MISSISSIPPI STATE UNIVERSITY

UNIVERSITY OF MISSOURI

Computer Vision: A Computational Intelligence Perspective – Part I



[1]Derek T. Anderson [2]James M. Keller [3]Chee Seng Chan

[1]Department of Electrical and Comp. Engineering, Mississippi State University, USA
[2]Department of Electrical and Comp. Engineering, University of Missouri, USA
[3]Computer Science and Information Technology, University of Malaya, Malaysia

IEEE WCCI – Vancouver, Canada – July 24 to 29th, 2016

What did you get yourself into!?

- Two part tutorial (2 hours each)
  - Part 1: Derek and Jim show
  - Part 2: Chan
- Open with some important concepts/terms
  - *Low*- to *mid*- to *high*-level computer vision (CV)
- Some topics are described at a high-level
  - Show the breadth of CI in CV
  - Help us discuss the overall role of CI in CV
- Go into more depth on a few topics
  - Hopefully leave with a few new *tricks*
- Discussion (debate?)

So, where/how does CI fit into CV?

- Our viewpoint
  - This talk is **NOT** a comprehensive survey
    - 2 hours would not do the field justice!
  - Scattered ideas
- Please don't be offended
  - If we do not acknowledge your research
  - Many examples come from our research
    - Yes, we are predominantly fuzzy guys …
- **But, let's indeed use today to debate the proper role of CI (FSs, EAs, NNs) in CV!**

Building blocks

- Today, we assume a basic understanding of

FSs

- Membership functions and operators (t-norm, t-conorm, etc.)
- Prop's: "U is A"
- Compositional rule of inference
- Fuzzy inference system (FIS)
- Extension principle

EAs

- Genetic algorithms (GAs)
  - Chromosome
  - Fitness fx's
  - Selection, mutation, crossover, …
- Helpful if you know about GPs, PSO, ACO, etc.

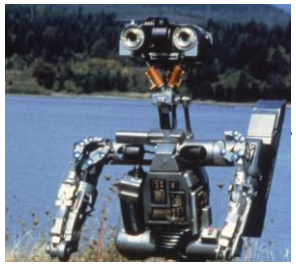
NNs

- Perceptron
- MLP
- Convolution

What is computer vision?

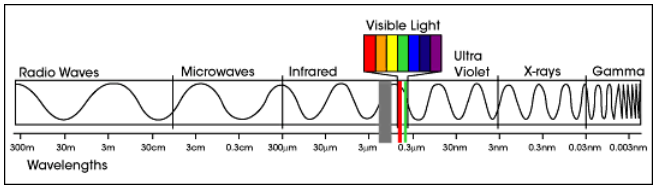
Computer

http://cs.brown.edu/courses/cs143/comp_vision_teaser_by_kirkh.jpg

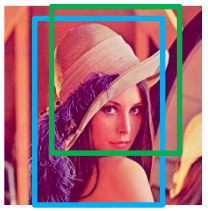


http://www.imdb.com/title/tt0091949/

GOALS … WHY DOES SYSTEM EXIST …?

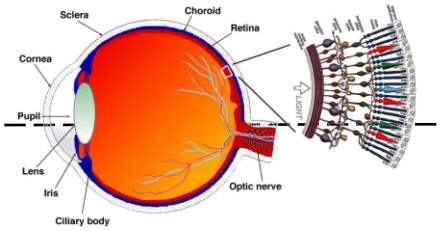


http://earthobservatory.nasa.gov/Experiments/ICE/panama/Images/em_spectrum.gif

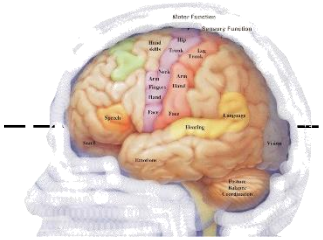


Hat
Lena
Posing

Biological



http://sips.inesc-id.pt/~pfzt/wordpress/wp-content/uploads/2010/12/Sagschem.jpeg

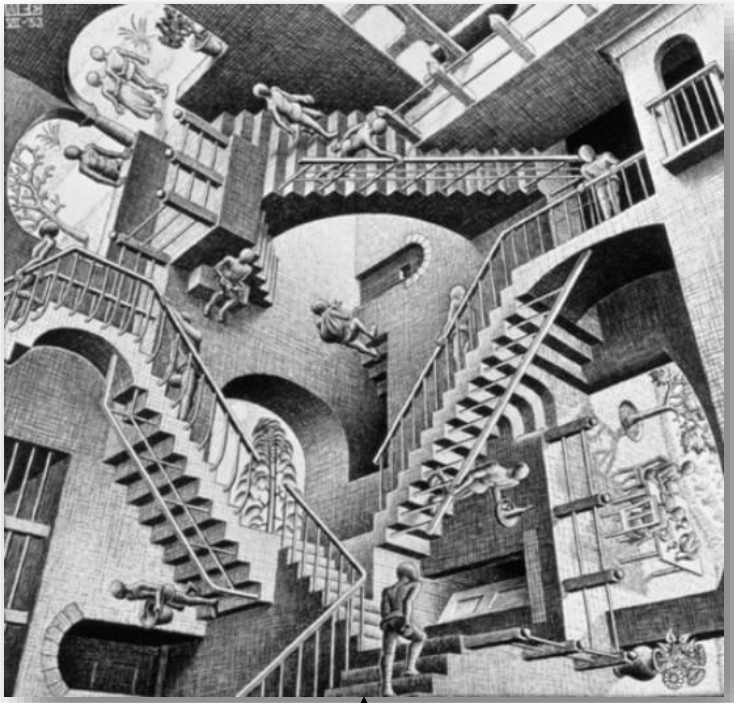


http://www.speed-light.info/miracles_of_quran/images/brain.jpg

Physics — Sensors — Low Level …. Mid Level …. High Level

Universe — *Observation* — *Processing (Data -> Decisions)*

What is computer vision?

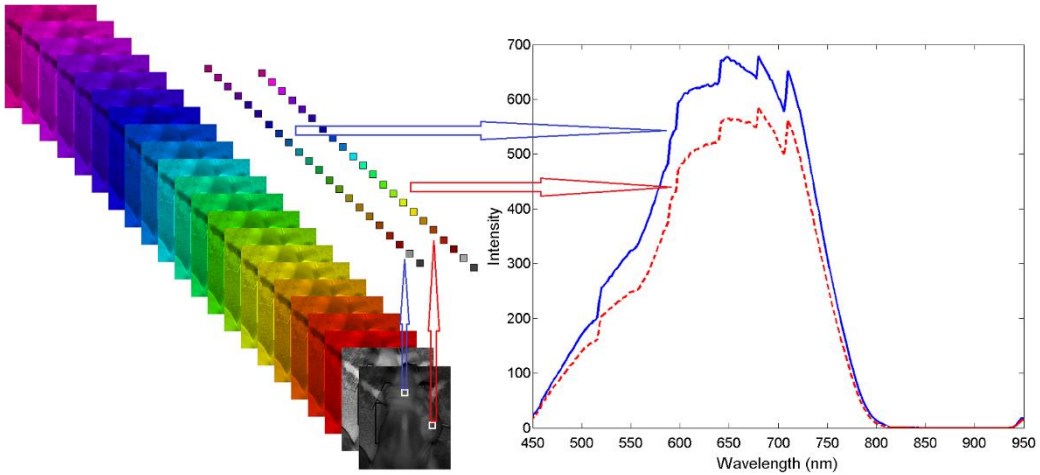


What's going on???

One take on CV: engineering algorithms, mathematics and/or systems to **understand** "signals"

"Signal" … complex concept

Hyperspectral (spectral, spatial, temp): $f_t(x,y,b)$

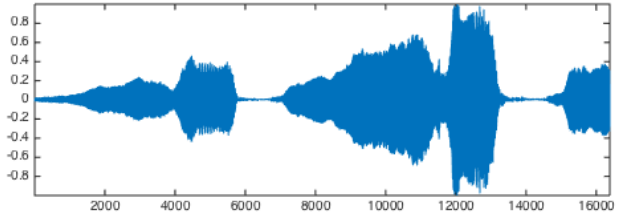


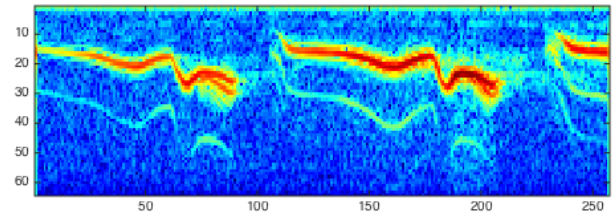
http://biomedicaloptics.spiedigitallibrary.org/data/Journals/BIOMEDO/24714/JBO_17_7_076005_f001.png

Temporal (and spatial): $f_t(x,y)$

Spatial: f(x,y) (and scary!)

Temporal: f(t)





STFT

http://www.numerical-tours.com/matlab/audio_1_processing/index_09.png

Shallow vs. *deep* questions

Detection

Recognition

Robert Luke

Derek Anderson

Jim Keller

Jim Bezdek

Good luck computer!
Not an easy image to understand …

Who?

4 people
Bob, J² and Derek

What?

Pirate party!!!

Where?

Columbia MO
Flat Branch Pub

When?

2007

Why?

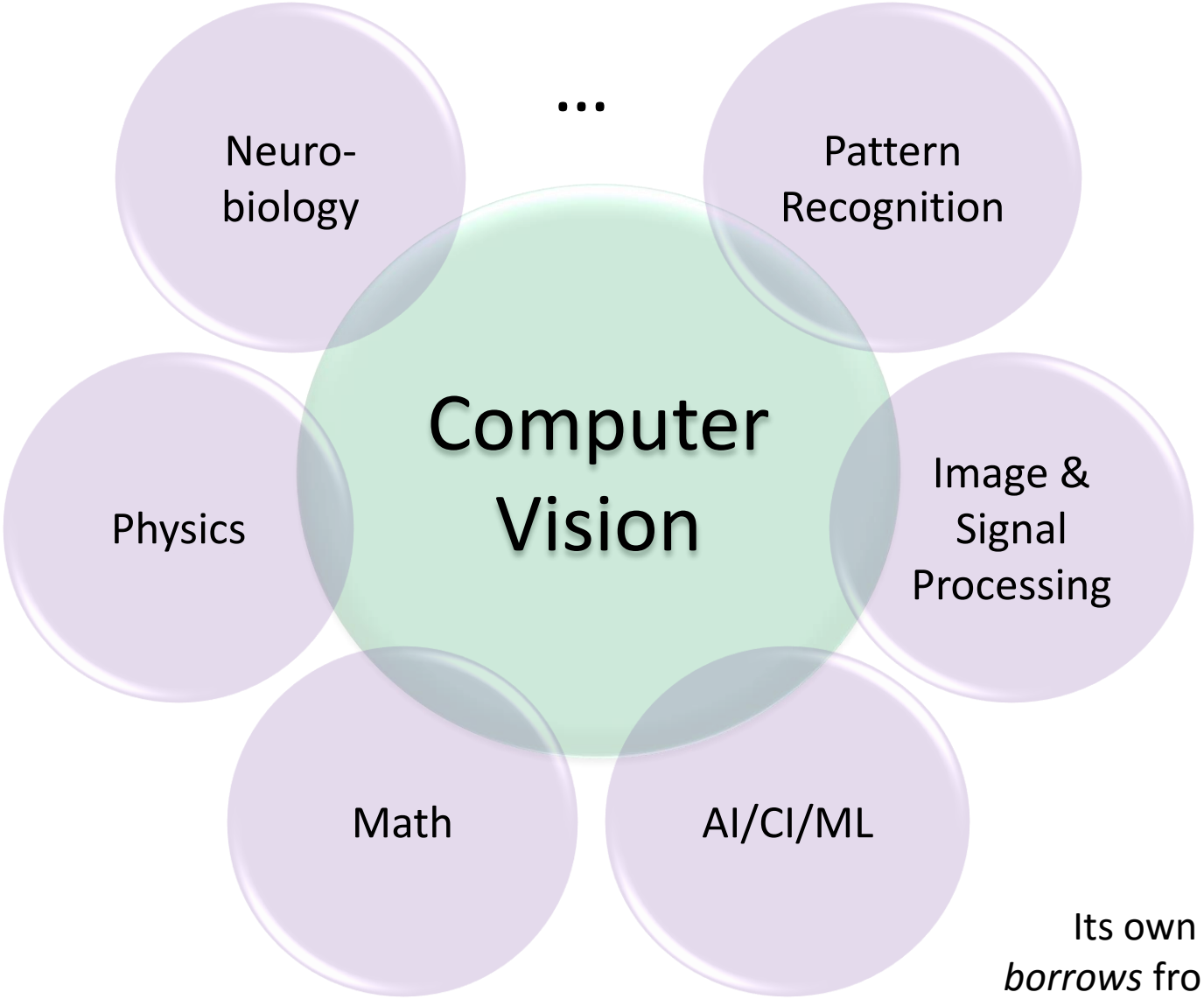
Good time!!!
TLAPD

Why study computer vision?

- Too many reasons to enumerate!
  - Entertainment
    - e.g., Kinect, augmented reality (hololens), …
  - Proliferation of mobile devices (smart phones)
    - MASSIVE amounts of images/videos
      - Photo stitch, face detection-based auto focus, biometrics, …
  - Ground vehicles and unmanned aerial vehicles (UAVs)
    - Smart cars, earth obv., intelligence, surveillance & resonance, …
  - Robotics
    - Control (e.g., industrial), navigation, Mars Rover, …
- Greater understanding of human vision
- Various dark, deep, attractive scientific/math mysteries …
  - e.g., how does detection and recognition *work*?

MARS Rover

Who *owns* CV?



Its own field, but
borrows from other fields

CV is steeped in statistical methods

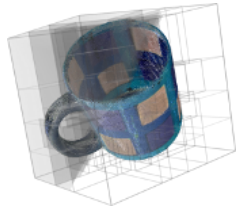
- Machine learning is a big component of CV
  - Statistics and statistical learning theory are big ML contributors
    - Probabilistic graphical models (e.g., Bayes-based, MRFs, etc.)
    - Support vector (and regression) machines (many fields claim these …)
    - Random decision forests, dictionary learning, etc.
    - List goes on and on and on and on …
  - One advantage, many of these techniques are “data driven”
- Recent activity in areas like (to name a few)
  - Deep learning, sparsity promotion, NMF, metric learning, manifold learning, online learning, etc.
- Talk about fuzzy …
  - Pattern recognition, CI, ML, Artificial Intelligence, …
  - Different approaches/tools …
  - Machine learning is MUCH more than just probability theory

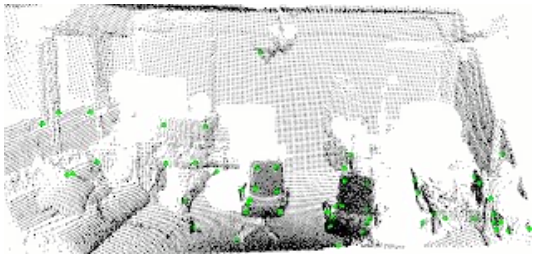
State-of-the art in CV

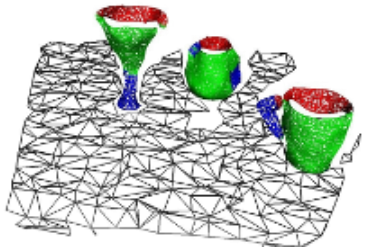
- Big research communities
  - PAMI, ICCV, CVPR, ECCV, NIPS, CVIU, …
  - **TFS, TNNLS and TEC?**
- Field of "tools"
  - Many open source CV tools avilable
    - OpenCV, VLFeat, SimpleCV, PCL, Torch7, OpenML, libSVM, Shogun, etc.

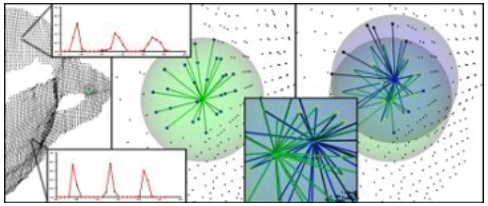
pointcloudlibrary

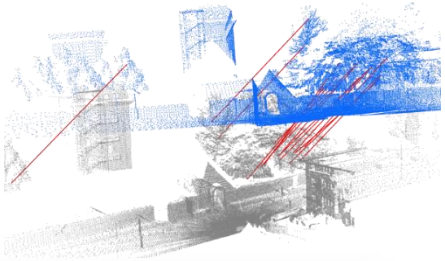
OpenCV

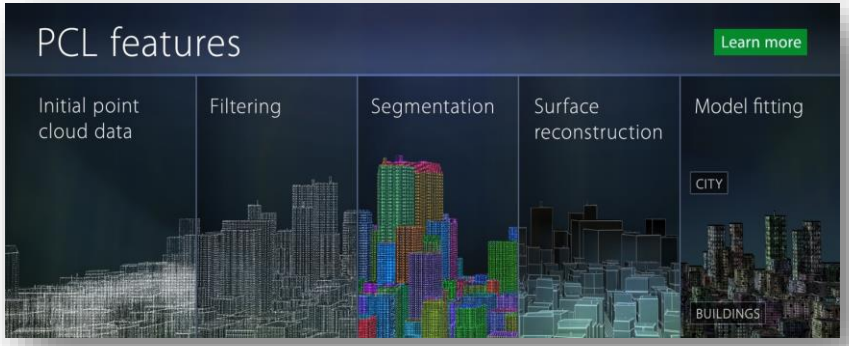












Publishing work in CV

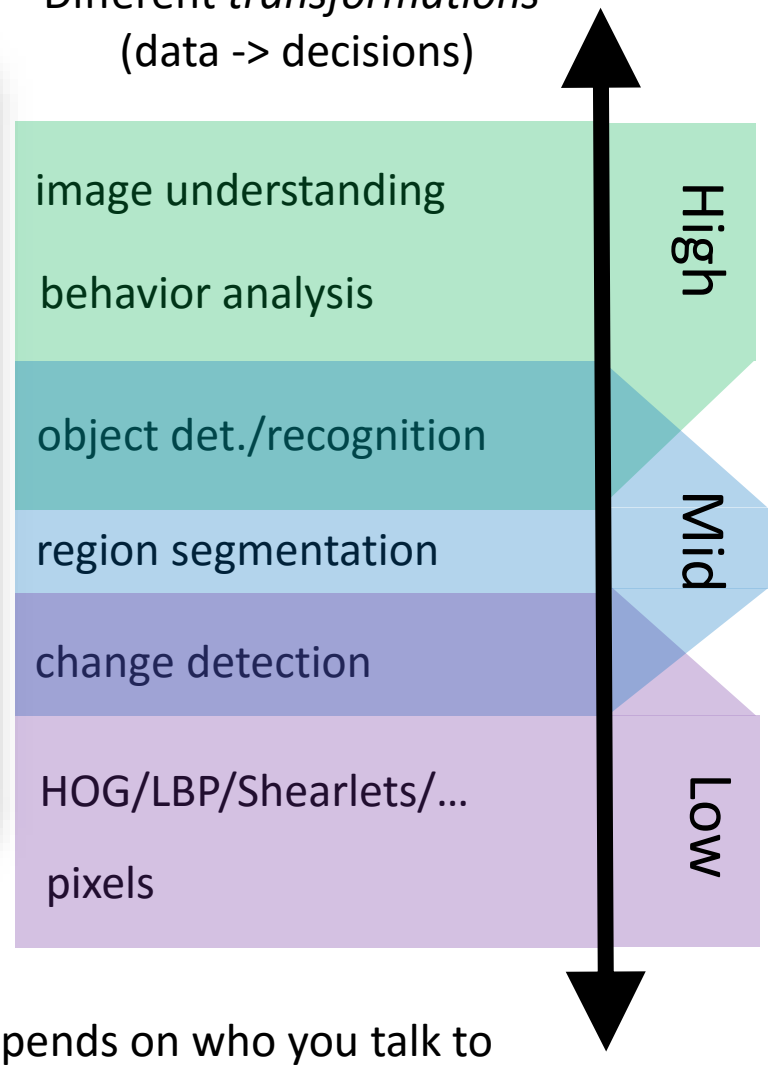
- Field is very **results** and **comparison heavy**
  - Need to process A LOT of data (but, variety vs. volume)
    - Month of data with just two events is a very sparse data set
  - Use community benchmark data sets
  - "Compare" to other related techniques
- Code sharing
  - Reproducible research
  - Better (in theory) benchmarking/comparison
- Depending on who you talk to …
  - Approach 1: do whatever it takes to get the job done
    - Looking for system s to solve CV task X
  - **Approach 2: theory that transcends beyond CV**
    - Solve CV task X but in reality we are focused on general theory t

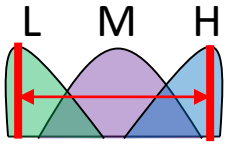
Always stick to your principles!

- David Marr
  - Neuroscientist and CV pioneer, 1945-1980
- Principle of least commitment (**PLC**)
  - Don't do something that may later have to be undone
  - "If the Principle of Least Commitment has to be disobeyed, one is either doing something wrong or something very difficult"
- Principle of graceful degradation (**PGD**)
  - Degrading the data will not prevent the delivery of at least some of the answer
  - Algorithms should be robust
  - Processes should possess some degree of continuity

Levels of CV

Different *transformations*
(data -> decisions)



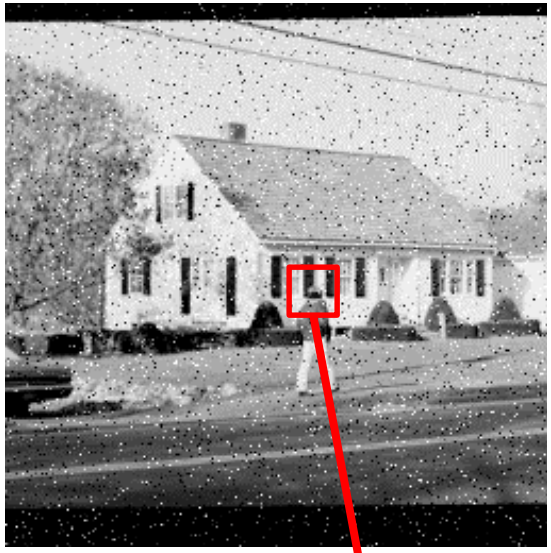


OK, its really sort of *fuzzy* and sort of depends on who you talk to

Image enhancement: fuzzy logic

Rules use pixel neighborhood characteristics to "select" strength of different enhancement/smoothing operators in combination – adaptive *fusion* of algorithms

A Robust Approach to Image Enhancement Based on Fuzzy Logic (Choi & Krishnapuram)

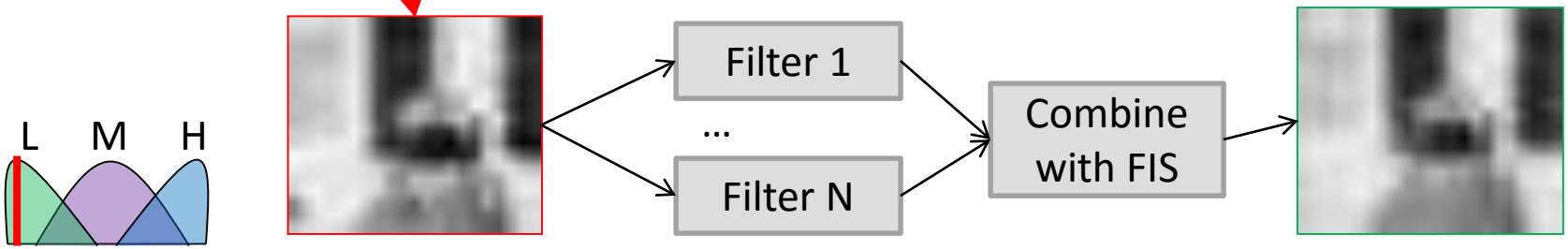


Noisy Original

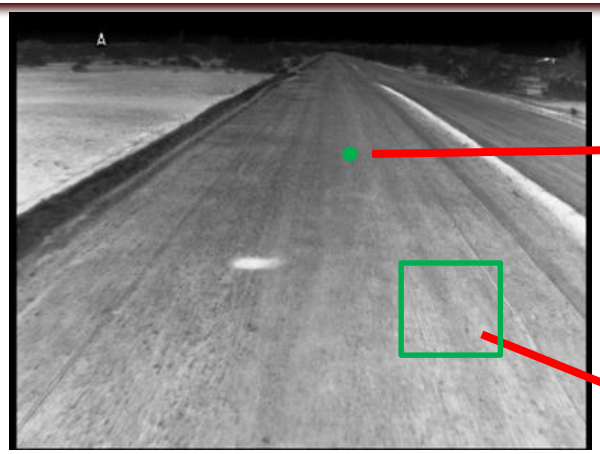
Median Filter

Fuzzy Rule Base

Which one do you like best?



Simple pixel features: local binary pattern (LBP)



Compute LBP Code for each pixel.

8	9	8
3	8	9
2	4	3

→

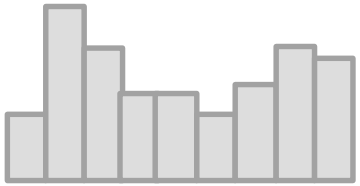
1	1	1
0		1
0	0	0

×

2^0	2^1	2^2
2^7		2^3
2^6	2^5	2^4

$$LBP_n = \sum_{k=0}^{n} s(i_k - i_c)2^k, \quad s(x) = \begin{cases} 1 \; if \; x \geq 0 \\ 0 \; if \; x < 0 \end{cases}$$

Compute histogram of codes inside a window

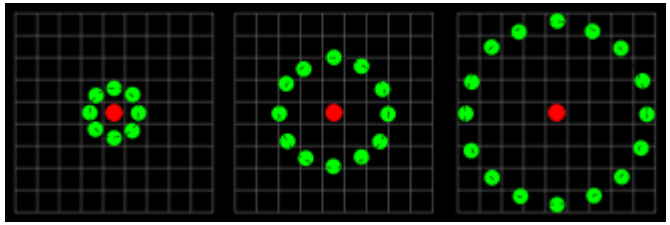


$LBP_{8,1}^{riu2}$

10 features per cell.

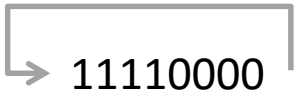
Local binary pattern (LBP)

Variations



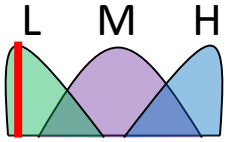
$LBP_{n,r}$

Rotation Invariant



Uniform

- Count 01 10 transitions.
- Combine codes with more than *u* transitions

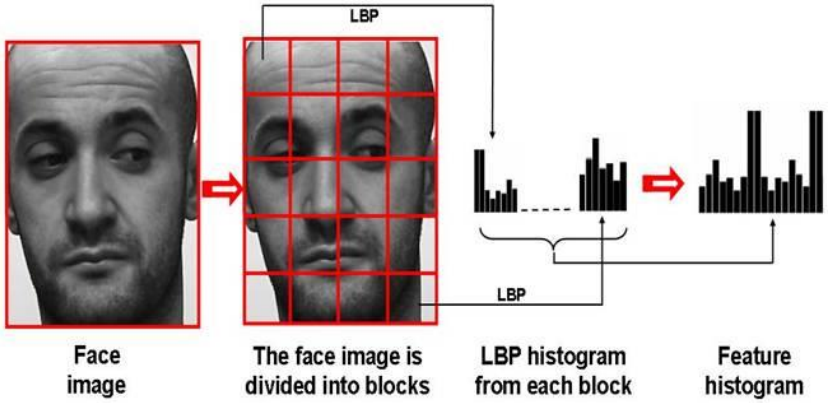


Robust to (uniform) changes in illumination

Ojala et al, PR'96, PAMI'02

Spatial context: histogram of LBPs

- Divide image chip into cells (e.g., 16x16 pixels for each cell)
- Compute histogram in each cell -> frequency of each "number"



* Can be sped up via integral image

- Optionally, normalize the histogram (e.g., l1 or l2 norm)
- Concatenate normalized histograms of all cells
- Spatial context is often important
- Can implement pyramid feature

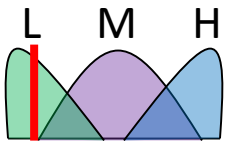
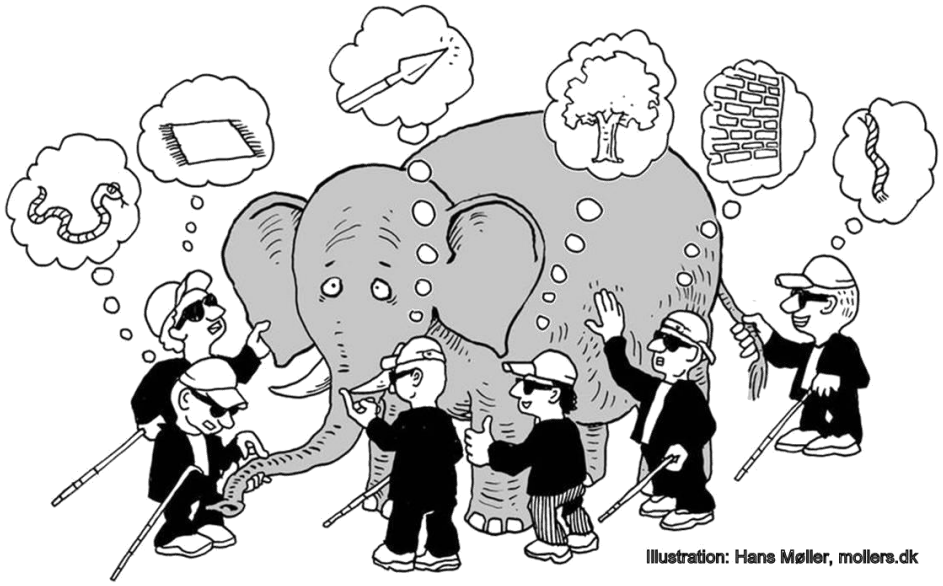


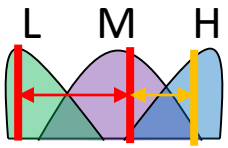
Image patches: move beyond pixels



Six blind men and an elephant

Bag of Words (BOWs)

http://upload.wikimedia.org/wikipedia/commons/0/08/Bag_of_words.JPG



Deformable Parts Model

Our 2013 FSs CV collaborator (Pirate Tony Han)

Examples of Levels

Keypoint detector and descriptor

Scale-Invariant Feature Transform (SIFT)

David G. Lowe
- Distinctive image features from scale-invariant keypoints, IJCV 2004
- Object recognition from local scale-invariant features, ICCV 1999

http://www.cs.ubc.ca/~lowe/keypoints/
http://www.vlfeat.org/~vedaldi/code/sift.html

Numerous applications: image stitching, object detection, etc.

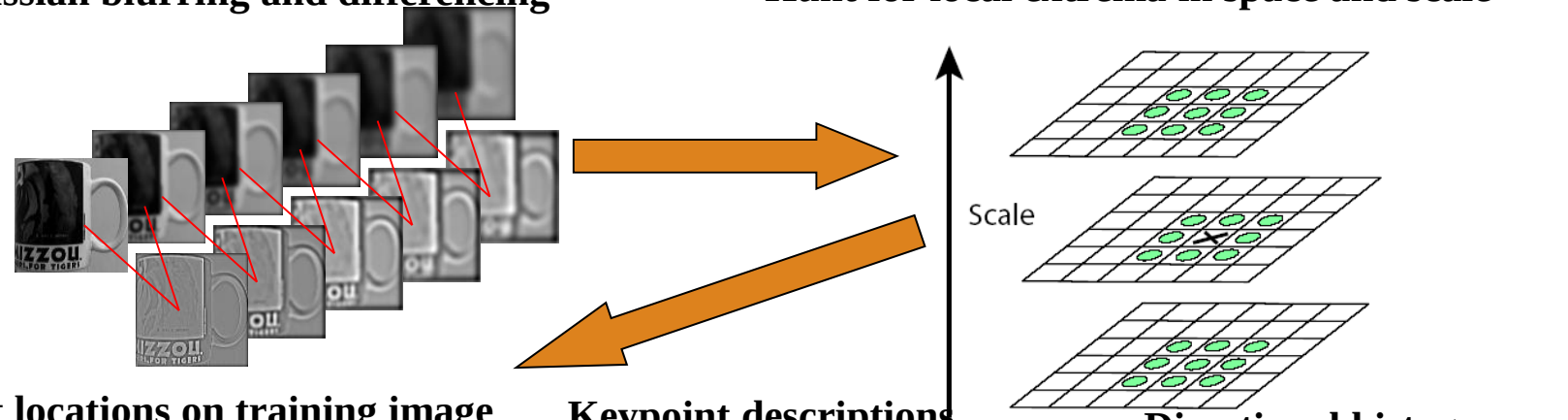
Others: SURF, ASIFT, etc.

