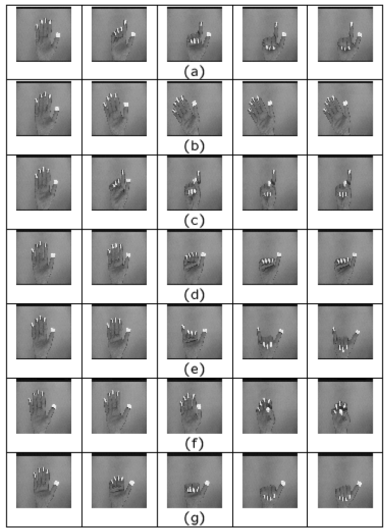
Hand Gesture Recognition, Aerobic exercises, Events

Lecture-15

Hand Gesture Recognition

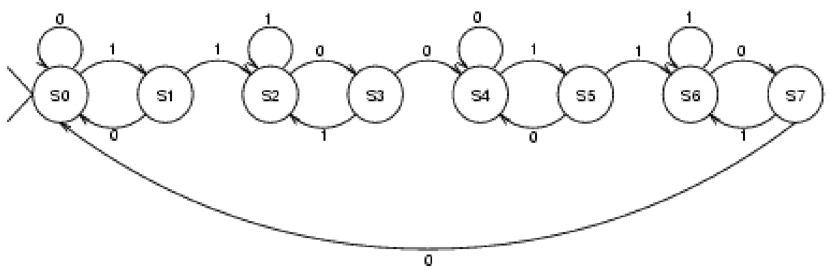
Seven Gestures



Gesture Phases

- Hand fixed in the start position.
- Fingers or hand move smoothly to gesture position.
- Hand fixed in gesture position.
- Fingers or hand return smoothly to start position.

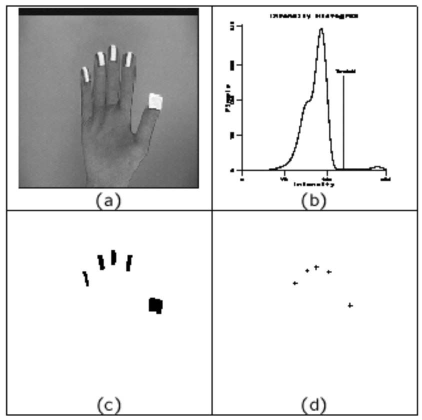
Finite State Machine



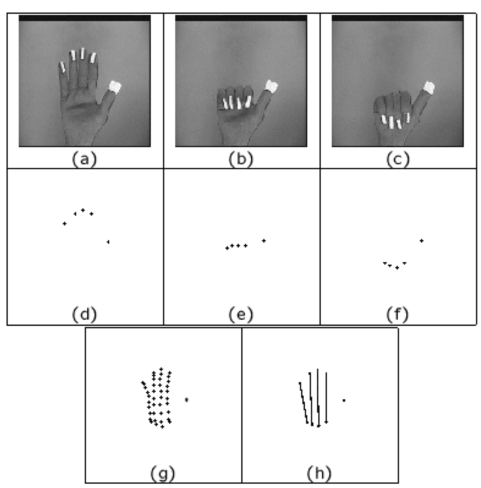
Main Steps

- Detect fingertips.
- Create fingertip trajectories using motion correspondence of fingertip points.
- Fit vectors and assign motion code to unknown gesture.
- Match

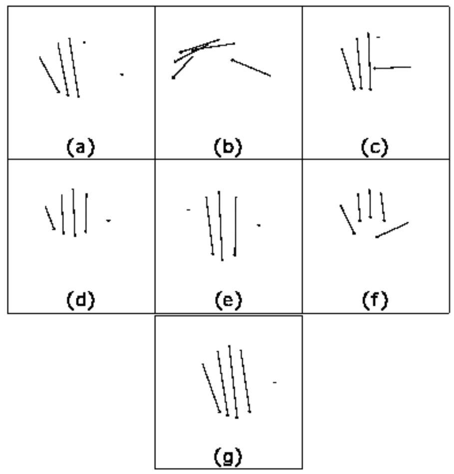
Detecting Fingertips



Vector Extraction



Vector Representation of Gestures



Results

Results

Run	Frames	L	R	U	D	T	G	S
1	200	√	√	√	√	√	√	√
2	250	√	√	√	√	√	√	√
3	250	√	√	√	X	√	√	√
4	250	√	√	√	√	√	√	√
5	300	√	√	√	√	√	√	√
6	300	√	√	√	√	√	√	√
7	300	√	√	√	√	√	√	√
8	300	√	√	√	√	√	√	√
9	300	√	√	√	√	*	*	*
10	300	√	√	√	√	√	√	√

L = Left, R = Right, U = Up, D = Down, T = Rotate, G = Grab, S = Stop, √ - Recognized, X - Not Recognized, * - Error in Sequence.

Publication

- http://www.cs.ucf.edu/~vision/papers/shah/94/DAS94.pdf (James Davis and Mubarak Shah. Visual Gesture Recognition, Vision, Image and Signal Processing, Vol 141, No. 2, April 1994.)

- http://www.cs.ucf.edu/~vision/papers/CS-TR-93-11.pdf (James Davis and Mubarak Shah. Gesture Recognition, European Conference on Computer Vision,1994.)

Action Recognition Using Temporal Templates

Jim Davis and Aaron Bobick

Main Points

- Compute a sequence of difference pictures from a sequence of images.
- Compute Motion Energy Images (MEI) and Motion History Images (MHI) from difference pictures.
- Compute Hu moments of MEI and MHI.
- Perform recognition using Hu moments.

