Shape Recognition for Multi-Touch Table

Yehoraz Kasher Project Competition

10/06/2012

Students: Itay Hubara
Amit Nishry

Supervisor: Yair Moshe

Outline

- Introduction
- System
- Algorithm
- Implementation
- Demo Game
- Conclusion

Background

- Multi-touch is starting to appear in every day computers
- New User Interfaces (NUI) are extensively pursued

Motivation

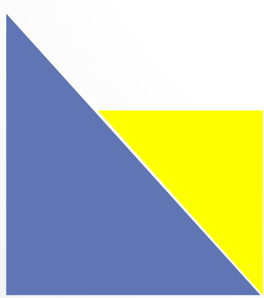
- Existing multi-touch surfaces are expensive
- Current technology is unable to recognize shapes
- Shape recognition enhances user interface

Goals

- Design a shape recognition algorithm
- Track and recognize combined shapes
- Integrate into open-source environment
- Real-Time performance

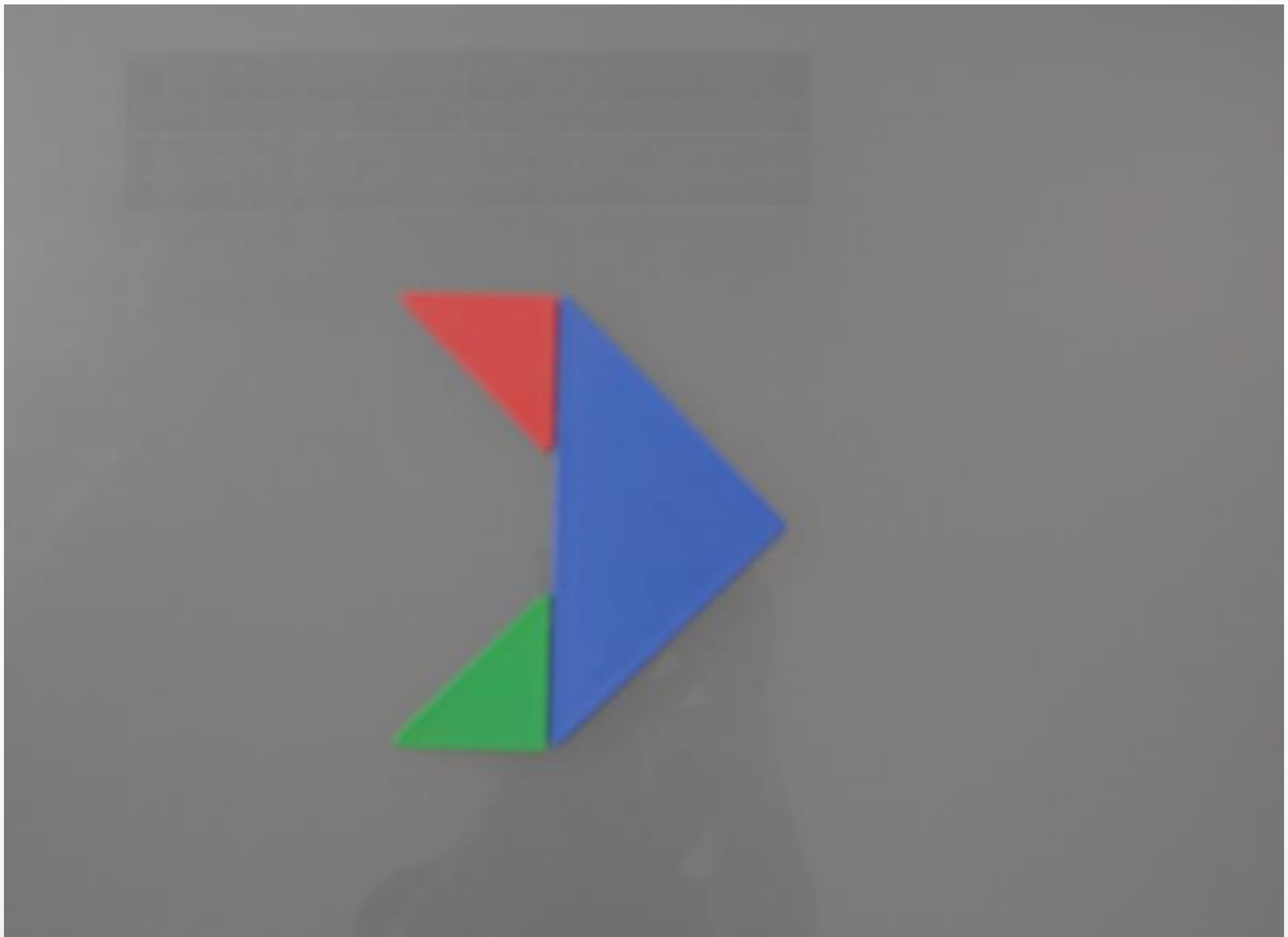
Challenges

- Noisy input image
- Real-time constrains
- Implementing into open-source environment
- Creating a complete system solution
- Shape combinations and movement



Challenges (cont.)