Examples of Levels

Scale invariant feature transform

Gaussian blurring and differencing

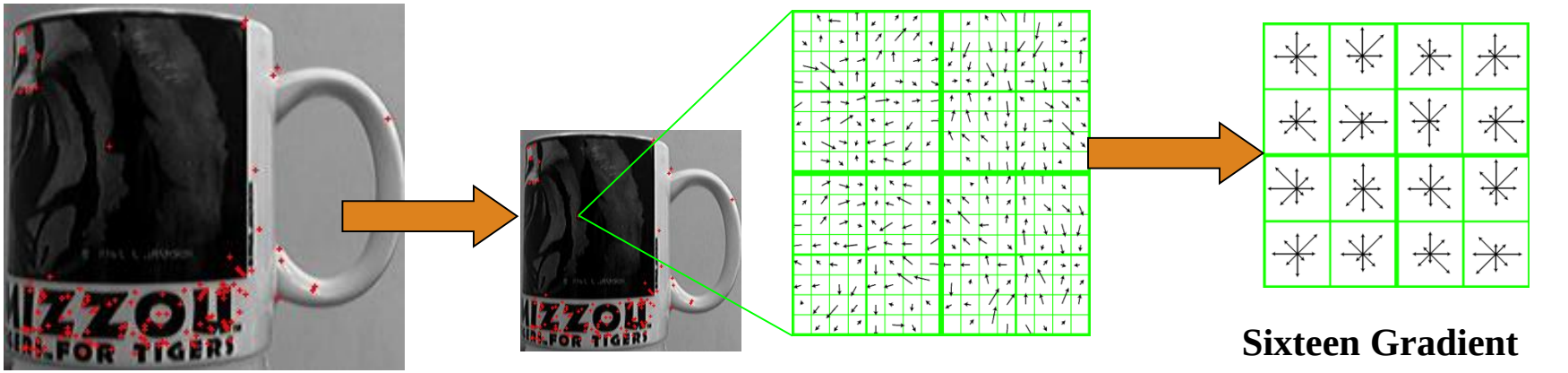
Hunt for local extrema in space and scale



Keypoint locations on training image

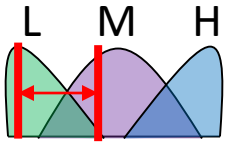
Keypoint descriptions

Directional histograms



Sixteen Gradient Histograms Created

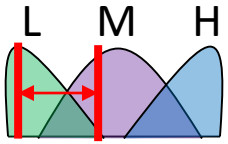
- **Major direction of gradients is determined**
- **Rotate gradient locations so that keypoint orientation is 0º.**
- **Rotate individual gradient directions to be consistent with orientation**



SIFT overview

- Detector (Determine *where to extract the feature*)
  1. Find Scale-Space Extrema (*using DOG*)
  2. Keypoint Localization & Filtering

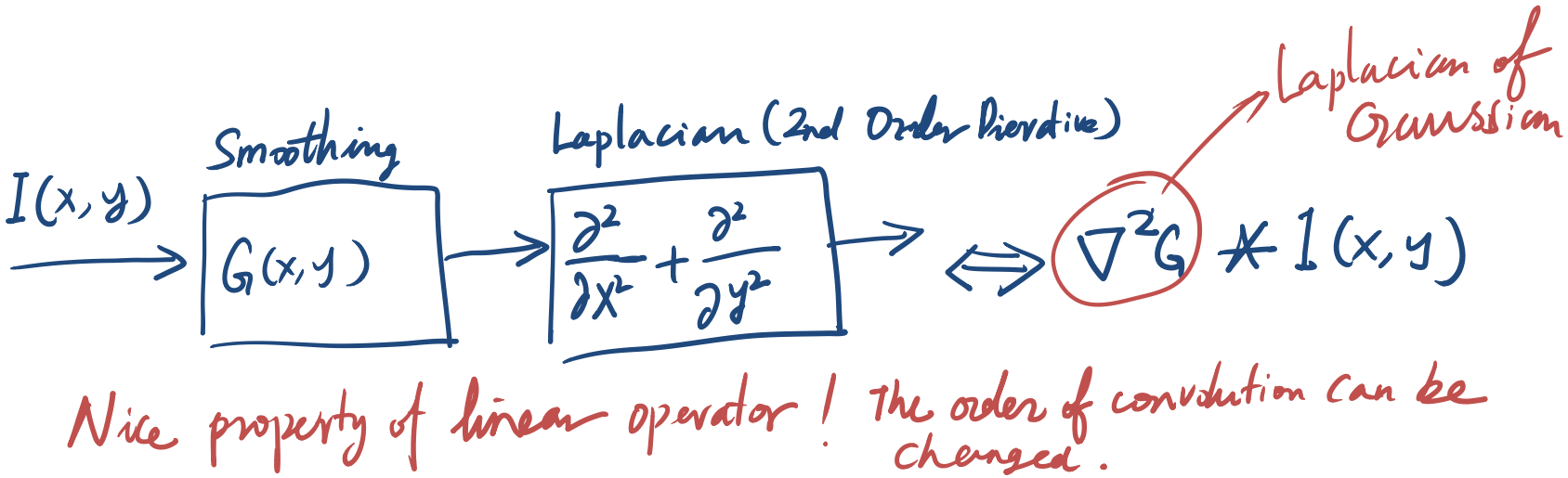
- Descriptor (*How to encode the feature*)
  1. Orientation Assignment (*remove effects of rotation and scale*)
  2. Create descriptor (*using histograms of orientations*)

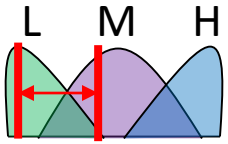


Examples of Levels
Scale selection

- The Laplacian-of-Gaussian (LoG) operator: $\sigma^2 \nabla^2 G$, where

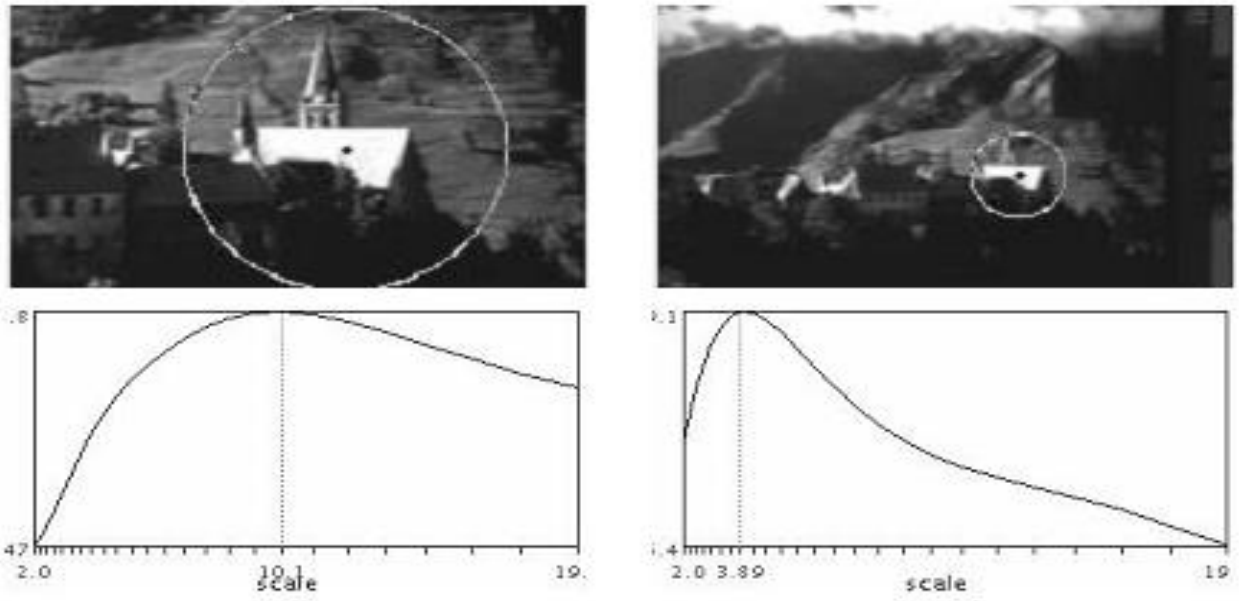
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

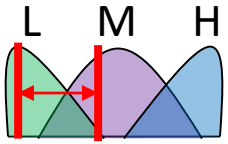




Examples of Levels
Scale selection

- Experimentally, maxima of Laplacian-of-Gaussian gives a notion of *scale*





Approx. LoG using diff-of-Gaussians (DoG)

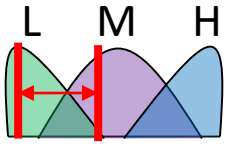
- LoG is *expensive*, so let's approximate it
  - Using the heat-diffusion equation:

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(k\sigma) - G(\sigma)}{k\sigma - \sigma}$$

$$(k-1)\sigma^2\nabla^2 G \approx G(k\sigma) - G(\sigma)$$

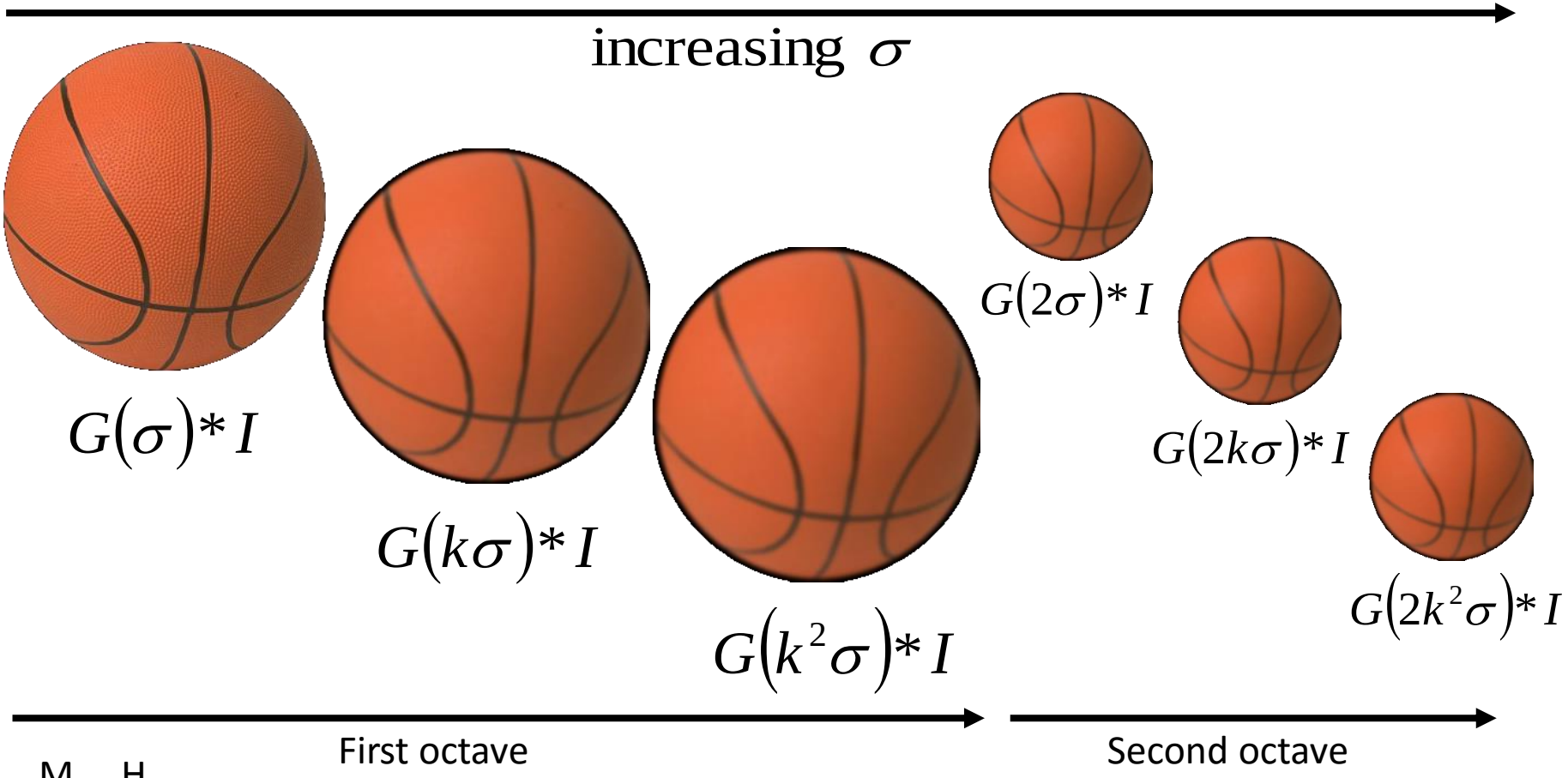
  - Define Difference-of-Gaussians (DoG):

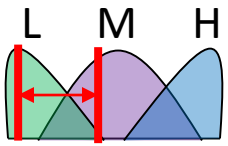
$$D(\sigma) \equiv (G(k\sigma) - G(\sigma)) * I$$



Scale space construction

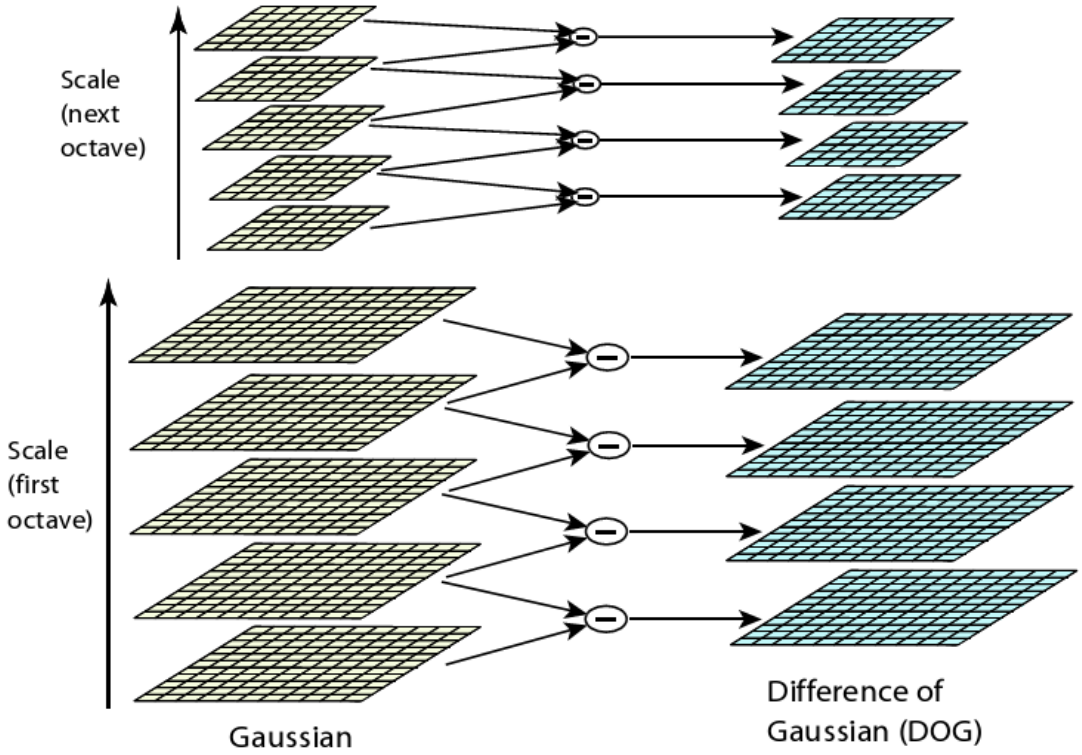
- First construct scale-space:

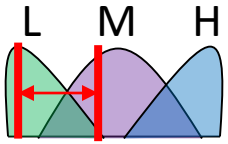




Examples of Levels
Difference-of-Gaussians

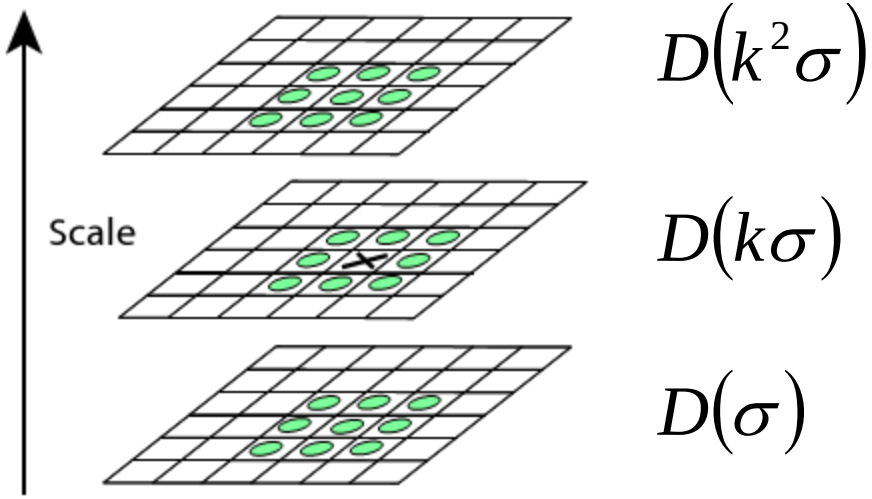
- Now take differences:

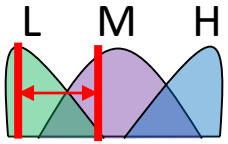




Scale space extrema

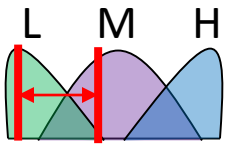
- Choose all extrema within 3x3x3 neighborhood.
- Low cost – only several usually checked





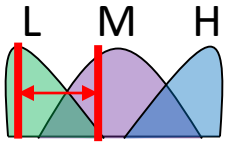
Examples of Levels
Keypoints

(a) 233x189 image
(b) 832 DOG extrema



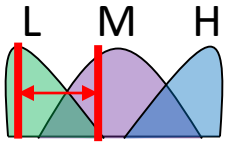
Properties of ideal descriptors

- Want descriptor to be robust w.r.t.
  - Affine transformations
    - Translation, scale, rotation
  - Lighting
  - Color distortion
  - Background clutter
- The descriptor should be discriminative



Orientation assignment

- Now we have set of good points
- Choose a region around each point
  - Removes (robust?) effect of scale and rotation



Orientation assignment

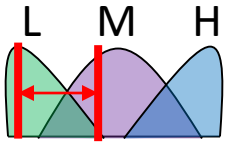
- Use scale of point to choose correct image:

$$L(x,y) = G(x,y,\sigma) * I(x,y)$$

- Compute gradient magnitude and orientation using finite differences:

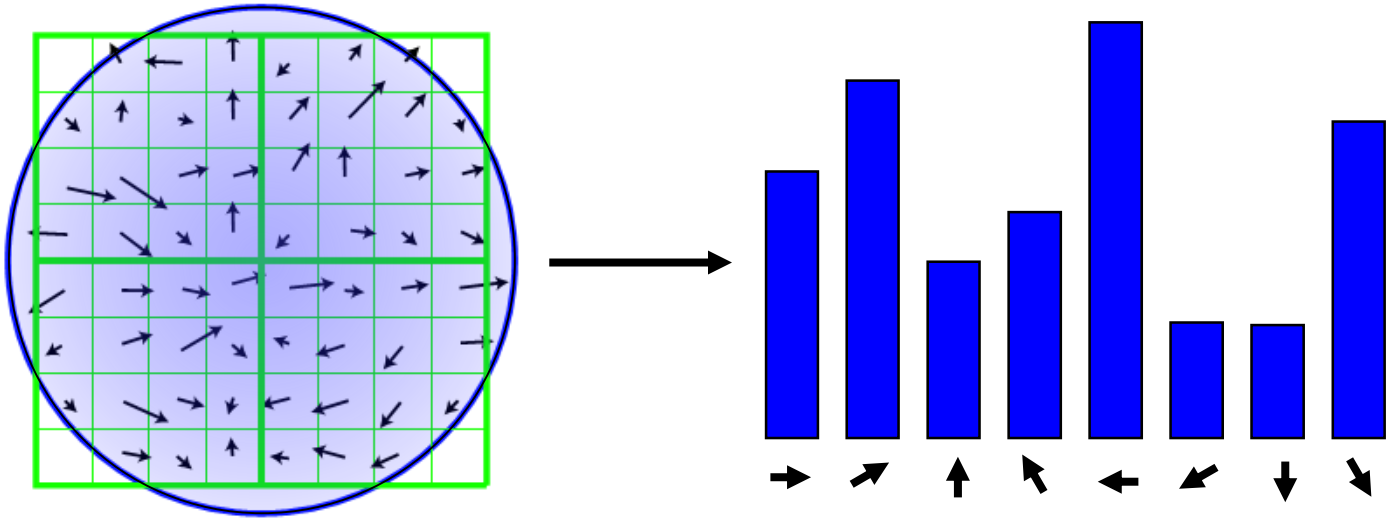
$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

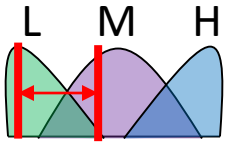
$$\theta(x,y) = \tan^{-1}\left(\frac{(L(x,y+1) - L(x,y-1))}{(L(x+1,y) - L(x-1,y))}\right)$$



Orientation assignment

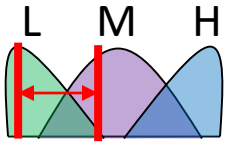
- Create gradient histogram (8 bins)
  - Weighted by magnitude and Gaussian window (sigma is 1.5 times that of the scale of a keypoint)





Orientation assignment

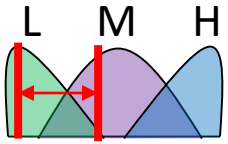
- Any peak within 80% of the highest peak is used to create a keypoint with that orientation
- ~15% assigned multiple orientations, but contribute significantly to the stability
- Finally a parabola is fit to the 3 histogram values closest to each peak to interpolate the peak position for better accuracy



Examples of Levels
SIFT descriptor

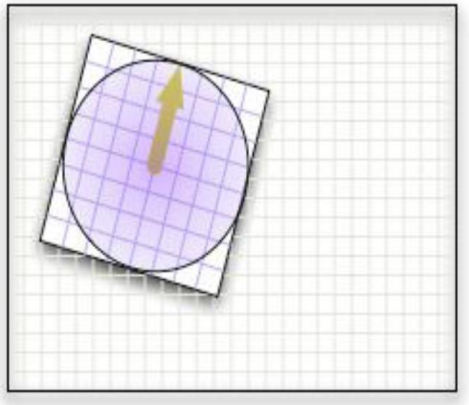
Edelman et al. 1997

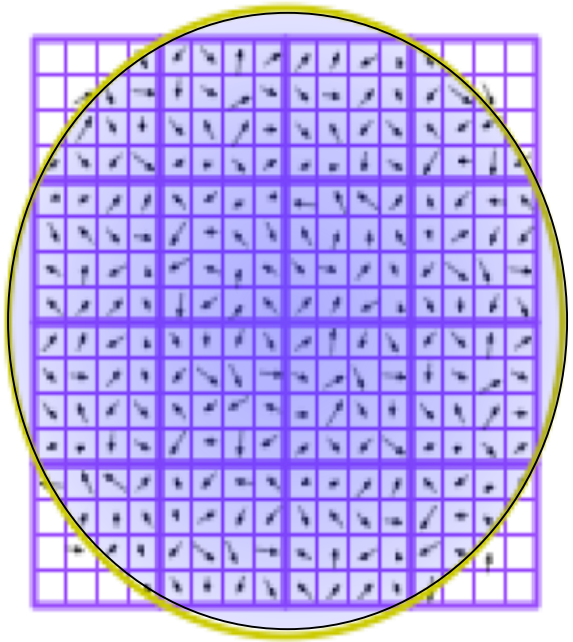
- Each point so far has x, y, σ, m, θ
- Now we need a descriptor for the region
  - Could sample intensities around point, but…
    - Sensitive to lighting changes
    - Sensitive to slight errors in x, y, θ
- Look to biological vision
  - Neurons respond to gradients at certain frequency and orientation
    - But location of gradient can shift slightly!



Examples of Levels
SIFT descriptor

- 4x4 gradient window
- Histogram of 4x4 samples per window in 8 directions
- Gaussian weighting around center
- 4x4x8 = 128 dimensional feature vector





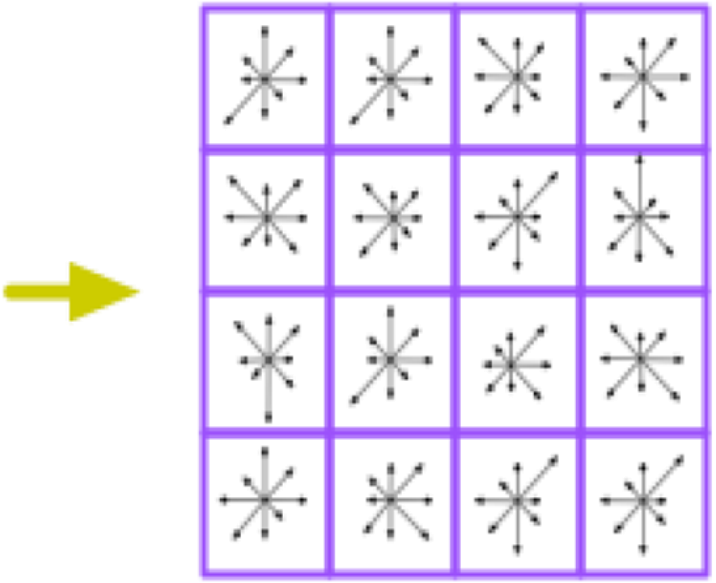


Image gradients

Keypoint descriptor

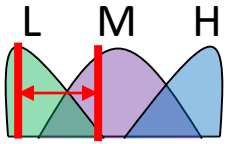
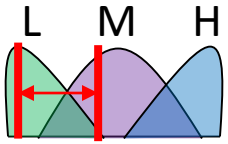


Image from: Jonas Hurrelmann

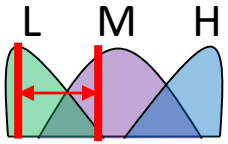
SIFT – lighting changes

- Gains do not affect gradients
- Normalization to unit length removes contrast
- Saturation affects magnitudes much more than orientation
- Threshold gradient magnitudes to 0.2 and renormalize



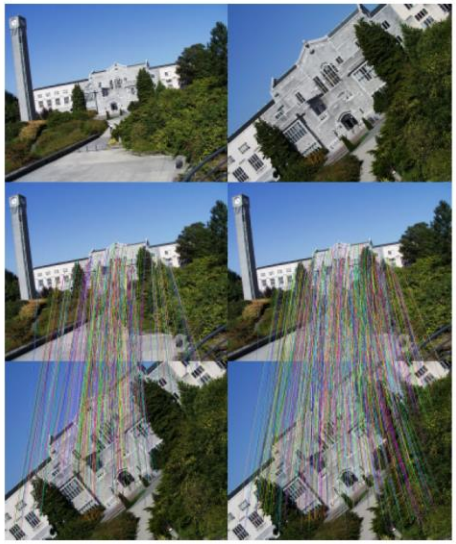
SIFT Performance

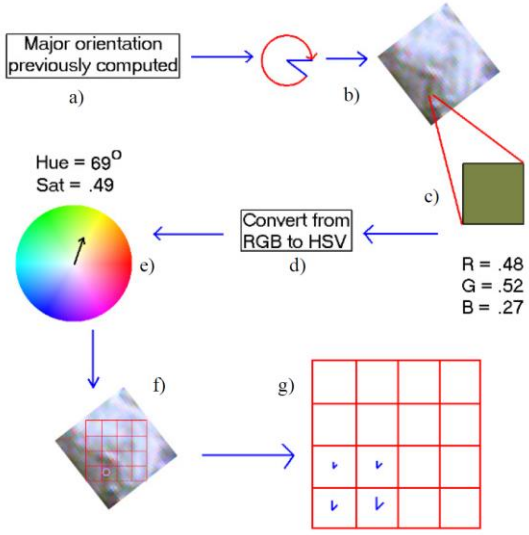
- Extremely robust
  - 80% repeatability at
    - 10% image noise
    - 45° viewing angle
    - 1k-100k keypoints in database
- Best descriptor in [Mikolajczyk & Schmid 2005]'s extensive survey
- According to Google Scholar, the SIFT paper has been cited for 35,325 times by 2014
- According to Microsoft academic search, the paper is the second mostly cited paper in CS in the last 10 years

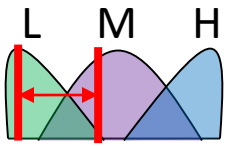


Examples of Levels
Color SIFT

- 2008: Extending the Scale Invariant Feature Transform into the Color domain
  - Robert Luke, James Keller, Jesus Chamorro-Martinez
- Summary
  - Extend SIFT into the color domain by using hue and saturation (from HSV) as the “gradient vector” and build the SIFT descriptor
    - Hue is angle and saturation is mag
  - Wait, three fuzzy guys and no fuzzy!

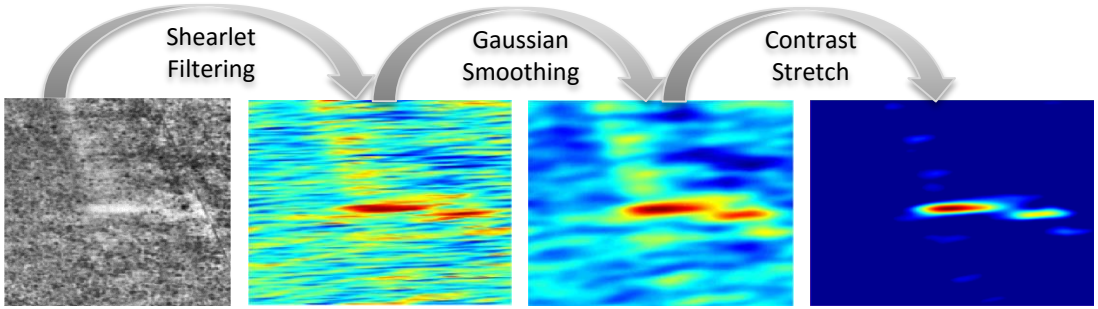




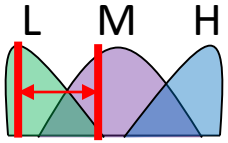


Soft features: overview

- Motivation
  - When we extract features for a region of interest (ROI), we typically extract **both** background and foreground information
  - Can confuse classifiers and make *transfer* difficult
    - Example: collection of car images, Jim in each one with a red hat …
- What if we had *evidence* about foreground locations?



