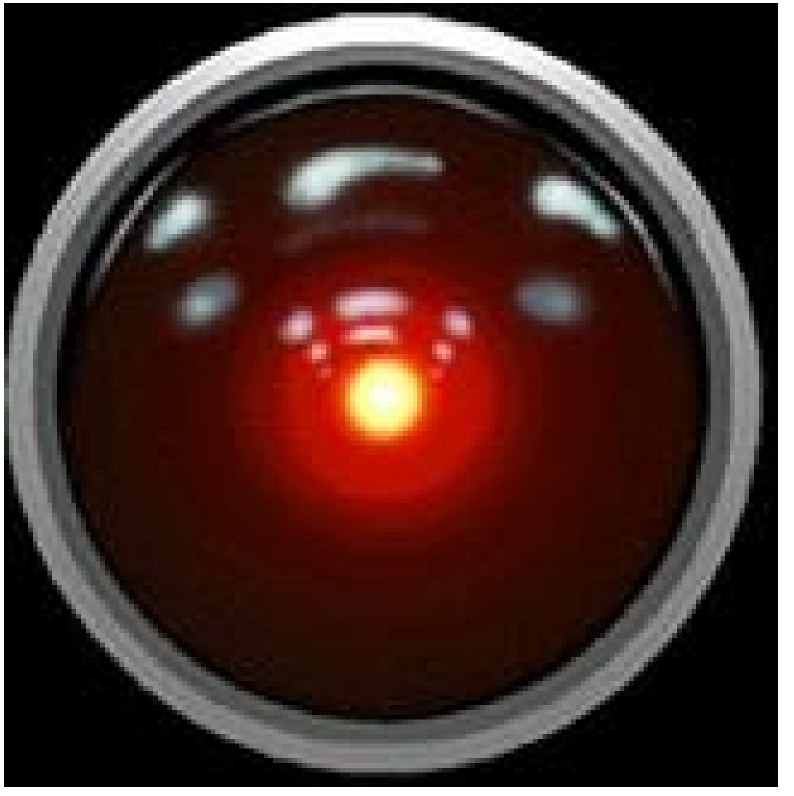


CS 378: Autonomous Intelligent Robotics

Instructor: Jivko Sinapov

http://www.cs.utexas.edu/~jsinapov/teaching/cs378/

Computer Vision: Motion

Announcements

- Homework 5 due today
- Homework 6 out today (due 4/5)

Announcements

Volunteers needed for another study!

As before, there will be extra credit

To sign up, email:

– Rodolfo Rodriguez <rcorona@utexas.edu>

– Jesse Thomason <thomason.jesse@gmail.com>

Final Project Timeline

- Project Proposal due: ~~Mar. 29th~~ Apr. 1st
- Project Presentations / Demos: Last Week of Class (May 3rd and 5th)
- Final Report due: May 11th

Project Proposal Guidelines

- Work in groups of 2-3 (it's OK to work on your own if you really want to)
- Preferably, team up with people with different skills than yours
- Purpose of the proposal is to give you an outline / roadmap

Project Proposal Guidelines

- Each proposal should be about 2-3 pages

- Each proposal should include:
  - What is the application / task / problem?
  - Any previous experience you may have in that area
  - What do you expect to achieve by the end of the semester?
  - How do you plan to evaluate whether it works or not?
  - A timeline / schedule of progress and milestones

Project Proposal Guidelines

- Organization: your proposal should have sections and headings (don't just submit one long essay)
- For example:
  - Introduction / problem formulation
  - Proposed approach / software
  - Proposed evaluation
  - Summary of anticipated end result

Project Ideas

Help the robot “see” something it currently cannot

Help the robot “hear” something (e.g., the elevator sound)

Help the robot “do” something (e.g., follow a person)

Final Project Timeline

The most important thing is to start early, and discuss your ideas with the TA, mentors and myself. We'll point you to a starting point, describe functionality that already exists, and help refine your ideas.

The rest of the semester...

- 3D Vision – processing point-cloud data from the kinect
- Multiple ways to control the robot:
  - Low-level velocity commands
  - **2D navigation goals**
  - High-level logical goals (e.g., “go to room 3.414”)

Sending 2D Navigation Goals

- During the training session, you sent 2D navigation goals to the robot using the rviz GUI

- Now, let's see how we can do the same from code

Installing our code base

- Github page:
  - https://github.com/utexas-bwi/bwi

2D Goal Example

Computer Vision: Motion

What is this?

What is this?

A. F. Bobick and J.W. Davis

``An appearance-based representation of action''.

In Proceedings of IEEE International Conference on Pattern Recognition 1996,

August 1996, pp. 307-312.

A. Davis, J. and A. Bobick

``The Representation and Recognition of Action Using Temporal Templates'',

In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,

June 1997, pp. 928-934.

What action is being performed?