- Real-world view

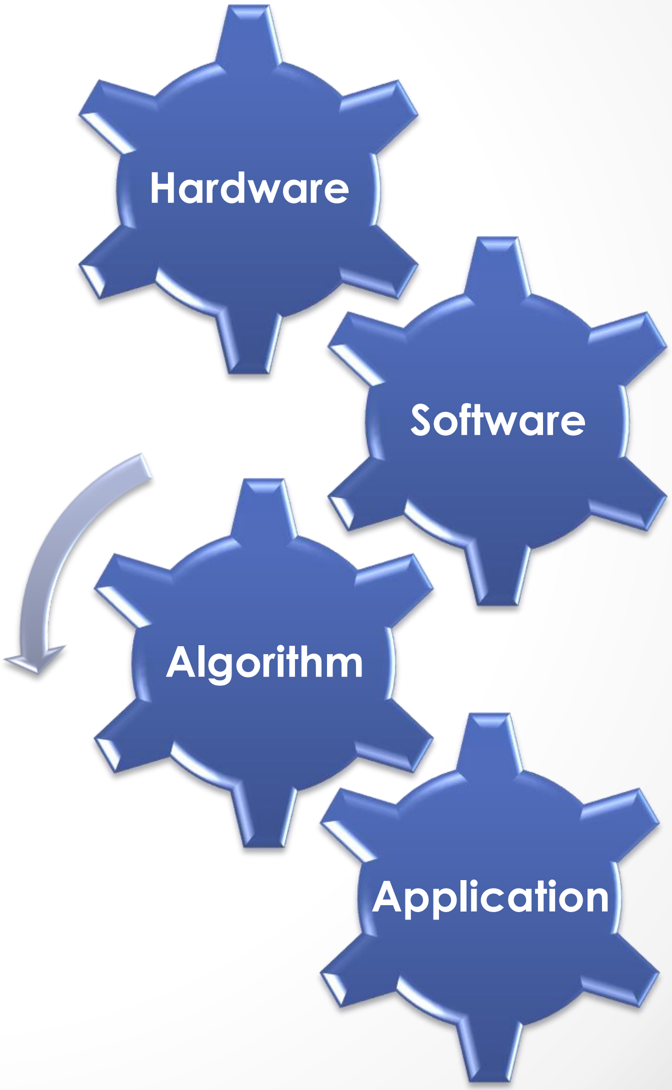


Outline

- Introduction
- System
- Algorithm
- Implementation
- Demo Game
- Conclusion

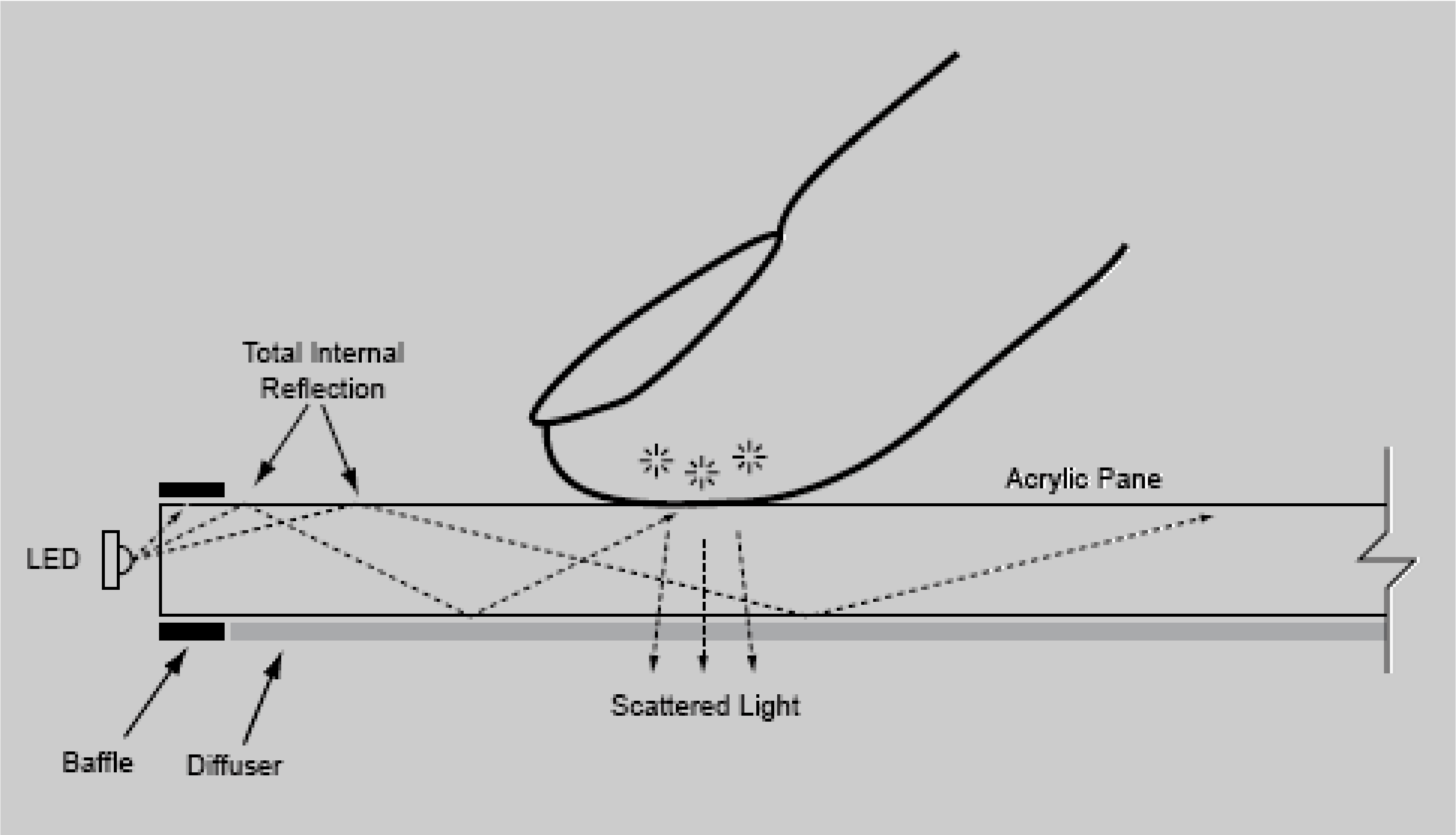
System

- Hardware
  - Multi-Touch Table
  - FTIR Technology
- Software
  - Community Core Vision
  - Client / Server Model
- Algorithm
  - Image Processing
  - Shape Detection
  - Tracking
- Application
  - Game