MEI and MHI

Motion-Energy Images (MEI)

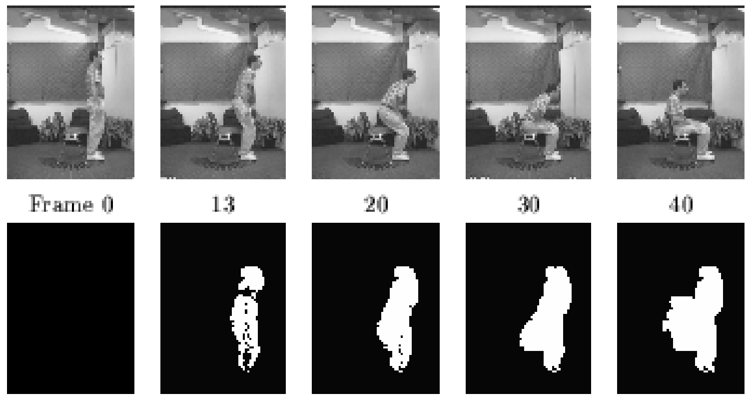
$$E_\tau(x,y,t) = \bigcup_{i=0}^{\tau-1} D(x,y,t-i)$$

Change Detected Images

Motion History Images (MHI)

$$H_\tau(x,y,t) = \begin{cases} \tau & if\ D(x,y,t)=1 \\ \max(0, H_\tau(x,y,t-1)-1) & otherwise \end{cases}$$

MEIs



Color MHI Demo

Summary

- Use seven Hu moments of MHI and MEI to recognize different exercises.
- Use seven views (-90 degrees to +90 degrees in increments of 30 degrees).
- For each exercise several samples are recorded using all seven views, and the mean and covariance matrices for the seven moments are computed as a model.
- During recognition, for an unknown exercise all seven moments are computed, and compared with all 18 exercises using Mahalanobis distance.
- The exercise with minimum distance is computed as the match.
- They present recognition results with one and two view sequences, as compared to seven view sequences used for model generation.

Moments

Binary image

General Moments

$$m_{pq} = \int \int x^p y^q \rho(x, y) dxdy$$

Central Moments (Translation Invariant)

$$\mu_{pq} = \int \int (x - \bar{x})^p (y - \bar{y})^q \rho(x, y) \; d(x - \bar{x}) d(y - \bar{y})$$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$$

centroid

Central Moments

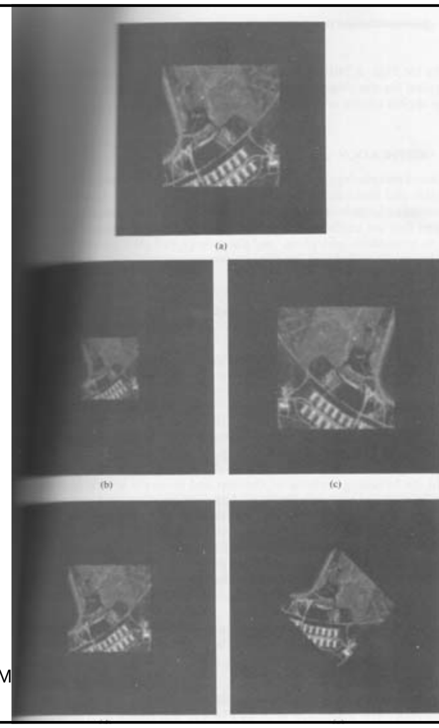
$$\mu_{00} = m_{00} \equiv \mu$$
$$\mu_{01} = 0$$
$$\mu_{10} = 0$$
$$\mu_{20} = m_{20} - \mu\bar{x}^2$$
$$\mu_{11} = m_{11} - \mu\overline{xy}$$
$$\mu_{02} = m_{02} - \mu\bar{y}^2$$
$$\mu_{30} = m_{30} - 3m_{20}\bar{x} + 2\mu\bar{x}^3$$
$$\mu_{21} = m_{21} - m_{20}\bar{y} - 2m_{11}\bar{x} + 2\mu\bar{x}^2 y$$
$$\mu_{12} = m_{12} - m_{02}\bar{x} - 2m_{11}\bar{y} + 2\mu\bar{x}y^2$$
$$\mu_{03} = m_{03} - 3m_{02}\bar{y} + 2\mu\bar{y}^3$$

Moments

Hu Moments: translation, scaling and rotation invariant

$$\upsilon_1 = \mu_{20} + \mu_{02}$$
$$\upsilon 2 = (\mu_{20} - \mu_{02})^2 + \mu_{11}{}^2$$
$$\upsilon_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{12} - \mu_{03})^2$$
$$\upsilon_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$
$$\vdots$$

Half size, mirror
Rotated 2, rotated 45

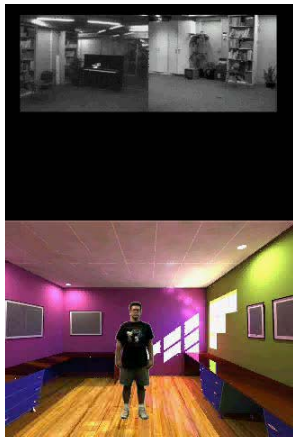


Copyright M

Table 8.2 Moment Invariants for the Images in Figs. 8.24(a)–(e)

Invariant (Log)	*Original*	*Half Size*	*Mirrored*	*Rotated 2°*	*Rotated* [illegible]
ϕ_1	6.249	6.226	6.919	6.253	6.318
ϕ_2	17.180	16.954	19.955	17.270	16.803
ϕ_3	22.655	23.531	26.689	22.836	19.724
ϕ_4	22.919	24.236	26.901	23.130	20.437
ϕ_5	45.749	48.349	53.724	46.136	40.525
ϕ_6	31.830	32.916	37.134	32.068	29.315
ϕ_7	45.589	48.343	53.590	46.017	40.470