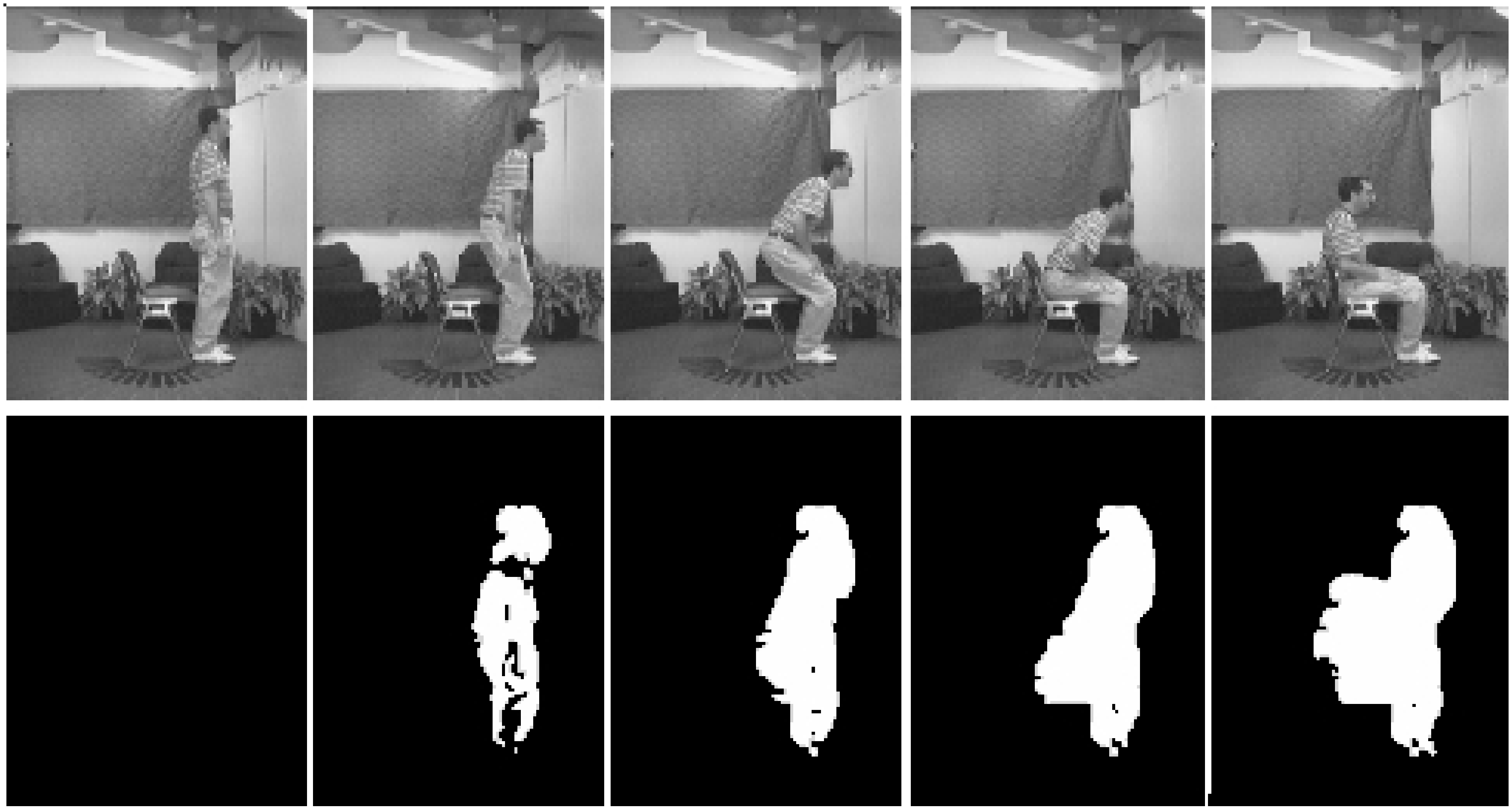

Frame 10


20


Frame 30

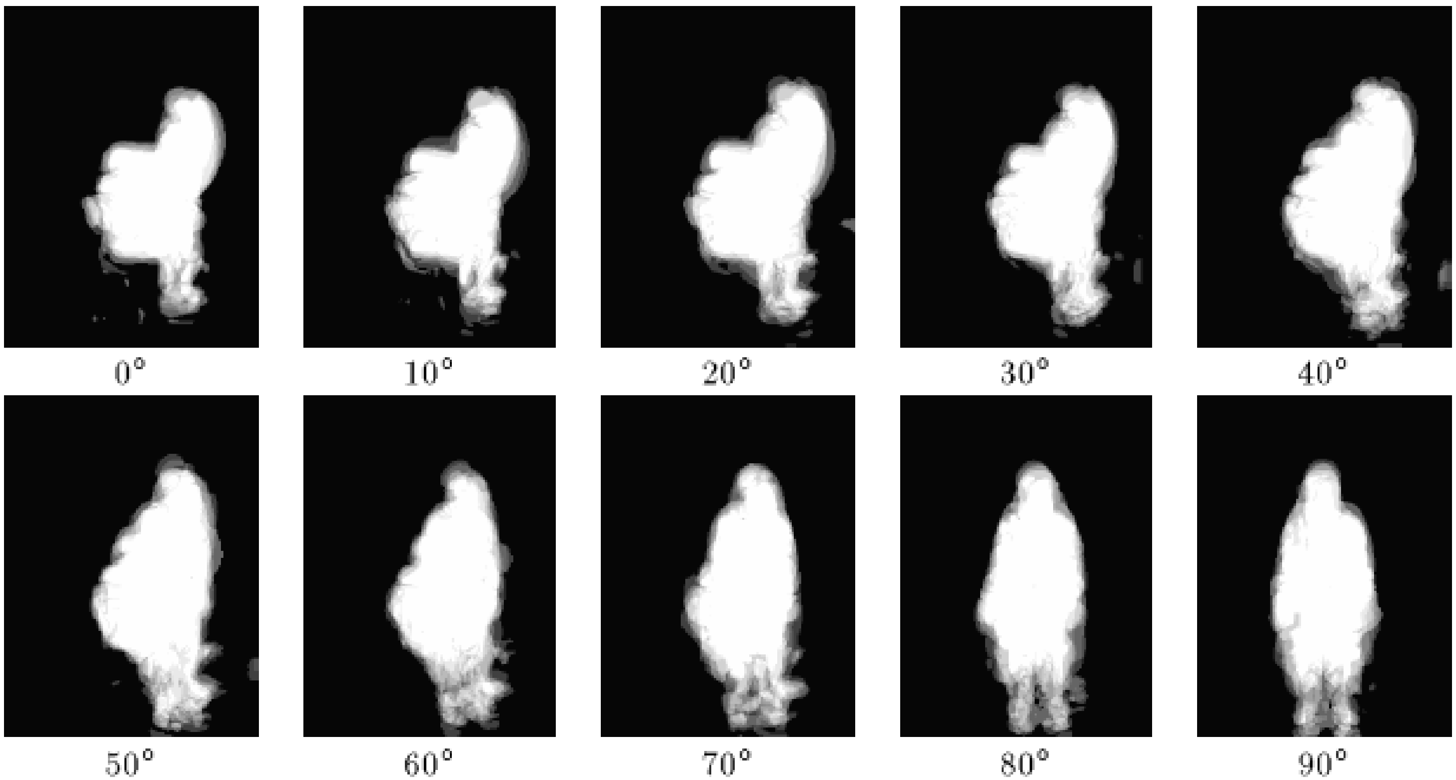

40

Motion Energy Image (MEI)