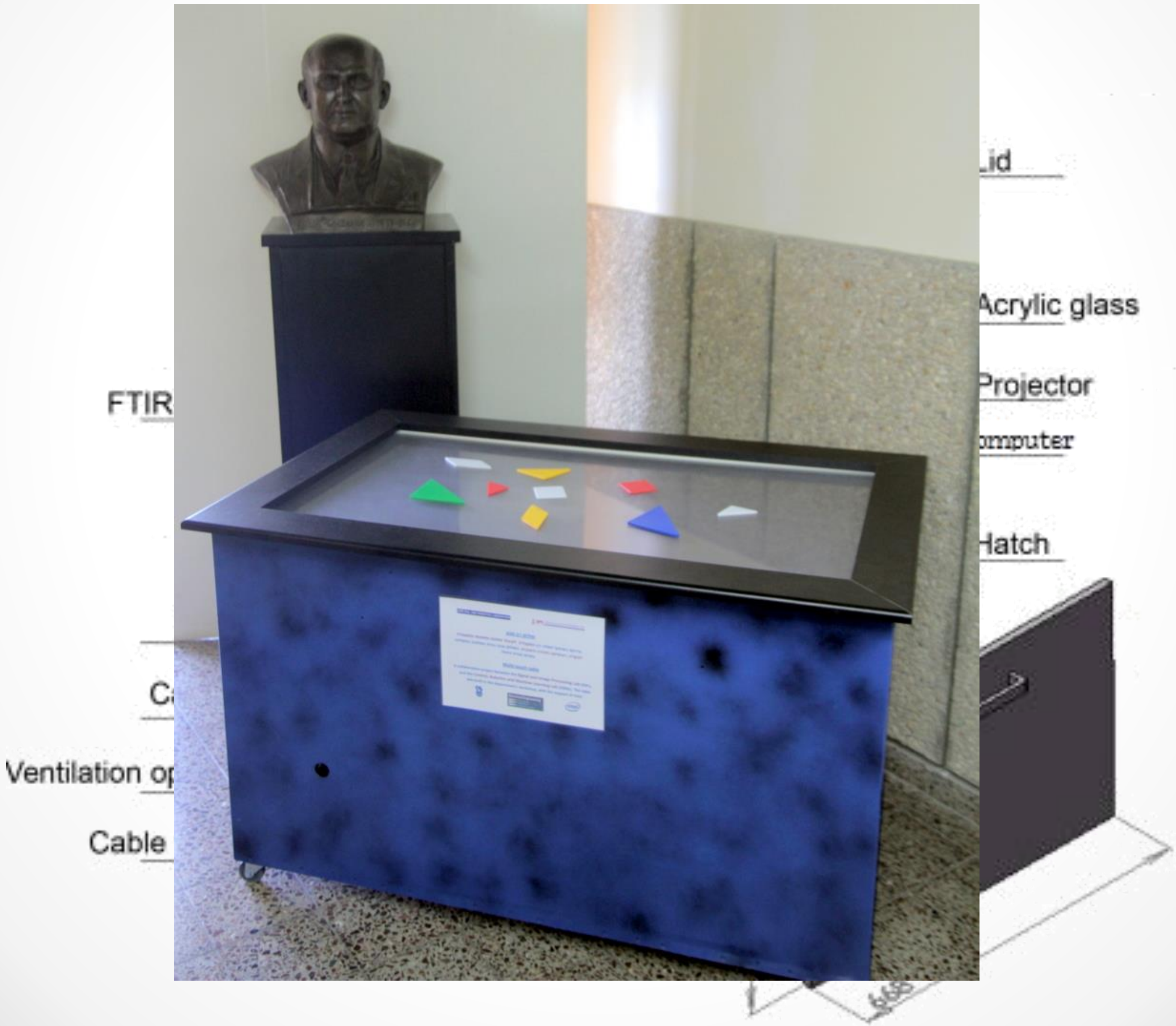


FTIR - Frustrated Total Internal Reflection

Frustrated Total Internal Reflection



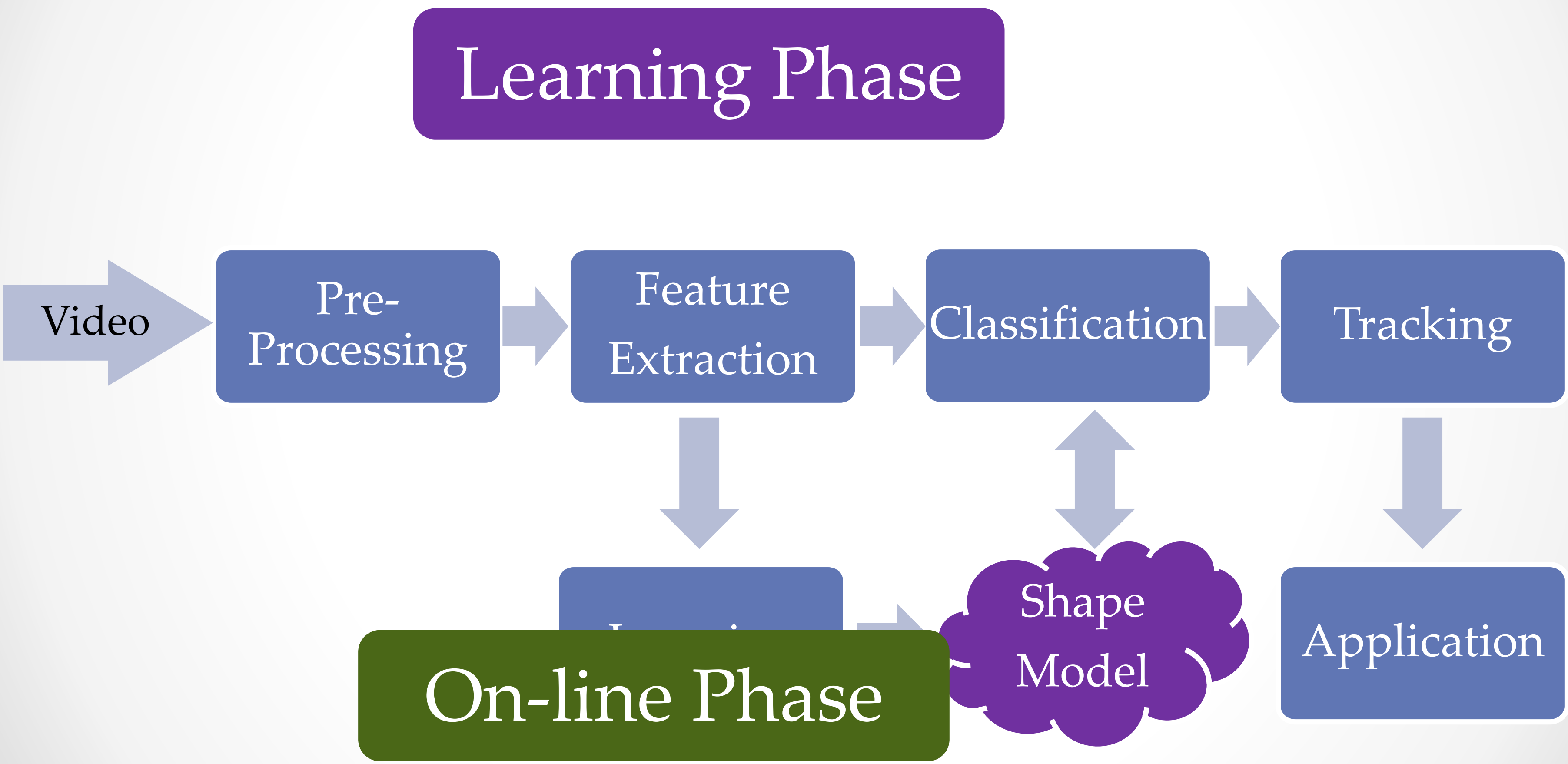
Multi-Touch Table



Outline

- Introduction
- System
- Algorithm
- Implementation
- Demo Game
- Conclusion

Shape Recognition



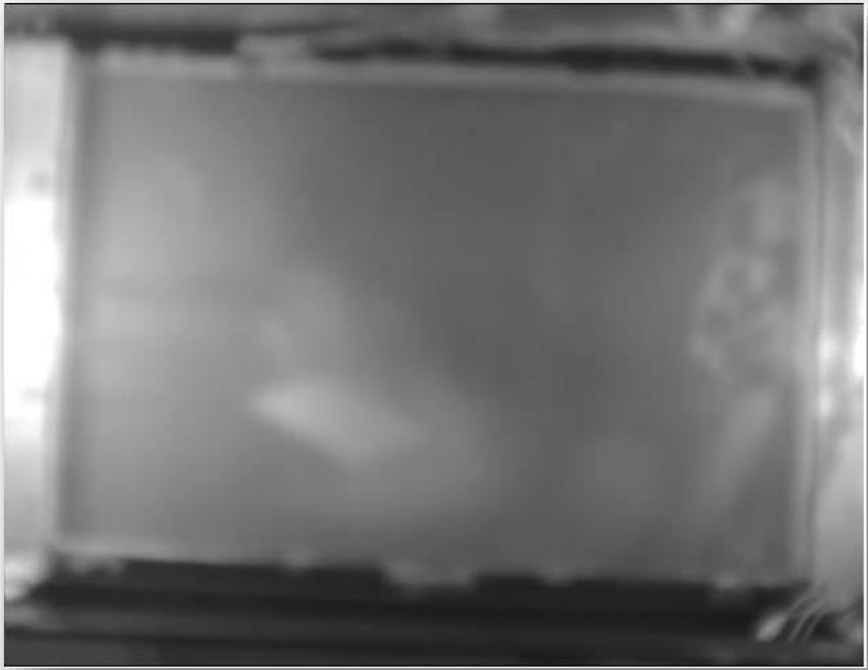
Pre-Processing

- Cropping
- Stitching
- Background removal
- Noise filtering
- Thresholding
- Segmentation

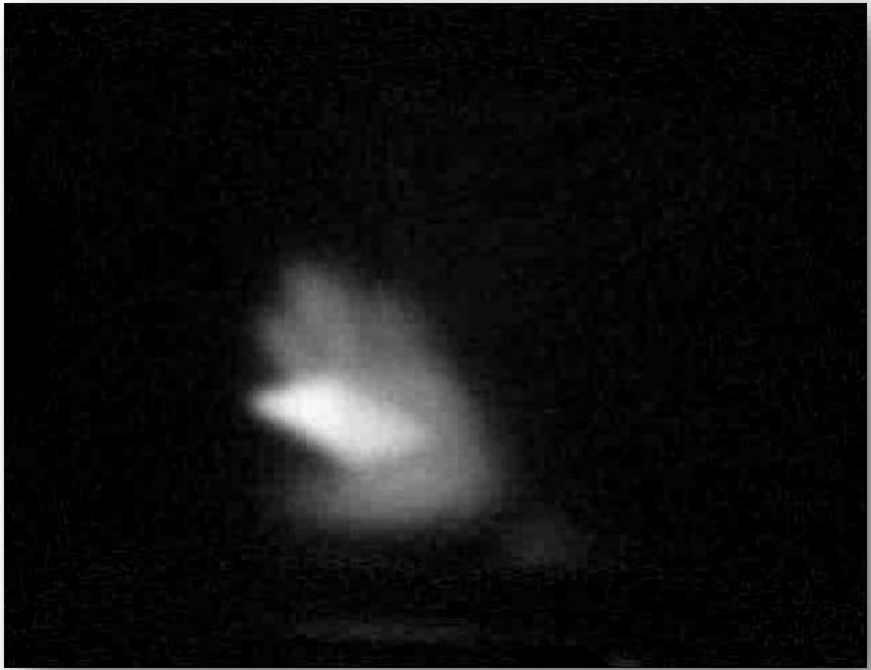
- Global thresholding leads to inconsistent performance
- Each frame has a different dynamic range
- Solution: Adaptive thresholding

Pre-processing (cont.)

IR input

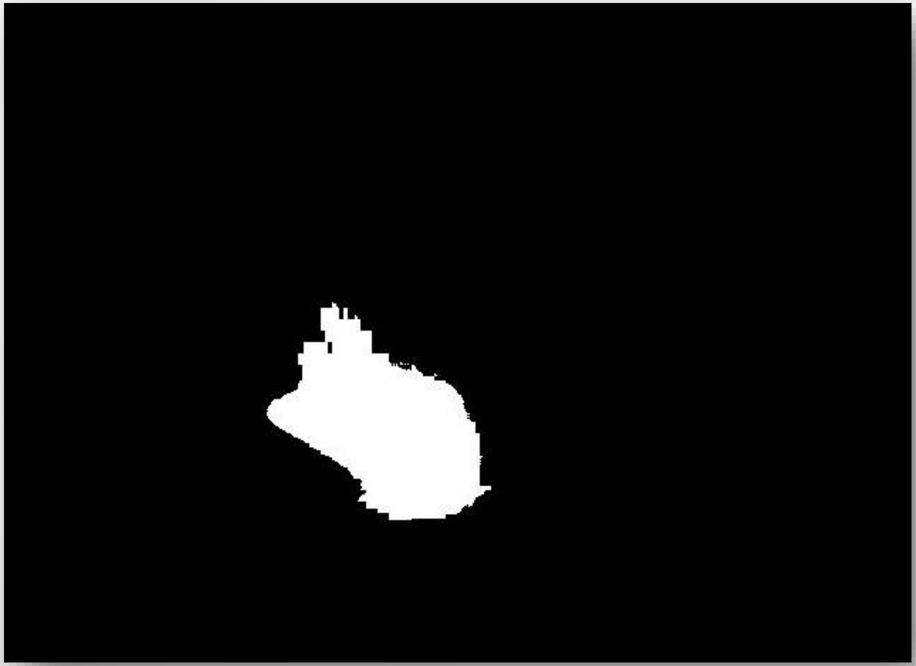


Background removed

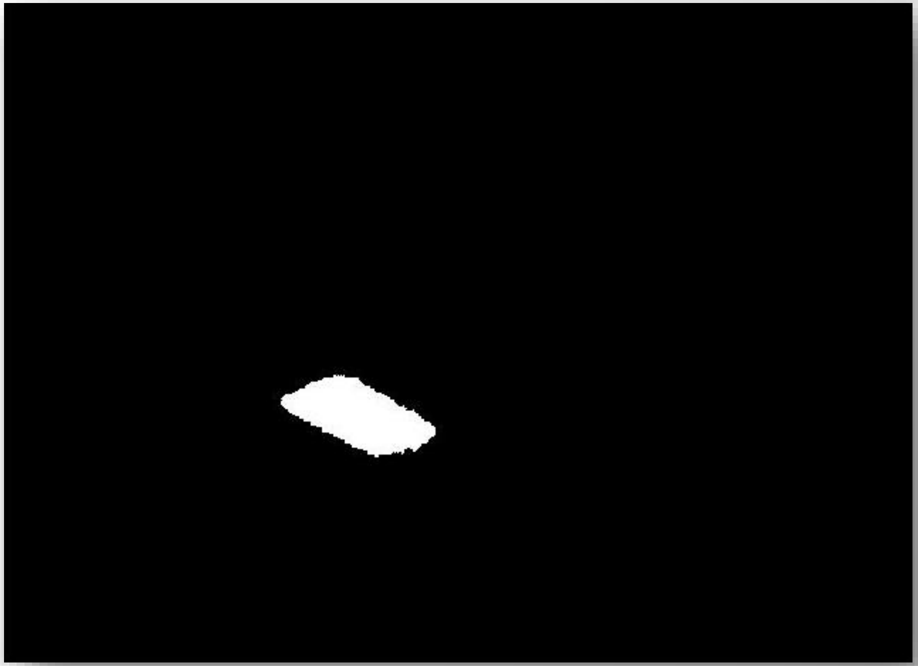


Pre-processing (cont.)