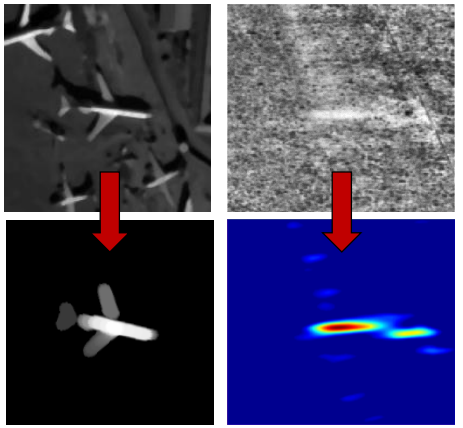
- Breakdown (filtering then soft feature extraction)
  - Identify *locations* that likely contain relevant information and extract our features there, but do so in a *soft* versus *hard* way

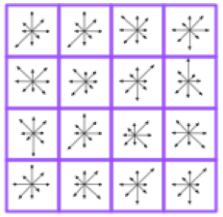


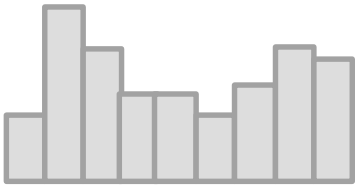
[Price, Anderson, Scott, et al., AIPR 2011, SPIE 2013, 2014, IGARSS 2012]

Soft features: general idea

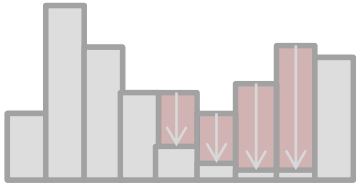
- Step 1 (supervised approach)
  - Identify domain specific filter(s)
    - E.g., ATR in aerial imagery (DMPs and FI)
      - Scott & Anderson: AIPR 2011 & IGARSS 2012
    - E.g., explosive hazard detection (Shearlets)
      - Price & Anderson: SPIE 2013 and 2014
- Step 2
  - Select feature(s) and descriptor(s)
    - E.g., HOG, LBP, Harr-like, IOM, etc.
- Step 3
  - "Extend" descriptors
    - E.g., weighted histogram versions



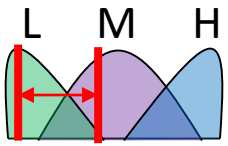




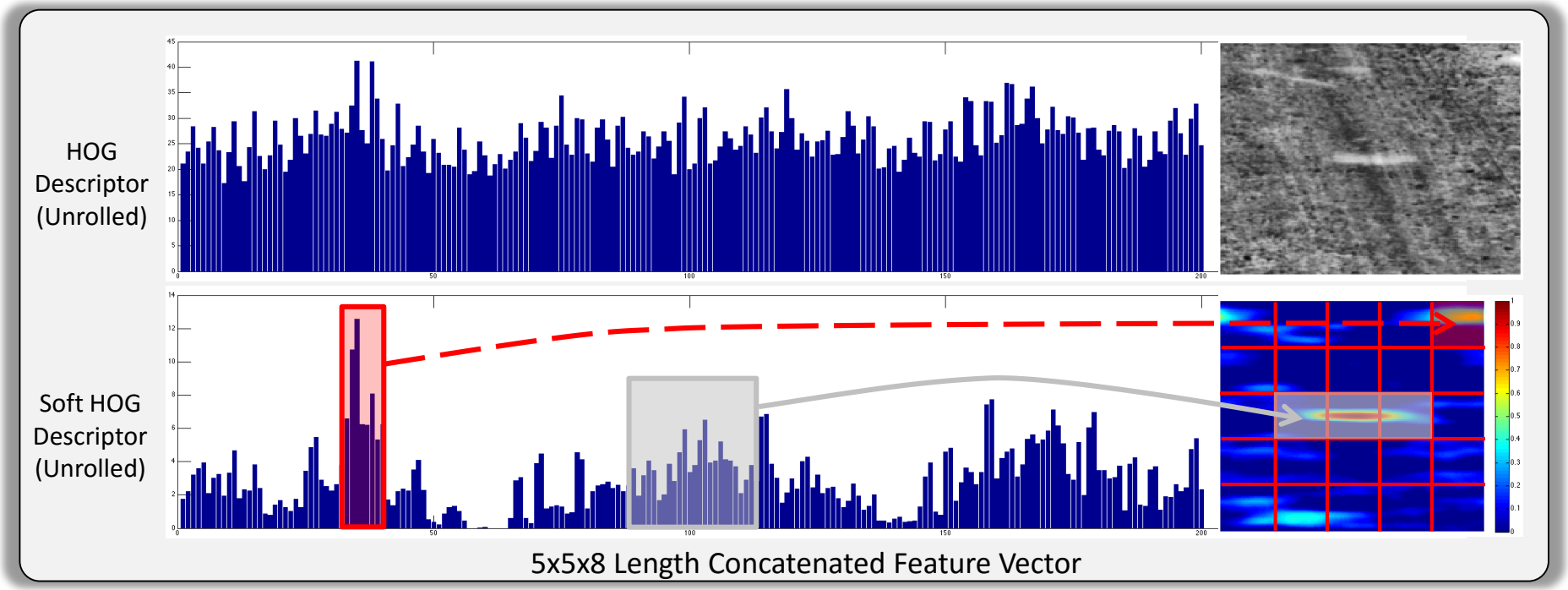
Before

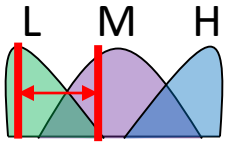


After



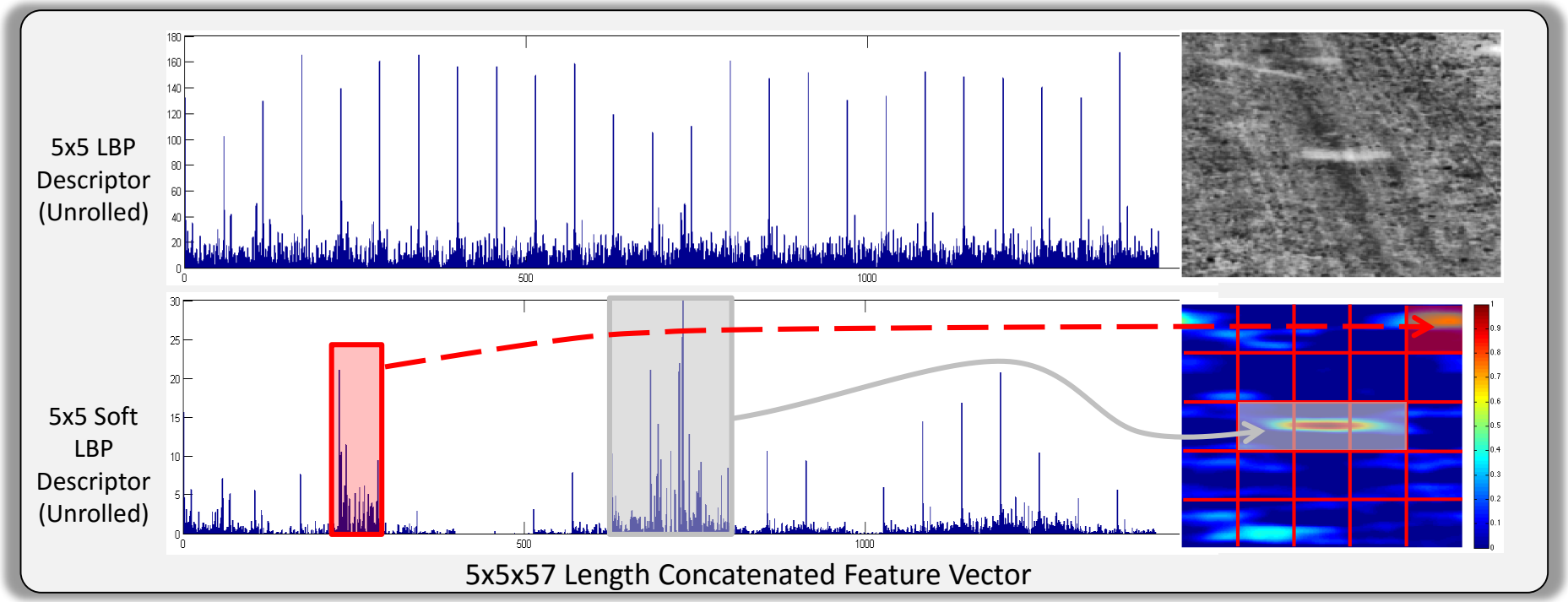
Soft features: explosive hazard detection

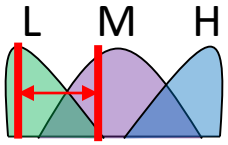




[Price, Anderson, SPIE 2013, 2014]

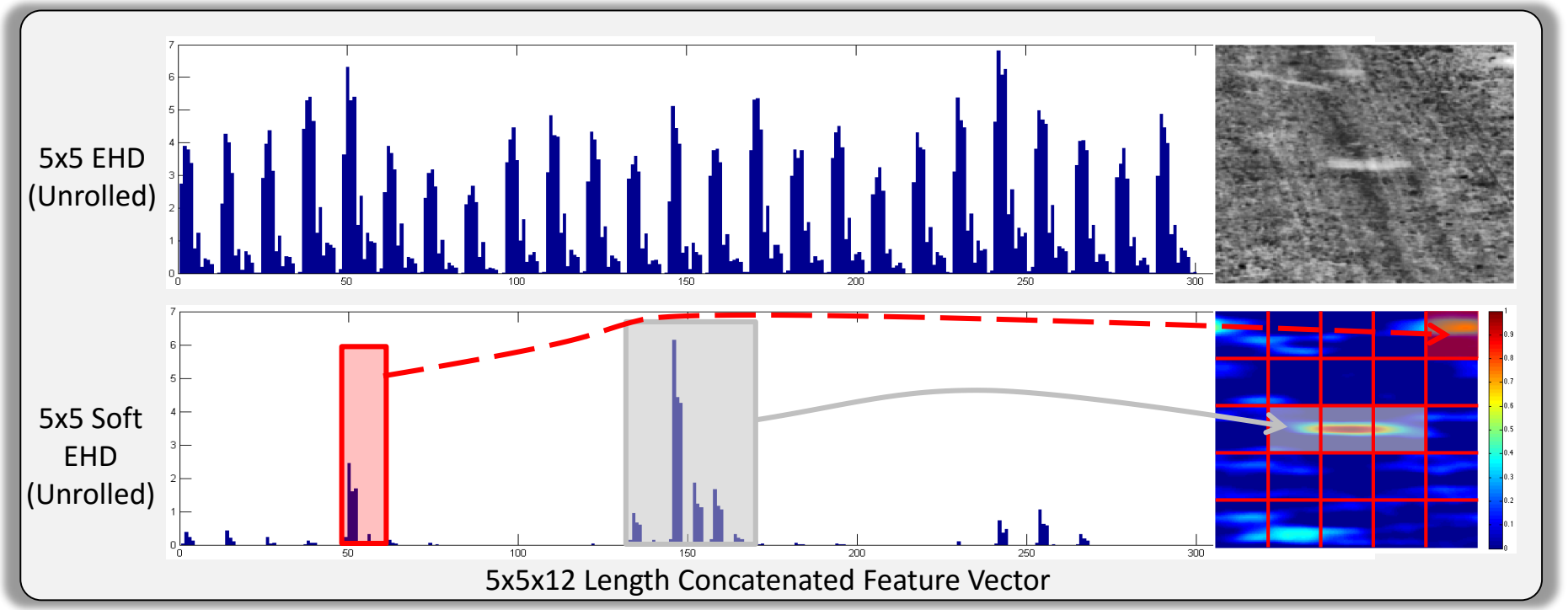
Soft features: explosive hazard detection

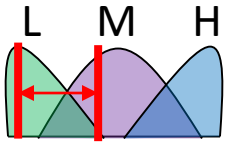




[Price, Anderson, SPIE 2013, 2014]

Soft features: explosive hazard detection

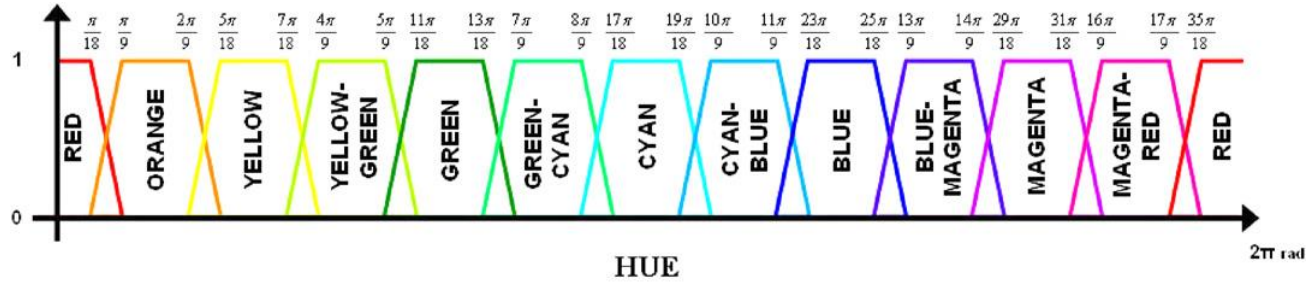


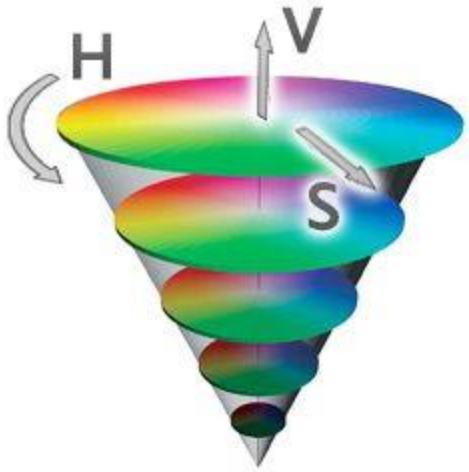


[Price, Anderson, SPIE 2013, 2014]

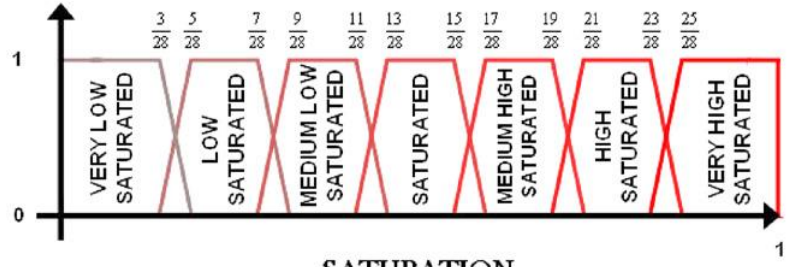
Examples of Levels

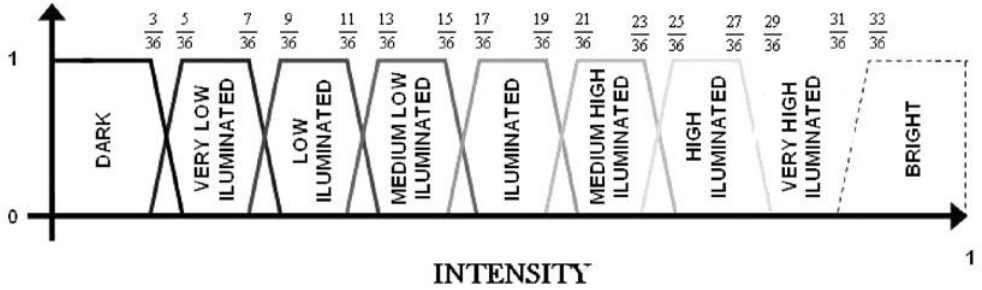
Fuzzy sets and color description

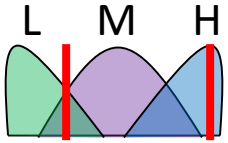




http://learn.colorotate.org/wp-content/uploads/2013/06/hsv.jpg



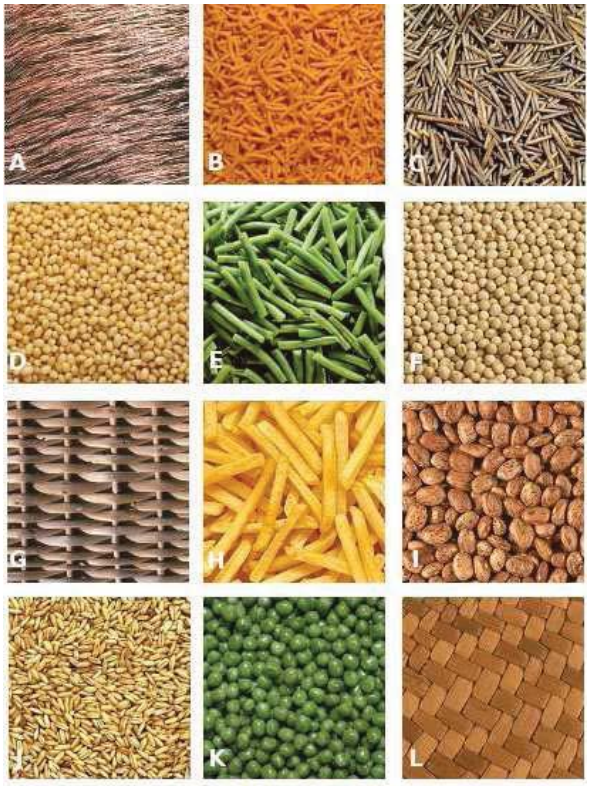




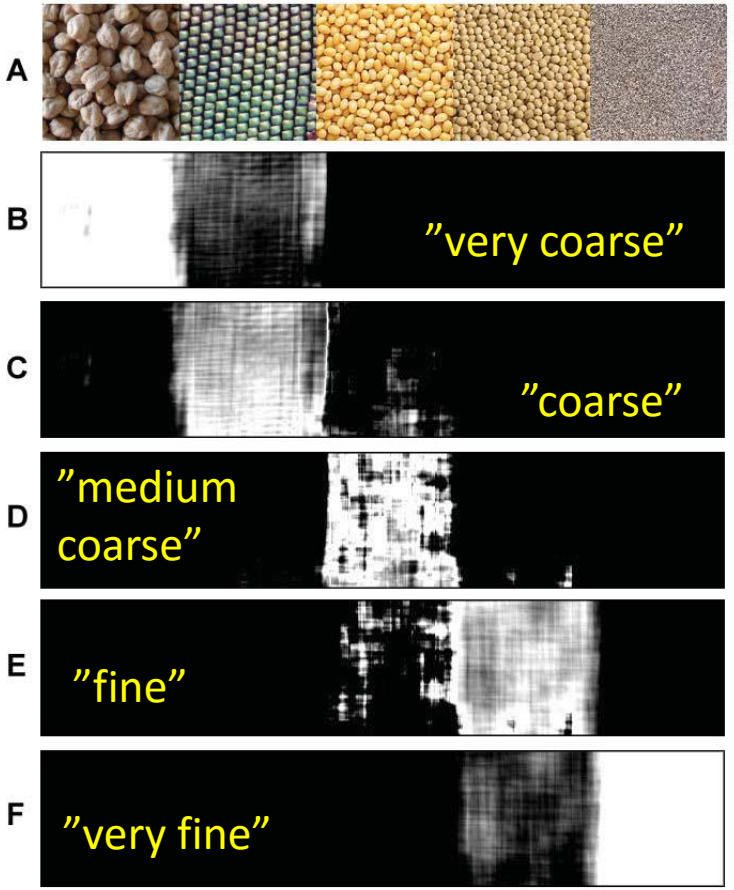
Jesus Chamorro-Martinez, “Retrieving images in fuzzy object-relational databases using dominant color descriptors”

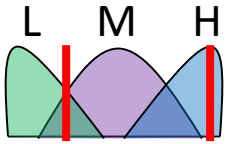
Examples of Levels
Linguistic descriptions

- Color and texture



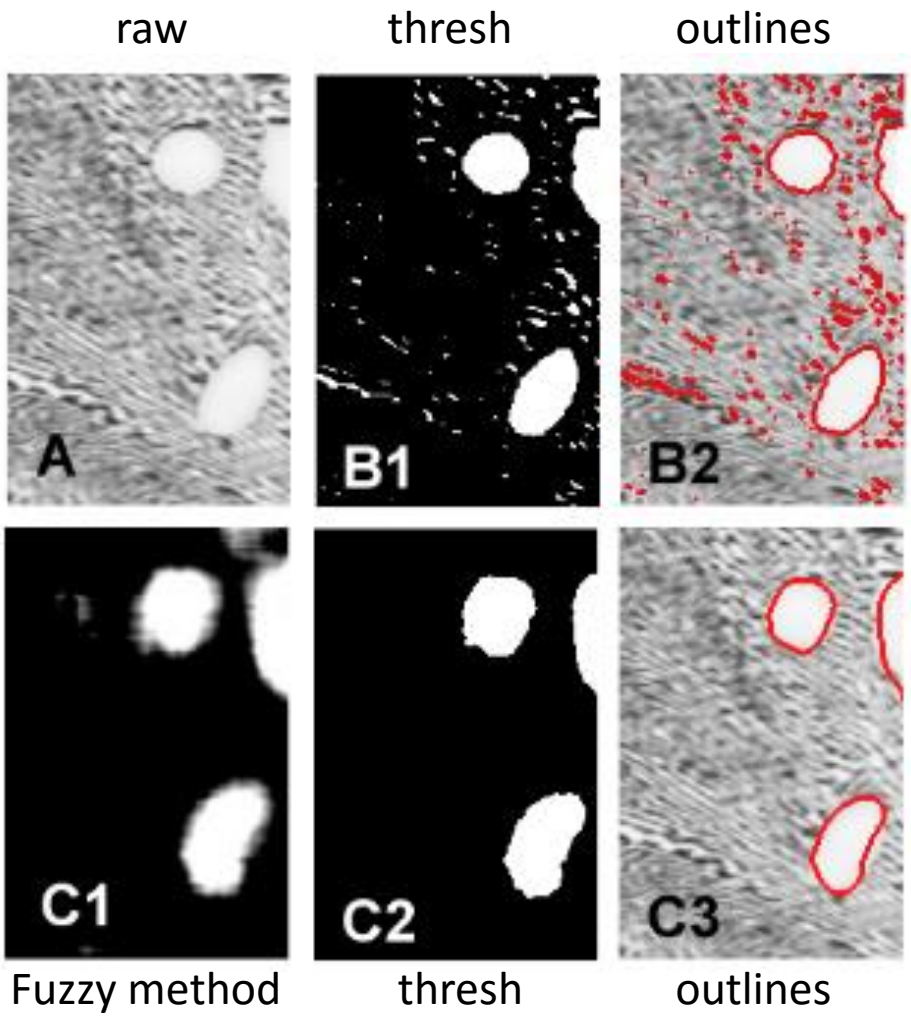
images with different degrees of fineness

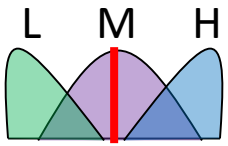




Jesus Chamorro-Martinez, "Defining A Fuzzy Partition for Coarseness Modelling in Texture Images", IFSA-EUSFLAT 2009

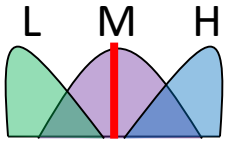
Examples of Levels
Use for segmentation





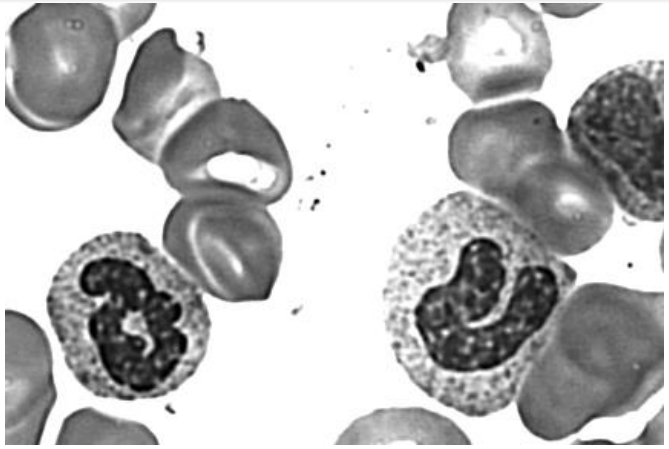
Mid-level CV

- Not going into great depth here today
- Hundreds of papers on
  - Crisp, fuzzy, probabilistic, possibilistic, neural network, swarm clustering for image segmentation
- All produce "good" results sometimes
  - **Trick is to know when**
  - Human consumption or on to next level
    - Principle of Least Commitment (PLC)

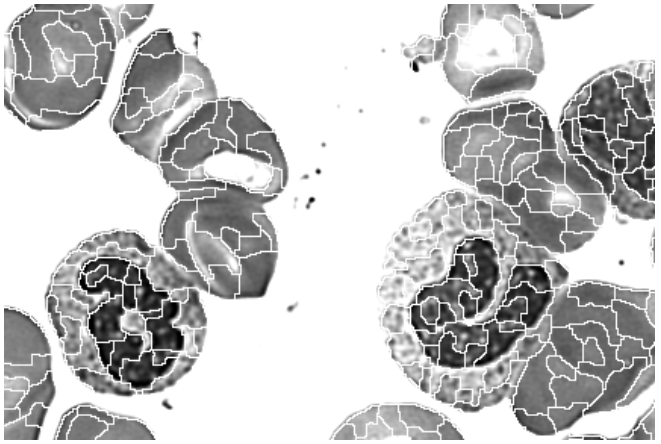


Examples of Levels

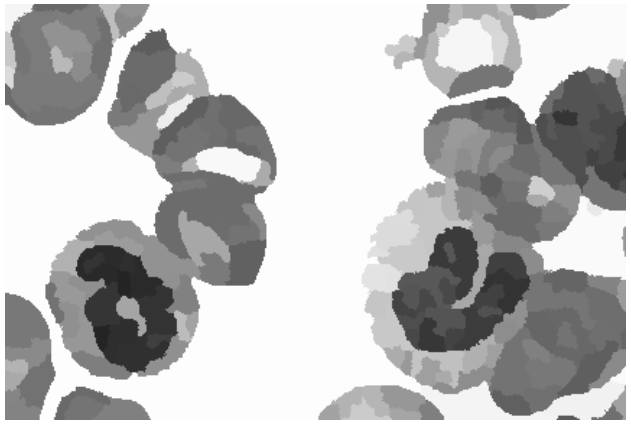
Fuzzy relaxation labeling



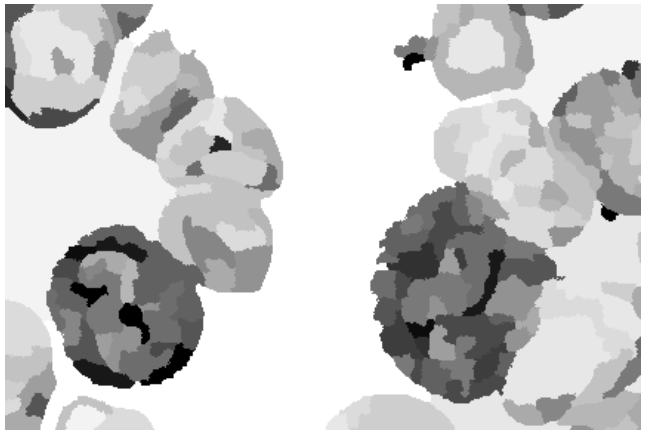
Original Image



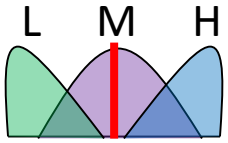
Watershed patches



Mean gray level



Patch texture



Simple Example Applied to Segmentation of Bone Marrow Cell Imagery

Fuzzy relaxation labeling

Simple Example Applied to Segmentation of Bone Marrow Cell Imagery

- **Initial Memberships of Patches Are Assigned Using Fuzzy Rules**

  Constructed with *apriori* Knowledge of Bone Marrow Cell Images (gray level and texture)

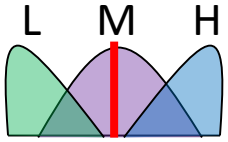
- **The rules for assigning memberships are :**

  IF Median Intensity of a Patch is *Bright*, THEN It *Is Part of* the Background

  IF Median Intensity of a Patch is *Gray* and It Is *Homogeneous*, THEN It *Is Part of* a Red Blood Cell

  IF Median Intensity of a Patch is *Gray* and It Is *Textured*, THEN It *Is Part of* the Cytoplasm

  IF Median Intensity of a Patch Is *Dark*, THEN It *Is Part of* the Nucleus

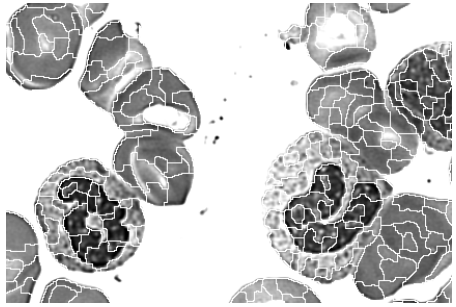


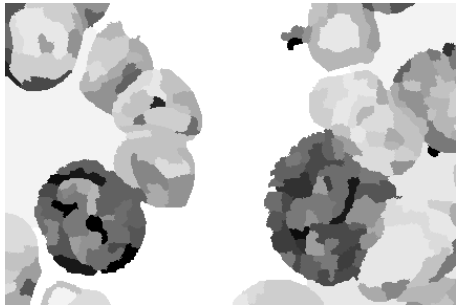
Park, J-S and Keller, J., "Fuzzy Patch Label Relaxation in Bone Marrow Cell Segmentation", *Proceedings, IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, FL, October, 1997, pp. 1133-1138

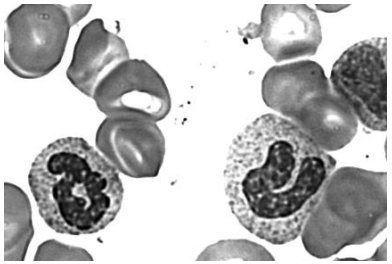
Fuzzy relaxation labeling

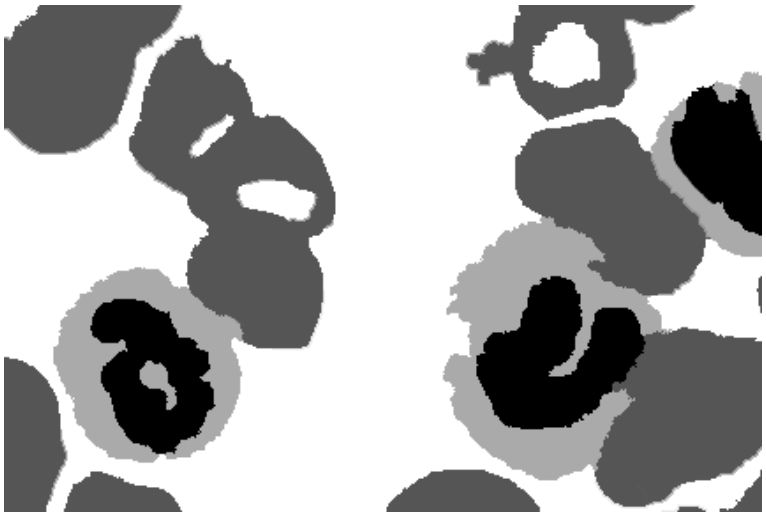
Integrate Fuzzy Rules Into the Updating Scheme
Membership Refinement Step – Determining Compatibility
Global Region Type Compatibility – c(j,k) ,i.e., Nucleus with Cytoplasm

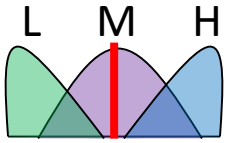
Local Support Region Compatibility:
IF a Patch is {NU, CY, RB, BG} And
Its Support Region in {NU, CY, RB, BG} is *Big* And
It is *Strongly Connected* to That Support Region,
THEN Its Neighbors Are *Compatible*
•Each Rule Is Fired to Update the Memberships for That Label







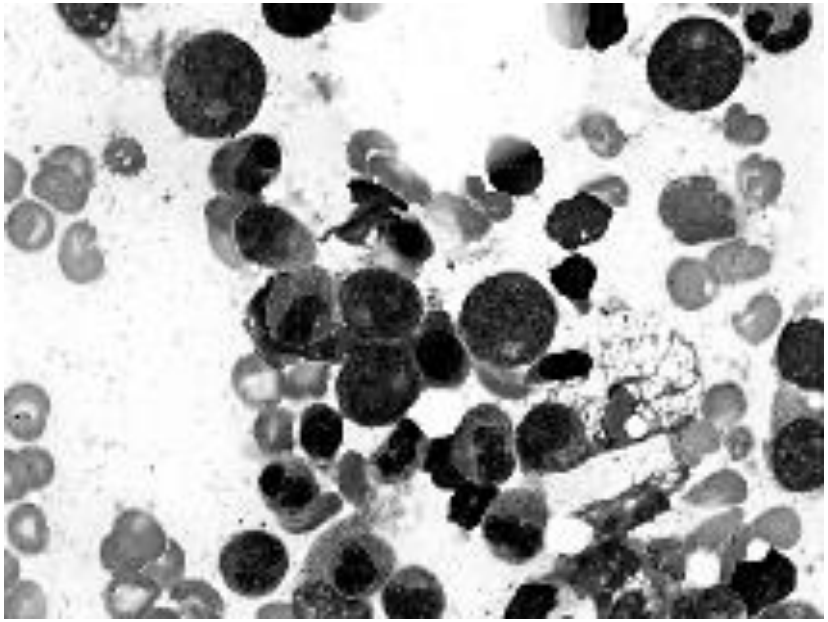




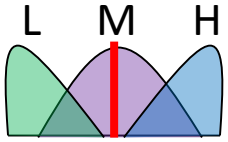
Final Segmentation

Examples of Levels
Fuzzy relaxation labeling

Many image types; here's a bone marrow slide



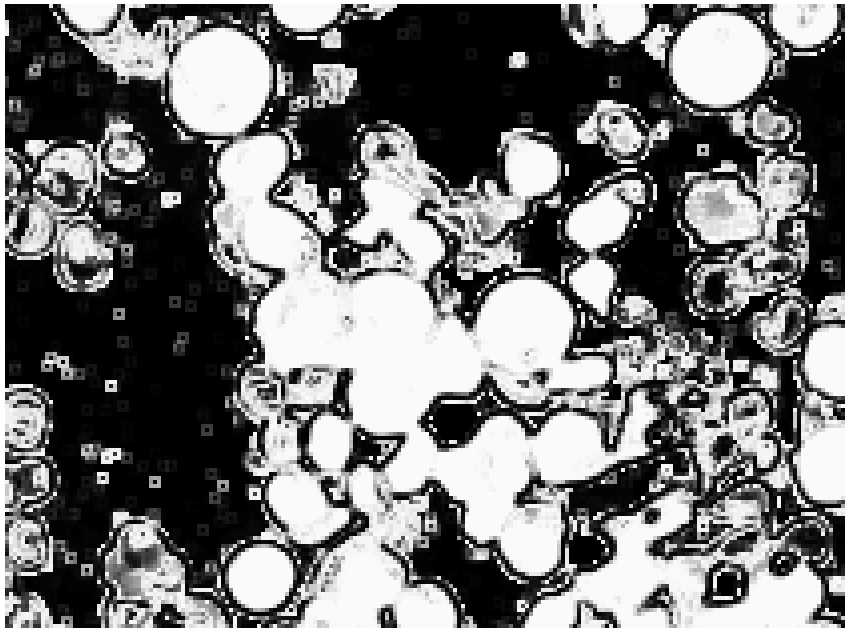
Goal is to find and count white blood cells



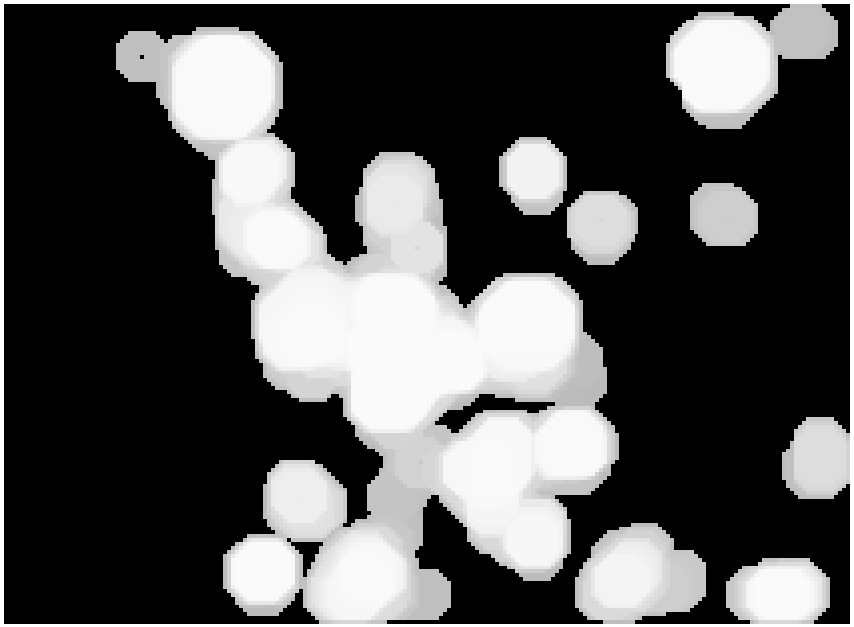
Examples of Levels
Fuzzy relaxation labeling

- Soften morphological operations

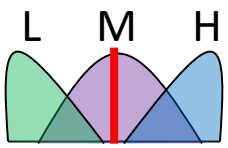
Results



Applying Fuzzy Set



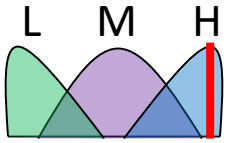
Applying Fuzzy Sets & MSE



Examples of Levels

High-level CV

- Not just pixels or image regions anymore
- High(er) level decision making
  - May or may not be motivated by humans
- Closer to AI than signal processing?
- Examples
  - Object recognition
  - Activity/behavior recognition
  - Scene/environment understanding
  - Linguistic summarization
  - 5 W's: who, what, when, where, why

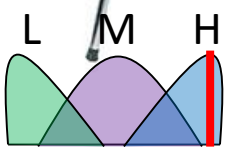


Examples of Levels

Example: what is a chair?