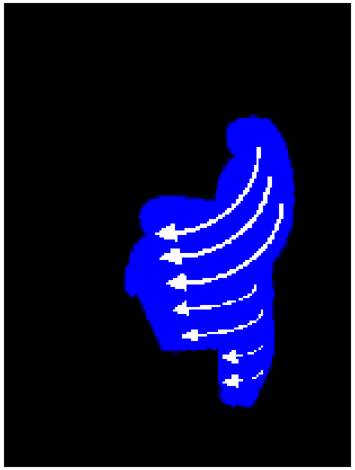


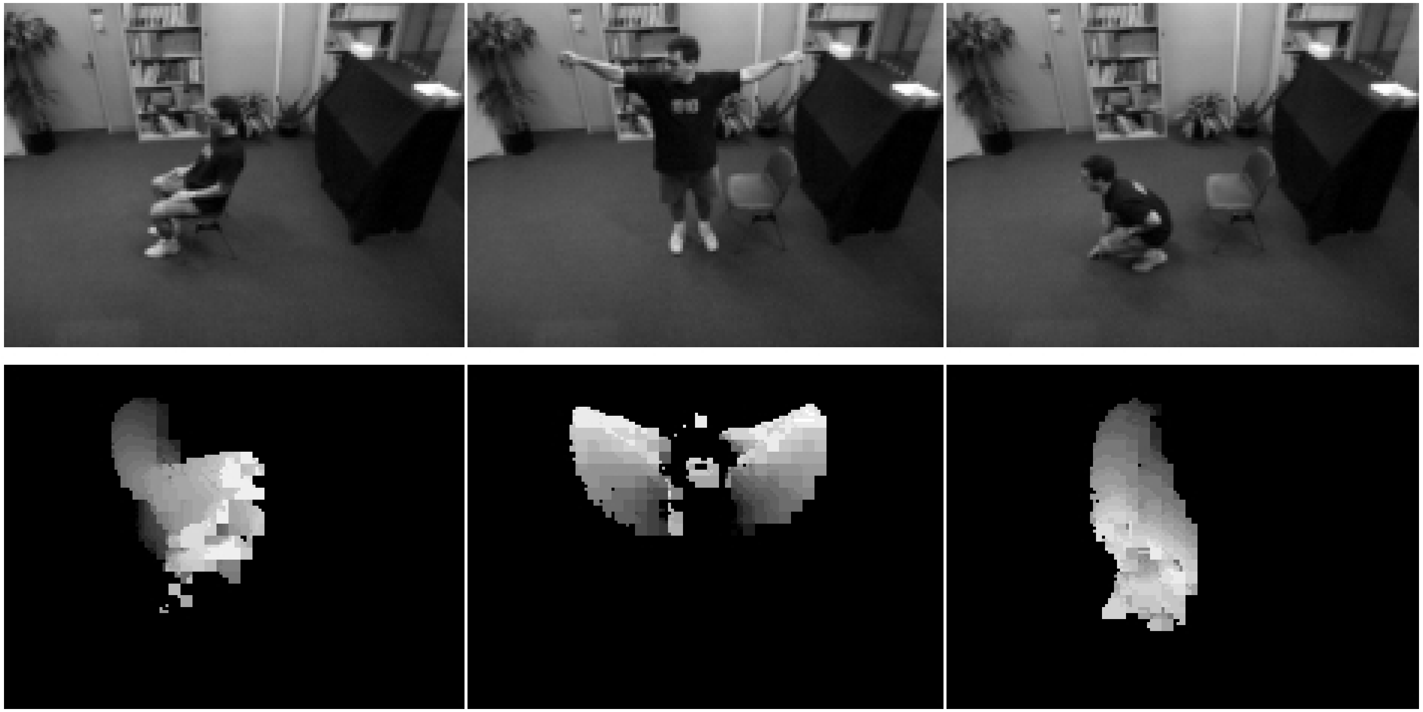
[http://www.cse.ohio-state.edu/~jwdavis/CVL/Research/MHI/mhi.html]

Average MEI for various viewing angles



Motion History Image (MHI)





[http://www.cse.ohio-state.edu/~jwdavis/CVL/Research/MHI/mhi.html]

Definitions

- Image Sequence $I(x, y, t)$
- Binary Images $D(x, y, t)$
  indicating regions of motion
- Binary Motion Energy Image $E_\tau(x, y, t)$

