Fingerspelling segmentation results:

Ground truth: 67 - 81

Result: 69 - 83



Results (cont.)

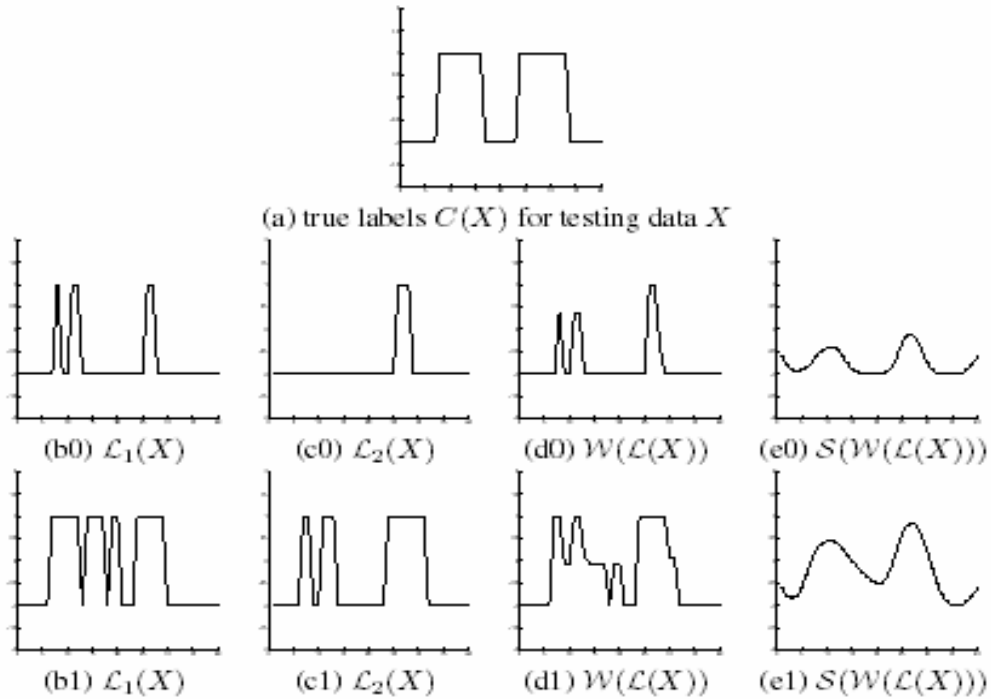
"MARY"

"JOHN"

Fingerspelling segmentation results:

Ground truth: 51 – 67 (MARY), 83 – 101 (JOHN)

Result: 49 – 63 (MARY) , 83 – 95 (JOHN)



Prediction Accuracy

	Classifier 1	Classifier 2	Final
Experiment 1	0.7308	0.5769	0.6923
	0.6538	0.6923	0.9231
Experiment 2	0.6500	0.6000	0.5250
	0.7750	0.8000	0.8500

Table 1: Prediction Accuracy. The fist row in each experiment is the result without the unlabeled data and the second row is the result with inclusion of chosen unlabeled data into the training set. Classifier 1 is trained on the curvature feature set. Classifier 2 is trained on the changes of curvature feature set.

What if …

No spatio-temporal properties?

Then we **only** present those “informative unlabeled data” for manually labeling.

In SVM: only the “support vectors” determine the final classifier. So if we had known which data are support vectors, then labeling only data is enough!