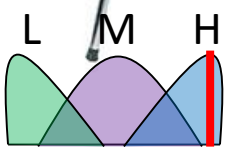
Examples of Levels

Object Different color, shape, texture, number of legs … optional armrests … some roll …



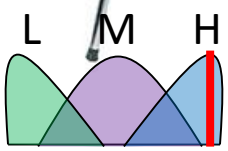
Examples of Levels

Objects do not necessarily bare any visual similarities



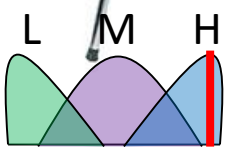
Examples of Levels

Current approaches: visual bag of words (BoW), HOG, SIFT, … SVM, mean shift, etc …



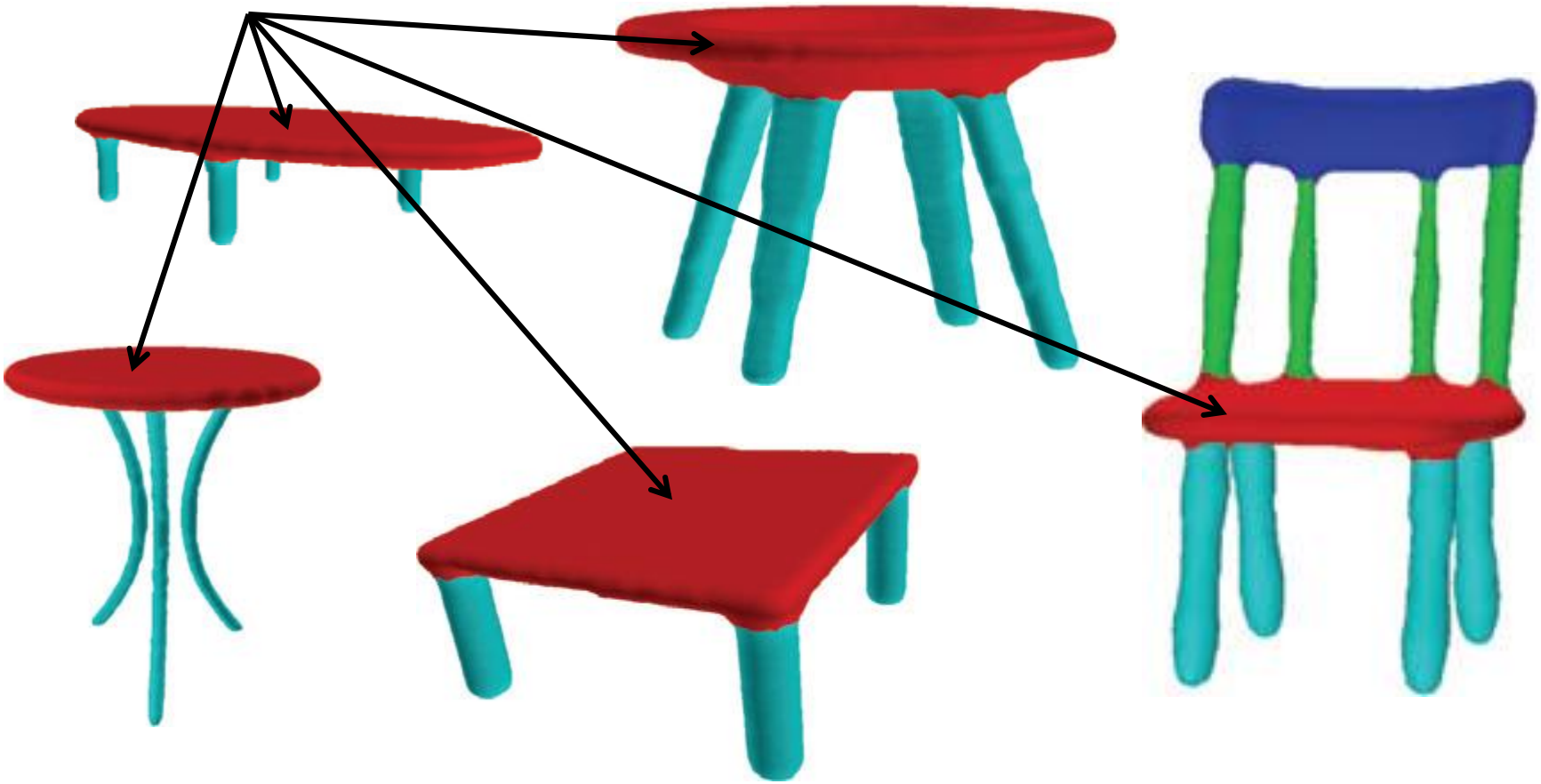
Examples of Levels

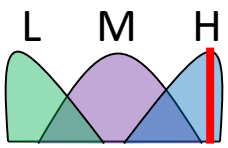
Again, what is a chair? *Something we can sit on!*



Examples of Levels

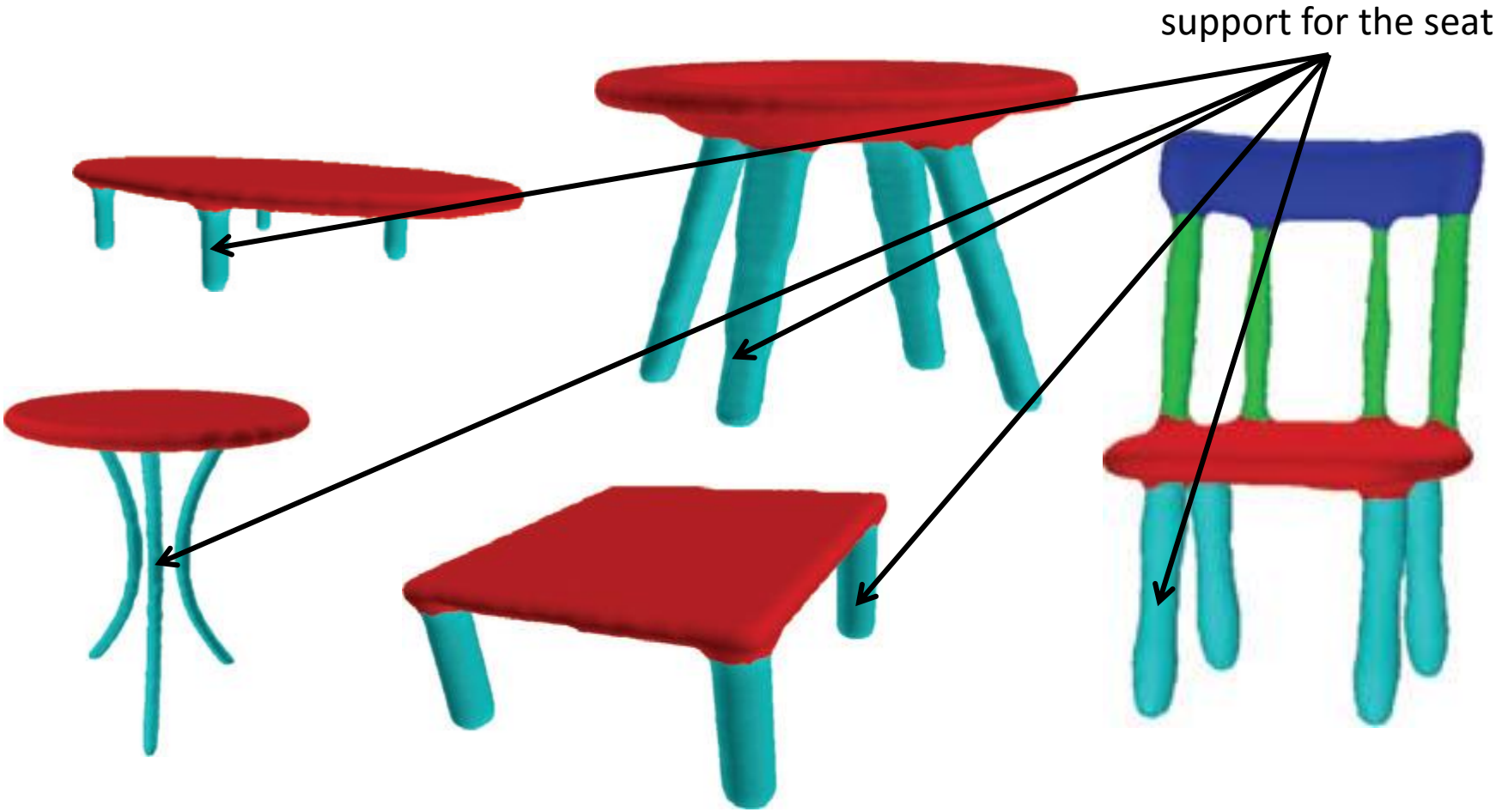
Function vs. visual appearance

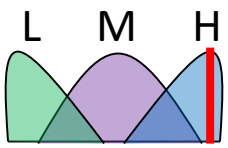




Confidence that Object is a Chair

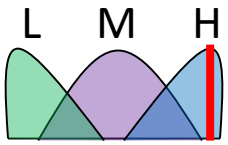
Examples of Levels
Function vs. visual appearance





Confidence that Object is a Chair

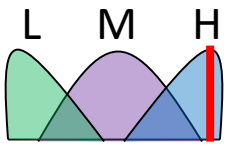
Examples of Levels
Function vs. visual appearance



Confidence that Object is a Chair

Object recognition: high level computer vision task

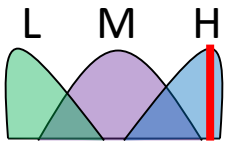
IF (flat horizontal surface near waist high)
AND (structural support below horizontal surface)
THEN (medium confidence that the object is a chair)
IF (flat horizontal surface near waist high)
AND (structural support below horizontal surface)
AND (vertical surface for back support)
THEN (high confidence that the object is a chair)



Object recognition: high level computer vision task

IF (flat horizontal surface near waist high)
AND (structural support below horizontal surface)
THEN (medium confidence that the object is a chair)
IF (flat horizontal surface near waist high)
AND (structural support below horizontal surface)
AND (vertical surface for back support)
THEN (high confidence that the object is a chair)

What about object size
What about object position (context) in room
What about object interaction with humans (how its used)



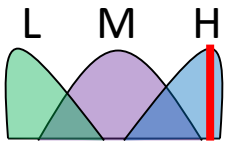
Examples of Levels

Object recognition: high level computer vision task

IF (flat horizontal surface near waist high)
AND (structural support below horizontal surface)
THEN (medium confidence that the object is a chair)
IF (flat horizontal surface near waist high)
AND (structural support below horizontal surface)
AND (vertical surface for back support)
THEN (high confidence that the object is a chair)

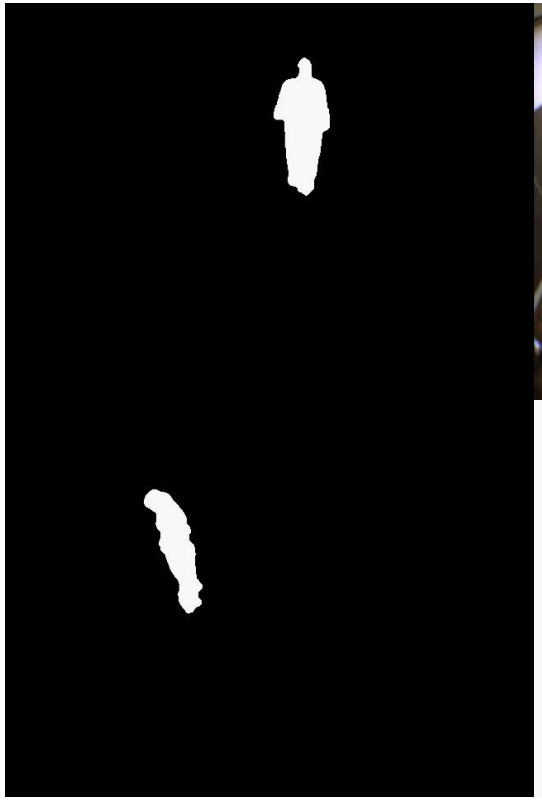
What about object size
What about object position (context) in room
What about object interaction with humans (how its used)

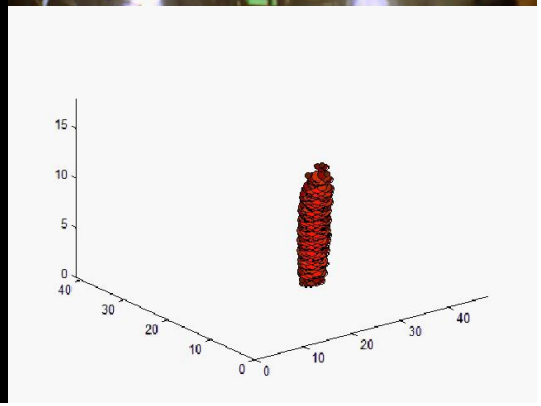
Can we learn the descriptions?

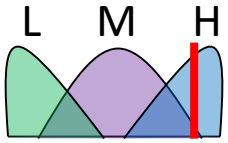


Aging-in-place and fall detection

- **Application**: eldercare
- **Task (short)**: people detection and activity recognition (e.g., falls)
- **Task (long)**: Daily, weekly and monthly trends (and deviations)
- **Consideration**: should we operate in two or three dimensions?
  - 2D is view dependent
- **Constraints**: Preserve privacy? (silhouettes and voxel person)

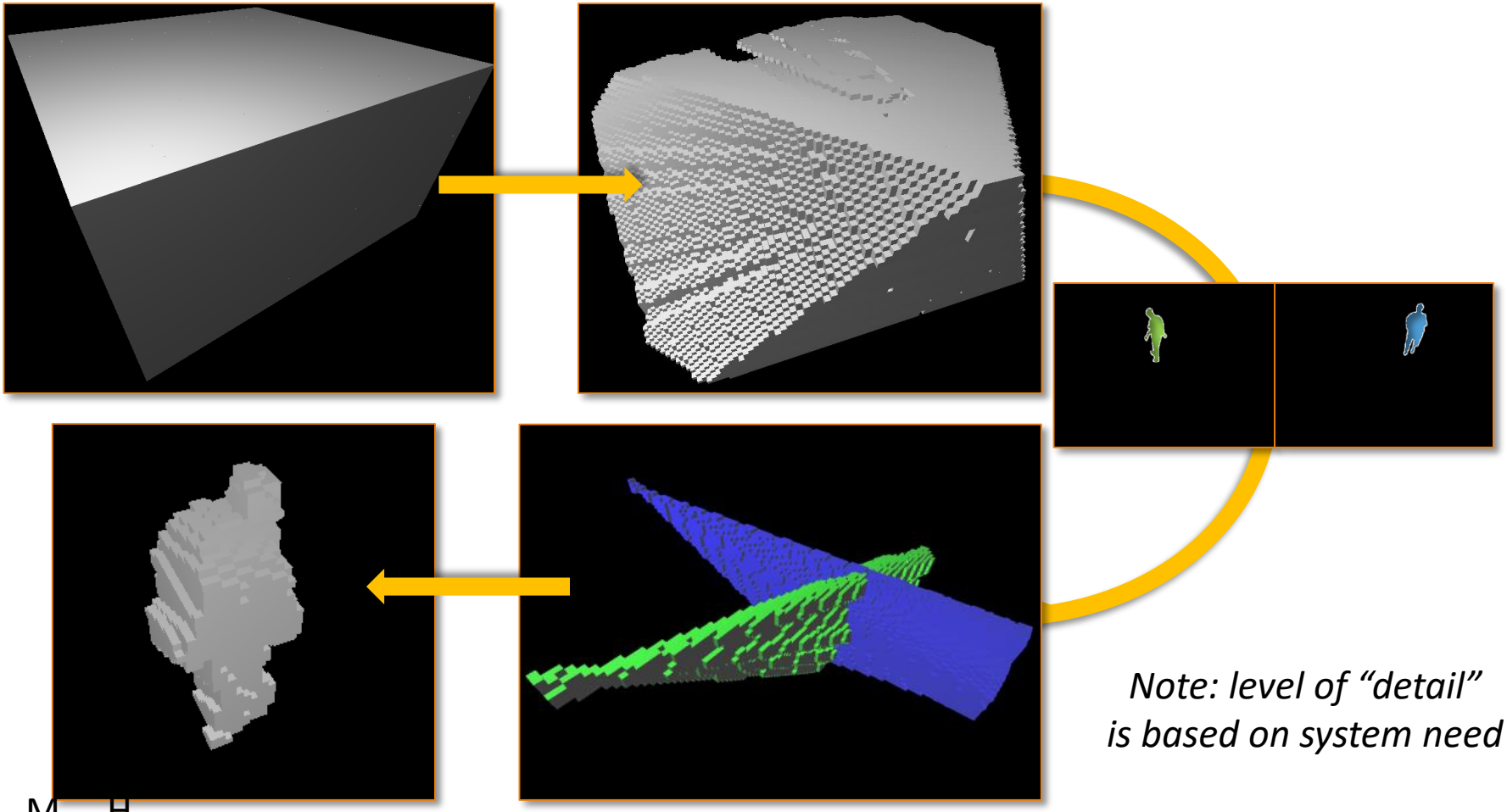




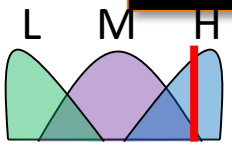


[Anderson, Keller, et al. TFS, CVIU, FUZZ-IEEE, …]

Examples of Levels
Volume element (voxel) space

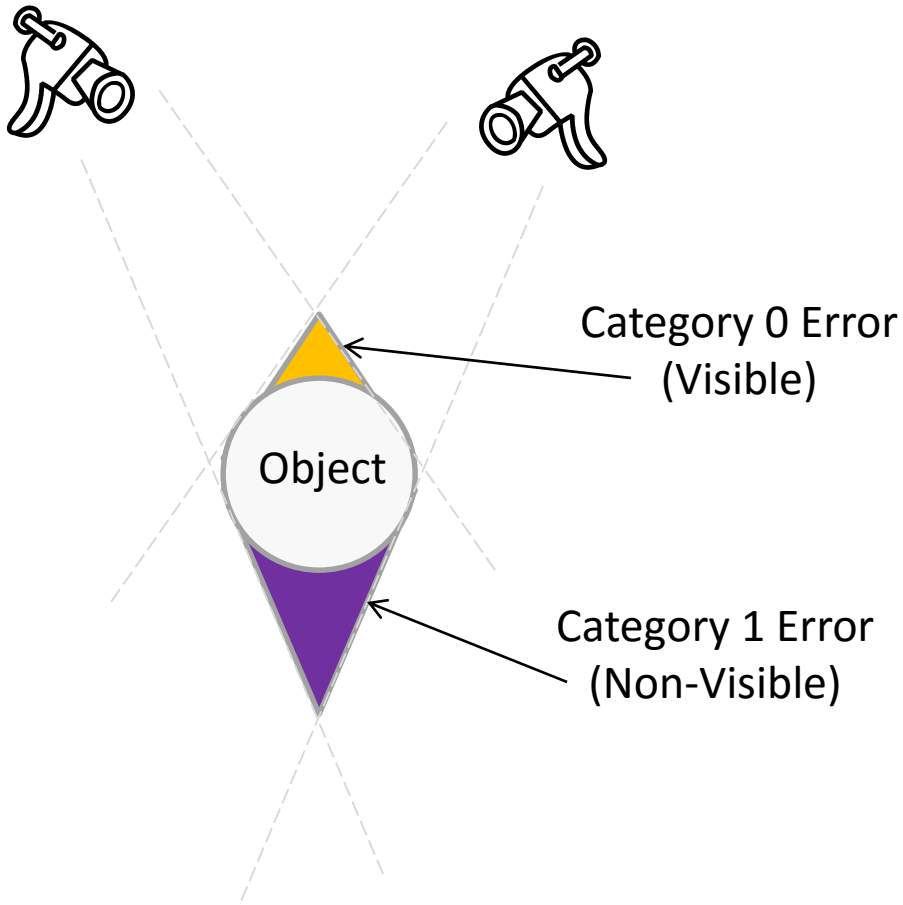


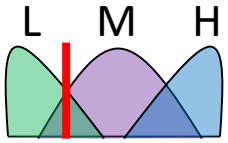
Note: level of "detail" is based on system need



[Anderson, Keller, Luke, et al. IFSA, 2009]

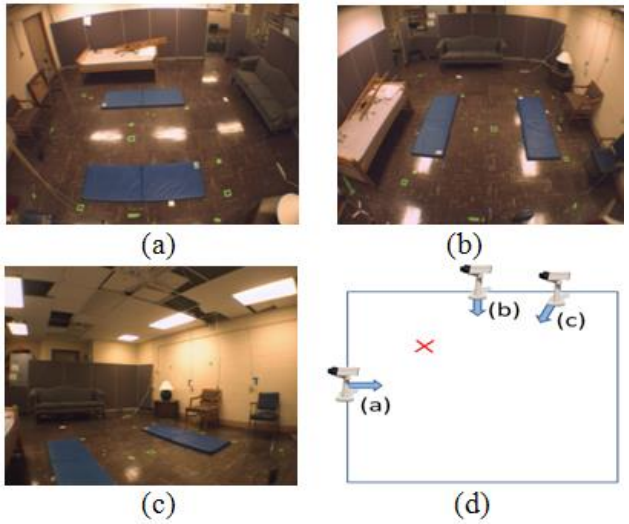
Back-projection construction error

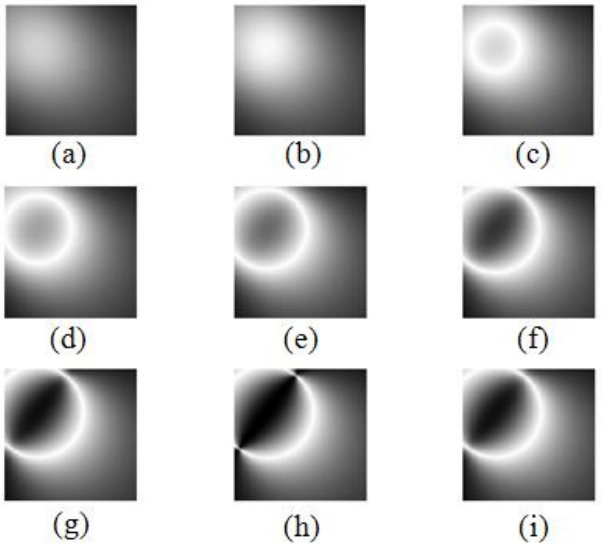




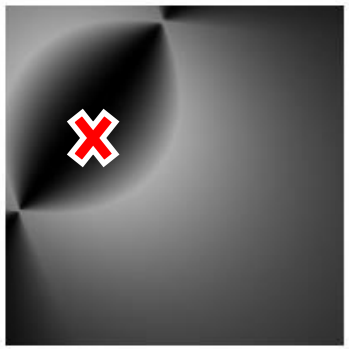
[Anderson, Keller, Luke, et al. IFSA, 2009]

Examples of Levels
Quality of monitoring

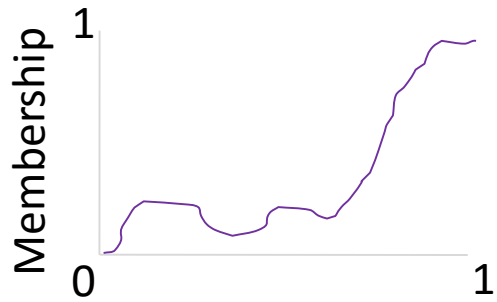




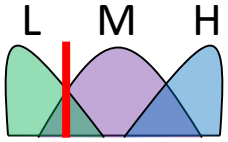
$$Q_{(i,j,l)} = 1 - \max_{\substack{C_g, C_h \\ g \neq h}} \left| (v_{(i,j,l)} - C_g)^T (v_{(i,j,l)} - C_h) \right|$$



$$Q_{(i,j)} = \min_k Q_{(i,j,k)}$$



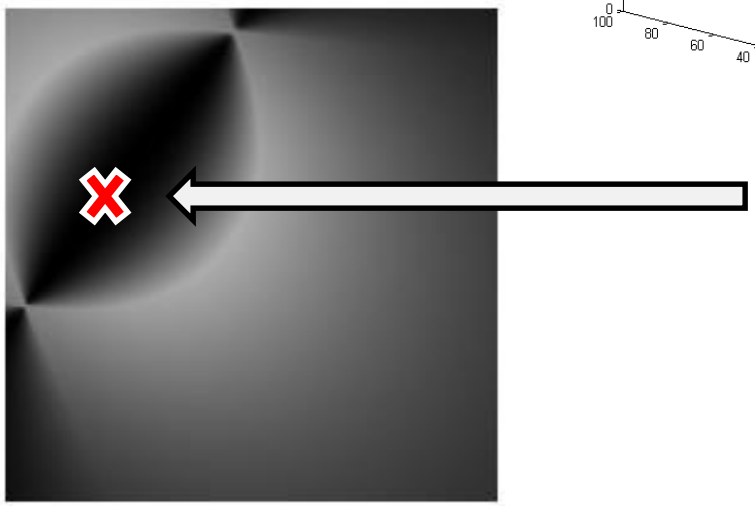
Histogram of object voxel memberships

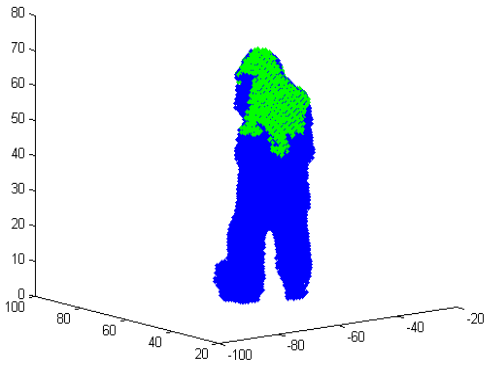


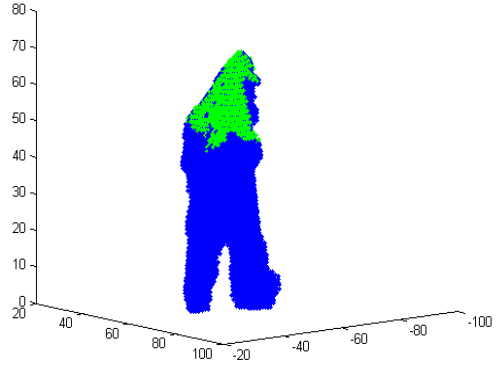
[Anderson, Keller, Luke, et al. IFSA, 2009]

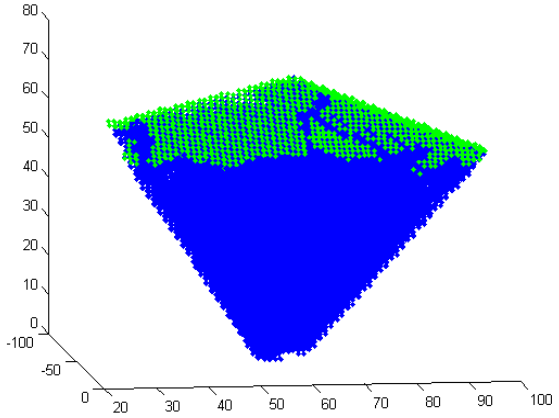
Examples of Levels

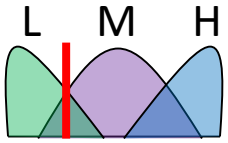
Some spots are hard to reconstruct











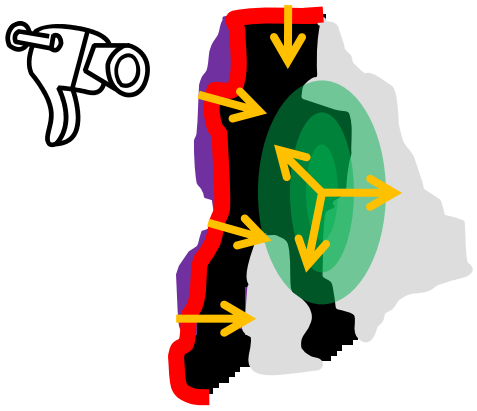
[Anderson, Keller, Luke, et al. IFSA, 2009]

Examples of Levels
Fuzzy voxel object

1. Initialize all $m_{(i,j,l)}$ to 0
2. $V' = V_t'$
3. $S' = S_t$
4. while $|V'| < 0$
   a. for each $\vec{v}'_{(i,j,l)} \in V'$
      i. add one to $m_{(i,j,l)}$
   b. end
   c. $V' = V' - S'$
   d. $S' = S' \oplus K$
5. end.

$$m'_{(i,j,l)} = 1 - \left(\frac{m_{(i,j,l)}}{\max_{m_{(i,j,l)}} m_{(i,j,l)}}\right)$$

Distance to what is observable

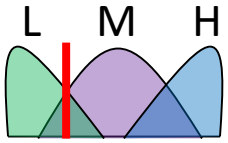


1. Initialize all $e_{(i,j,l)}$ to 0
2. $V' = V_t'$
3. while $|V'| < 0$
   a. for each $\vec{v}'_{(i,j,l)} \in V'$
      i. add one to $e_{(i,j,l)}$
   b. end
   c. $V' = V' \ominus K$
4. end.

$$e'_{(i,j,l)} = (1-\beta)\frac{e_{(i,j,l)}}{\max_{e_{(i,j,l)}} e_{(i,j,l)}} + \beta$$

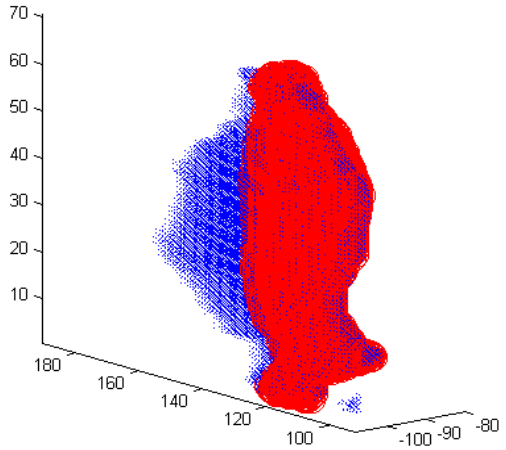
Density

$$\mu_{(i,j,l)} = m'_{(i,j,l)} \wedge e'_{(i,j,l)}$$

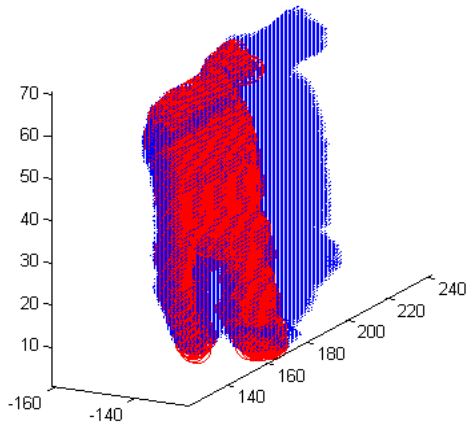


[Anderson, Keller, Luke, et al. IFSA, 2009]

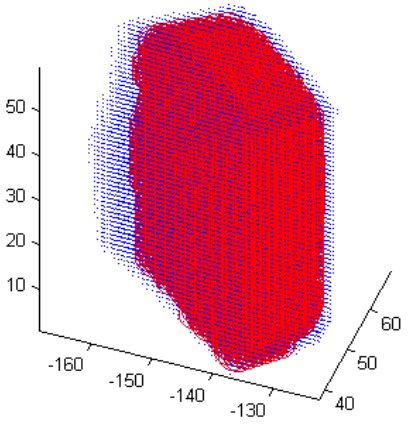
Examples of Levels
Fuzzy voxel object



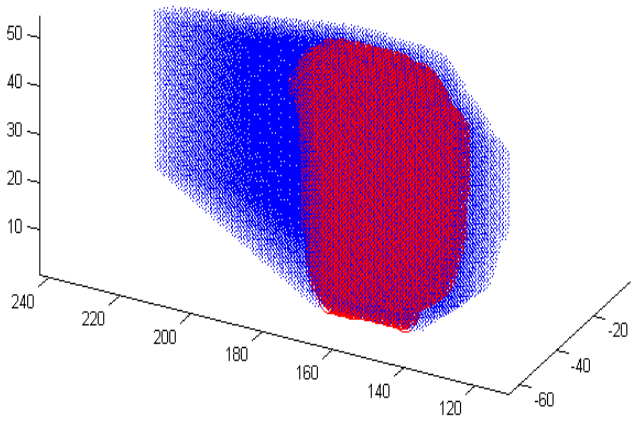
Human



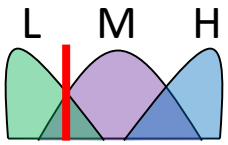
Human



File Cabinet



File Cabinet

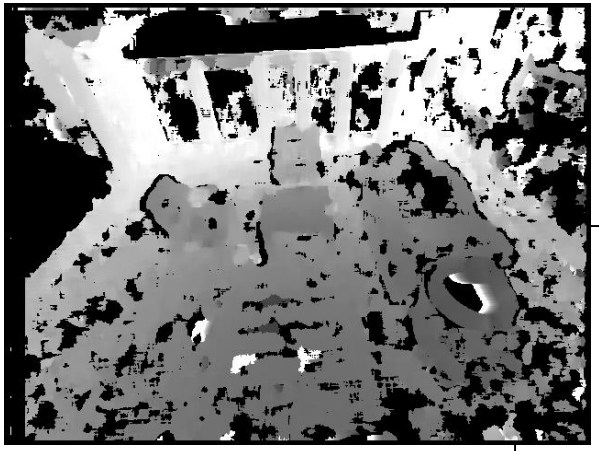


[Anderson, Keller, Luke, et al. IFSA, 2009]