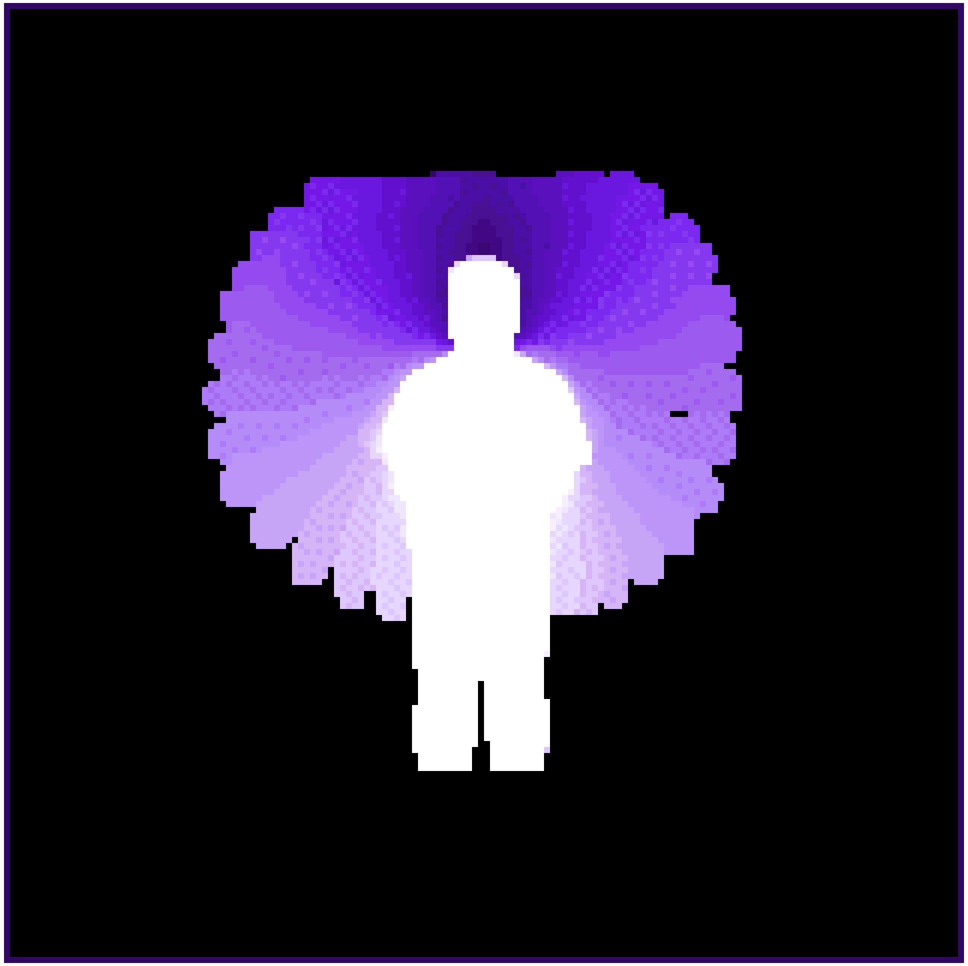
Motion Energy

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

Motion History

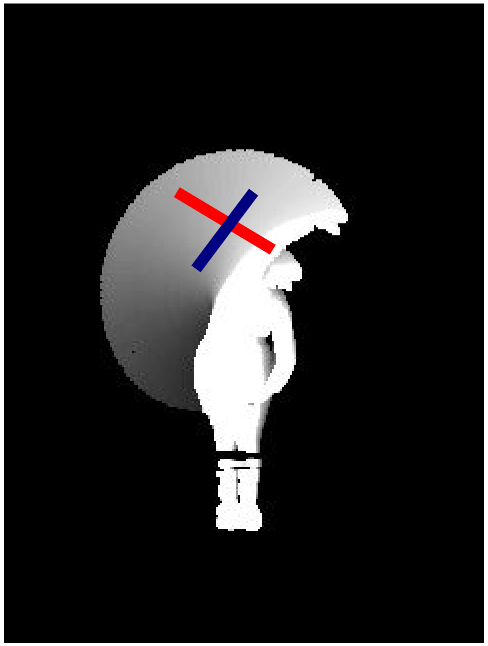
$$H_\tau(x,y,t) = \begin{cases} \tau & \text{if } D(x,y,t) = 1 \\ \max\ (0, H_\tau(x,y,t-1) - 1) \\ \quad \text{otherwise} \end{cases}$$

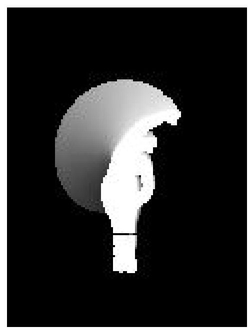
The result: more recently moving pixels appear brighter

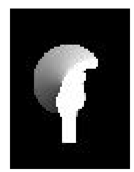


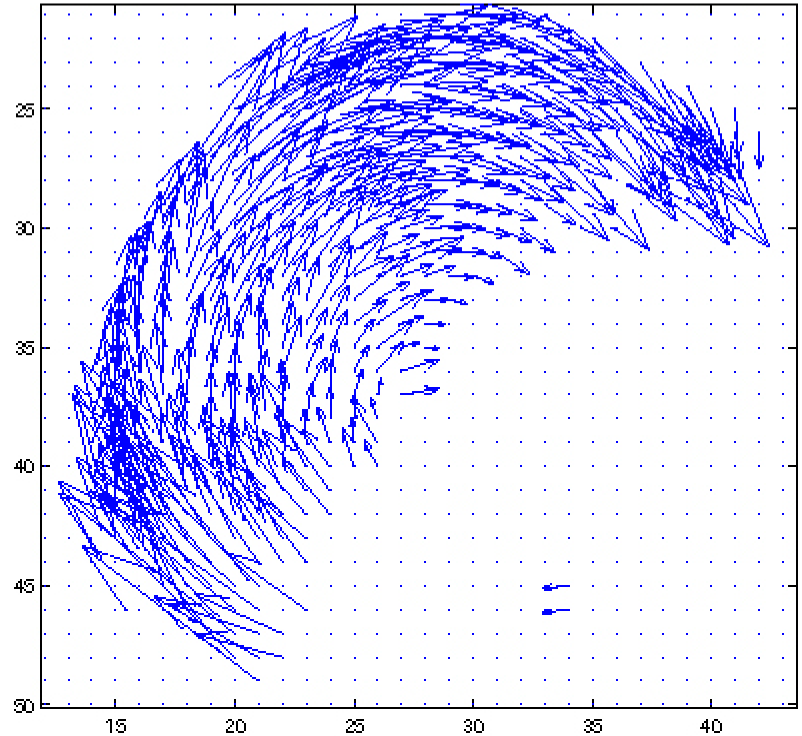
[http://www.cse.ohio-state.edu/~jwdavis/CVL/Research/MHI/mhi.html]

MHI pyramid



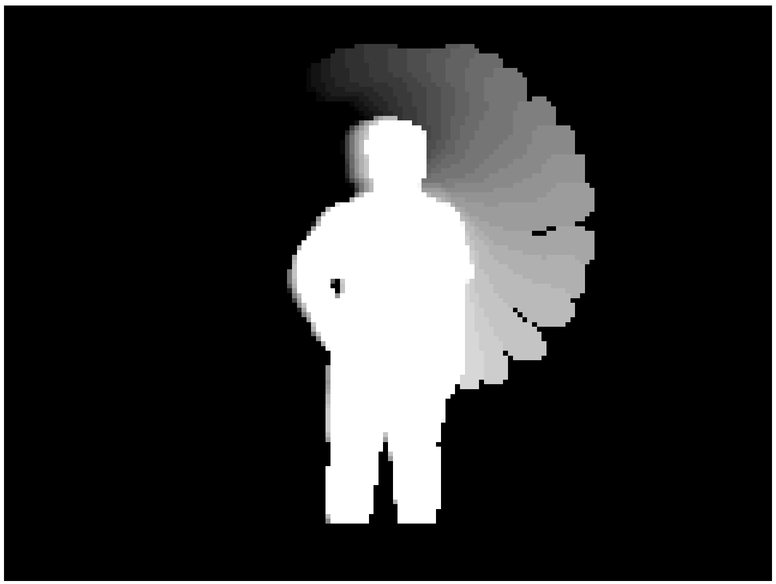


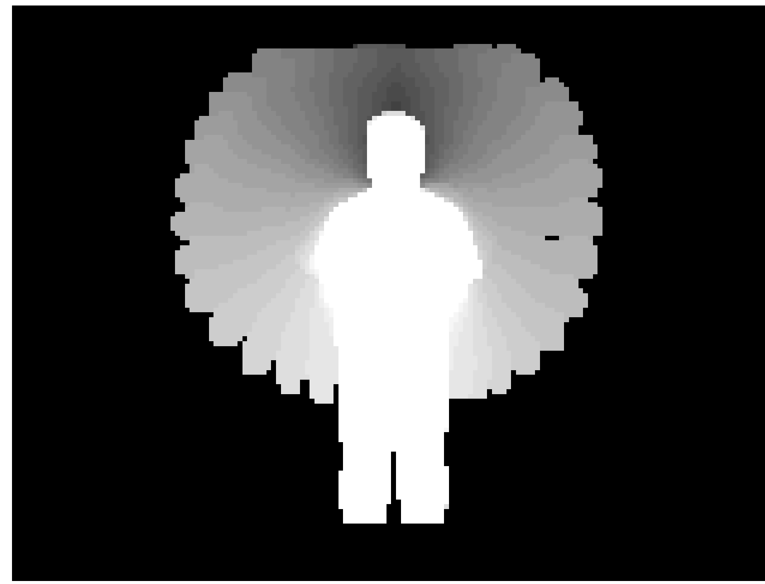




[http://www.cse.ohio-state.edu/~jwdavis/CVL/Research/MHI/mhi.html]

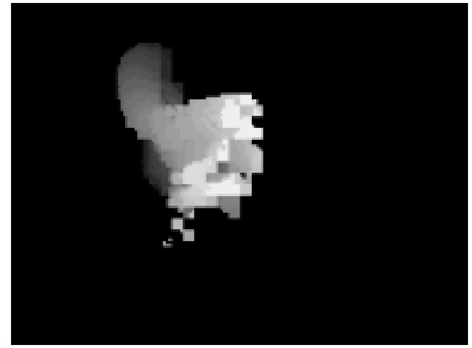
Motion templates for finishing LEFT-ARM-RAISE and FAN-UP-ARMS.