Hu moments



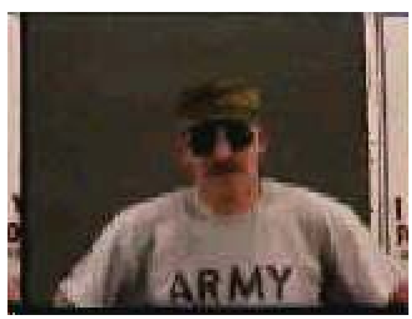
PAT (Personal Aerobic Trainer)

Copyright Mubarak Shah 2003

PAT (Personal Aerobic Trainer)

http://vismod.www.media.mit.edu/vismod/demos/actions/mhi_generation.mov

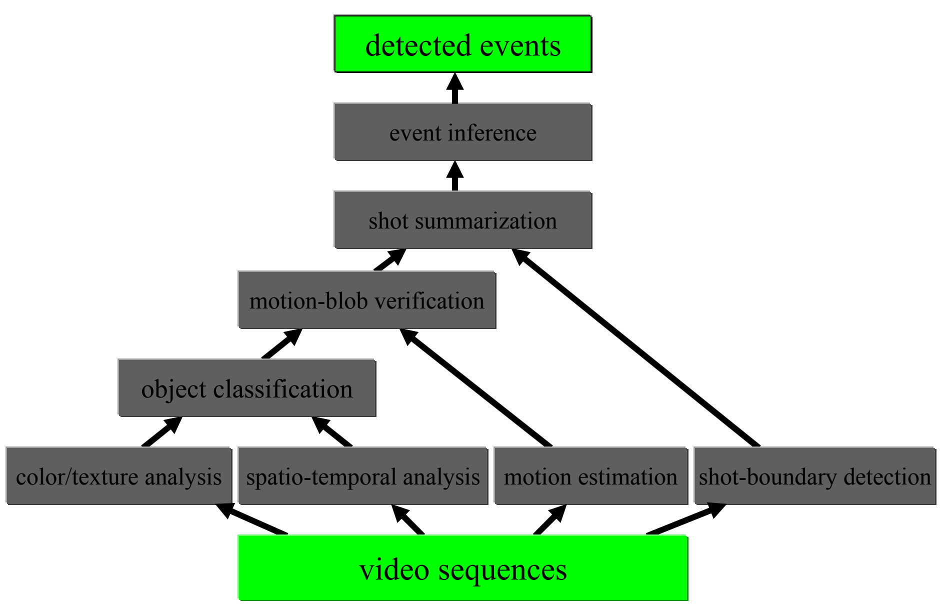
PAT (Personal Aerobic Trainer)



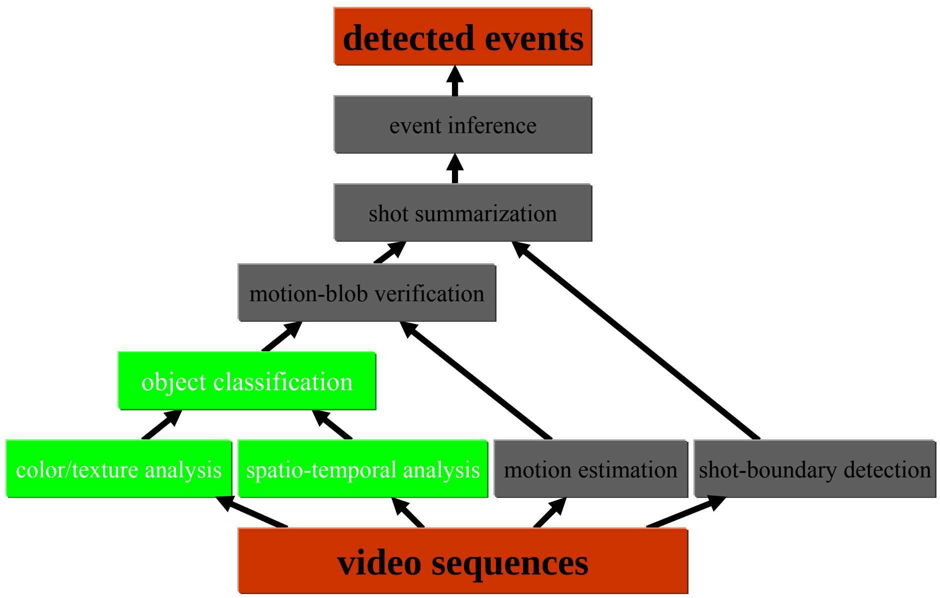
A Framework for the Design of Visual Event Detectors