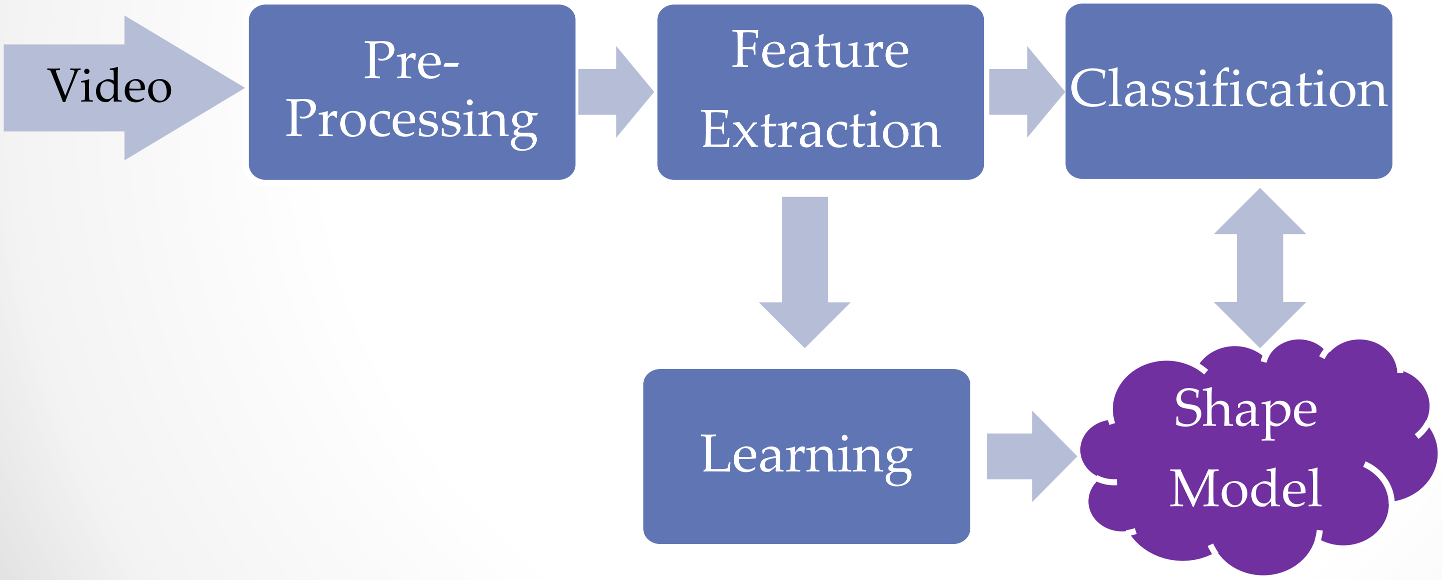
Global threshold



Adaptive threshold

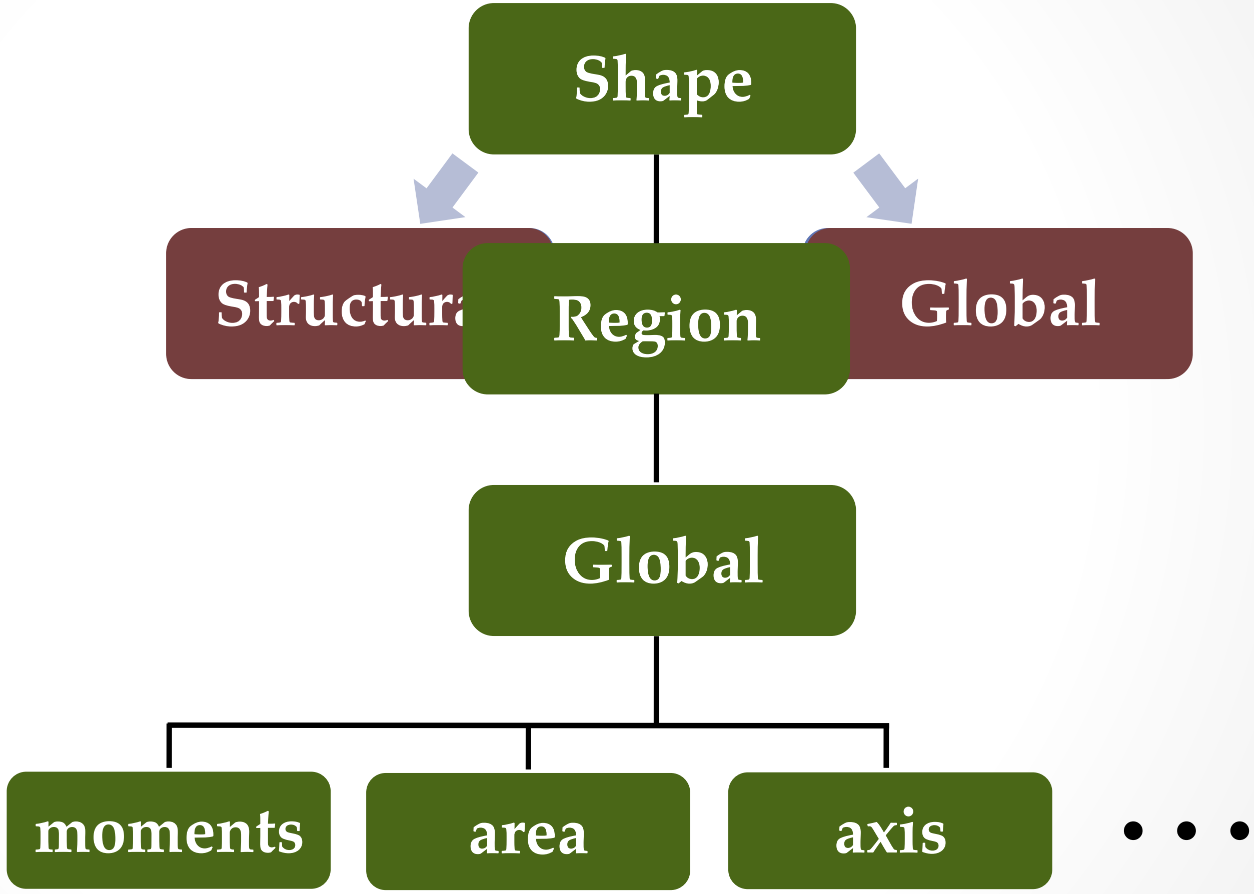


Shape Recognition



Feature Extraction

- Shape features



[Zhang & Lu, 2004]

Candidate Features

- Area
- Major & minor axis
- Orientation
- Eccentricity
- 6 Hu moments
- 7 complex moments

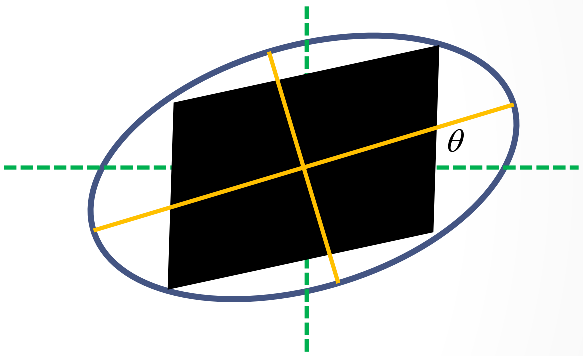


Image Moments

- Complex moments:

$$c_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x+iy)^p (x-iy)^q \chi(x,y)\, dxdy$$

$\chi(x,y)$ - image indicator function

- Moments are centralized for translation invariance

Rotation Invariance

- In polar coordinates:

$$c_{pq} = \int_0^{\infty}\int_0^{2\pi} r^{p+q+1} e^{i(p-q)\theta} \chi(r,\theta)\, dr d\theta$$

- Rotation by α $\quad c'_{pq} = e^{i(p-q)\alpha} c_{pq}$
- In general $\quad I = \prod_{j=1}^{N} c_{p_j q_j}^{k_j}$
- Which leads to $\quad \boxed{\sum_{j=0}^{n} k_j \left(p_j - q_j\right) = 0}$

Hu Moments

- Seven rotation and translation invariant moments
- Useful for visual pattern recognition
- Rotation invariance via complex moments

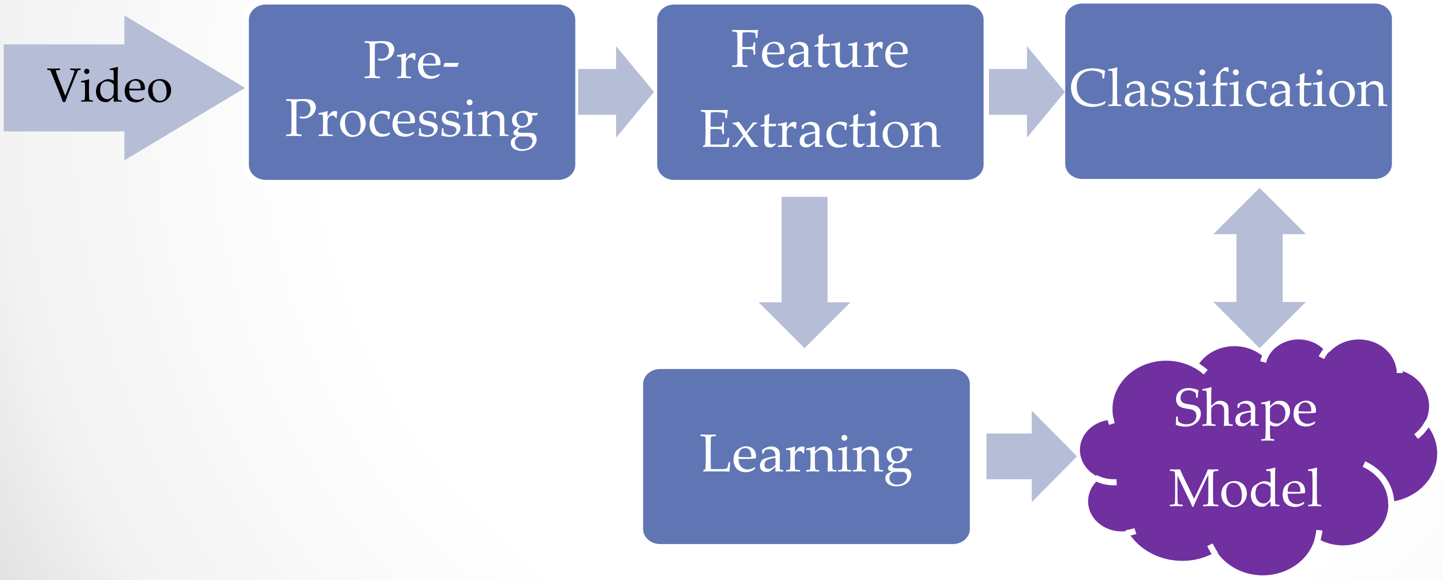
$$\varphi_1 = c_{11}$$

$$\varphi_2 = c_{20} \cdot c_{02}$$

$$\vdots$$

[Hu, 1962]
[Flusser, Suk & Zitová, 2009]