[http://www.cse.ohio-state.edu/~jwdavis/CVL/Research/VirtualAerobics/aerobics.html]

sit-down

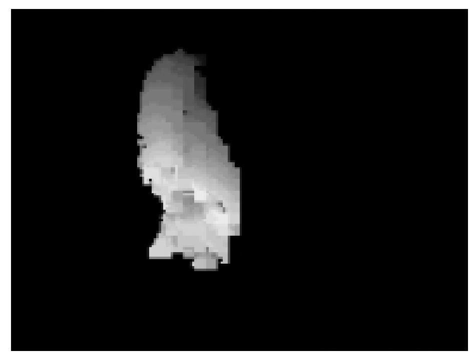


sit-down MHI

arms-wave

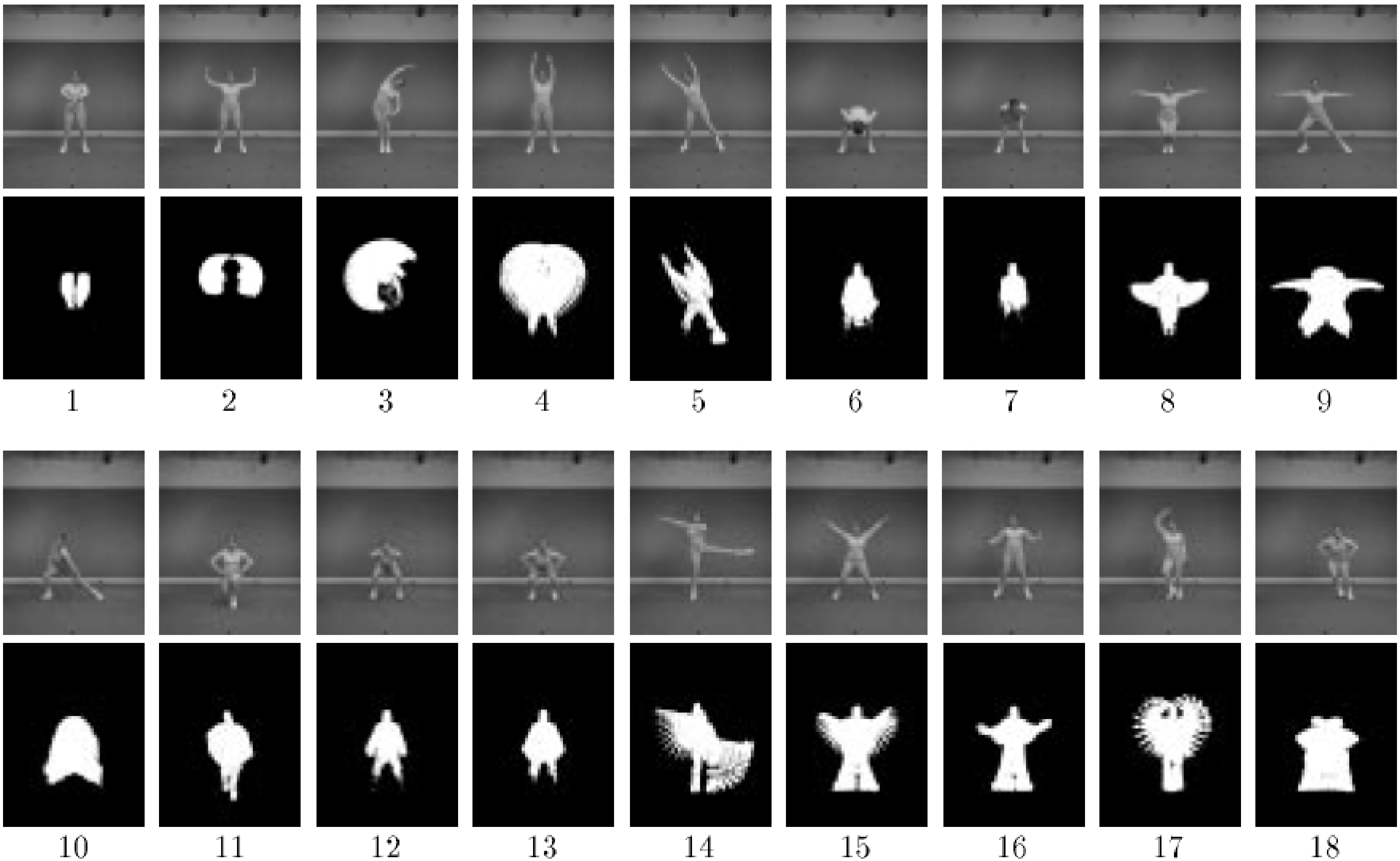
arms-wave MHI

crouch-down

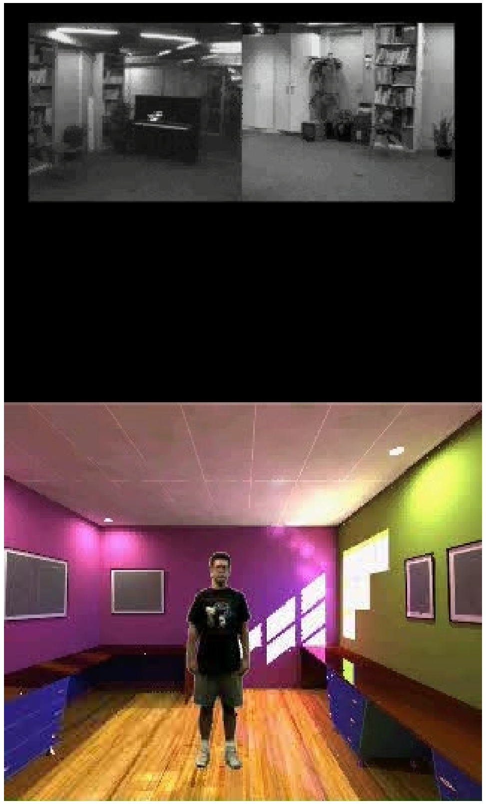


crouch-down MHI

Aerobics Dataset



Video

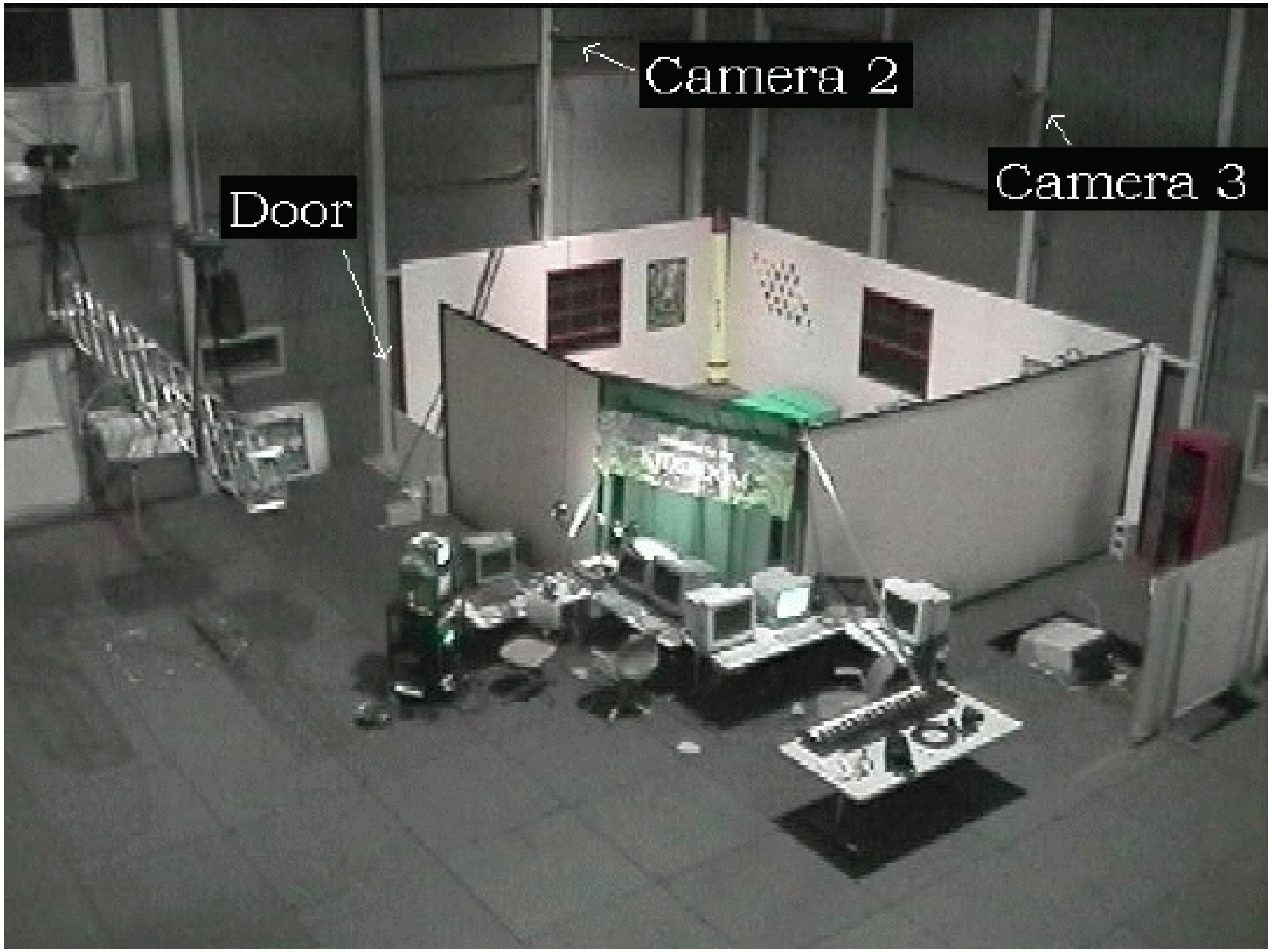


A. Bobick, S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schutte, and A. Wilson (1999)