Niels Haering

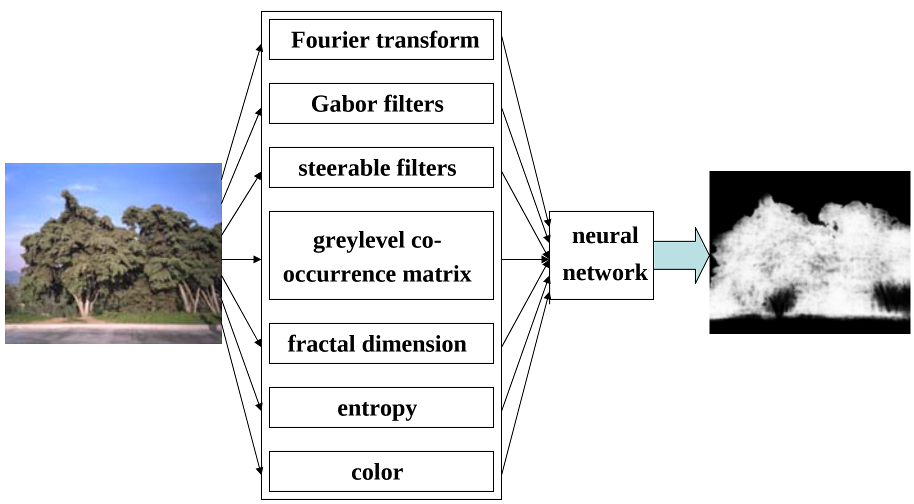
Our Framework



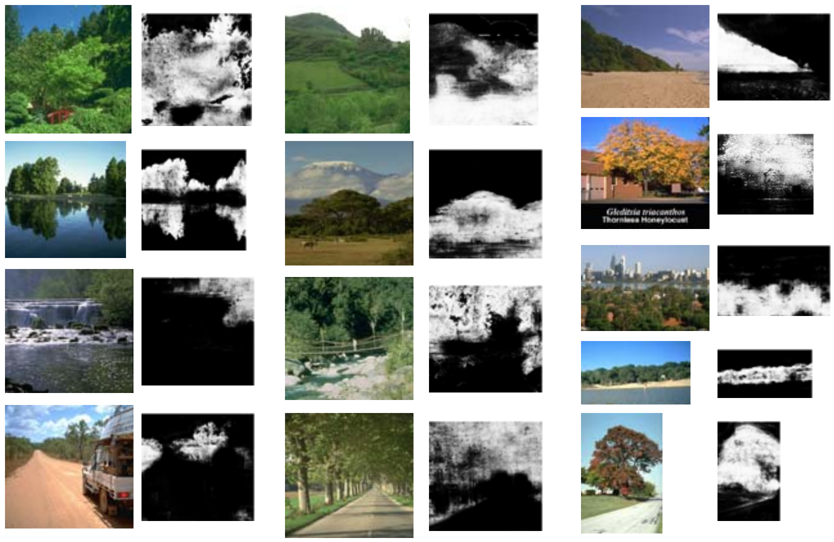
Object Classification



Object Classification

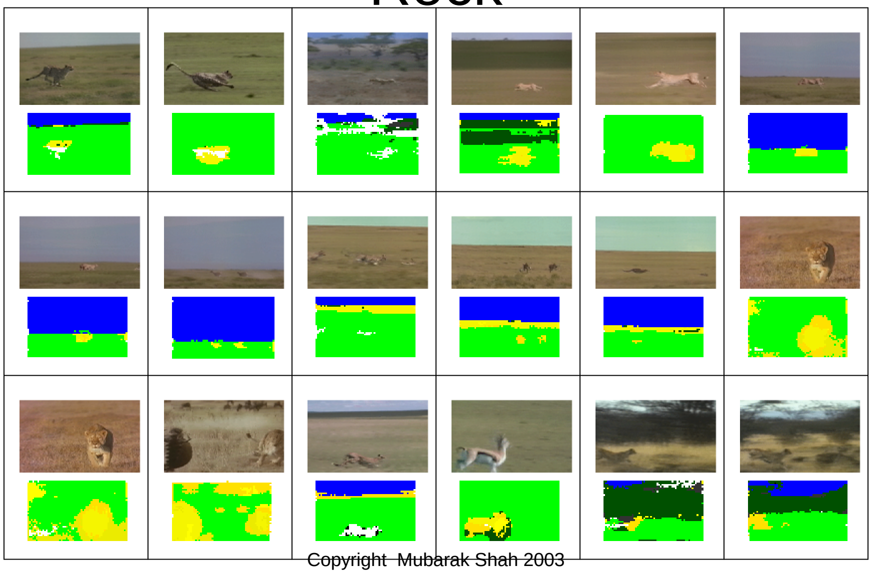


Deciduous Trees



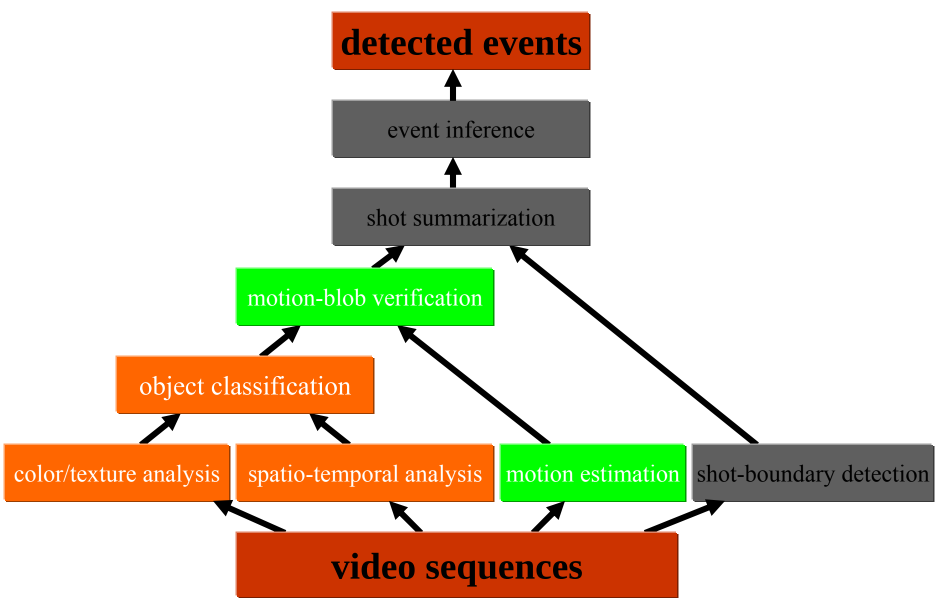
Sky

Animals, Sky, Grass, Trees, Rock

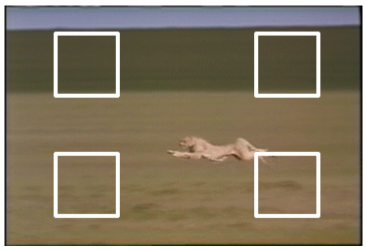


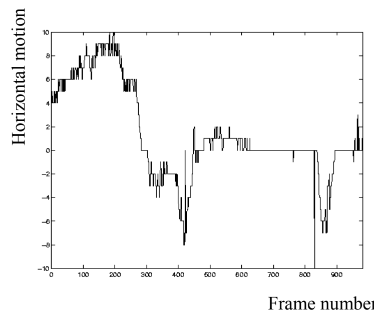
Copyright Mubarak Shah 2003

Motion-blob Verification



Motion Estimation

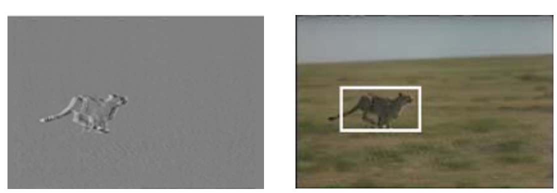




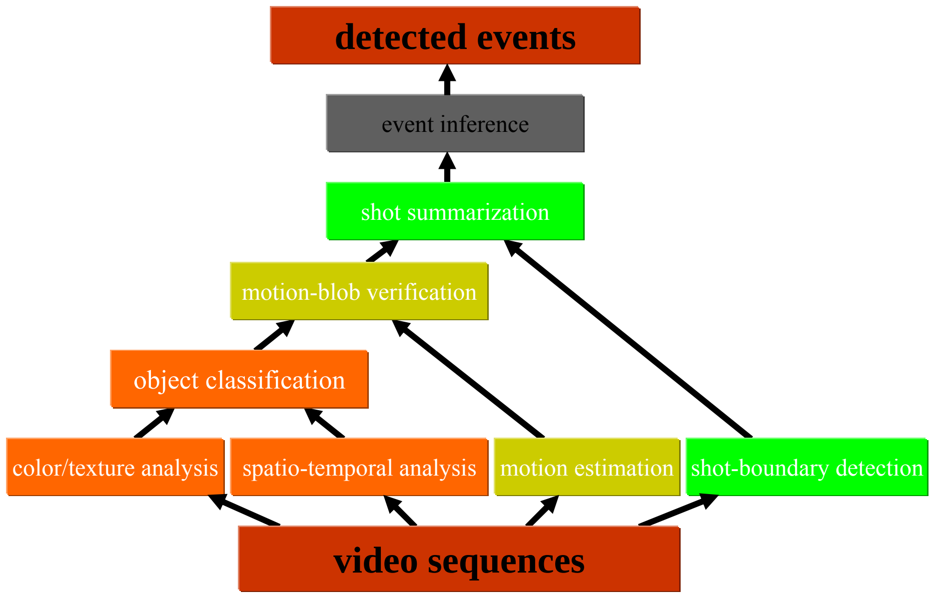
- three parameter system: x-, y-translation, and zoom,