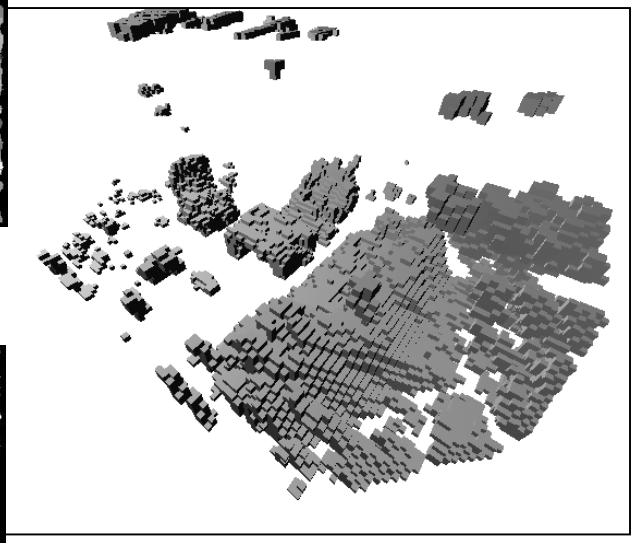
Examples of Levels

Multiple stereo cameras

Camera 1

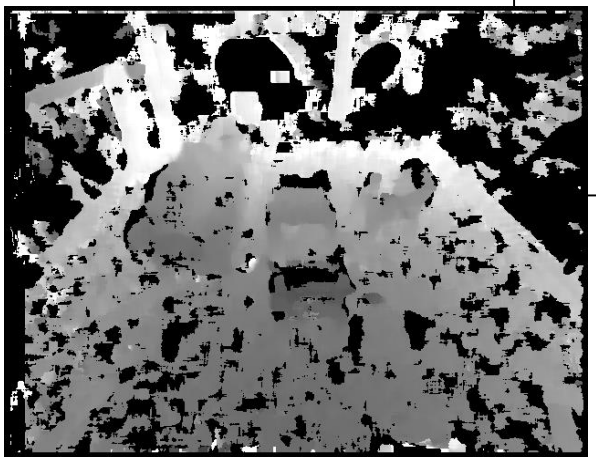


Depth Map 1

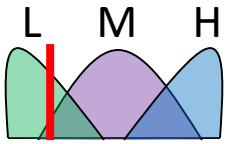


3D stereo space

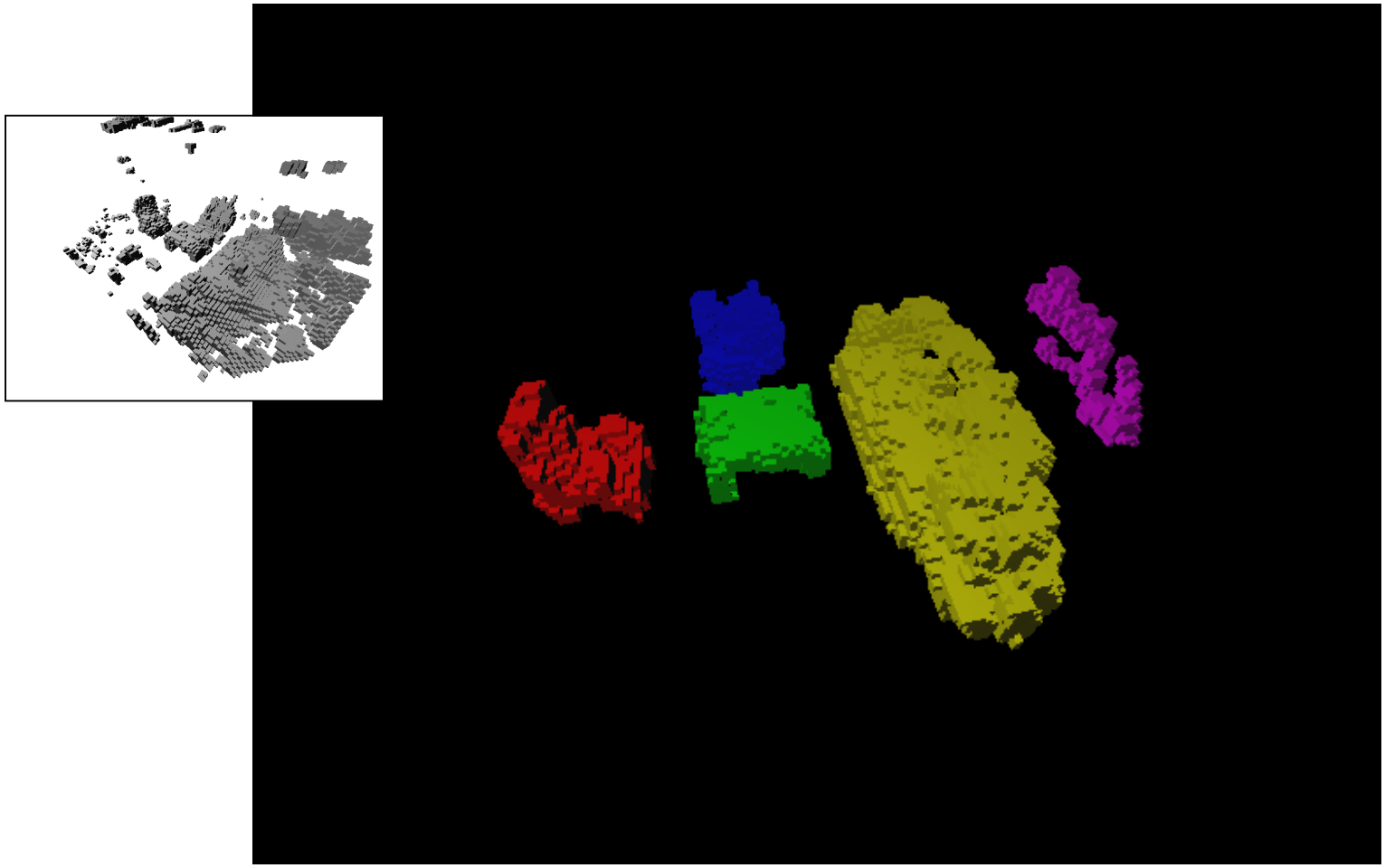
Camera 2

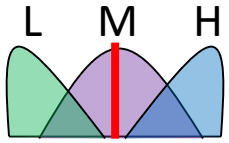


Depth Map 2



Examples of Levels
Automatic segmentation

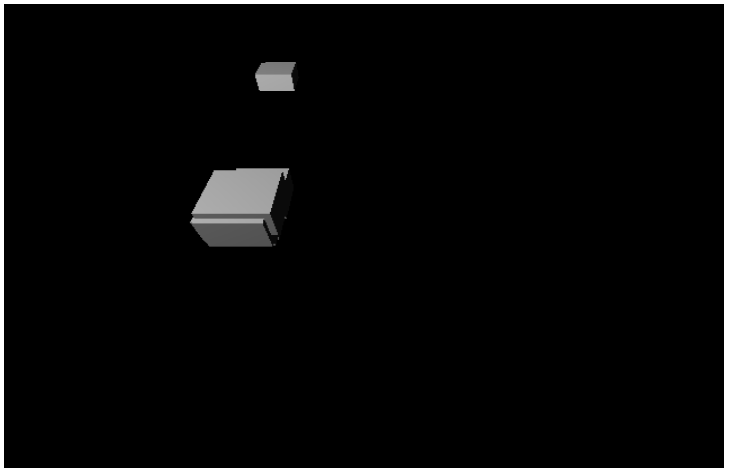




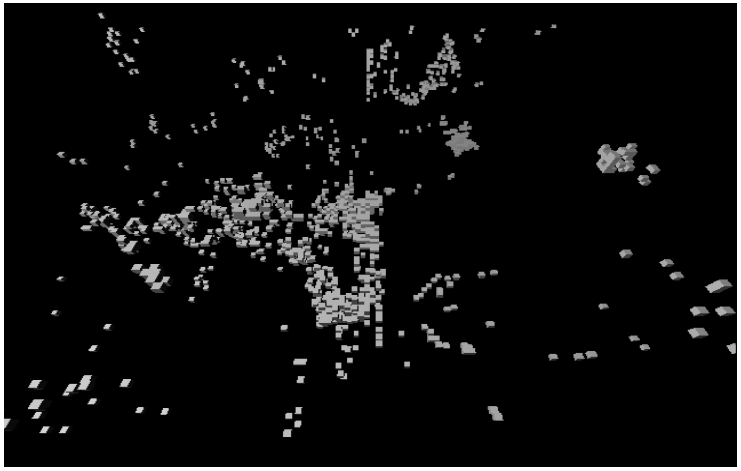
[Anderson, Keller, Luke, et al. FUZZ-IEEE, 2010]

Examples of Levels
Weak classifiers

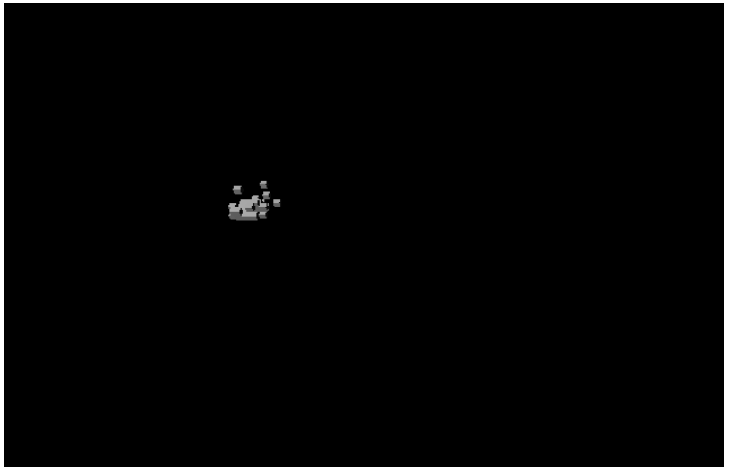
Find skin



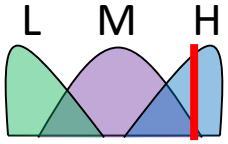
Find heads



Find skin

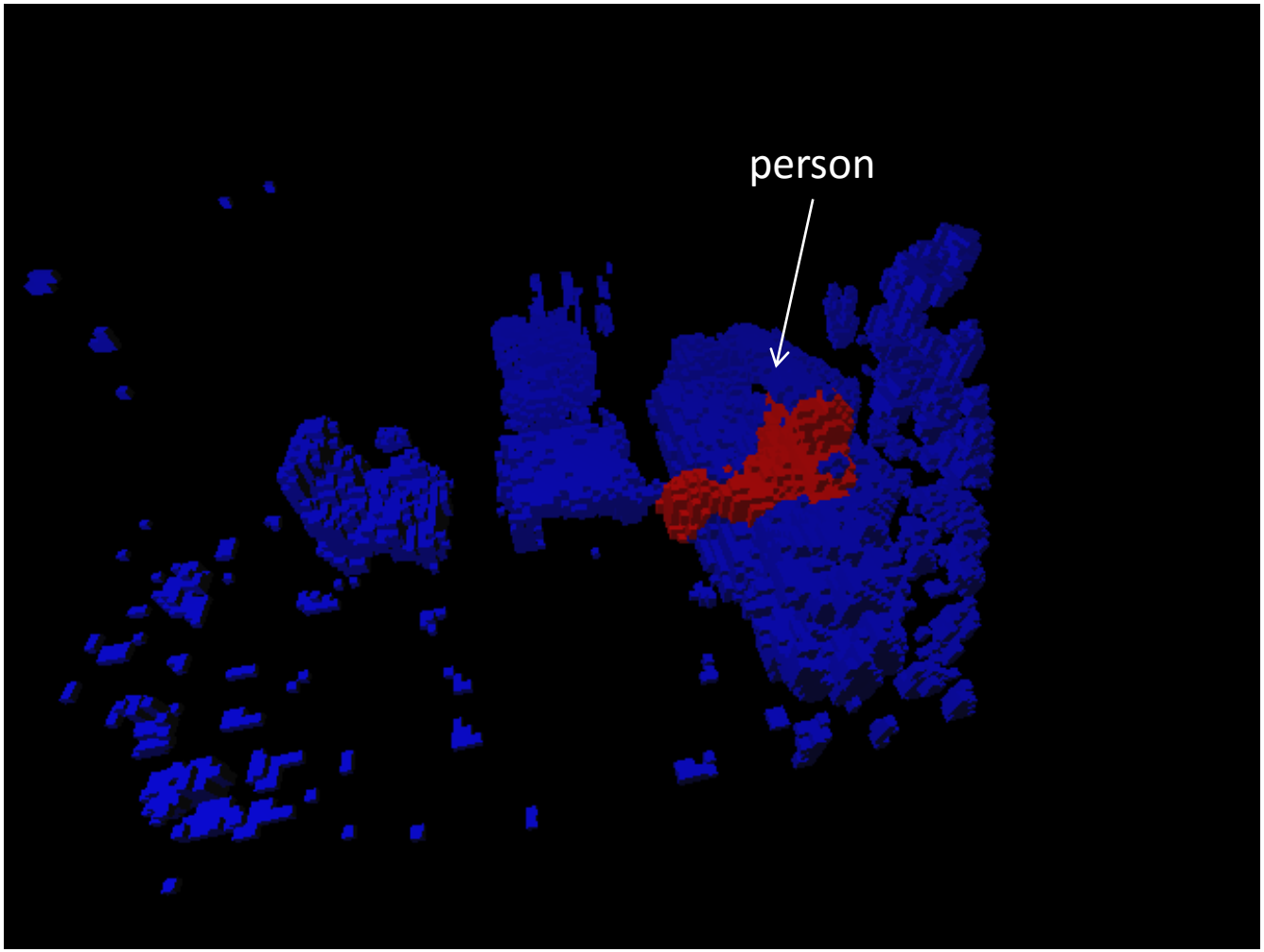


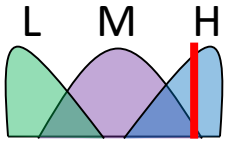
skin + head



[Robert Luke, PhD]

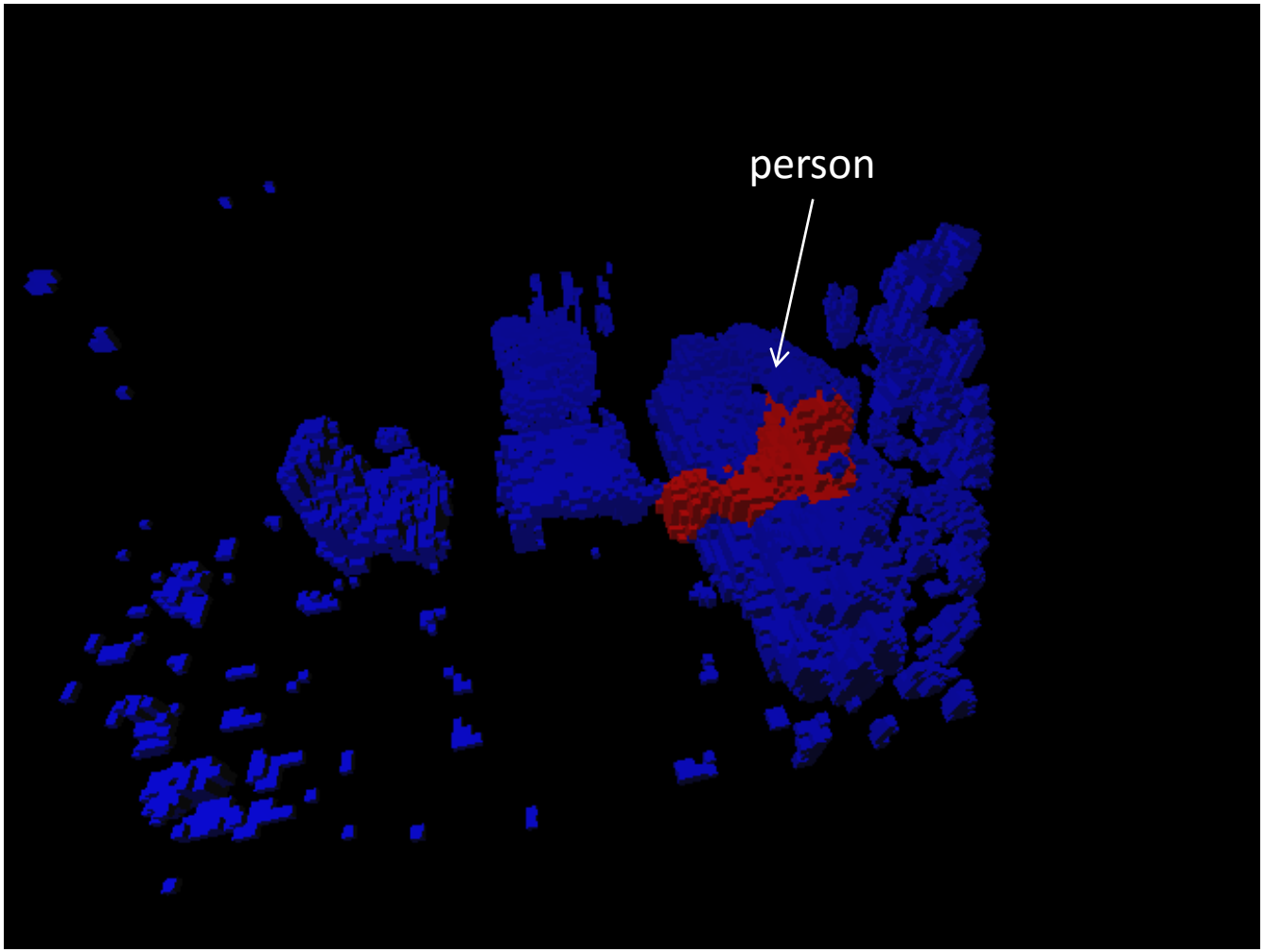
Change detection in voxel space

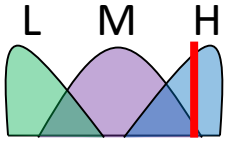




[Robert Luke, PhD]

Change detection in voxel space

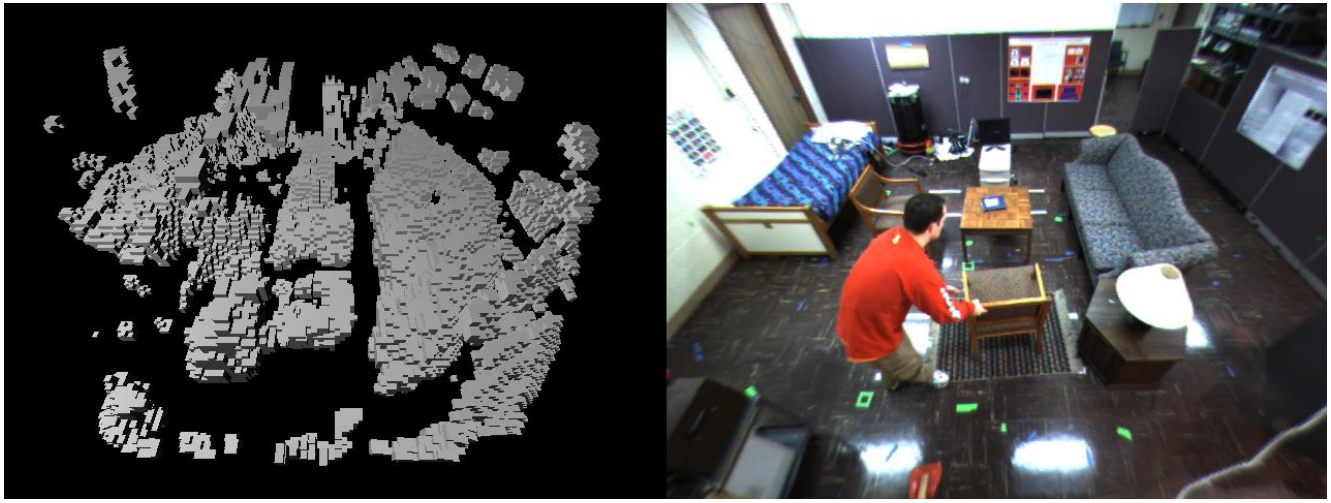


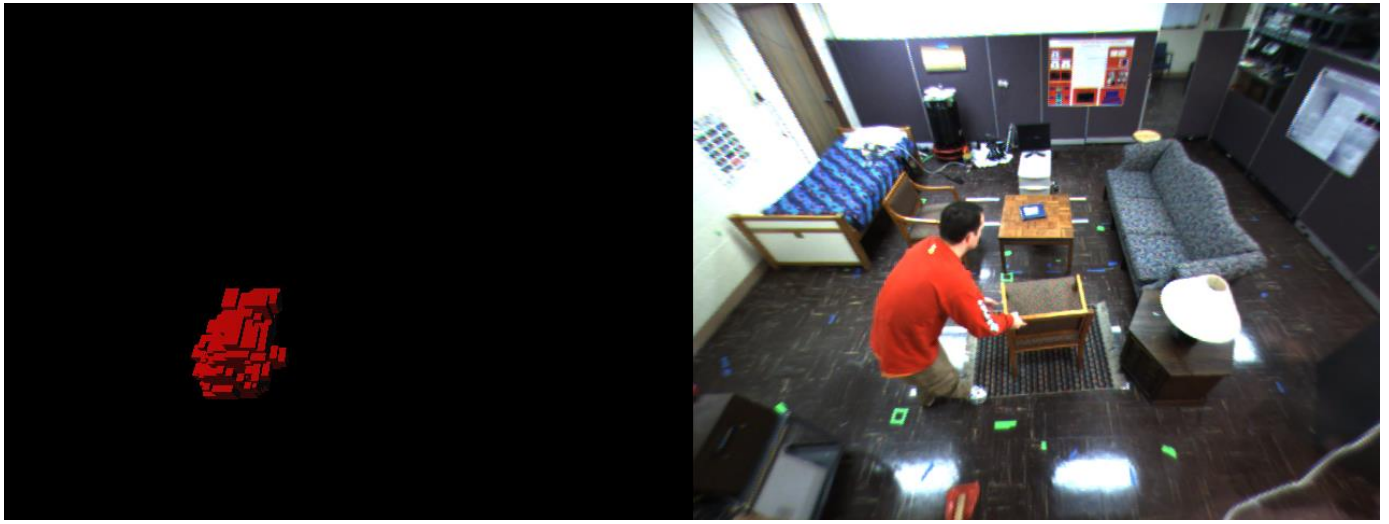


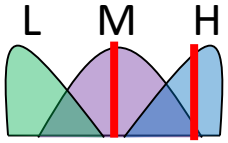
Would fuzzy modeling help here? Maybe not

[Robert Luke, PhD]

Model background and find Bob



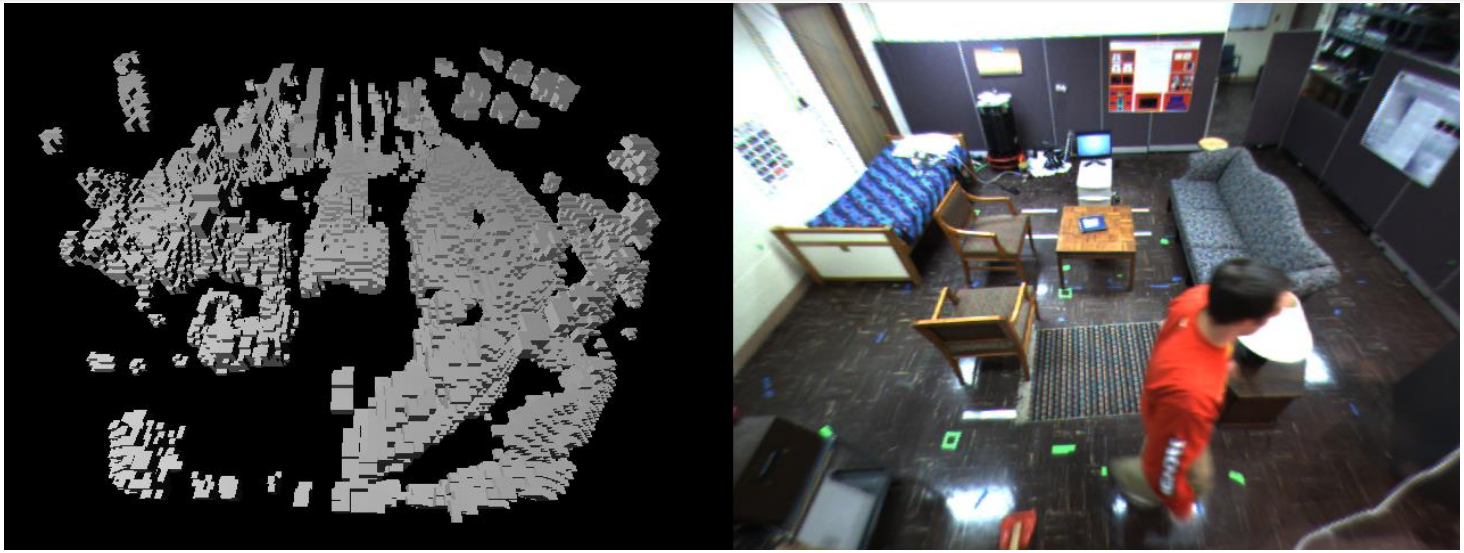


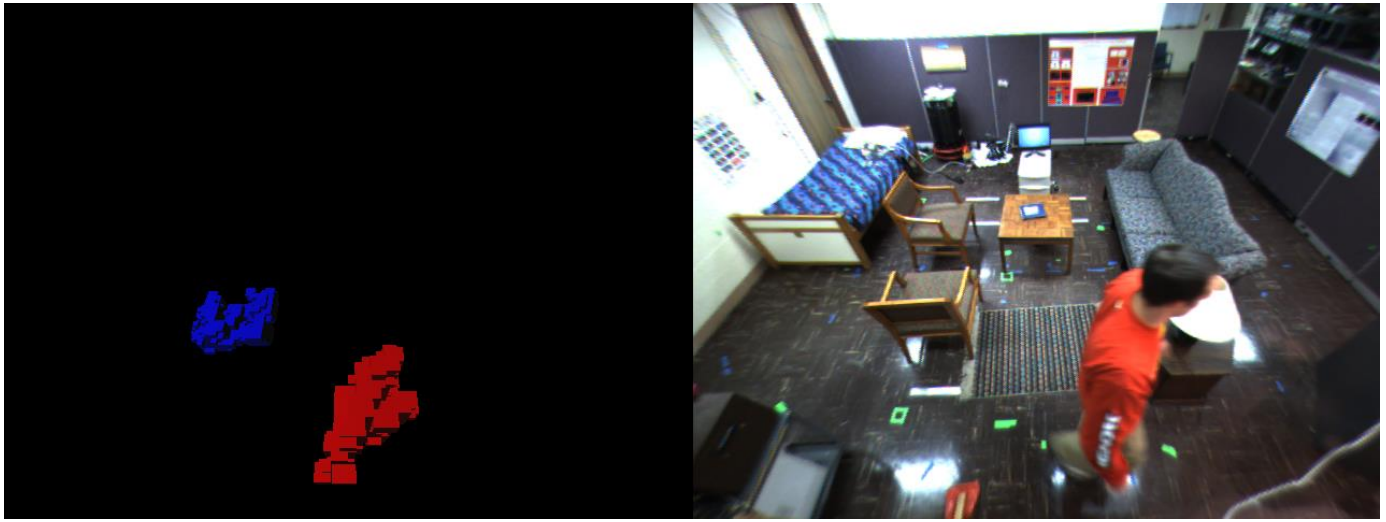


[Robert Luke, PhD]

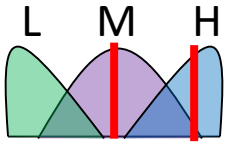
Examples of Levels

Bob moves the chair





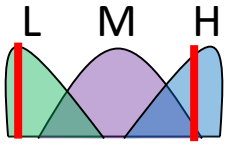
Moved object is not a human; absorb into background

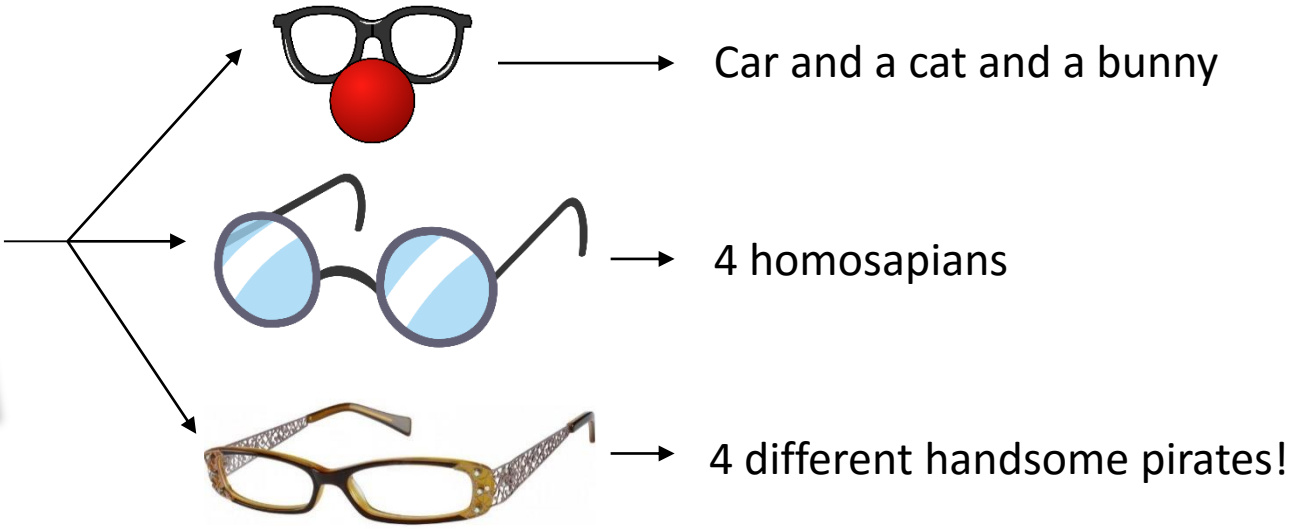


[Robert Luke, PhD]

Feature learning

- Classical approach
  - "**Hand crafted**" features/transforms (designed by people)
    - Fourier-based, Wavelets, Shearlets, Curvelets, etc.
    - Histogram of oriented gradients, LBPs, etc.
- New(er) approach
  - Let the **machine learn it** versus the human





Real world

Transformation

What we "see"

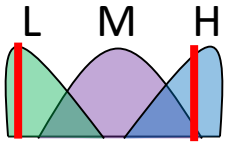
Car and a cat and a bunny

4 homosapians

4 different handsome pirates!

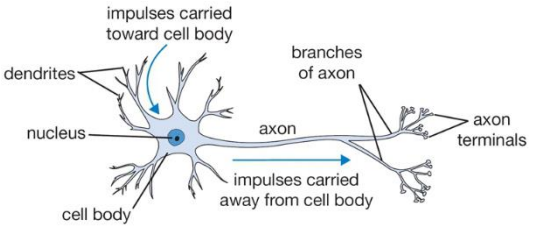
Feature learning

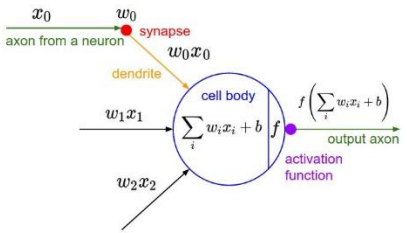
- Different approaches
  - No one **owns** feature learning
  - Few examples
    - Deep learning
      - Geoffrey Hinton and deep belief networks (2006 ish?)
      - Morphological Shared weight Neural Network (MSNN)
        - Gader et al., 1999-ish
      - Convolutional neural networks (CNNs)
    - Evolutionary approaches
      - ECO (Lillywhite 2012)
      - iECO (Stanton Price, Anderson, et al., SSCI 2014, SPIE 2015 and 2016)



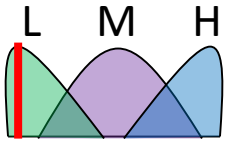
CNNs: context

- Biologically inspired variant of MLP
- Hubel and Wiesel (1968) - cat's visual cortex
  - Complex cell arrangement in visual cortex
  - Cells are sensitive to *small* sub-regions in visual cortex
    - Referred to as a receptive field
  - Sub-regions are *tiled* cover full visual field
    - Cells act as local filters
    - Good at exploiting strong spatially local correlation
- Two basic types of cells were identified
  - Simple cells
    - Respond to edge-like patterns in their receptive field
  - Complex cells
    - Have larger receptive fields and locally invariant to position of the pattern



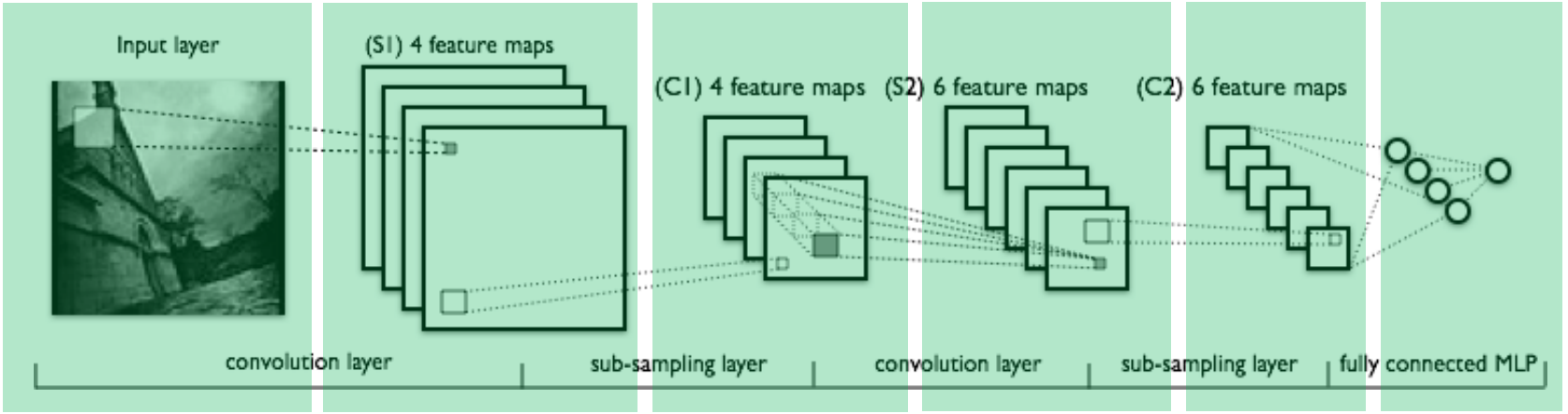


http://cs231n.github.io/neural-networks-1/



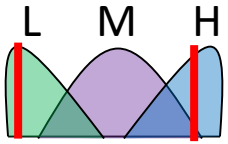
CNNs: vanilla architecture

- Multiple layers
  - Convolution (filters and feature maps)
  - Sub-sampling (sum/max pooling operations)
  - Normalization
  - Classification (MLP, not covering today)
  - Others
- FYI, many good tutorials out there! (very hot topic)



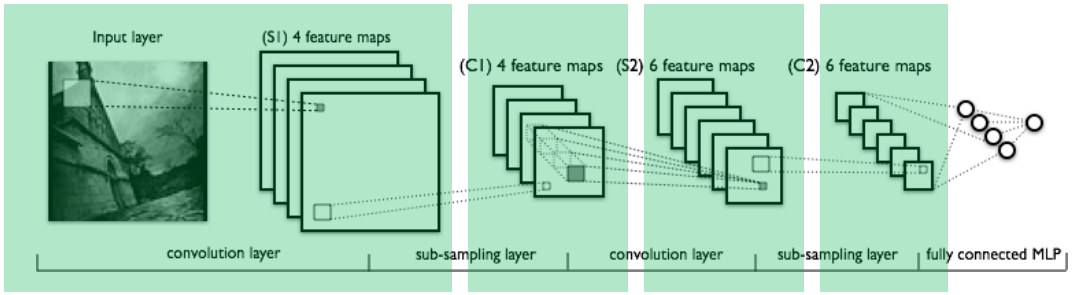
http://deeplearning.net/tutorial/lenet.html

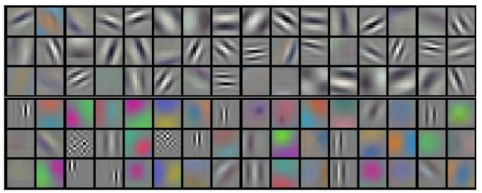
Beyond articles, giving tutorials and code references



CNNs: basic concepts

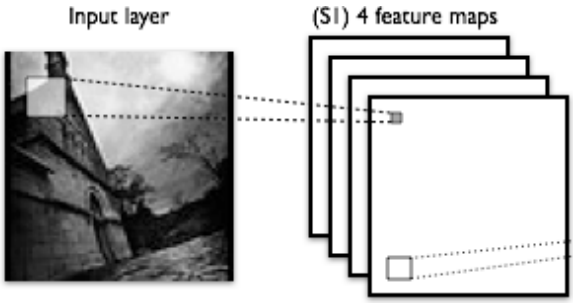
- CNNs exploit spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers



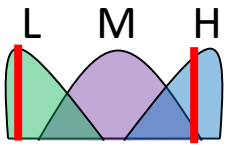


96 [11x11x3] filters
http://cs231n.github.io/convolutional-networks/

- Shared weights: each filter replicated across entire visual field (results in our "feature maps")

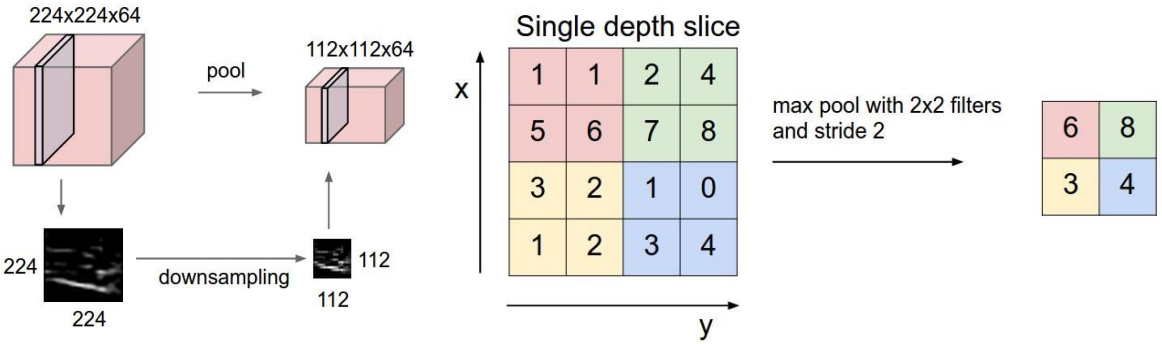


Notes:
- Features can be detected regardless of their exact position in the visual field
- Also, a lot more efficient!
- Can lead to more generalizable solutions



CNNs: basic concepts

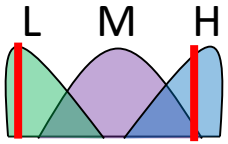
- Max pooling
  - Non-linear down sampling
  - Partitions image into set of non-overlapping rectangles
  - In each region, output is the maximum value



- Why?