``The Kidsroom: A Perceptually-Based Interactive and Immersive Story Environment''

Presence: Teleoperators and Virtual Environments, Vol. 8, No. 4, 1999, pp. 367-391.

The Kid's Room



[Bobick et al. 1996]

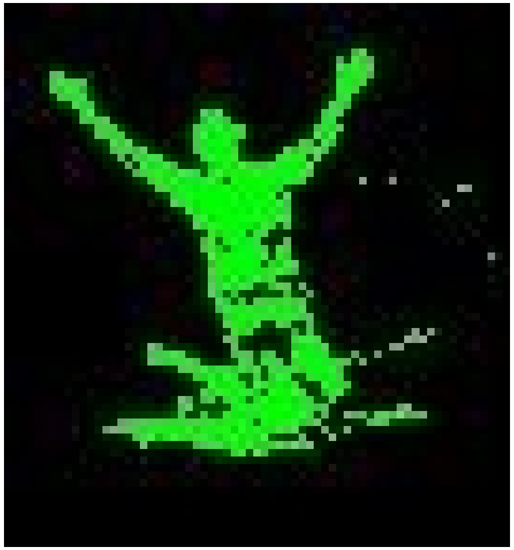
The Blue Monster

[http://vismod.media.mit.edu/vismod/demos/kidsroom/kidsroom.html]

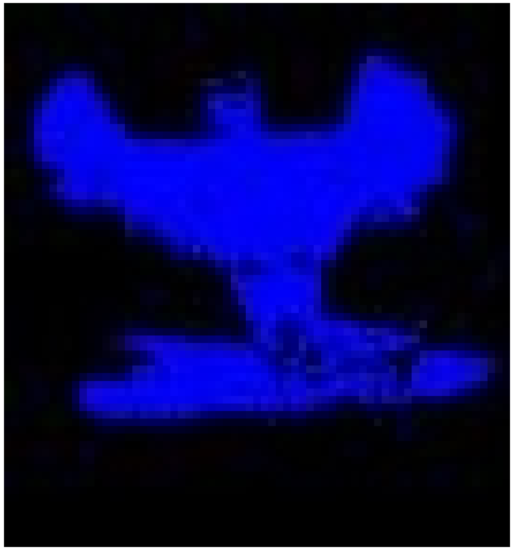
The Technology

[http://vismod.media.mit.edu/vismod/demos/kidsroom/kidsroom.html]