- 4 motion estimates based on pyramid,
- 4 motion estimates based on previous best match,
- “texture” measure prevents ambiguous matches

Motion-blob detection

Motion estimate
Δx = -7
Δy = 0
zoom = 1.0



Shot Summarization

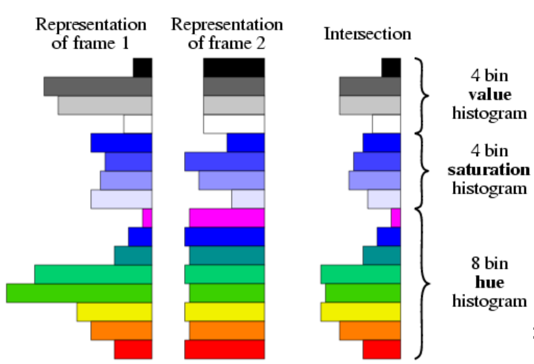


Shot Detection

Characteristics of shot boundaries:

- Change of camera/viewpoint
- Change of color characteristics

4 Bins for Value
4 Bins for Saturation
8 bins for hue



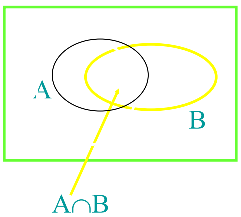
= 0.79

$$\cap = \frac{\sum_{n=0}^{15} \min(f_1(n), f_2(n))}{\min(\sum_{n=0}^{15}(f_1(n), \sum_{n=0}^{15} f_2(n))}$$

Shot Summaries

[redacted]

[redacted]