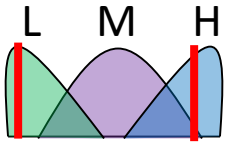
http://cs231n.github.io/convolutional-networks/

  - Eliminate non-maximal values - reduce computation in upper layers
  - More translation robust



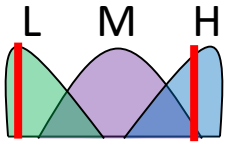
CNNs: basic concepts

- Rectified linear unit (ReLU) [non-linear gating]
  - f(x) = max(0,x)
    - Activation is simply thresholded at zero
  - Comments
    - Was found to accelerate (factor of 6 in Krizhevsky et al.) convergence of stochastic gradient descent compared to sigmoid/tanh functions
    - Simple to implement (versus alternative solutions, e.g., tanh/sigmoid)
- *Leaky* ReLU
  - Fix the "dying ReLU" problem
    - f(x)=1(x<0)(ax)+1(x>=0)(x) [a is a small constant]
- Other types of units as well in the literature

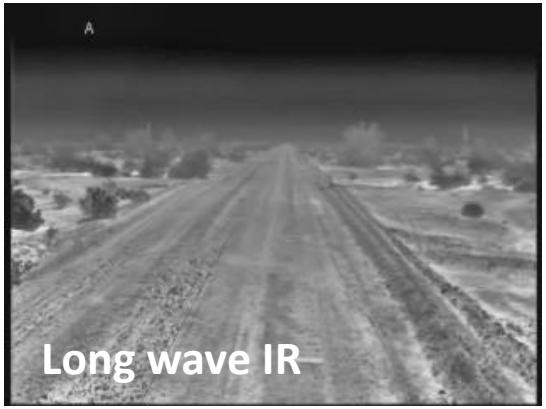


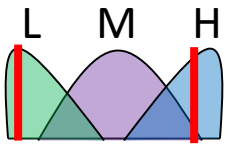
Code for deep learning/CNNs

- Of course, do it yourself
  - That way you know it all!!!
- But, if looking for some state-of-the-art tools
- TensorFlow
  - https://www.tensorflow.org/
  - Examples
    - Deep MNIST
      - https://www.tensorflow.org/versions/r0.8/tutorials/mnist/pros/index.html
    - CNNs
      - https://www.tensorflow.org/versions/r0.8/tutorials/deep_cnn/index.html
- Vlfeat
  - http://www.vlfeat.org/matconvnet/

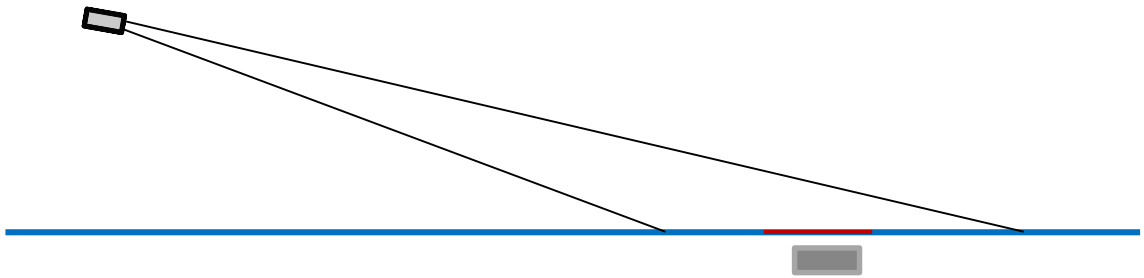


CNN example: EHD in LWIR

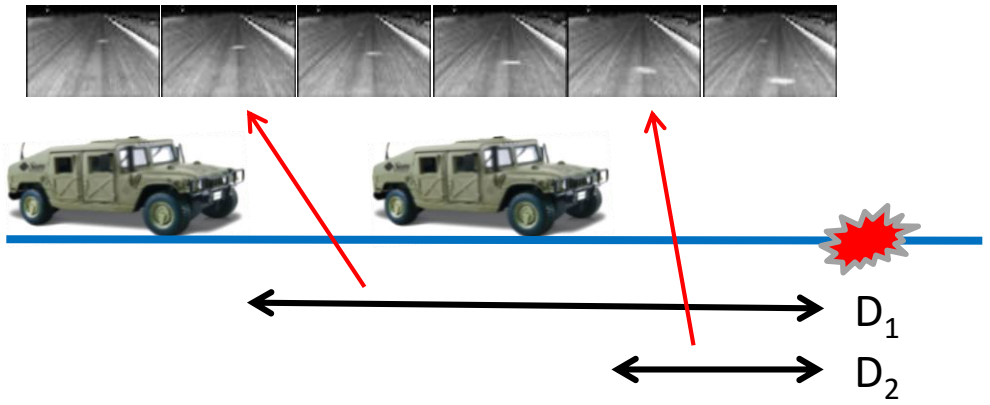




Single look



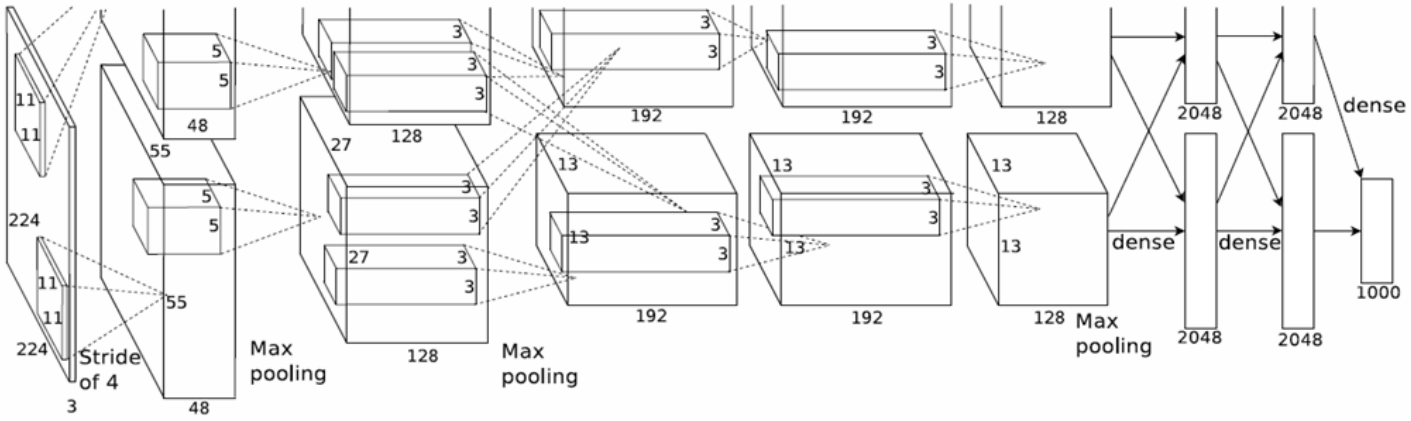
Utilizing Multiple Looks

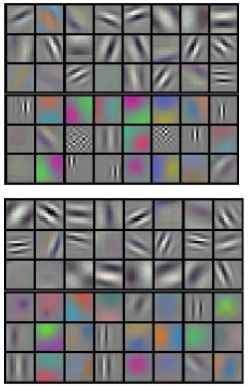


[Stone and Keller, SPIE 2014]

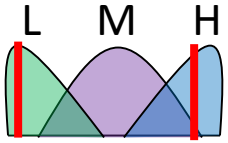
Stone & Keller et al.: CNNs for EHD

- Architecture from **Krizhevsky et al.** (2012) [1]
  - Won the ImageNet Large Scale Visual Recognition Challenge 2012
    - 1000 object categories and over 1.2 million training images
  - 5 convolutional layers, 3 fully connected layers, 1000-way softmax
    - Convolutional layers followed by ReLU activation, response normalization, and max-pooling





- Python implementation and trained model parameters from **Donahue et al.** (2013) [2]. DeCAF - Deep Convolutional Activation Feature.
- Image chips (ROIs in IR) are cropped to the required input size of 227x227

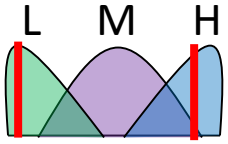


[1] "ImageNet Classification with Deep Convolutional Neural Networks," NIPS (2012)
[2] "Decaf: A deep convolutional activation feature for generic visual recognition," arXiv:1310.1531 (2013)

Stone & Keller et al.: CNNs for EHD

FC7 (ReLU) = second fully connected layer (and after ReLU activation)
FC6 (ReLU) = first fully connected layer (and after ReLU activation)
POOL5 = final convolutional layer after max-pooling
CONV5 = final convolutional layer before max-pooling
RNORM1 = first convolutional layer after max-pooling and response normalization
POOL1 = first convolutional layer after max-pooling

DEEP PRETRAINED CNN – NAUC @ 0.01									
	# FEATURES	DVE CAMERA				SELEX CAMERA			
		ALL LANES	Lane A	Lane B	Lane C	ALL LANES	Lane A	Lane B	Lane C
FC7-ReLU	4096	0.435	0.321	0.420	0.556	0.451	0.355	0.418	0.573
FC7	4096	0.458	0.333	0.451	0.581	0.460	0.359	0.430	0.582
FC6-ReLU	4096	0.479	0.353	0.480	0.598	0.469	0.365	0.454	0.582
FC6	4096	0.513	0.383	0.515	0.632	0.489	0.372	0.489	0.597
POOL5	9216	0.557	0.404	0.600	0.658	0.501	0.386	0.500	0.609
CONV5	43264	0.615	**0.471**	0.655	**0.712**	**0.566**	**0.423**	**0.604**	**0.662**
RNORM1	69984	0.623	0.454	0.709	0.699	0.525	0.389	0.553	0.624
POOL1	69984	**0.624**	0.458	**0.710**	0.695	0.519	0.386	0.545	0.617
CONV5 + BASELINE	75360	0.676	0.508	0.748	0.764	0.607	0.449	0.649	0.714

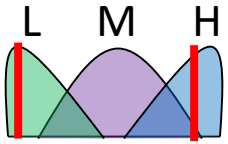


Fully connected layers do not perform well. Convolutional layers that convey spatial position have similar performance to the best hand-engineered image features.

Combining CONV5 features with the Baseline features doesn't improve results.

Stone & Keller et al.: CNNs for EHD

- **EHD results suggest a deep model is not necessary on this data**
  - RNORM1 and POOL1 features perform quite well
  - Task does not require much invariance to translation, orientation, or scale
    - All image chips are extracted at similar distance and angle from camera
- **Shallow vs. deep CNN model**
  - Single convolutional layer connected to output neuron using sigmoid activation
  - Train - stochastic gradient descent with momentum and cross entropy error fx
- Modifications to basic CNN structure
  - Learned frequency domain representation of filters
  - We do this using the inverse FFT (IFFT) as a transformation function
    - $\mathrm{OUTPUT} = \mathrm{CONV}\left(\mathrm{INPUT}, \mathrm{IFFT}(\mathrm{X})\right)$
    - Easier to enforce zero-mean and zero-phase constraints
      - Fix DC frequency component at 0 (zero-mean)
      - Force X to be real and positive (zero-phase)



[Stone and Keller, SPIE 2014]

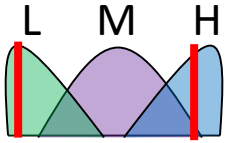
Stone & Keller et al. and CNNs for EHD

- Results
  - 8 neuron network with 0-mean, 0-phase filters learned in frequency domain
    - Vary convolution filter radius

TABLE 14: SHALLOW CNN – NAUC @ 0.01				
Convolution Filter Radius	ALL LANES	Lane A	Lane B	Lane C
DVE				
2	0.617	0.428	0.725	0.689
3	**0.635**	0.464	**0.734**	**0.700**
5	0.616	**0.478**	0.694	0.670
7	0.612	0.460	0.697	0.673
CONV5	0.615	0.471	0.655	0.712
RNORM1	0.623	0.454	0.709	0.699
SELEX				
2	0.531	0.356	0.611	0.616
3	**0.562**	0.397	**0.626**	**0.656**
5	0.559	**0.413**	0.611	0.645
7	0.557	0.409	0.628	0.626
CONV5	0.566	0.423	0.604	0.662
RNORM1	0.525	0.389	0.553	0.624

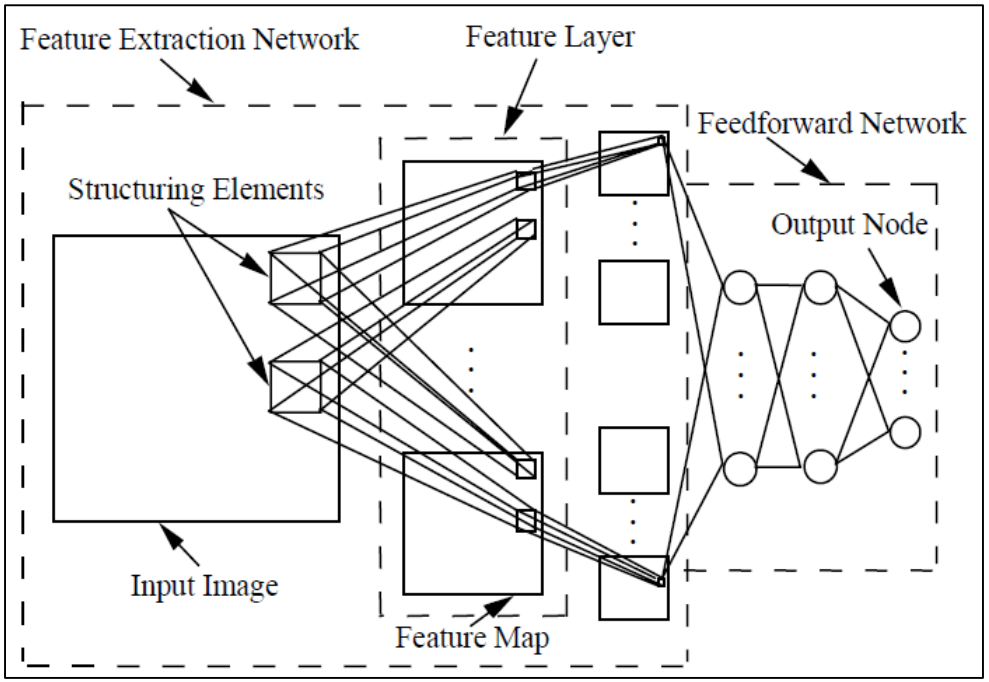
DVE: Shallow CNN outperforms all of the pretrained (deep) CNN features.

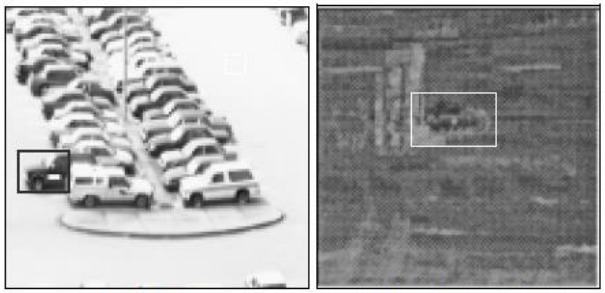
SELEX: Shallow CNN outperforms the first layer pretrained (deep) CNN features. Very close to fifth layer feature performance.

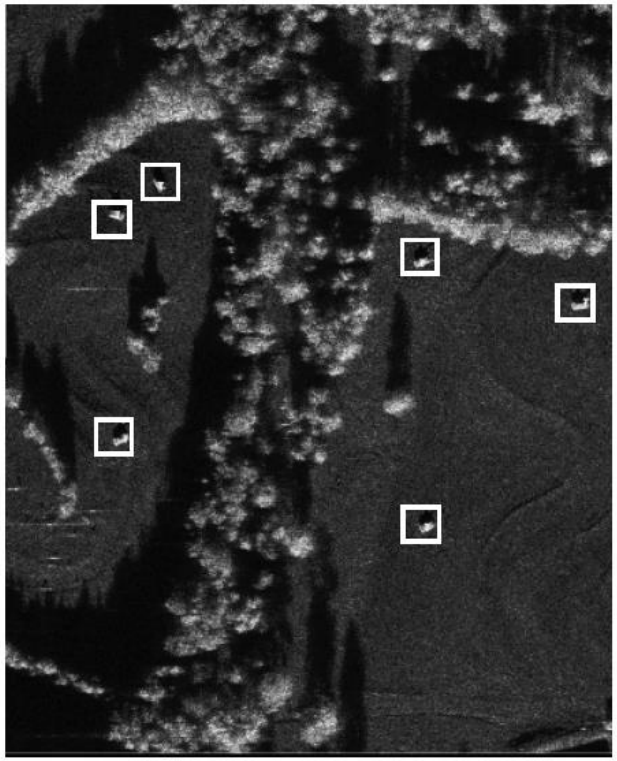


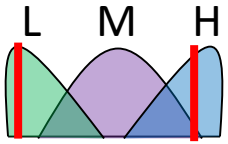
Feature learning: MSNN

- Look familiar?
  - Morphological shared weight NN



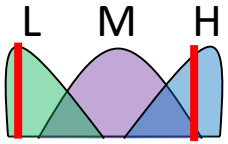

(a) Visible Image (b) FLIR Image


(c) SAR Image



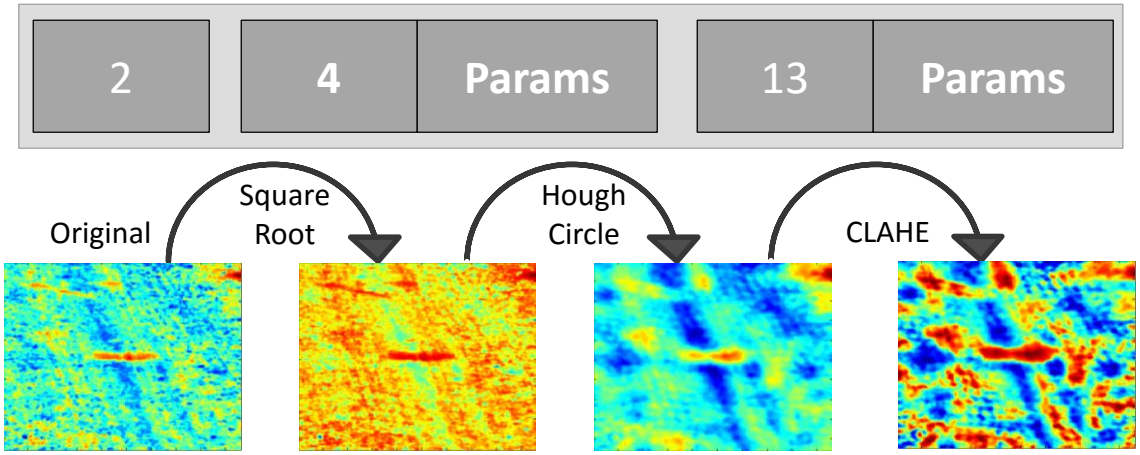
Feature learning: ECO and iECO

- Motivation
  - Composition of heterogeneous transformations
    - Not married to convolution
  - Is it possible to learn explicit (not black box) "features"
    - Try to reverse engineer what was learned (if desired)
  - Use EAs versus "NNs" to optimize the solution(s)
- Initial ECO algorithm
  - Composition of hetero transforms + EA
  - Features == simply unroll image
  - No real diversity promotion nor *complexity* management
- improved Evolutionary COnstructed (iECO) features
  - **Three parts**: (i) pre-processing/conditioning step, (ii) descriptor(s) and (iii) fusion of diverse individuals in different populations



iECO chromosome

- Supervised learning
- Chromosome
  - Variable length
  - Composition of functions -> f2(f4(f13(image)))
  - What functions, parameters, …



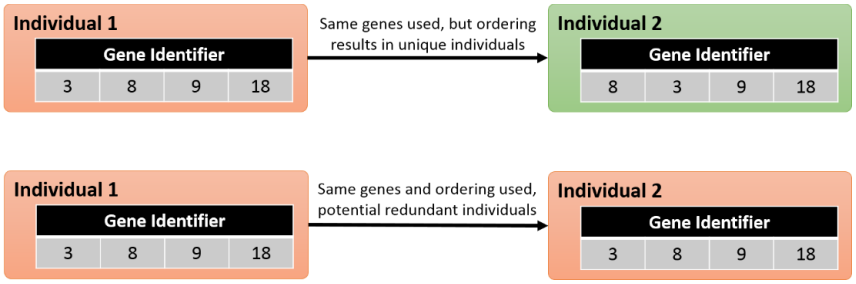
- Big and complex search space …
  - Redundant -> f1(f2(…)) == f4(f1(f3(…)))

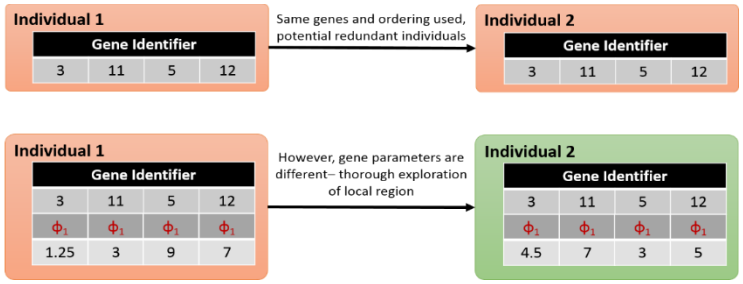
Gene ID	Image Transform
0	Harris Corner Detector
1	Gradient
2	Square Root
3	Gaussian Blur
4	Hough Circle
5	Median Blur
6	Canny
7	Rank Transform
8	Log
9	Sobel
10	Difference of Gaussian
11	Erode
12	Dilate
13	CLAHE
14	Distance Transform
15	Histogram Equalization
16	Laplacian Edge
17	MSER detector
18	Shearlet Filter
19	Gabor Filter

[Price, Anderson, SSCI 2014, SPIE 2015-2016, IGARSS 2016]

iECO specifics

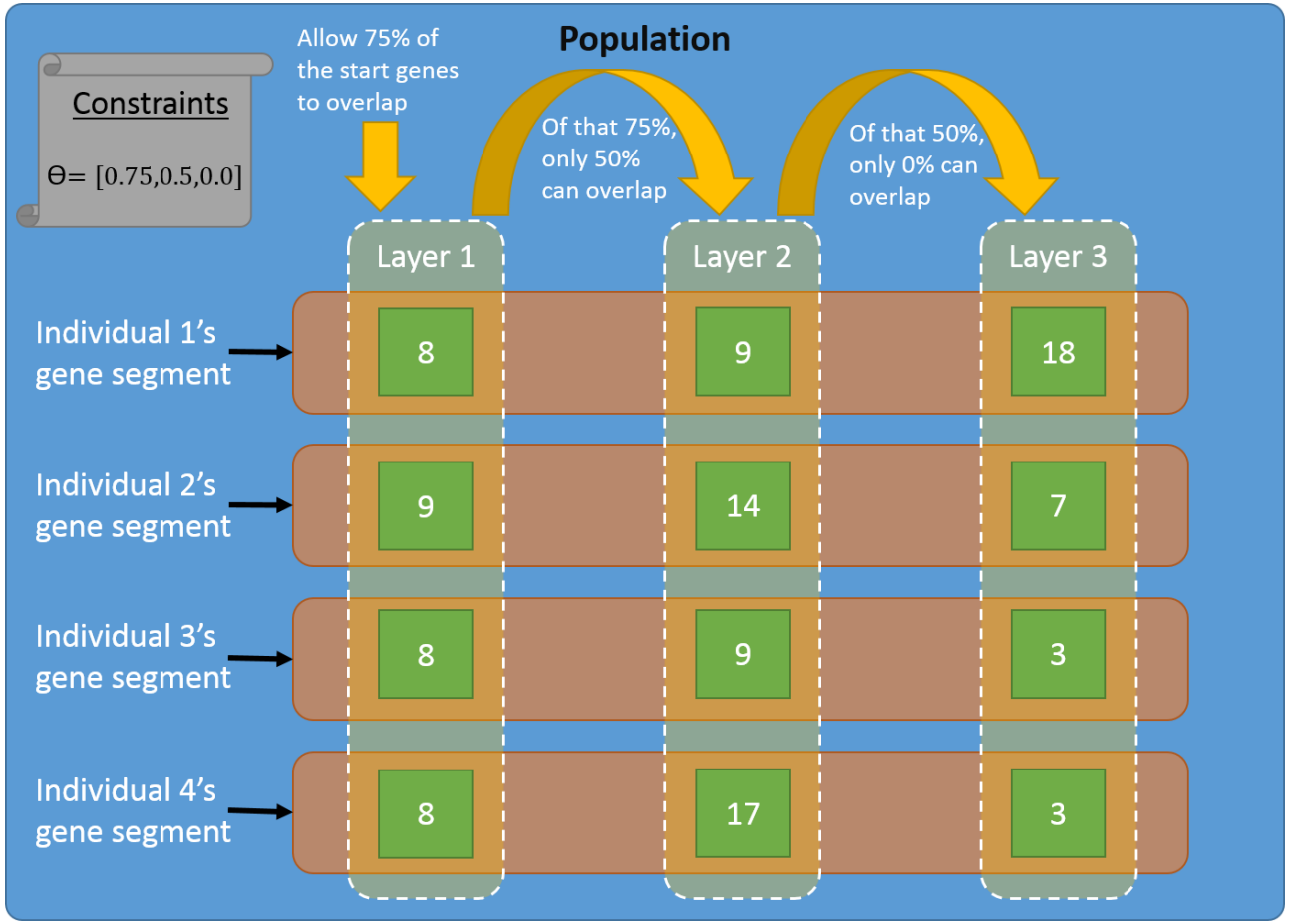
- Learn transforms wrt different descriptors
  - Feature descriptor (LBP, HOG, etc.) dependent
  - Big performance degradation if used globally or out of context
  - Learned each population independent of each other
- Introduce diversity promoting constraints
  - Thorough search
  - Reduce/control genotype redundancy
  - Hasten learning time
    - Explore new solutions sooner
    - Enforces more exploration than standard mutation operator



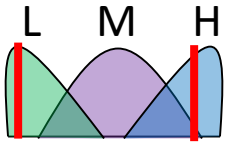


[Price, Anderson, SSCI 2014, SPIE 2015-2016, IGARSS 2016]

Enforcing constraints (diversity)



* Constraints are only at the "gene level"
Still possible similarity in underlying final "expression"

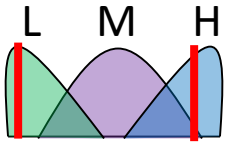


[Price, Anderson, SSCI 2014, SPIE 2015-2016, IGARSS 2016]

Feature learning: iECO

- How to “use it”?
- Route 1: know your ROI

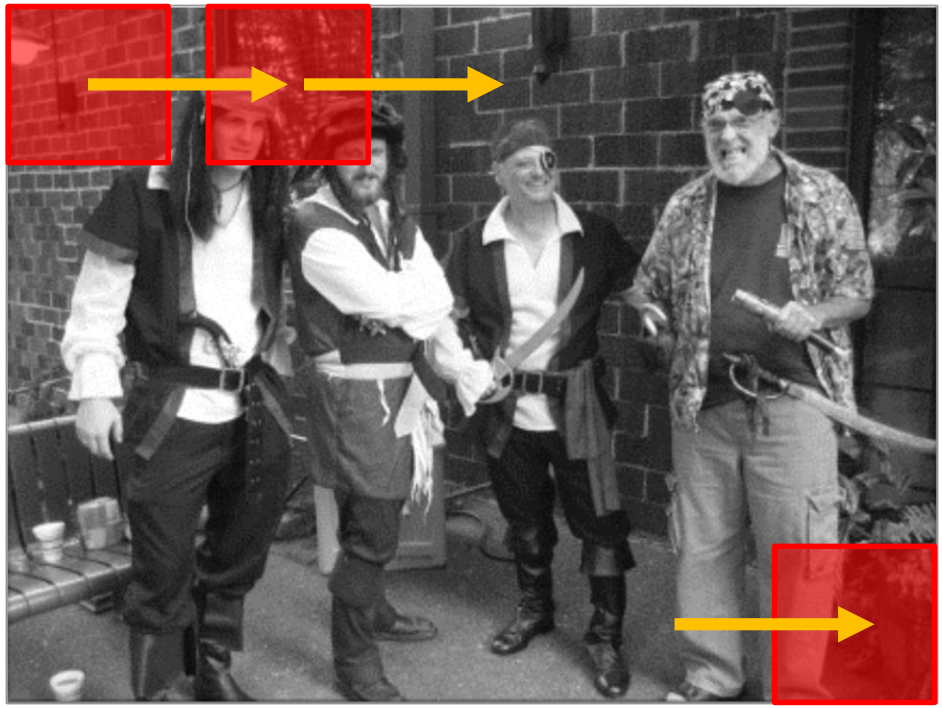
Very suspicious …

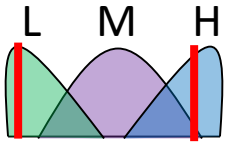


[Price, Anderson, SSCI 2014, SPIE 2015-2016, IGARSS 2016]

Feature learning: iECO

- How to “use it”?
- Route 2: fixed window size and slide-and-detect

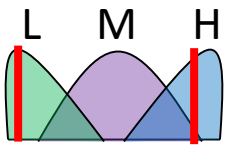




[Price, Anderson, SSCI 2014, SPIE 2015-2016, IGARSS 2016]

Feature learning: iECO

- How to “use it”?
- Route 3: multi-scale (pyramid) and slide-and-detect



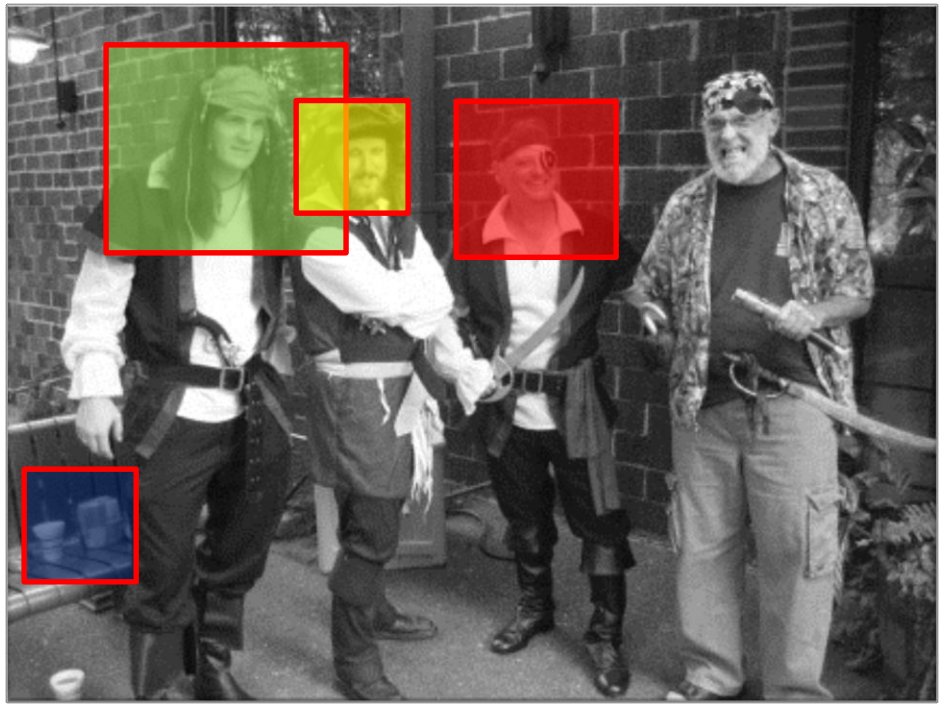
Scale 1

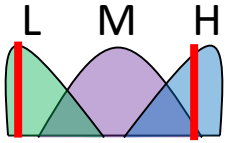
Scale 2

[Price, Anderson, SSCI 2014, SPIE 2015-2016, IGARSS 2016]

Feature learning: iECO

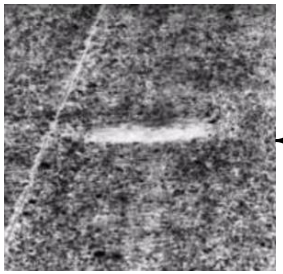
- How to “use it”?
- Route 4: keypoint based (MSER, SIFT, etc.)