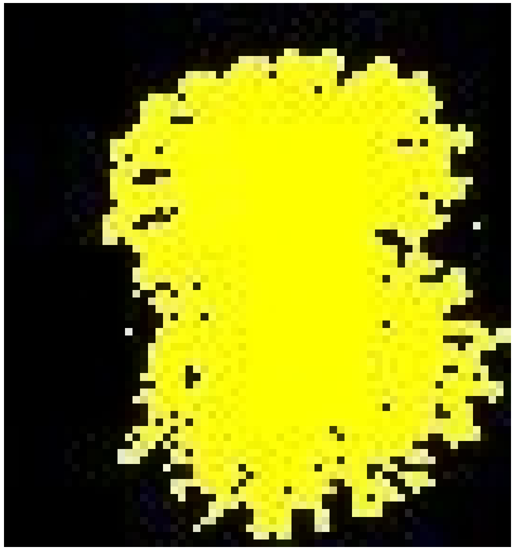
Motion History Templates



Making a 'Y'



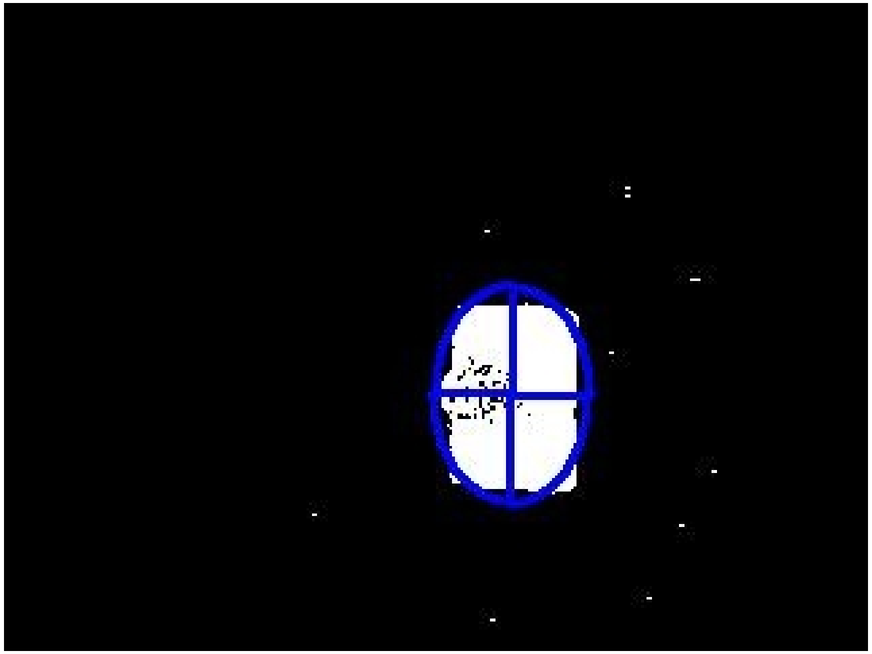
Flapping



Spinning

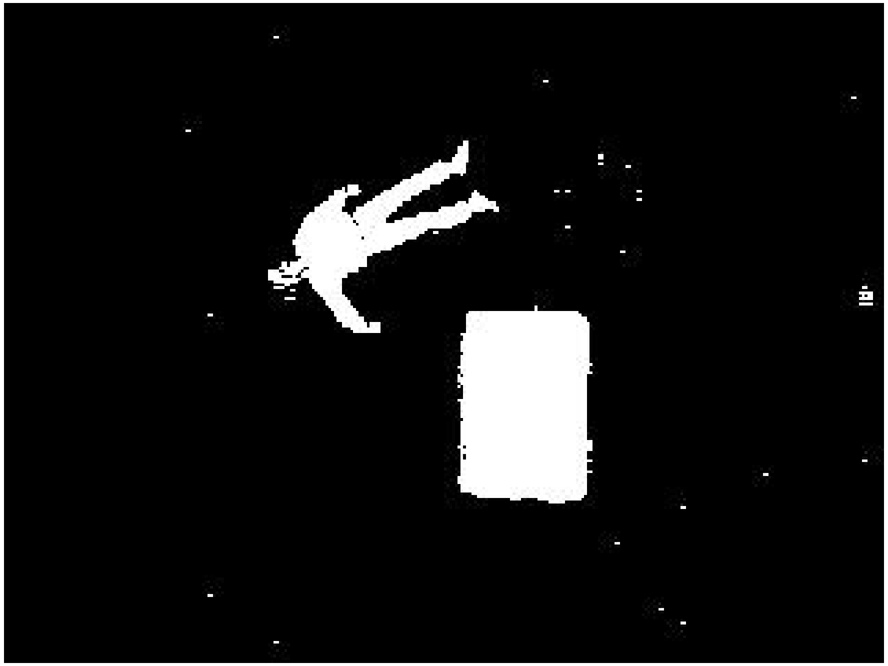
[http://vismod.media.mit.edu/vismod/demos/kidsroom/kidsroom.html]

Detecting the Bed



[http://vismod.media.mit.edu/vismod/demos/kidsroom/kidsroom.html]

Man Overboard Detector

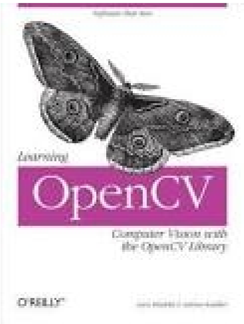


[http://vismod.media.mit.edu/vismod/demos/kidsroom/kidsroom.html]

Motion History Code Example

OpenCV Book and Code

- “Learning OpenCV”