A = Ground Truth

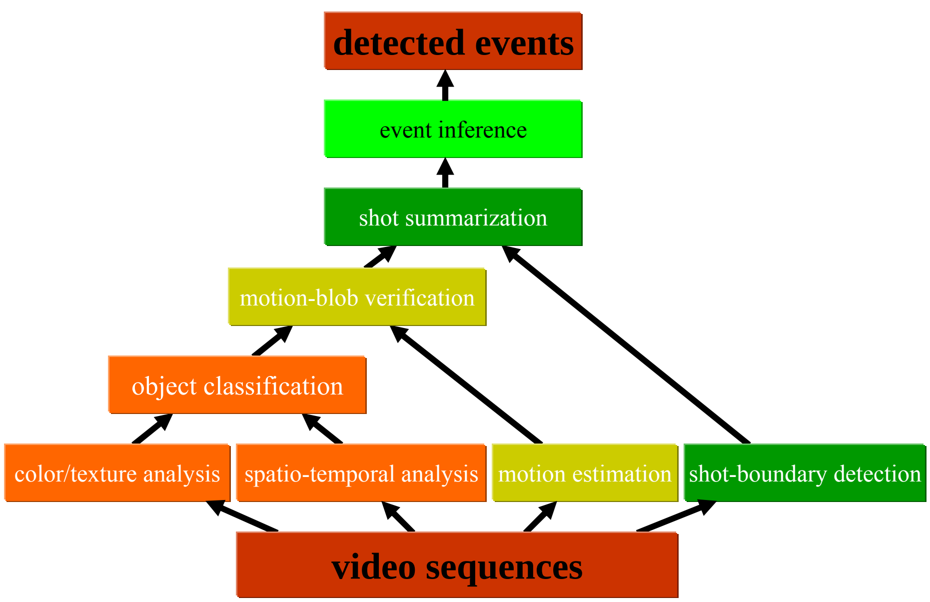
B = Result of Algorithm

A∩B = Correct detection

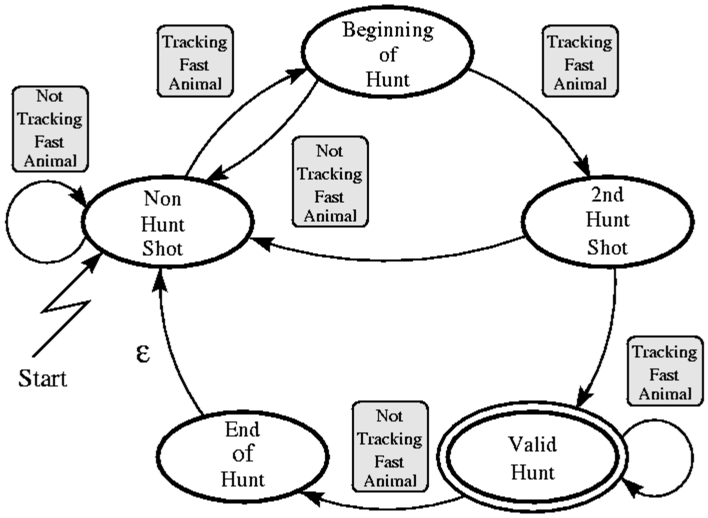
$$recall = \frac{A \cap B}{A}$$

$$precision = \frac{A \cap B}{B}$$

Event Inference



Hunt events



Hunts

Hunt

Non-hunt

Hunts

Non-hunt

Hunt

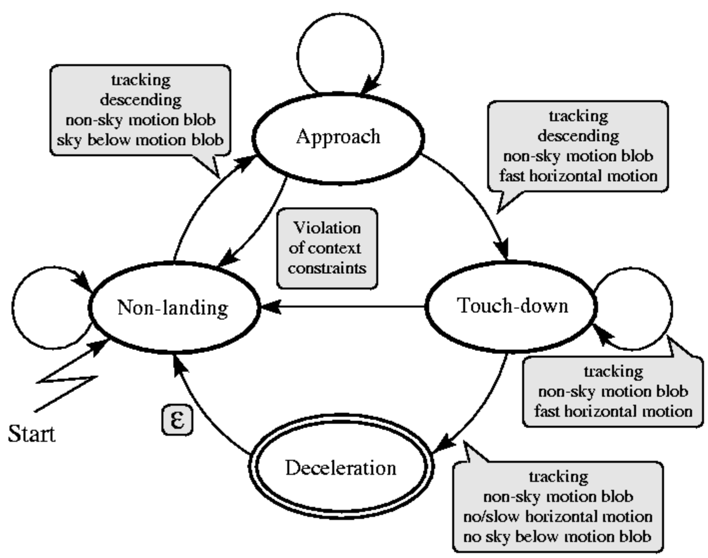
Non-hunt

Event Detection

Sequence Name	Actual Hunt Frames	Detected Hunt Frames	Precision	Recall
Hunt1	305 – 1375	305 – 1375	100%	100%
Hunt2	2472 – 2696	2472 – 2695	100%	99.6%
Hunt3	3178 – 3893	3178 – 3856	100%	94.8%
Hunt4	6363 – 7106	6363 – 7082	100%	96.8%
Hunt5	9694 – 10303	9694 – 10302	100%	99.8%
Hunt6	12763 – 14178	12463 – 13389	67.7%	44.2%
Hunt7	16581 – 17293	16816 – 17298	99.0%	67.0%
average			95.3%	86.0%

Landing Events

Landing Events



Landing Events

Non-landing

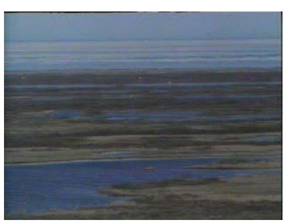
Approach

Touch-down

Deceleration

Landing Events

Non-landing



Approach

Touch-down

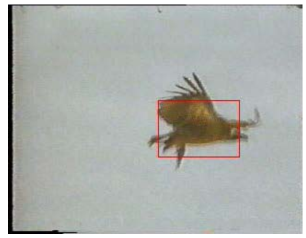
Deceleration

Non-landing

Landing Events

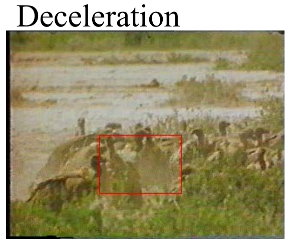
Non-landing

Approach

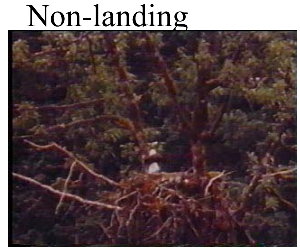


Touch-down

Deceleration



Non-landing



Conclusions

- Many natural objects are easily recognized by their color and texture signatures (shape is often not needed)
- Many events are easily detected and recognized by the classes of the comprising objects and their approximate motions
- The proposed visual event detection is robust to changes in scale, color, shape, occlusion, lighting conditions, view points and distances, and image compression

Publications

- Niels Haering and Niels da Vitoria Lobo. Features and Classification Methods to Locate Deciduous Trees in Images, Journal of Computer Vision and Image Understanding, 1999.
- Niels Haering, Richard Qian, and Ibrahim Sezan. A Semantic Event Detection Approach and Its Application to Detecting Hunts in Wildlife Video , IEEE Transactions on Circuits and Systems for Video Technology, 1999.
- VISUAL EVENT DETECTION, Niels Haering and Niels da Vitoria Lobo, Kluwer Academic Publishers, 2001.