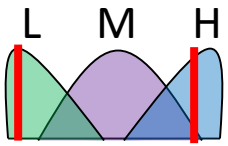


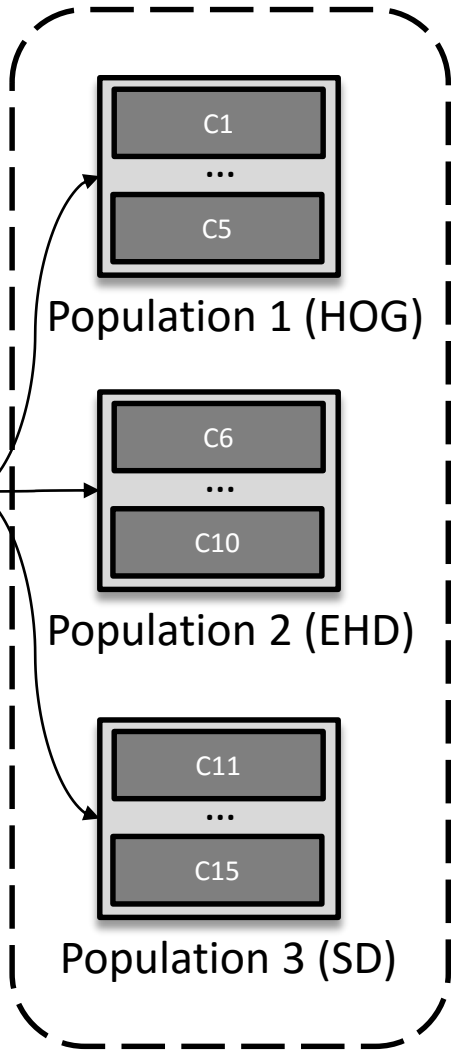
[Price, Anderson, SSCI 2014, SPIE 2015-2016, IGARSS 2016]

Price & Anderson et al.: iECO for FL-EHD in IR



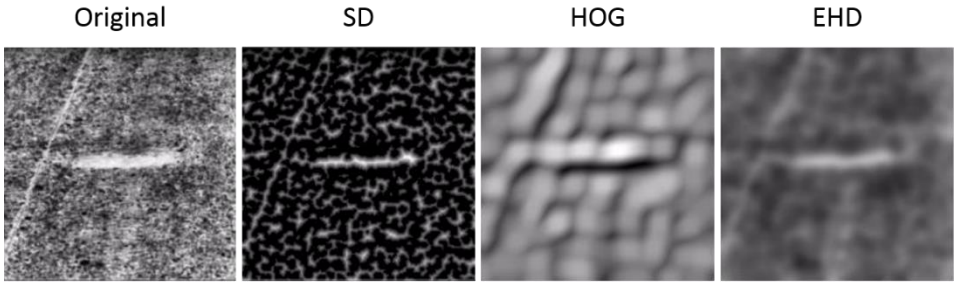
Candidate Chip (acquired using a ROI detection algorithm – prescreener)



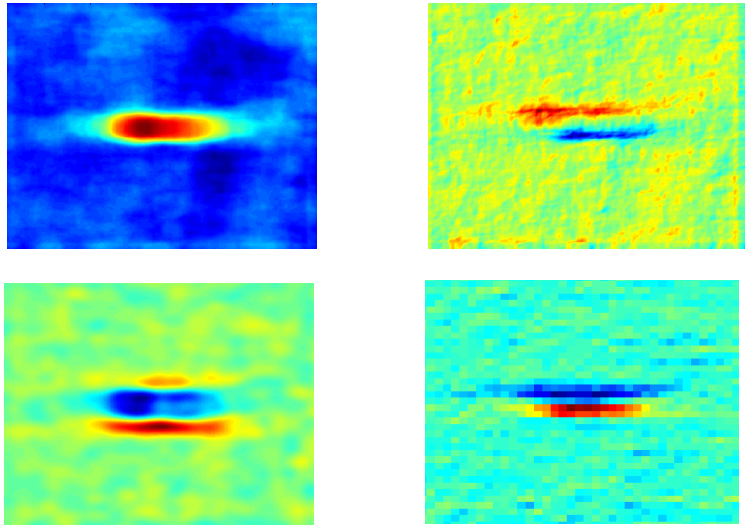


iECO

Single input, input for each descriptor

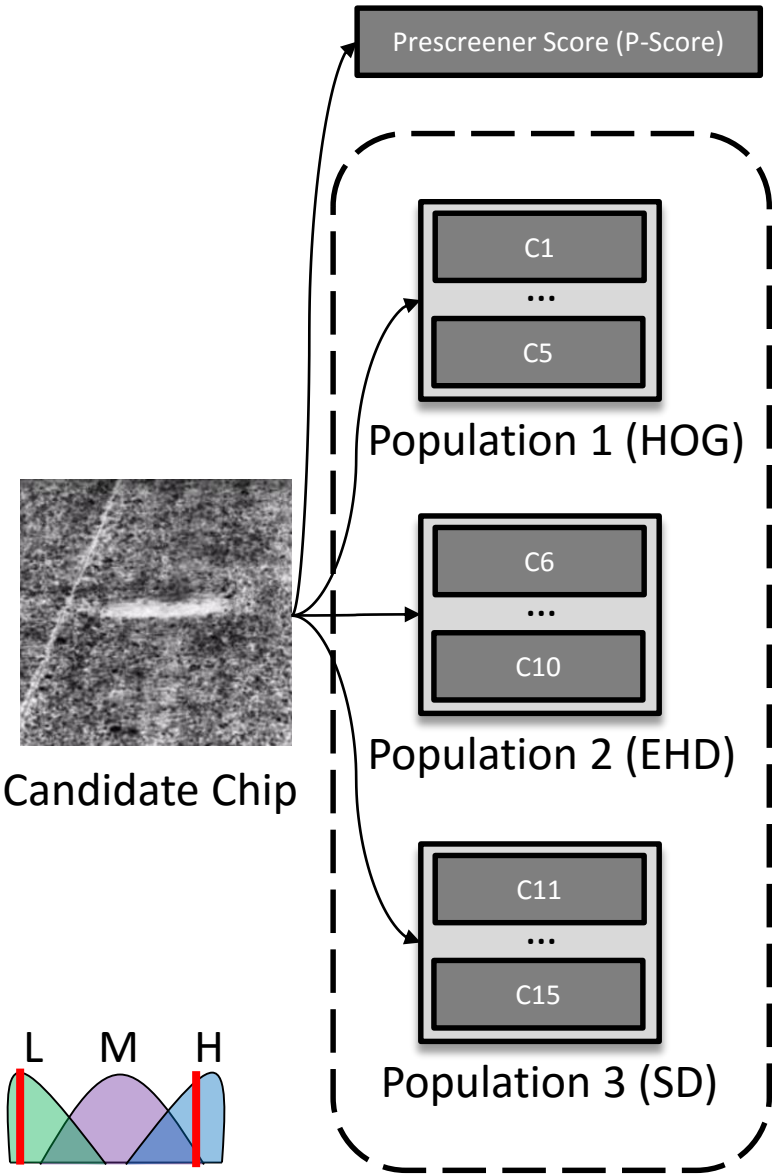


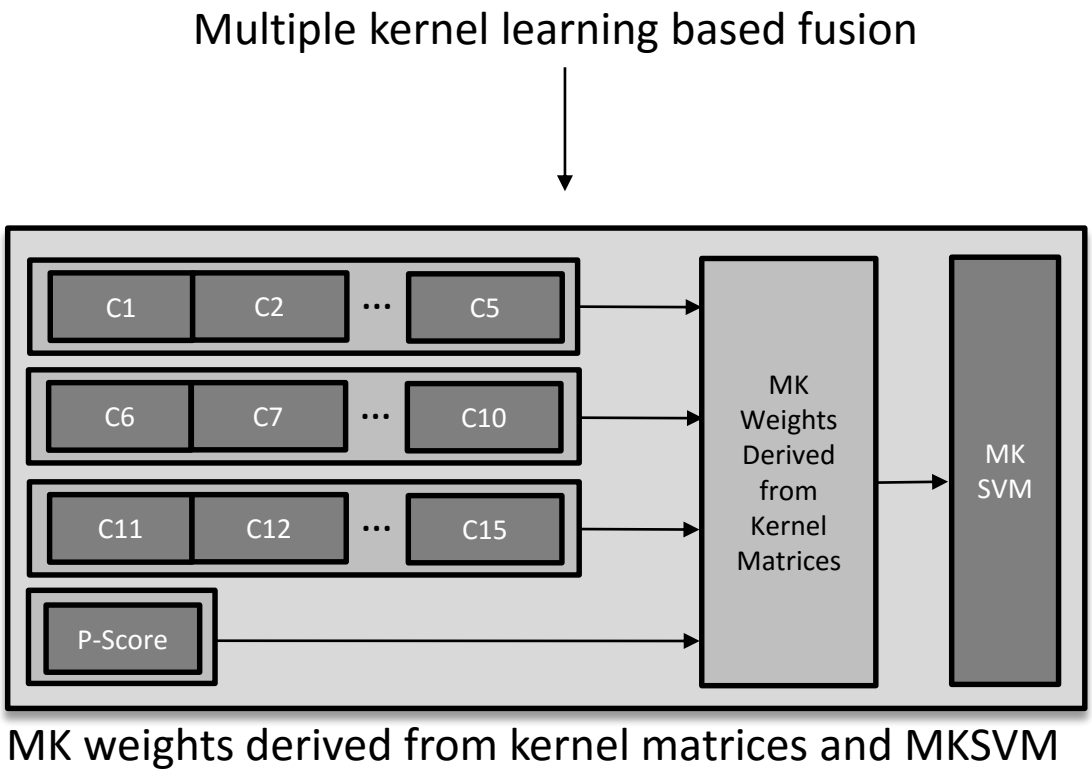
Average response across true targets (4 individuals)



They all sort of “look similar” at different aspects

Price & Anderson et al.: iECO for FL-EHD in IR

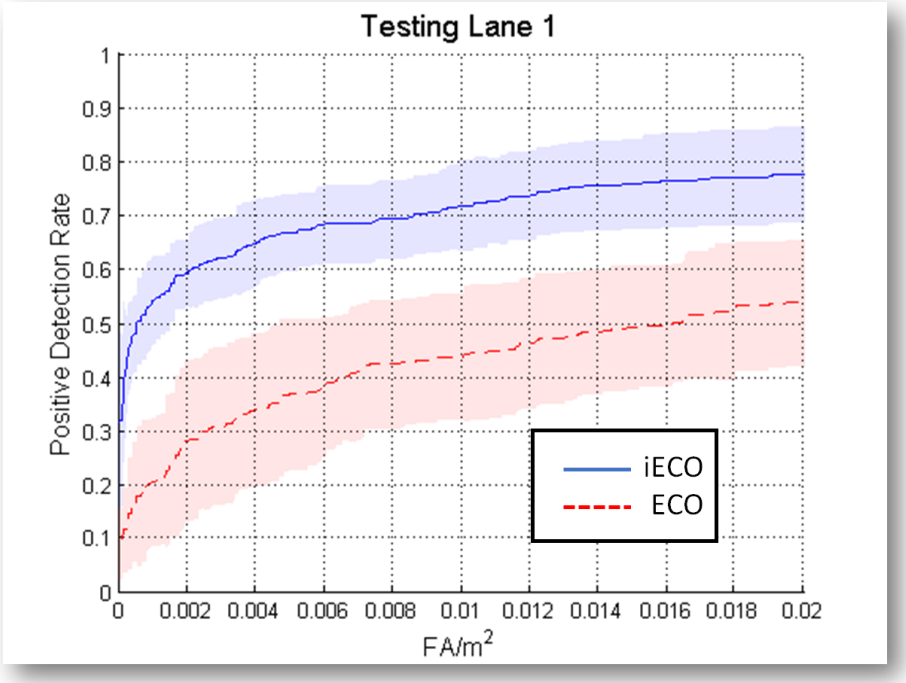


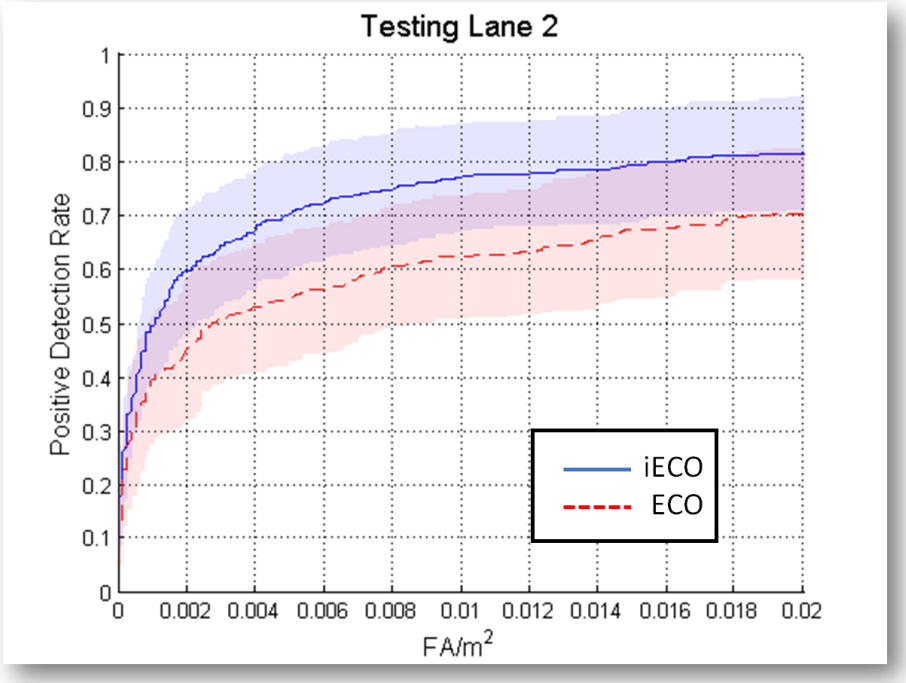


MK weights derived from kernel matrices and MKSVM

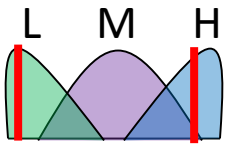
[Price, Anderson, SSCI, SPIE]

Price & Anderson et al.: iECO for FL-EHD in IR

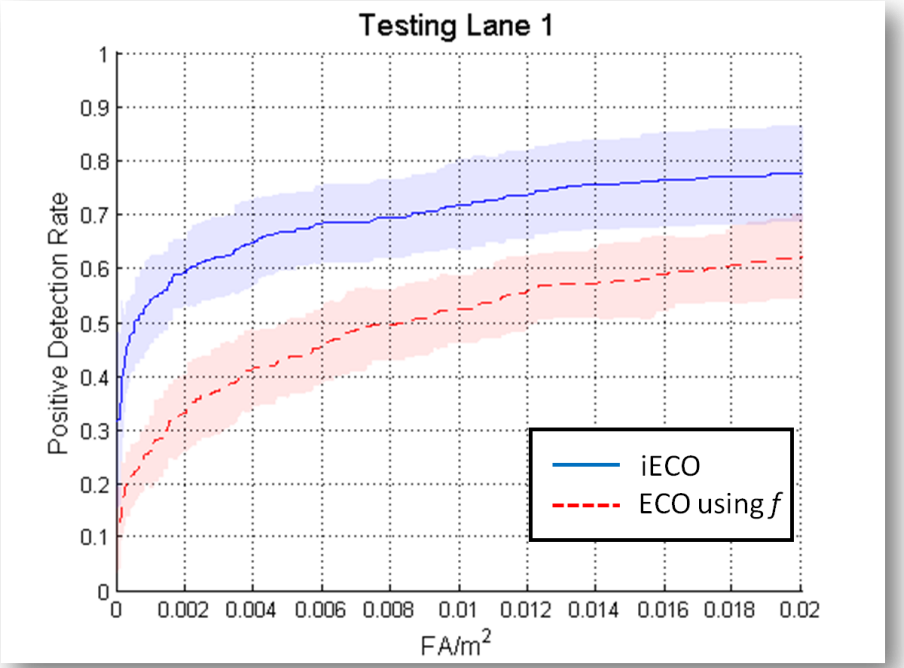


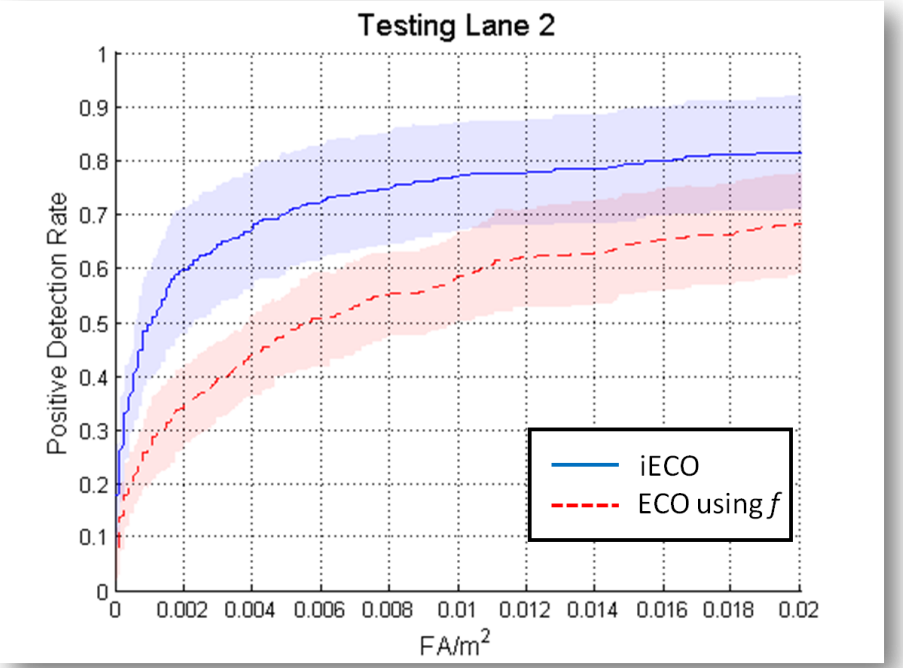


U.S. Army test site
Cross validation
Two lanes (16 total runs)
See Price et al. SPIE 2014-2016 and SSCI 2014



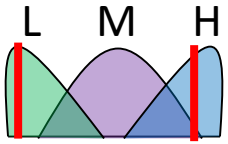
Price & Anderson et al.: iECO for FL-EHD in IR





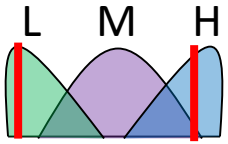
What impact was the descriptor?

Put descriptor on ECO and compare



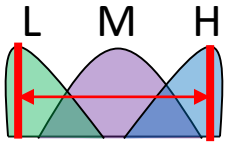
Feature learning: summary

- Deep learning
  - Still state-of-the-art
  - Wonderful performer on numerous data sets
  - **NNs** are at the heart of this approach
- iECO
  - **EAs** are at the heart of problem
  - Combination of “machine learned” (composition of functions), “human learned” (descriptors) and fusion
- Two ways to crack this egg (feature learning)
- Examples of NNs and EAs, how about FSs …?
  - Can the case be made for uncertainty?



Fusion

- Data/information "fusion"
  - Complicated and extremely diverse field
  - Today, we will focus on **aggregation operators**
  - General hope: obtain "better" solution (more robust, higher information content, etc.) versus just the individuals
- Why is fusion needed in CV?
  - Multiple classifiers, features/descriptors, "parts", views/cameras, etc.
- Various "levels"
  - Early fusion (exploit rich data correlations if/when present)
    - Spectrum level fusion (e.g., fusion of bands in hyperspectral imagery)
    - Input/feature space level fusion (e.g., MKL)
  - Late fusion (combining different decision makers)
    - Classifier/decision level fusion (e.g., ensembles, FI, DeFIMKL, etc.)

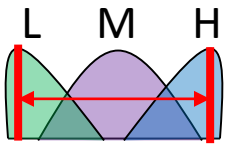


Fusion: “typical approach”

Inputs

Fusion
(e.g., fuzzy integral)

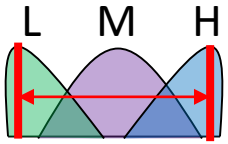
“N -> 1” operation
BUT
Was there conflict?
How much conflict?
How confident in our decision?
…
Not a simple topic whatsoever



Improvement over individuals?
Us – pattern recognition
Petry – information theoretic indices

Fusion: fuzzy integral

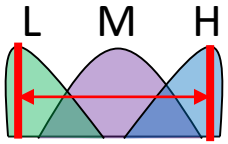
- Set of "sources" (X)
  - Experts, sensors, algorithms, evidence, etc.
    - Discrete set of N=3, X = { x1=Derek, x2=Jim, x3=Chan }
- Data/information coming from our inputs
  - h: partial support function (the integrand)
  - h(x1) = value in [0,1]
    - Can be other ranges as well, e.g., [-inf,inf], [0,inf], etc.
- What is the *worth/reliability* of our sources?
  - g: fuzzy measure (normal and monotone capacity)
  - Simple example
    - g({x1}) = worth of Derek
    - g({x1,x2}) = worth of the group Derek and Jim



[See Sugeno, Grabisch, Anderson, Keller, Havens, MANY MORE]

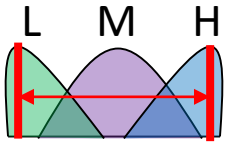
Fusion: fuzzy integral

- The fuzzy integral (FI) is a creative way to combine these two pieces of information (h and g)
  - E.g., could be objective values from sensors with subjective (or objective) values of worth
- What is the fuzzy integral? [one take on it!]
  - Real-value line, we get measure, for [a,b], g is b-a (length)
    - For higher dimensions, area, volume, hyper-volume …
  - What about when X is not the reals but set of experts?
    - Need a different way of specifying/obtaining/etc. the FM
  - The FM is a way to MODEL our problem (sets of sources)
  - The FI USES the FM to combine our inputs and get an answer
  - Really creative and slick idea (leveraging the integral)



Fusion: fuzzy integral

- Often, it is important to model and exploit the rich interactions between sources
- Move beyond independence assumptions and additivity property of probability
- Can obtain behaviors such as “the whole is more than the sum of its parts”
- Another take on the fuzzy integral
  - Just a parametric aggregation operator
    - Do not care about the “meaning” of g, it just gives us different aggregation operators [generator function]



Fusion: fuzzy measure

What is the 'worth' of these (handsome) subsets of sources?

g({x1,x2,x3})=0.900001 … 1?

g({x1,x2})=0.9

g({x1,x3})=0.9

g({x2,x3})=0.9

g(x1)=0.8

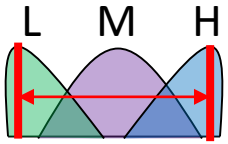
Derek

g(x2)=0.8

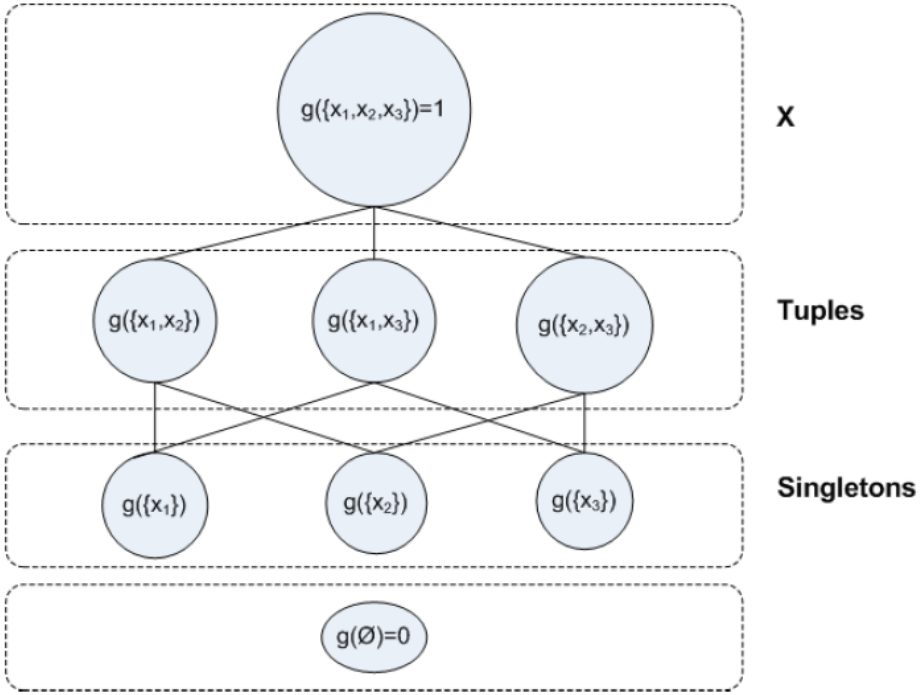
Tim Havens

g(x3)=0.8

Jim

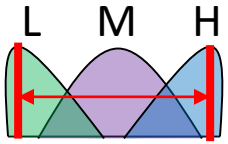


Fusion: fuzzy measure



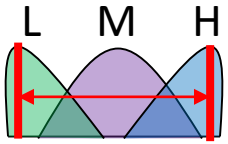
$$\text{(Boundary conditions) } g(\phi) = 0 \text{ and } g(X) = 1$$
$$\text{(Monotonicity) If } A, B \in \Omega \text{ and } A \subseteq B, \; g(A) \leq g(B)$$

* note: third condition (but we are discussing discrete sets here)



Fusion: fuzzy measure

- Non-additive FMs are useful in general
  - Probability measures (PM) are fuzzy
  - Upper/lower PM (belief theory) are fuzzy
  - Possibility and necessity measures are fuzzy
- The FM is not small
  - For |X|=N inputs, 2^N terms
    - Grows fast
  - How to get the FM?
    - Specify by an expert, learn from data, impute
- There are non-monotonic measures and CI



Fusion: well-known FMs

- Assume we are just given the densities
  - Worth of just the singletons
- Impute the rest of the FM from this info
- Sugeno lambda-FM

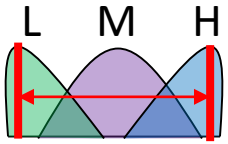
$$For \text{ all } A, B \subseteq X \text{ with } A \cap B = \phi,$$

$$g(A \cup B) = g(A) + g(B) + \lambda \cdot g(A) \cdot g(B) \text{ for some } \lambda > -1$$

$$1 + \lambda = \prod_{i=1}^{n} \left(1 + \lambda g^i\right)$$

- S-Decomposable (e.g., via max)

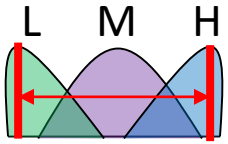
$$g(A \cup B) = \max(g(A), g(B))$$



Fusion: interesting observations

- For Sugeno lambda-FMs

- If $\sum_{i=1}^{n} g^i = 1$, then $\lambda = 0$; Probability measure
- If $\sum_{i=1}^{n} g^i > 1$, then $\lambda < 0$; Dempster-Shafer Belief function
- If $\sum_{i=1}^{n} g^i < 1$, then $\lambda > 0$; Dempster-Shafer Plausibility function



Fusion: (discrete) fuzzy integral

Sugeno integral

$$\int_s h \circ g = \bigvee_{i=1}^{n} (h(x_{(i)}) \wedge G_{(i)})$$

Choquet integral

$$\int_c h \circ g = \sum_{i=1}^{n} \omega_i h(x_{(i)})$$

Fuzzy measure

$$g : 2^X \rightarrow [0, 1]$$

Partial support function

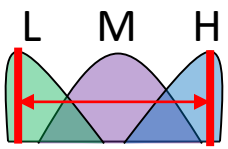
$$h : X \rightarrow [0, 1]$$

$$\omega_i = (G_{(i)} - G_{(i-1)})$$

$$G_{(i)} = g(\{x_{(1)}, \ldots, x_{(i)}\})$$

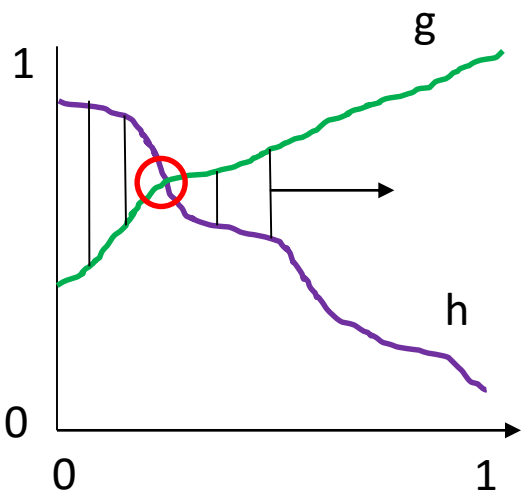
$$G_{(0)} = 0$$

$$h(x_{(1)}) \geq h(x_{(2)}) \geq \ldots \geq h(x_{(n)})$$

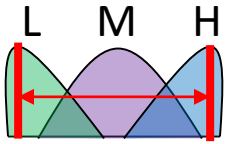


CIs and SIs

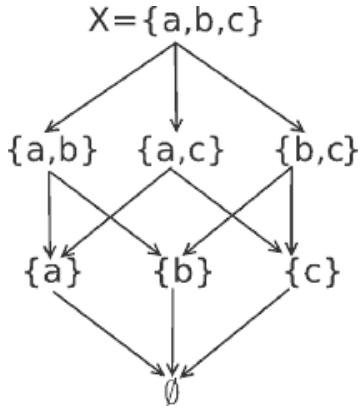
- CI is *nice* (mathematically speaking)
  - Continuous, differentiable, …
  - Bounded (between min and max)
  - Idempotent, monotonic, …
  - …
  - Attractive property
    - For additive measure, recover Lebesque integral



- The SI is "spotty" while CI "fills the spectrum"
  - Meaning, SI is one of the input values or one of the values from the FM (max-min), whereas CI can go from [min,max[]
- SI has a pretty story though!
  - The "best pessimistic agreement"



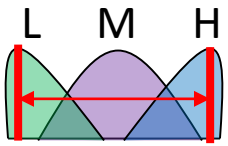
Fusion: CI



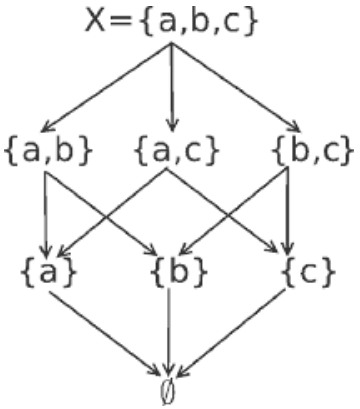
	1			1		
1	1	1	0	0	0	
1	1	1	0	0	0	
	0			0		

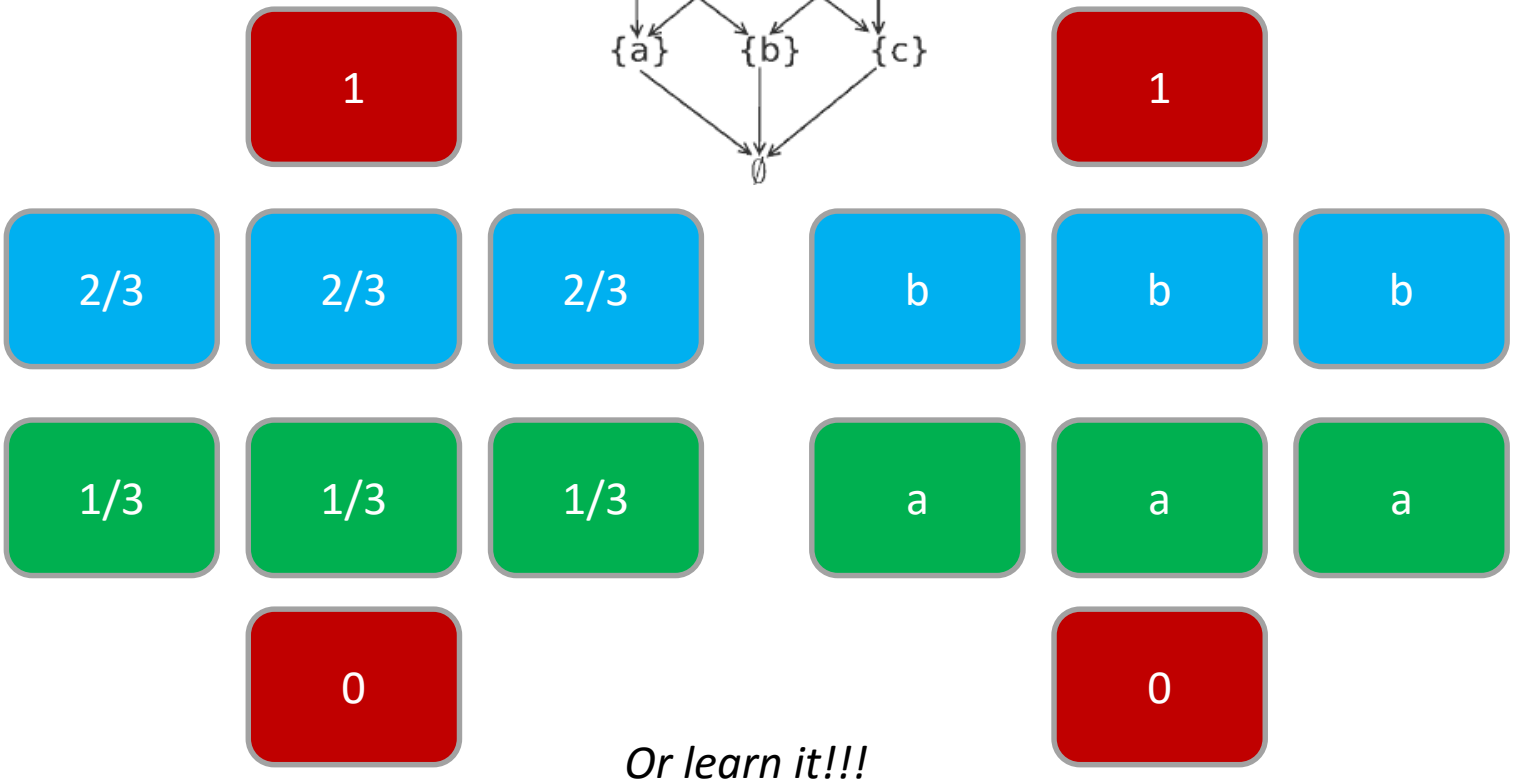
max

min

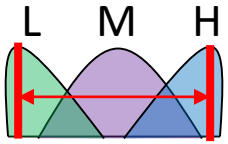


Fusion: CI





Or learn it!!!