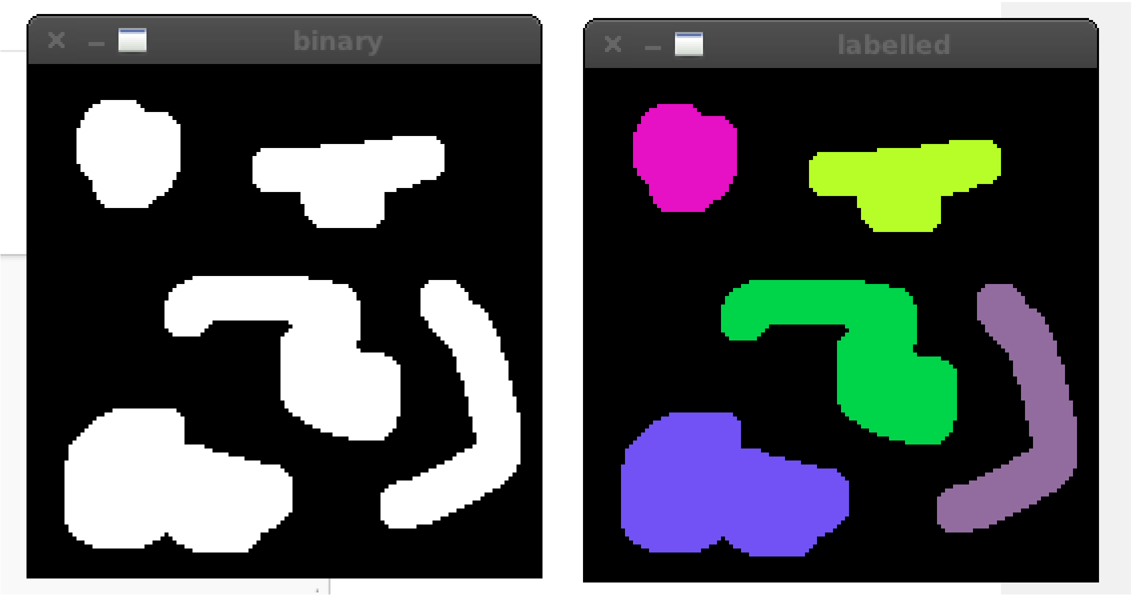


- Code from book is on github:

  https://github.com/Itseez/opencv_extra/tree/master/learning_opencv_v2

OpenCV Tutorials

- Connected Components:
  - http://nghiaho.com/?p=1102
  - https://davidlavy.wordpress.com/opencv/connected-components-in-opencv/



OpenCV Tutorials

- Circle Detection:
  - http://docs.opencv.org/3.1.0/d4/d70/tutorial_hough_circle.html#gsc.tab=0

OpenCV Tutorials

- Face Detection:
  - http://stackoverflow.com/questions/20757147/detect-faces-in-image
  - https://github.com/Itseez/opencv_extra/blob/master/learning_opencv_v2/ch13_ex13_4.cpp

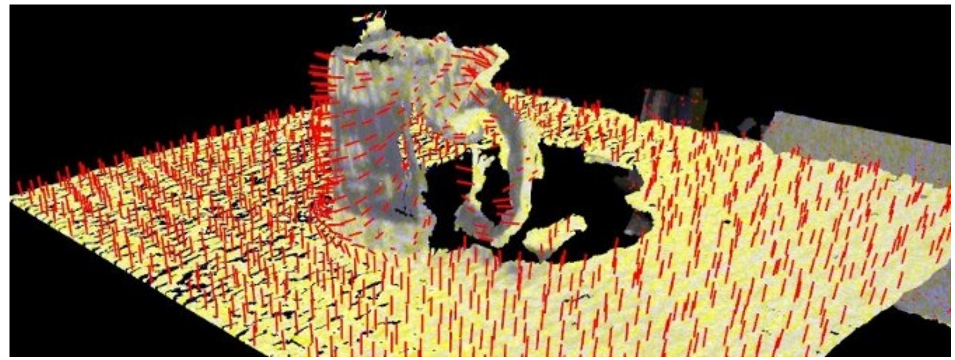
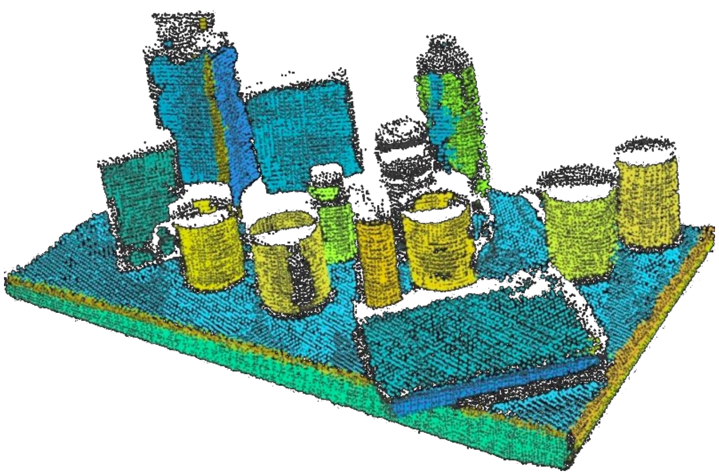
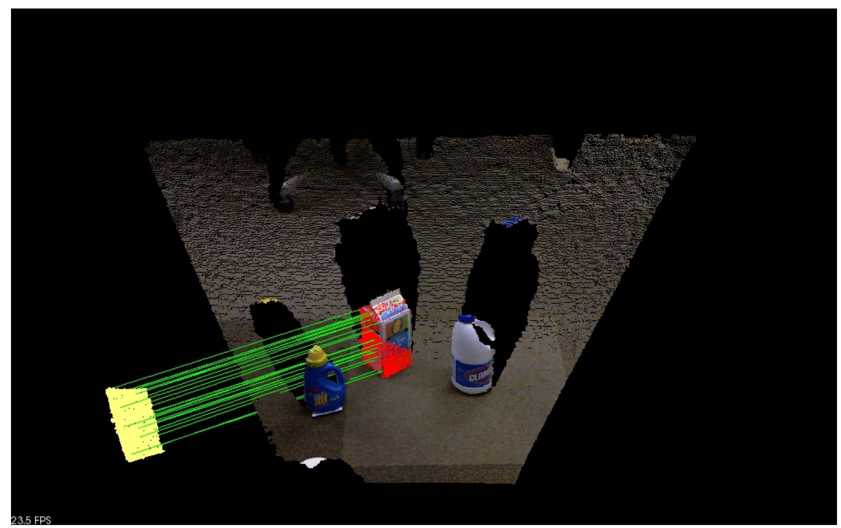
OpenCV Tutorials

- Blog full of OpenCV examples:
  - http://opencvexamples.blogspot.com/

Resources

- OpenCV in ROS:
  - http://wiki.ros.org/vision_opencv
  - http://wiki.ros.org/cv_bridge/Tutorials
  - http://docs.opencv.org/2.4/doc/tutorials/tutorials.html

Next time...3D Vision

Homework 6

Part 1: Color Detection – given a color image, detect the location of the pink hat

Part 2: One the hat has been detected, turn the robot in the direction of the hat

Homework 6

Part 1 can be completed off-line: I have recorded 3 .bag files containing sequences of images in which the pink hat appears

To demonstrate your solution, your code should draw a circle around the hat in each frame

Homework 6

Once you have completed Part 1, you will have to use the example from today to turn the robot in the direction of the hat (i.e., left or right).

The angle of turning should be larger if the hat is further to the left/right

Because we're working in 2D, you will not be able to compute the exact desired angle; instead, use a heuristic

THE END