Monitoring Human Behavior

Lecture-16

Monitoring Human Behavior

http://www.cs.ucf.edu/~vision/projects/Office/Office.html

Goals of the System

- Recognize human actions in a room for which **prior knowledge** is available.
- Handle multiple people
- Provide a textual description of each action
- Extract “key frames” for each action

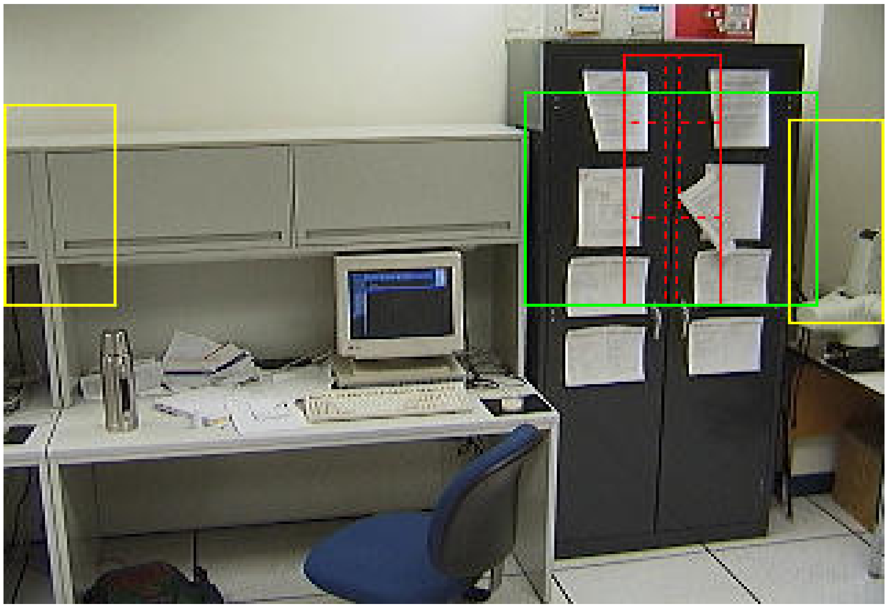
Possible Actions

- **Enter**
- **Leave**
- **Sitting** or **Standing**
- **Picking Up Object**
- **Put Down Object**
- **…..**

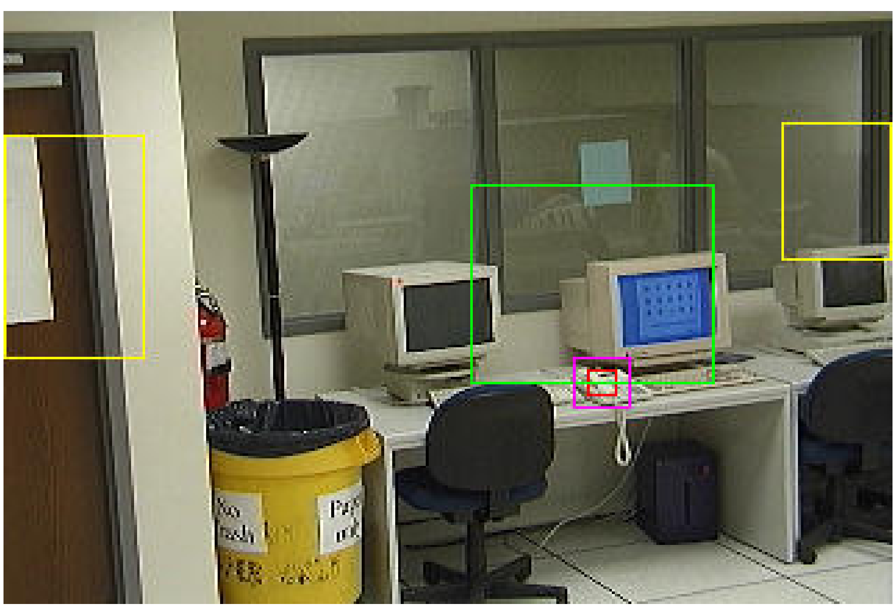
Prior Knowledge

- Spatial layout of the scene:
  - Location of **entrances** and **exits**
  - Location of **objects** and some information about how they are use
- Context can then be used to improve recognition and save computation