Shape Recognition



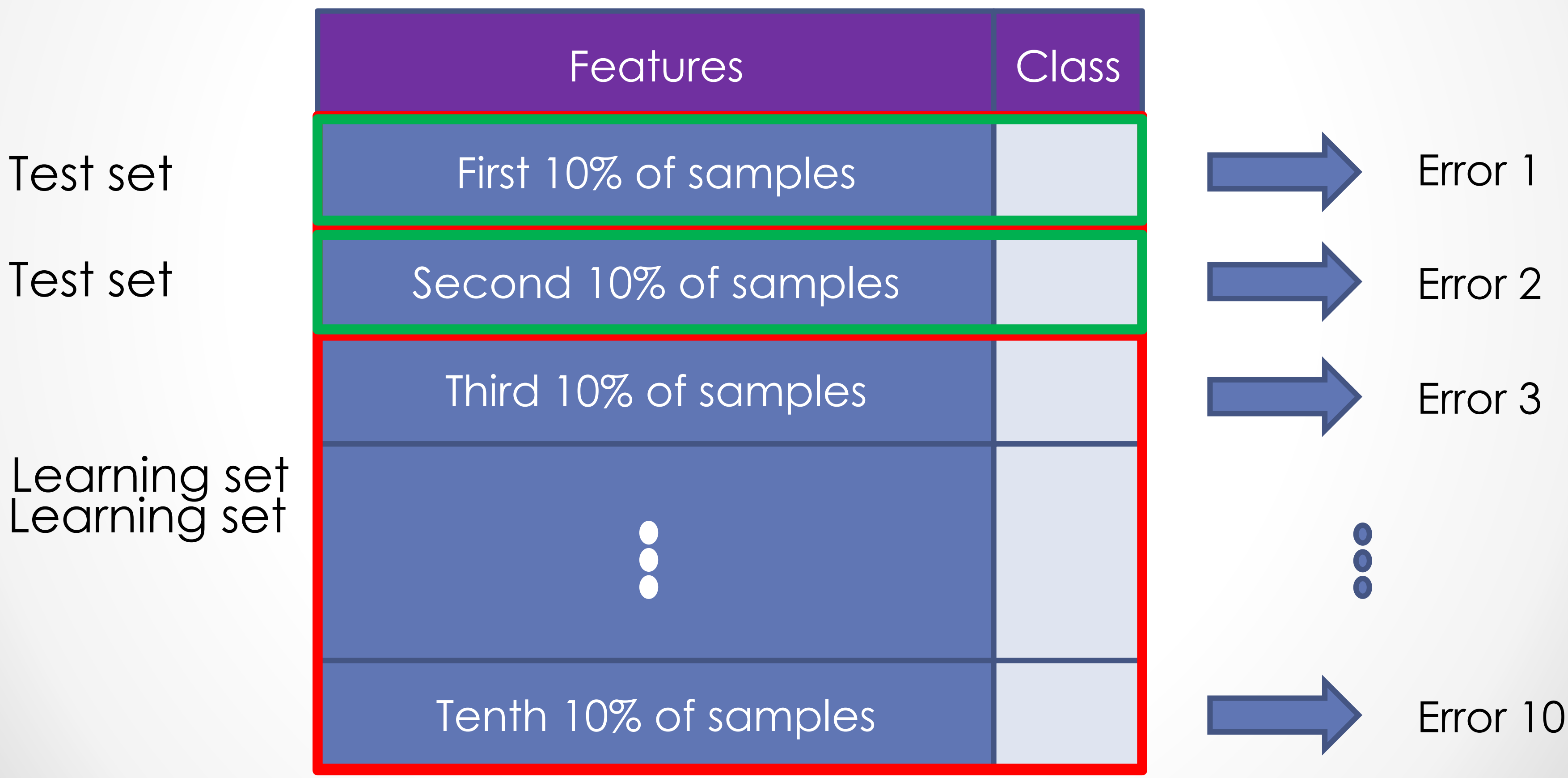
Learning

- Feature selection
- Shape model

- Many features $\Rightarrow$ high computation
- Dimensionality reduction required
- Extract features at various locations
- Approx. 50 samples per shape
- Independent Gaussian model

Feature Selection

- Exhaustive algorithm based on cross-validation



Result: $e_{total} = mean\left\{e_1, e_2, \ldots, e_{10}\right\}$

Feature Selection (cont.)

- Initialization
  - Set n=1 feature(s) out of a total of N features

- Step
  - Calculate e_{total}^{n} for all possible combinations $\binom{N}{n}$
  - Set n=n+1

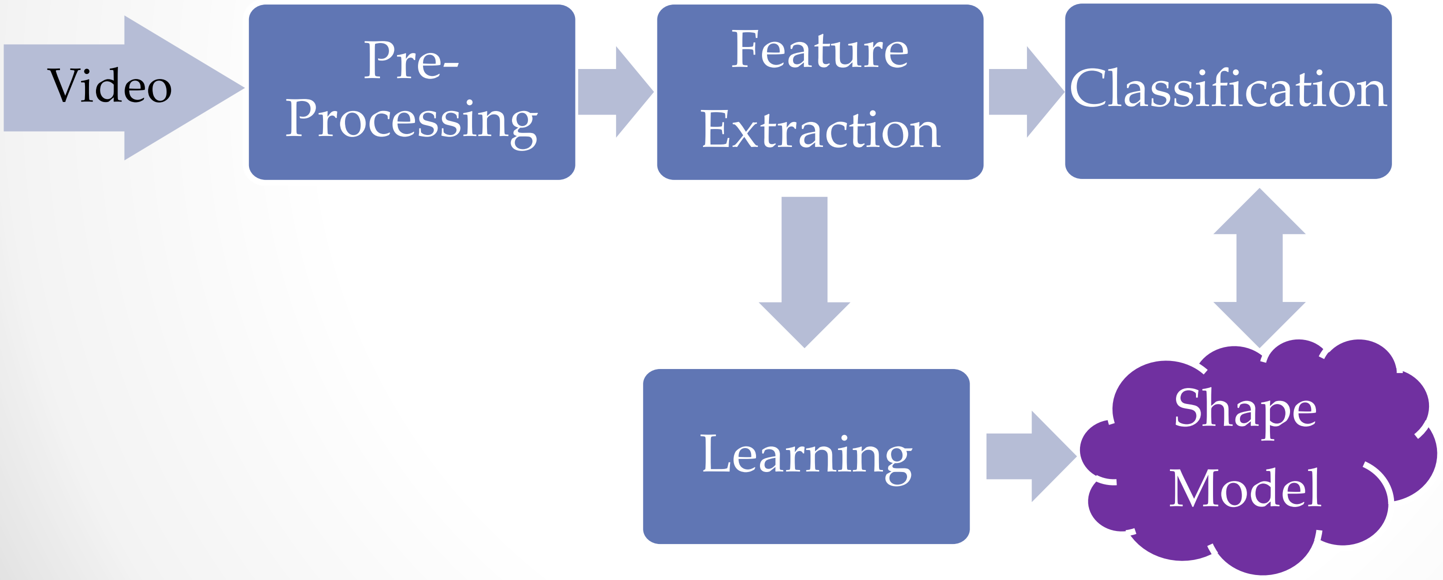
- Stopping condition
  - When $e_{total}^{n+1} > e_{total}^{n} - \varepsilon$

Feature Selection (cont.)

- Chosen features:
  - Area
  - Major & minor axis
  - Eccentricity
  - 4 Hu moments: $\varphi_1, \varphi_4, \varphi_6, \varphi_7$
- Empirical selection
- Similar to other shape recognition works

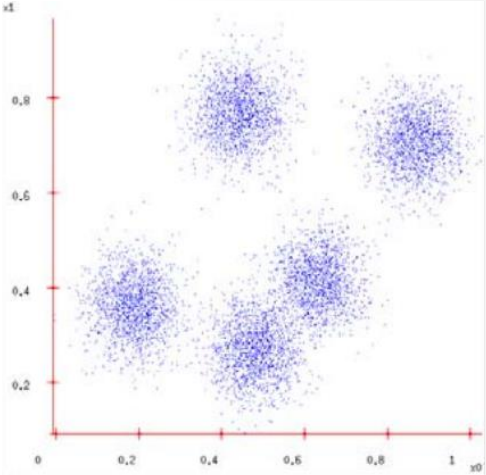
[Hu, 1962]
[Sarfraz, 1993]

Shape Recognition



Classification

- Independent Gaussian distributions
- Naïve Bayes Classifier
- For each shape s_j and feature f_i



assume: $p(f_i \mid s_j) \sim N\left(\mu_i, \sigma_i^2\right)$

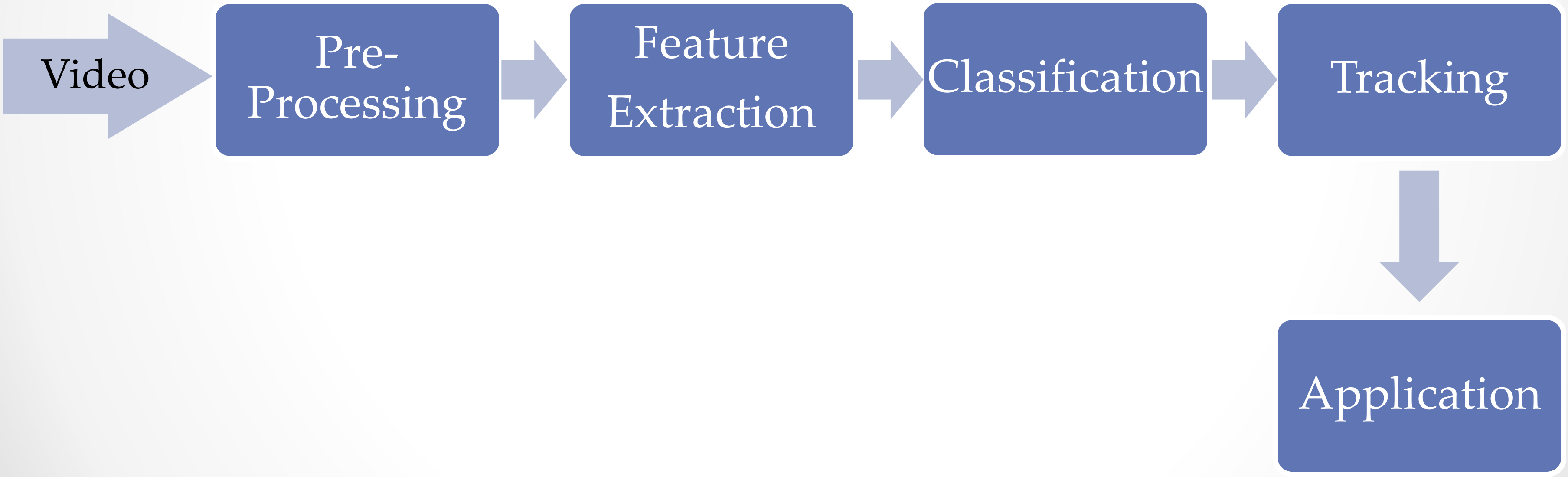
$$p(s_j \mid f_1, \ldots, f_m) \propto p(s_j) \cdot \prod_{i=1}^{m} p(f_i \mid s_j)$$

$$\hat{s} = \text{classify}(f_1, \ldots, f_m) = \underset{s_j}{\operatorname{argmax}}\ p(s_j \mid f_1, \ldots, f_m)$$