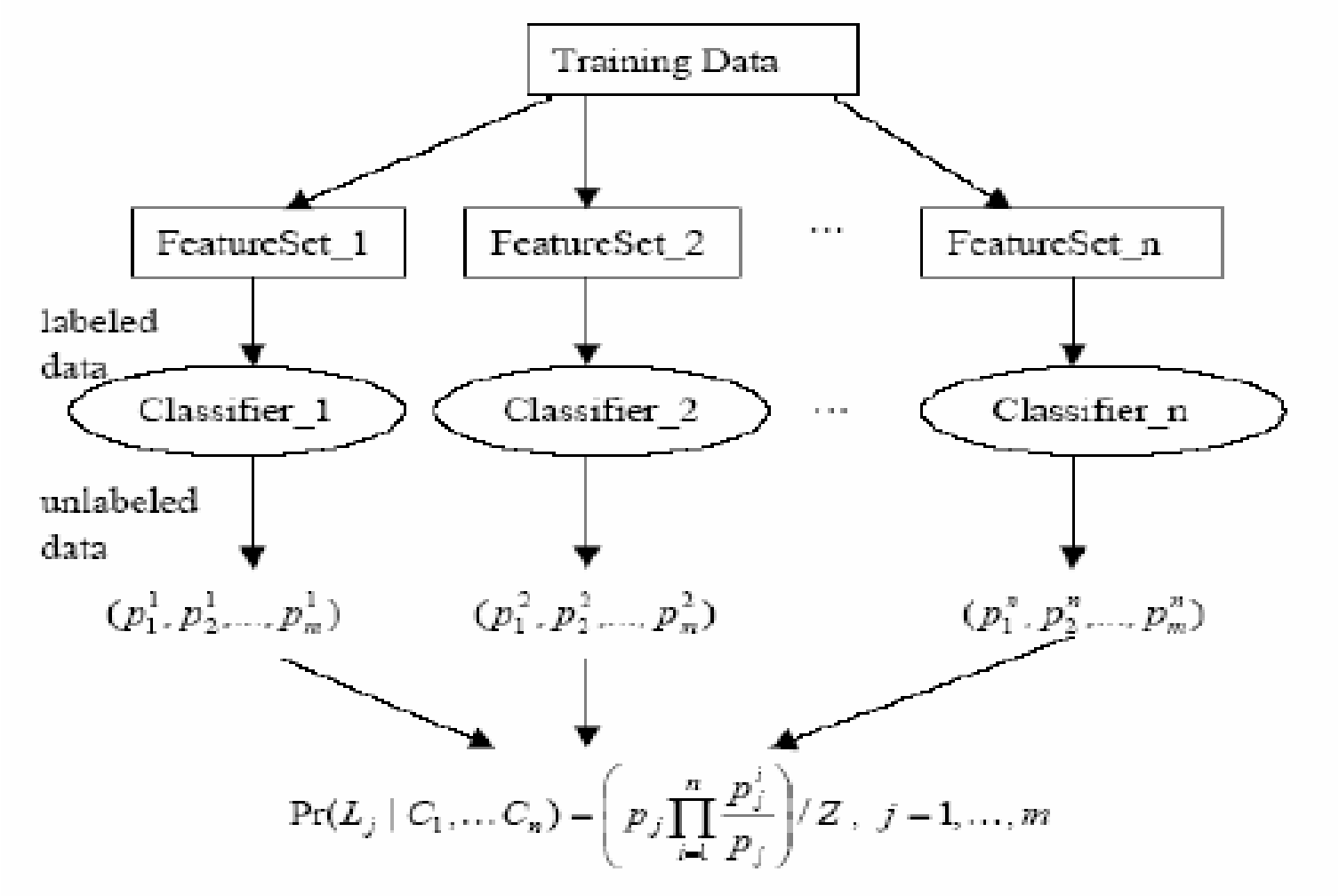
Discover informative unlabeled data

- Observation:

Support vectors are near the boundaries between two classes. where the classifier does not predict well about their labels.

Therefore, the probabilities given by the classifier can be used to discover those informative unlabeled data (for example, we can use logistic regression).

The scheme


Training Data
FeatureSet_1
FeatureSet_2
...
FeatureSet_n
labeled data
Classifier_1
Classifier_2
...
Classifier_n
unlabeled data
$(p_1^1, p_2^1, \ldots, p_m^1)$
$(p_1^2, p_2^2, \ldots, p_m^2)$
$(p_1^n, p_2^n, \ldots, p_m^n)$
$\Pr(L_j \mid C_1, \ldots C_n) = \left(p_j \prod_{i=1}^{n} \frac{p_j^i}{p_j} \right) / Z, \quad j = 1, \ldots, m$

Future Work

- Currently: We are applying our the new learning method to the 3D + 2D extracted data.

- Make tracking fast, close to real-time
- Build extensive database - “dictionary”
- Track two-handed signs
- Dominant hand recognition
- Continuous signing recognition based on the dictionary
- Fingerspelling recognition: retrieve the word from the first, last and some intermediate fingerspelled letters