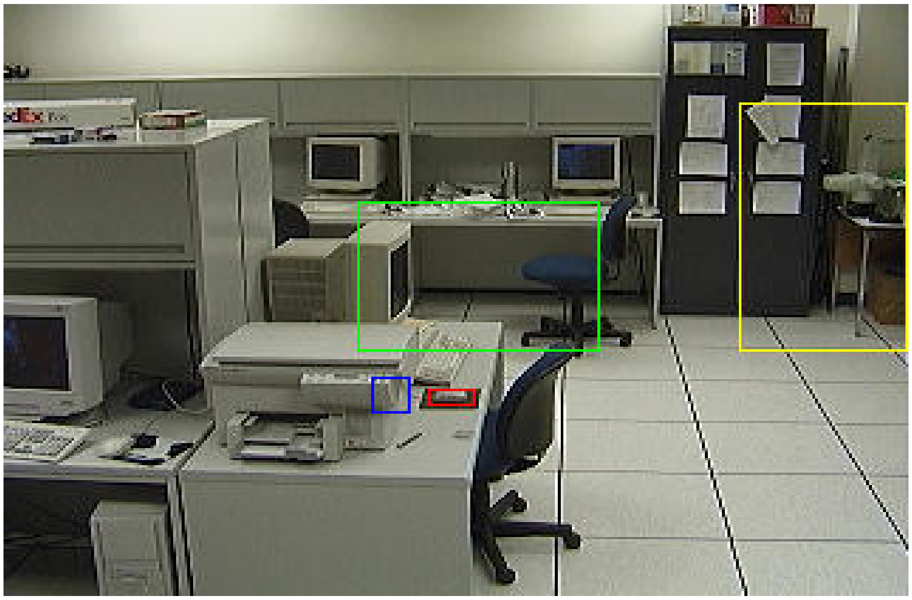
Layout of Scene 1



Layout of Scene 2



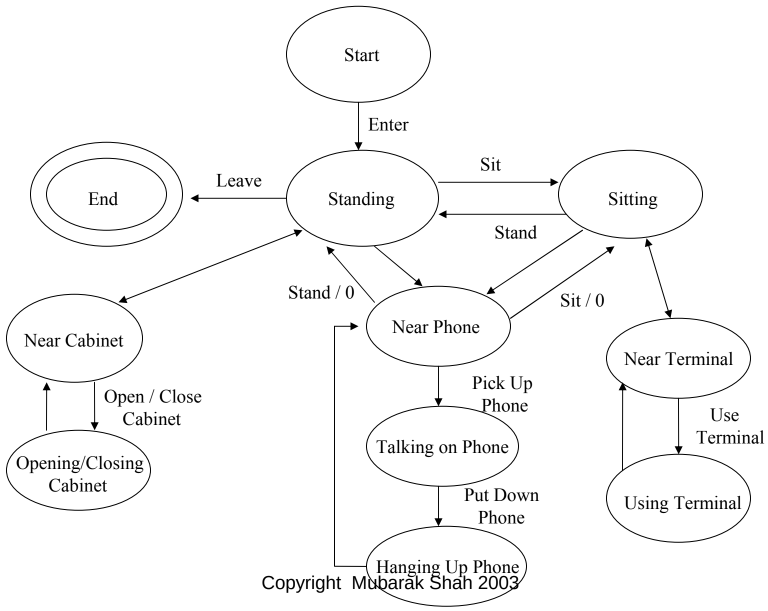
Layout of Scene 4



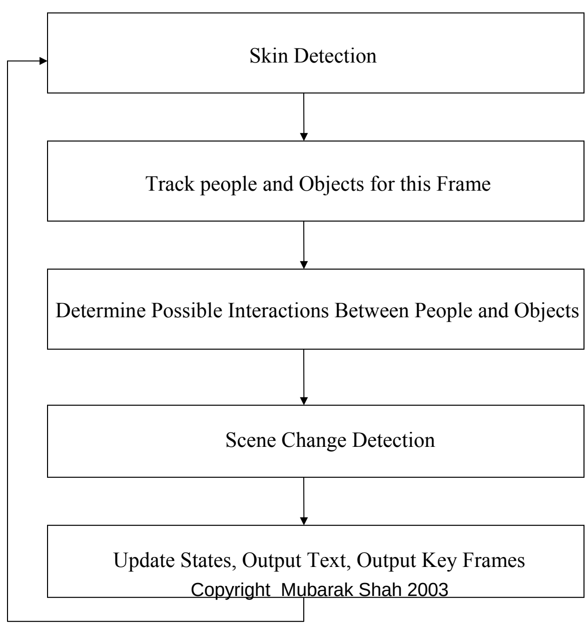
Major Components

- Skin Detection
- Tracking
- Scene Change Detection
- Action Recognition

State Model For Action Recognition



Flow of the System



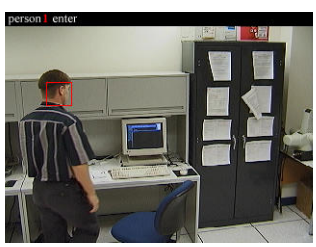
Key Frames

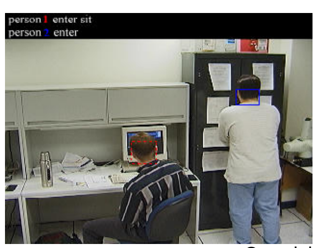
- Why get key frames?
  - Key frames take less space to store
  - Key frames take less time to transmit
  - Key frames can be viewed more quickly
- We use heuristics to determine when key frames are taken
  - Some are taken before the action occurs
  - Some are taken after the action occurs

Key Frames

- "Enter" key frames: as the person leaves the entrance/exit area
- "Leave" key frames: as the person enters the entrance/exit area
- "Standing/Sitting" key frames: after the tracking box has stopped moving up or down respectively
- "Open/Close" key frames: when the % of changed pixels stabilizes

Key Frames Sequence 1 (350 frames), Part 1

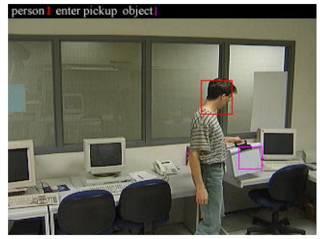




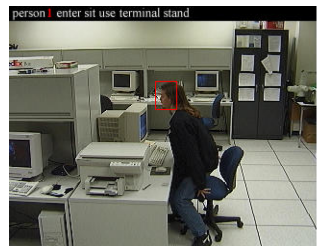
Key Frames Sequence 1 (350 frames), Part 2

Key Frames Sequence 2 (200 frames)

Key Frames Sequence 3 (200 frames)



Key Frames Sequence 4 (399 frames), Part 1



Key Frames Sequence 4 (399 frames), Part 2