Naïve Bayes Classifier

- Advantages:
  - Generic
  - Fast
  - Allows thresholding
  - Good performance
- Disadvantages:
  - Long offline learning process

Shape Recognition



Tracking

- ➢NN tracking
- ➢Combined shapes
- ➢Internal movement
- ➢False positive detection

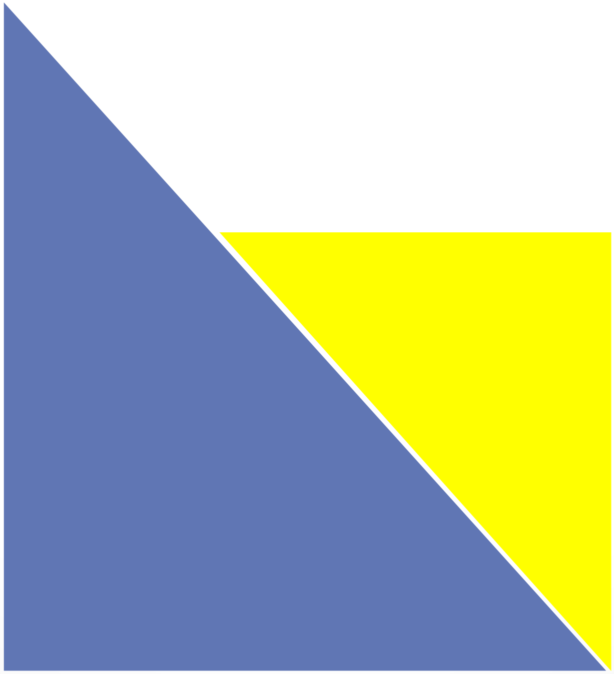
- Nearest neighbors tracking
- Simplifies data handling
- Prevents false detections
- Allows complex situations
  - Shape combinations
  - Internal movement in a combination

Complex Situations

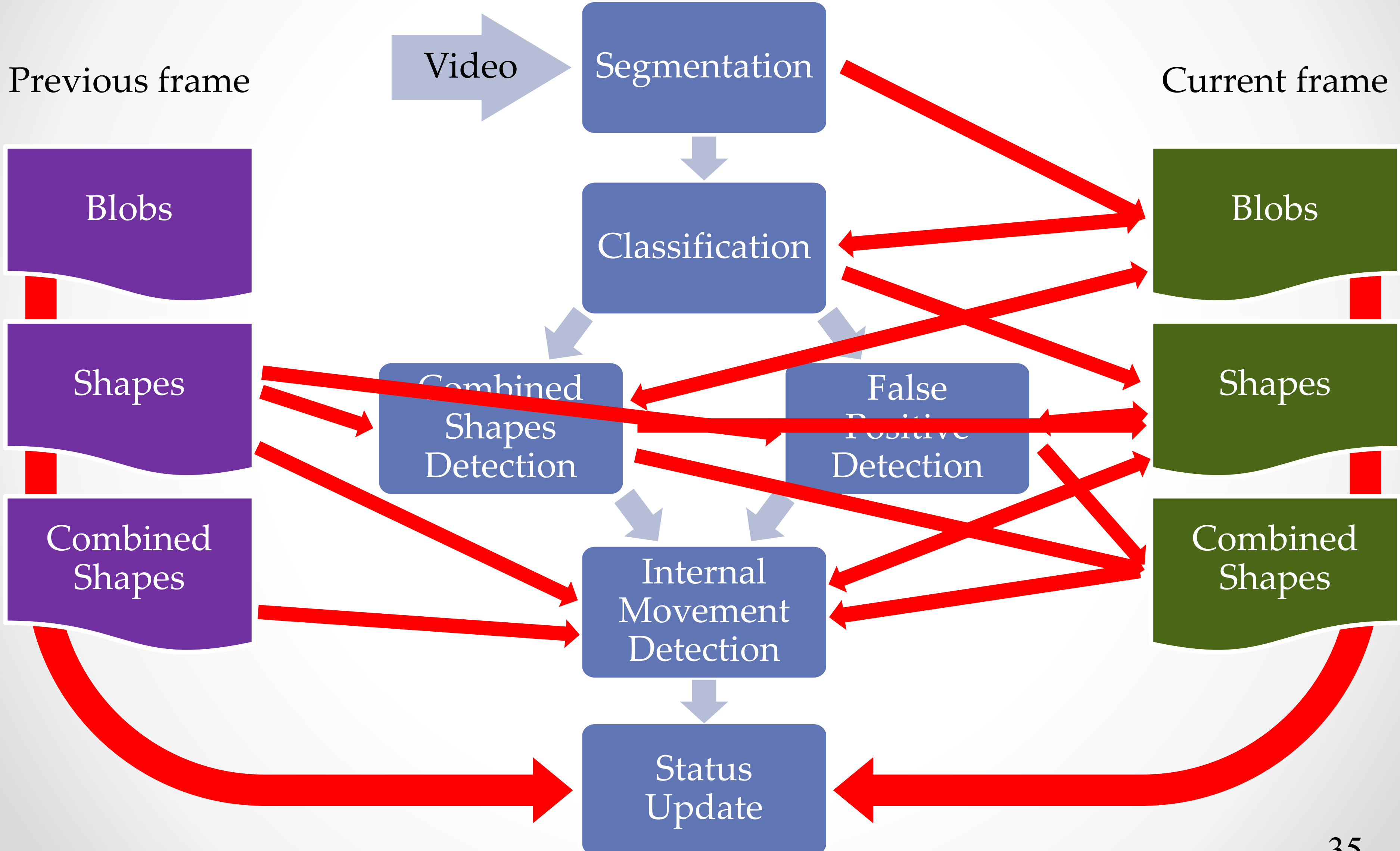
- Shape combinations can appear as one (legitimate) shape
- Possible classification mistakes:

Internal Shape Movement

- Complex shape modification
- Combined shape structure is maintained

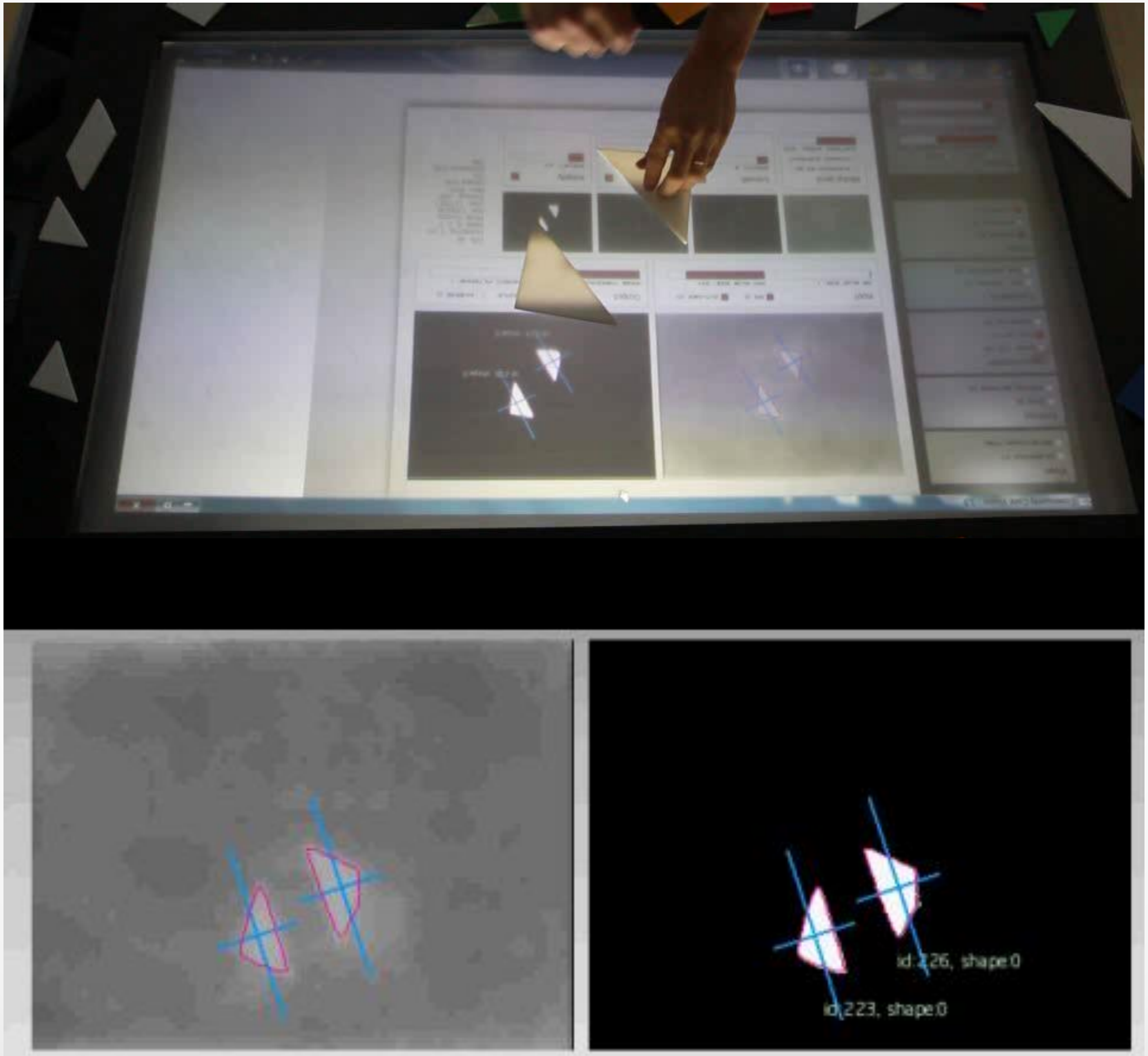


Tracking

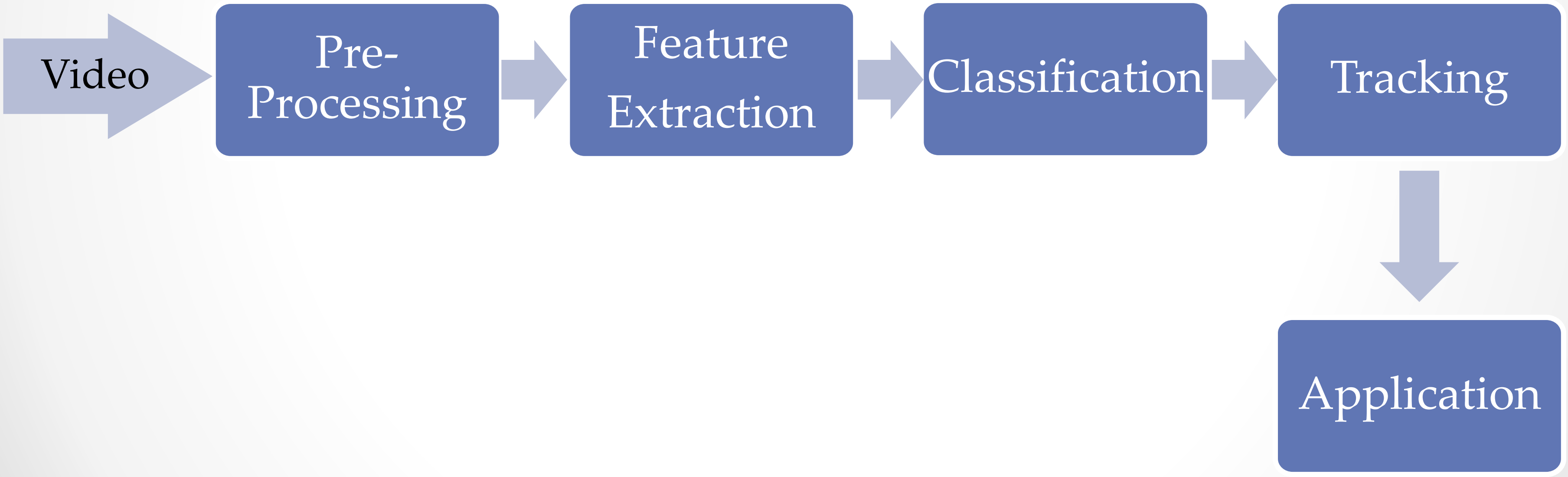


Previous frame
Video
Segmentation
Current frame
Blobs
Classification
Blobs
Shapes
Combined Shapes Detection
False
Detection
Shapes
Combined Shapes
Internal Movement Detection
Combined Shapes
Status Update

Complex Situations



Shape Recognition



Application

- Client / Server
- Data transfer protocol

- Many possible applications
- Client / Server model established
- Community Core Vision
- Tangible User Interface Objects (TUIO) protocol for data transfer

Community Core Vision

- Open-source / cross-platform solution for computer vision and machine sensing
- Supports multi-touch input
- Supports multi-camera feed
- Commonly used by developers

TUIO Protocol

- Open framework communication protocol and API for tangible multi-touch surfaces

- Supports various multi-touch information
  - Location
  - Orientation
  - Acceleration
  - …

- Commonly used with Community Core Vision

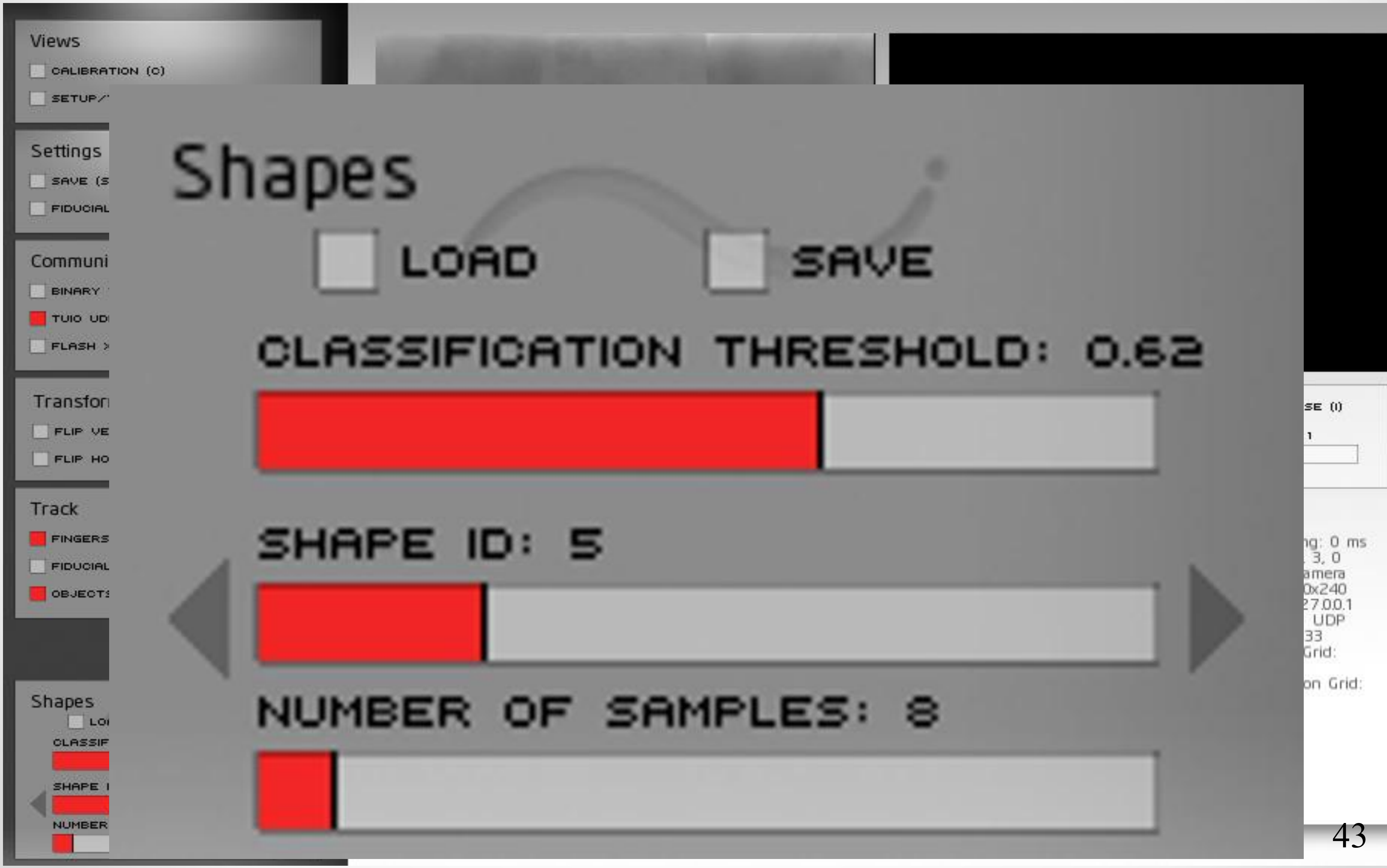
Outline

- Introduction
- System
- Algorithm
- **Implementation**
- Demo Game
- Conclusion

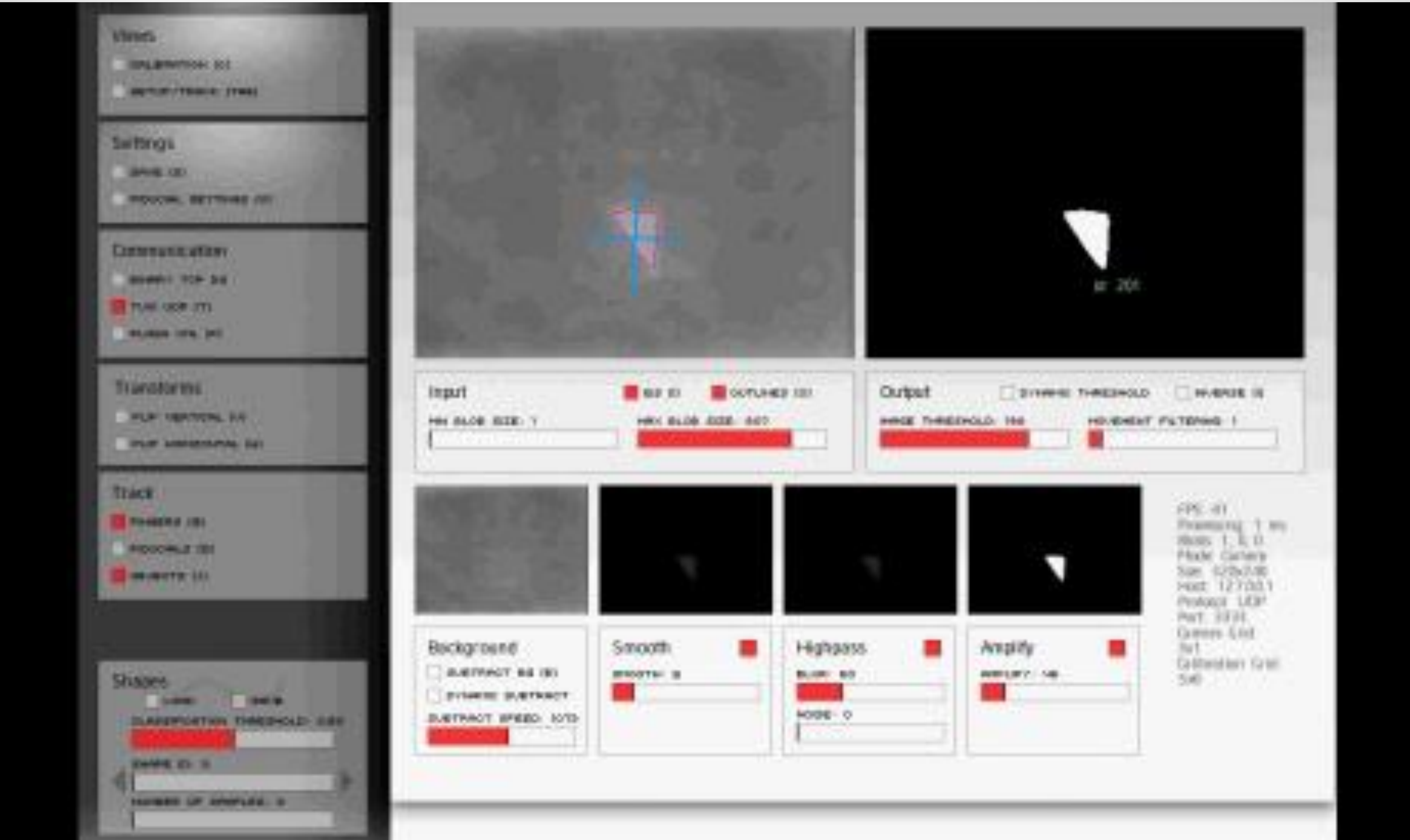
Implementation

- Simulations (Matlab)
- Implemented into CCV (C++)
  - Adaptive thresholding
  - Learning process
  - Classification
  - Shape tracking
  - Import / export classifiers
  - Additions to GUI
- Implemented into TUIO (C++ & C#)
  - Sending shape information
  - Distinguishing between touch and shape

Graphical User Interface



Learning

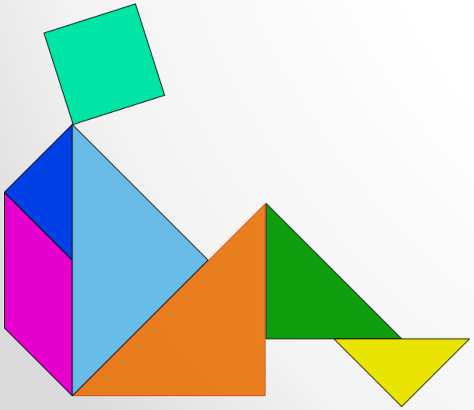

Views
Settings
Communication
Transforms
Track
Shapes
Input
Output
Background
Smooth
Highpass
Amplify

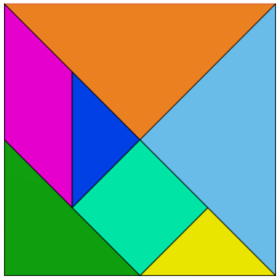
Outline

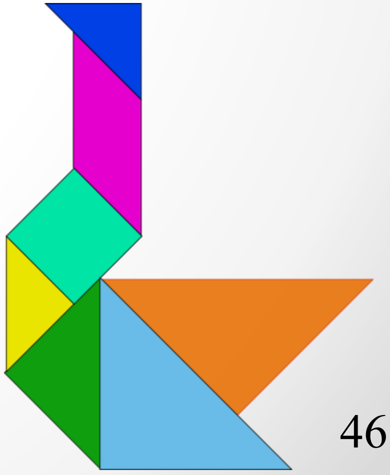
- Introduction
- System
- Algorithm
- Implementation
- **Demo Game**
- Conclusion

Tangram

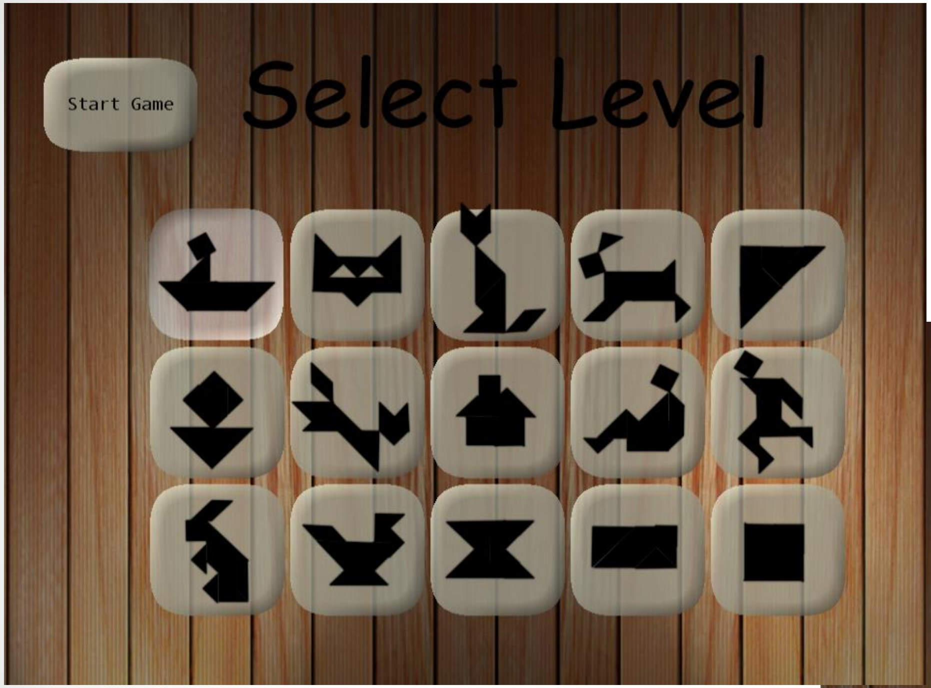
- Chinese dissection puzzle with seven shapes
- The objective is to fill a given silhouette
- Demonstrates the algorithm capabilities

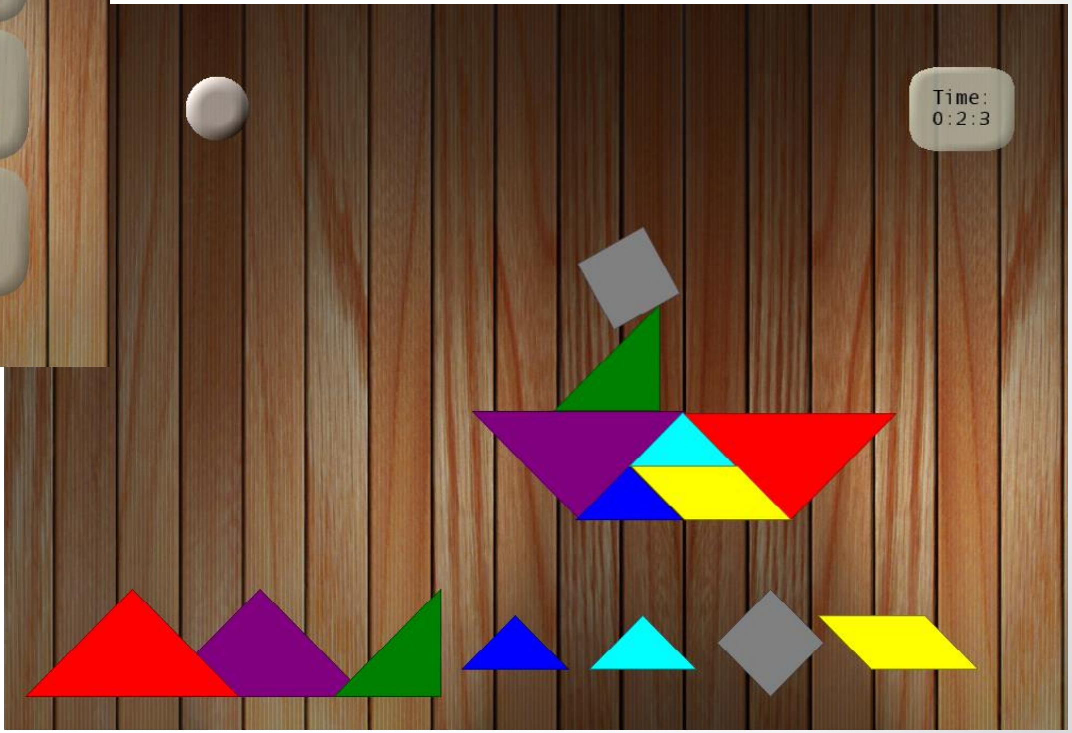






SIPL Tangram





Outline

- Introduction
- System
- Algorithm
- Implementation
- Demo Game
- Conclusion

Demo

Performance Analysis

- Resources
  - Intel i7 processor, 8 GB of memory
  - Three IR cameras, 60 fps

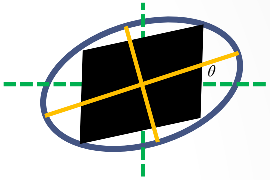
- Real-Time performance

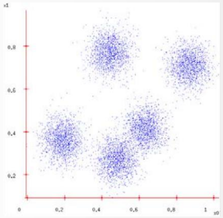
- Computing time < 2 miliseconds per frame

Summary

Multi-Touch table

Feature Extraction





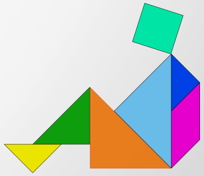
Shape Classification

Tracking

Shapes

Complex situations

CCV & TUIO



Tangram

Future Work

- Extending shape recognition
  - Additional shapes
  - Shape color
- Improving camera registration
- Cross-Platform implementations
- Finalize implementation of the demo game

Thanks

Kobi

Yair

Avi

References

- D. Zhang and G. Lu, "Review of shape representation and description techniques“, Pattern Recognition, vol. 37, pp. 1–19, 2004.
- M. K. Hu, "Visual Pattern Recognition by Moment Invariants“, IRE Trans. Information Theory, vol. 8, pp. 179–187, 1962.
- J. Flusser, T. Suk and B. Zitová, “Moments and Moment Invariants”, John Wiley & Sons Ltd., 2009.
- C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2007.
- "CCV, Community Core Vision“, [Online], http://ccv.nuigroup.com/.
- "TUIO.org“, [Online], http://www.tuio.org/