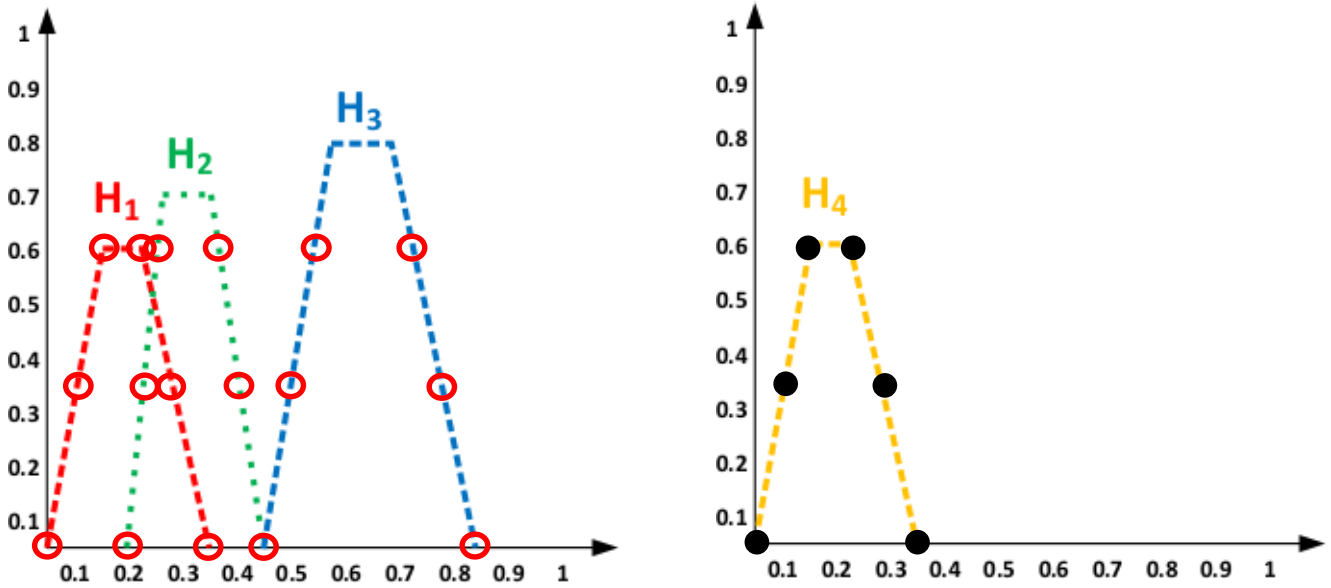


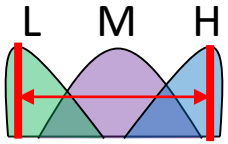
average

linear order statistics

Fusion: FI extensions

- Can deal with fuzzy sets as well
  - Probability distributions, possibility distributions, etc.
- Extensions: gFI, SuFI and NDFI
  - Anderson, Havens, Keller [TFS, FUZZ-IEEE, WCCI, IPMU, Info Science]





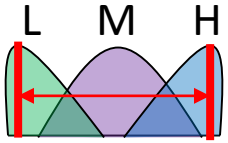
$$\alpha_1 = 0, \alpha_2 = 0.35, \alpha_3 = 0.6$$

$$g(\{h_1\}) = g(\{h_2\}) = g(\{h_3\}) = g(\{h_1, h_3\}) = g(\{h_2, h_3\}) = 0, g(\{h_1, h_2\}) = 0$$

(so min operator)

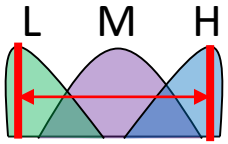
Fusion: FI learning

- Different approaches
  - Densities
    - Keller, Gader, etc.
      - Reward/punishment, genetic algorithms, etc.
  - Full capacity
    - Anderson, Havens, Grabisch, Eyke, Belikov, Islam, etc.
      - Quadratic programming, regularization, etc.
      - E.g., Anderson and Havens, FUZZ-IEEE 2014
- Most exploit labeled data
  - D = data and L = set of labels
  - Create objective function, e.g., SSE, maximize/minimize
  - BUT, 2^N values and even more constraints!



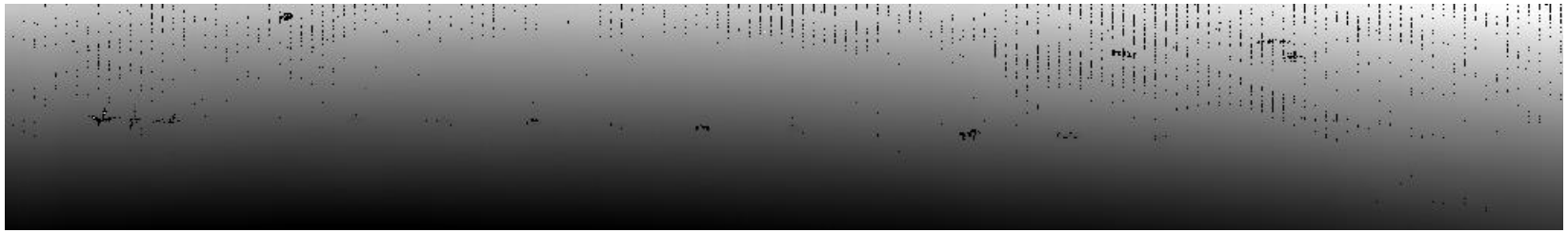
Keller et al.

- SI and CI used for non-linear image filtering
- Can implement morphological filters, all linear and order statistic filters, all linear combination of order statistics filters, etc.
- Used instead of means and variances for “size contrast” filters (SCF)
- Used to segment and to fuse multiple detectors (Mid? High?)
- Next slide is simple low-level example, but they are great at mid- to high- level fusion as well (FIs)

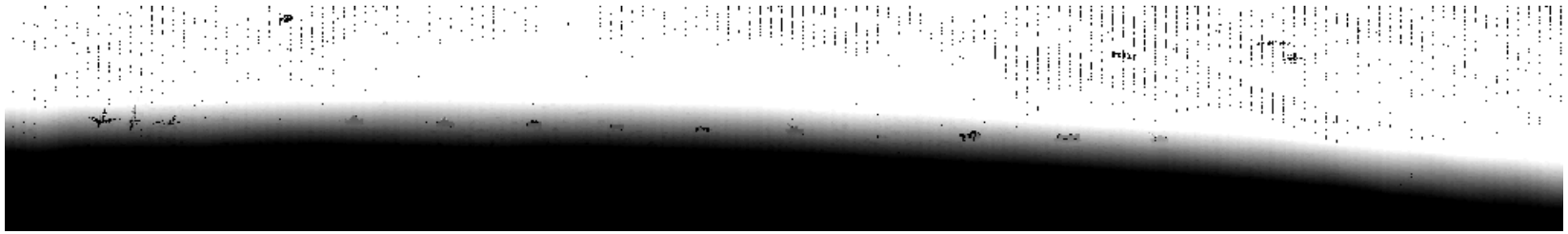


Keller et al., image processing

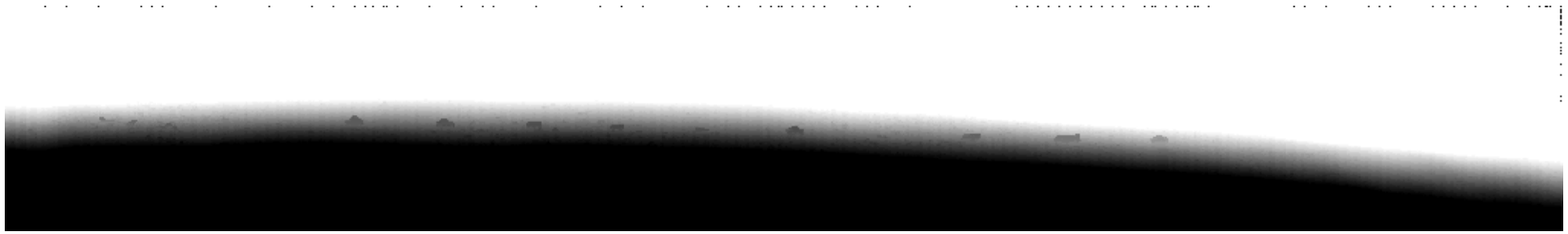
LADAR = LAser Detection And Ranging

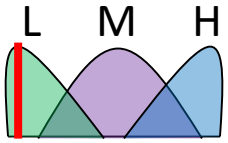


Original



Scaled (for Viewing)

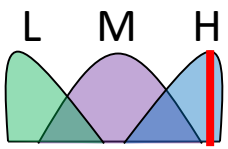




Preprocessed by Choquet Filter

CI learning in light of uncertainty

- Xiaoxiao, Zare, Anderson and Keller
  - Paper is Tue (two days from now) in EC
    - In Comp. Intelligence in Security & Def. SS
      - Multiple Instance Choquet Integral for Classifier Fusion
  - Learn the capacity under uncertainty
    - What if our labeling is poor?
      - E.g., localization error in UAS/airborne system?
    - For each observation, we get a "bag"
      - "Positive bags" – one or more instances are good
      - "Negative bags" – all instances are bad
    - Multiple instance learning (MIL) based FI learning
    - Applications
      - Multi-sensor/algorithm fusion, hyperspectral image processing



CI learning in light of uncertainty

- Xiaoxiao, Zare, Anderson and Keller

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{a=1}^{B^-}\sum_{i=1}^{N_b^-} \ln\left(1 - \mathscr{N}\left(C_{\mathbf{g}}(\mathbf{x}_{ai}^-)|1,\sigma^2\right)\right) + \sum_{b=1}^{B^+} \ln\left(1 - \prod_{j=1}^{N_b^+} 1 - \mathscr{N}\left(C_{\mathbf{g}}(\mathbf{x}_{bj}^+)|1,\sigma^2\right)\right)$$

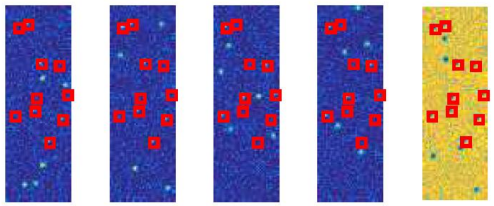


Fig. 3: One example of the simulated lane-based target detection data set. Each rectangular image represents one detector output - Detectors No. 1 to No. 5 from left to right. Each rectangular image has $120m$ (120 pixels) in the vertical direction and $40m$ (40 pixels) in the horizontal direction. The red boxes mark the true target locations.

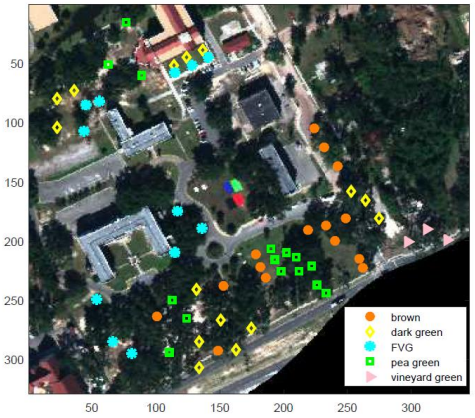


Fig. 6: The RGB image from Flight 2. Orange circle marks the true brown target locations, yellow diamond marks the true dark green target locations, cyan asterisk marks the true FVG target locations, green square marks the true pea green target locations, and pink triangle marks the true vineyard green target locations.

Algorithm 1 MICI Algorithm

TRAINING

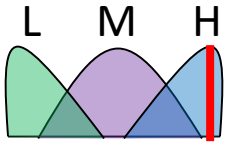
Require: Training Data, Training Labels, Parameters

Initialize a population of measures ▷ III-B1
$F^* = max(\mathbf{F}_P^0)$, $\mathbf{g}^* = \arg\max_{\mathscr{G}} \mathbf{F}_P^0$
for $t := 1 \rightarrow I$ **do**
  for $p := 1 \rightarrow P$ **do**
    Evaluate valid intervals of $\mathscr{G}\{p\}$ ▷ III-B2
    Randomly sample $z \in [0, 1]$
    if $z < \eta$ **then** ▷ III-B3
      Update $\mathscr{G}\{p\}$ by small-scale mutation
    else
      Update $\mathscr{G}\{p\}$ by large-scale mutation
    end if
  end for
  Evaluate fitness of updated measures using (6)
  Select measures ▷ III-B4
  if $max(\mathbf{F}_P^t) > F^*$ **then**
    $F^* = max(\mathbf{F}_P^t)$, $\mathbf{g}^* = \arg\max_{\mathscr{G}} \mathbf{F}_P^t$
  end if
end for
  return $\mathbf{g}^*$

TESTING

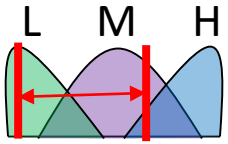
Require: Testing Data, $\mathbf{g}^*$

$TestLabels \leftarrow$ Choquet integral output computed based on Equation (1) using the learned $\mathbf{g}^*$ above
  return $TestLabels$



Mathematical morphology

- Roots in set theory
  - *Pre/post-processing* of imagery, obj soft segmentation, feature detection (and object detection), etc.
- Binary and grayscale variants
  - Briefly discuss binary
  - Numerous fuzzy set approaches
- Many *operations*
  - Erosion and dilation
  - Opening and closing
  - Reconstruction
  - Hit and miss transform

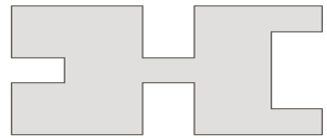


Mathematical morphology

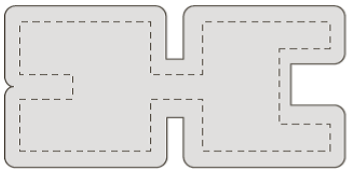
- Structuring element
- Operations

off	on	off
on	on	on
off	on	off

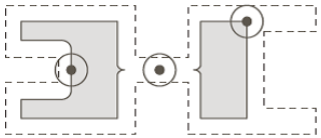
Original



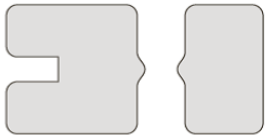
Dilation



Erosion

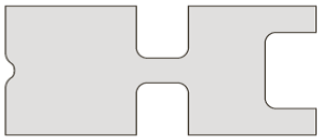


Opening

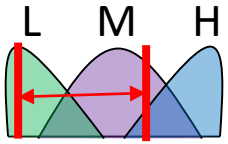


$A \circ B = (A \ominus B) \oplus B$

Closing

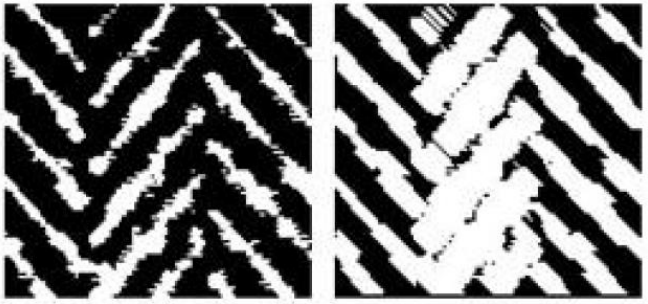


$A \cdot B = (A \oplus B) \ominus B$

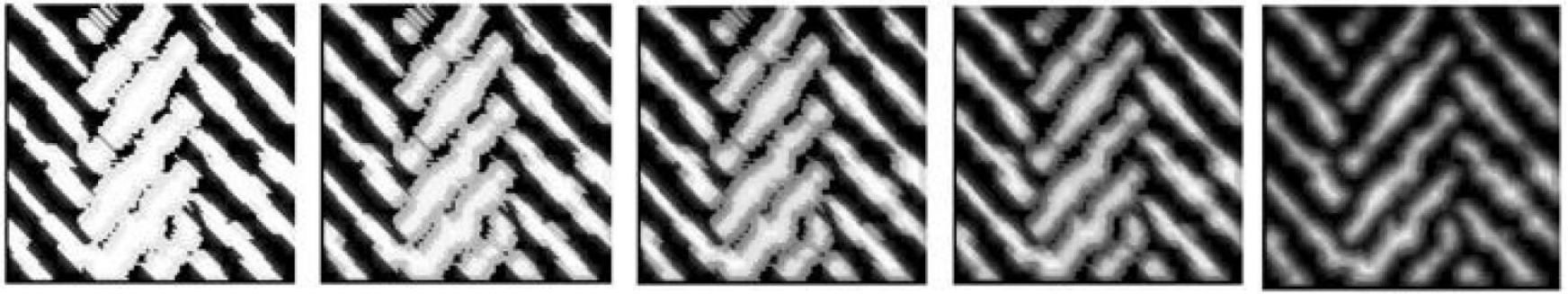


Morphology and fuzzy integral

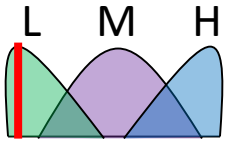
- Via the fuzzy integral [1994]
  - Grabisch showed that quasi-Sugeno fuzzy integral defined w.r.t. a proper fuzzy measure can form an algebraic dilation or erosion



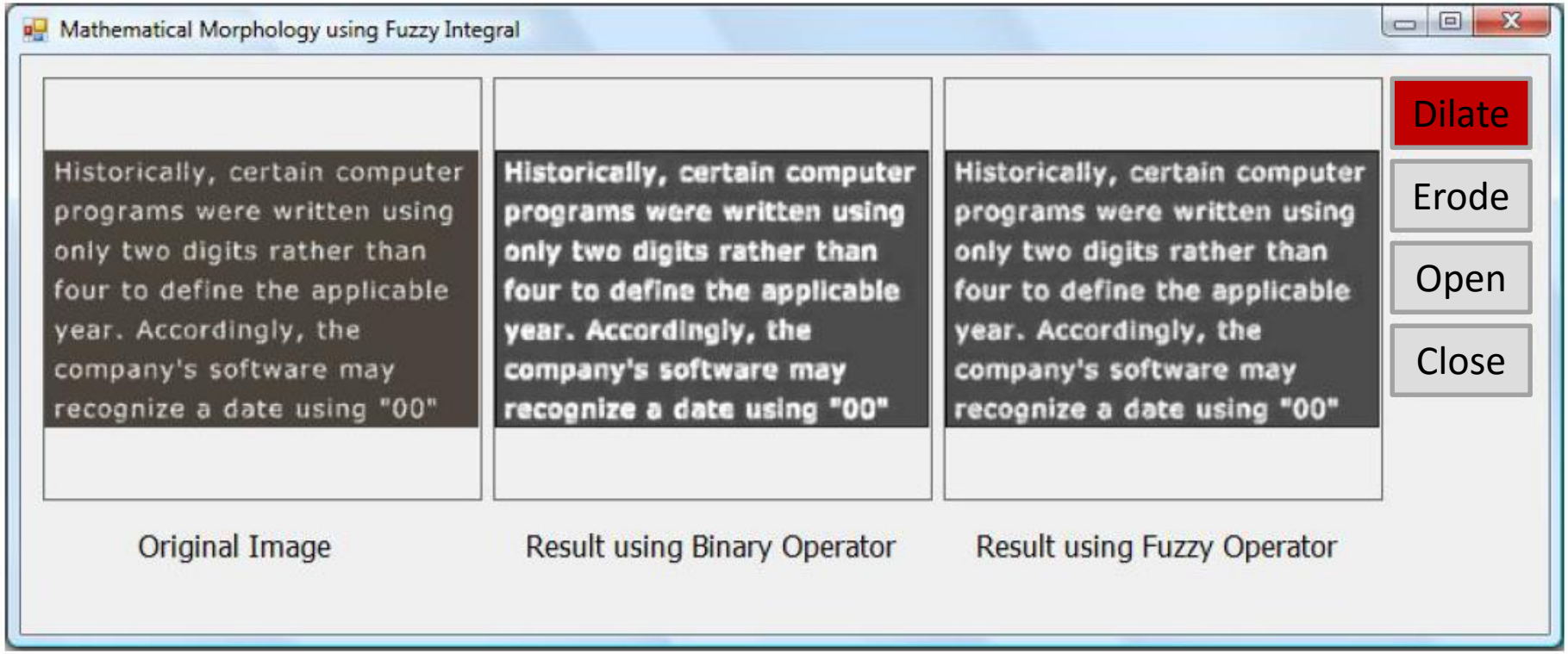
original binary morphology



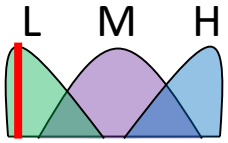
fuzzy morphology



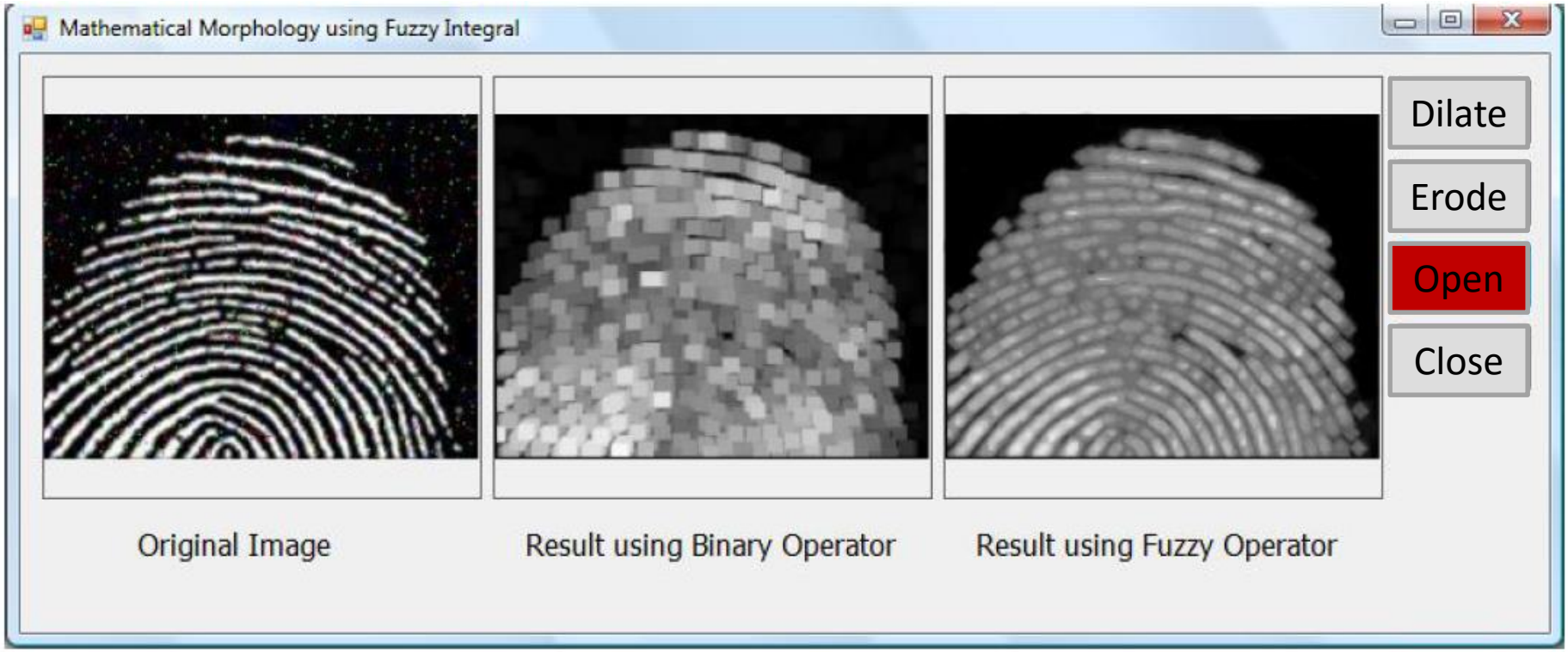
Morphology and fuzzy integral



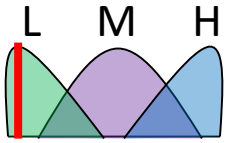
Program built by a graduate student of mine (Mohammad Al Boni)



Morphology and fuzzy integral

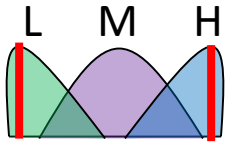


Program built by a graduate student of mine (Mohammad Al Boni)



Multiple kernel learning

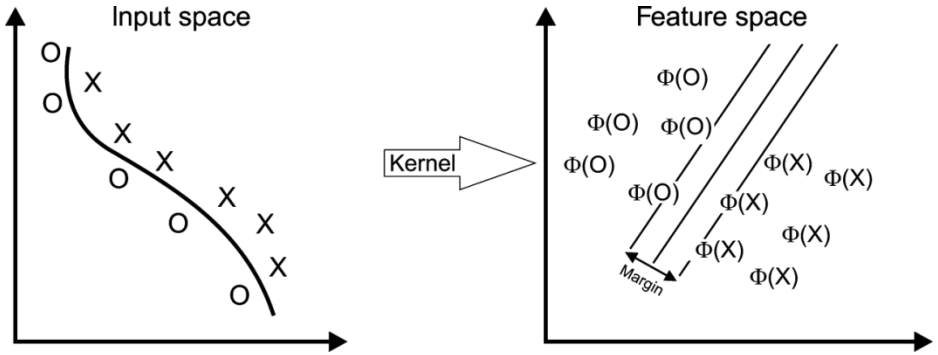
- Kernel theory – useful tool for pattern rec.
  - Both classification and clustering
  - Used extensively in signal/image processing, CV, …
- The reality (kernel theory)
  - Existence theorem … in practice, which one …
- Where do we go from there … ?
  - Multiple kernels – building blocks (valid kernels)
  - Aggregation operators (that preserve conditions)
  - Learning algorithms to optimize their combining
  - Where to do aggregation?
    - Feature-level (homogenization then aggregate)
    - Decision-level



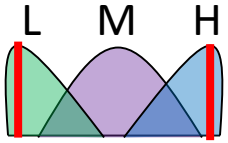
MKL: kernel (crash course)

Definition 1 (kernel). Suppose we are given a feature mapping $\phi : \mathcal{R}^d \rightarrow \mathcal{R}^{\mathcal{H}}$, where d is the dimensionality of the input space and $\mathcal{R}^{\mathcal{H}}$ is some (higher-)dimensional space, which is called the *Reproducing Kernel Hilbert Space* (RKHS). A kernel is the inner product function $\kappa : \mathcal{R}^d \times \mathcal{R}^d \rightarrow \mathcal{R}$, which represents the inner-product in $\mathcal{R}^{\mathcal{H}}$,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle, \quad \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}.$$



http://peds.oxfordjournals.org/content/21/1/37/F1.large.jpg



$$\kappa(\mathbf{x}, \mathbf{y}) = \exp(\sigma \|\mathbf{x} - \mathbf{y}\|^2)$$

MKL: Mercer kernel

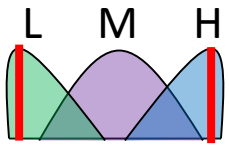
Definition 2 (Mercer kernel). Assume a kernel function κ and finite data $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$. The $n \times n$ matrix $K = [K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)]$, $i, j = [n]$, is a Gram matrix of inner products. Therefore, K is symmetric and *positive semi-definite* (PSD), $\mathbf{x}_i^T K \mathbf{x}_i \geq 0$, $\forall \mathbf{x}_i \in \mathcal{X}$. Note, since K is PSD, all eigenvalues are non-negative.

Its all about the matrix ;-)

L M H

MKL: existing work

- Existing MKL approaches
  - In general
    - Fixed rule: e.g., uniform weight
    - Heuristic: e.g., SPIE 2015 and 2016 - Price, Anderson, Havens, et al. derive from kernel matrices and do not anchor to SVM formula
    - Optimization: e.g., solve relative to SVM formula
  - Linear convex sum and feature space fusion
    - SimpleMKL and MKLGL
    - FIGA and GAMKLp
      - Hu, Anderson, Havens, Pinar
        - IPMU 2014, FUZZ-IEEE 2013
  - Non-linear
    - DeFIMKL: FI-based decision-level fusion
      - Pinar, Havens, Anderson, Hu
        - FUZZ-IEEE 2015



MKL: linear convex sum

$$K(x_i, x_j) = \sum_{e=1}^{E} \omega_e K_e(x_i^e, x_j^e)$$

Linear weighted sum of base Mercer kernels

$$\omega_e \in \mathcal{R}^+ \quad \sum_{e=1}^{E} \omega_e = 1$$

Constraints

$$\langle \Phi_{\vec{\omega}}(x_i), \Phi_{\vec{\omega}}(x_j) \rangle = \sum_{e=1}^{E} \omega_e K_e(x_i^e, x_j^e)$$

Inner product

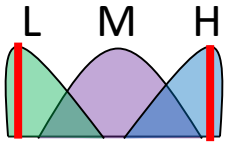
What is it really doing?

$$\begin{pmatrix} \sqrt{\omega_1}\Phi_1(x_i^1) \\ \sqrt{\omega_2}\Phi_2(x_i^2) \\ \vdots \\ \sqrt{\omega_E}\Phi_E(x_i^E) \end{pmatrix}^T \begin{pmatrix} \sqrt{\omega_1}\Phi_1(x_j^1) \\ \sqrt{\omega_2}\Phi_2(x_j^2) \\ \vdots \\ \sqrt{\omega_E}\Phi_E(x_j^E) \end{pmatrix}$$

1) Simple weighted sum of inner products
2) Concatenation of transformed spaces
3) Inner product of the scaled feature spaces
4) Weights often interpreted as “importance”
5) No interaction between sources exploited

…

(overly) simple approach



MKL: valid math operators

(Sum)

$$\mathcal{K}_{ij} = (K_1)_{ij} + (K_2)_{ij}$$

(Scalar Product)

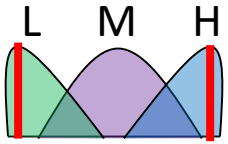
$$\mathcal{K}_{ij} = c(K_1)_{ij}, \forall c > 0$$

(Addition by Constant)

$$\mathcal{K}_{ij} = (K_1)_{ij} + c, \forall c > 0$$

(Product)

$$\mathcal{K}_{ij} = (K_1)_{ij}(K_2)_{ij}$$

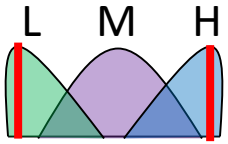


MKL: SVM and MK

- Simple extension, SKSVM -> MKSVM

$$\min_{\sigma \in \Delta} \max_{\alpha} \left\{ \mathbf{1}^T \alpha - \frac{1}{2} (\alpha \circ \mathbf{y})^T \left(\sum_{k=1}^{m} \sigma_k K_k \right) \alpha \circ \mathbf{y}) \right\}$$

$$0 \leq \alpha_i \leq C, \; i = 1, \ldots, n; \; \alpha^T \mathbf{y} = 0$$



Group Lasso: MKLGL

- Alternating optimization

Algorithm 1: MKLGL Classifier Training

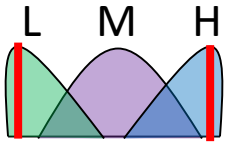
Data: (x_i, y_i) - feature vector and label pairs; K_k - kernel matrices
Result: α - MKLGL classifier solution; σ - kernel weight vector
Initialize $\sigma_k = 1/m,\ k = 1, ..., m$ - set kernel weights equal
while *not done* **do**

Solve unbalanced SKSVM for kernel matrix $K = \sum_{k=1}^{m} \sigma_k K_k$ for the optimal solution α

Update the kernel weights, σ_k using

$$\sigma_k = \frac{f_k^{2/(1+p)}}{\left(\sum_{k=1}^{m} f_k^{2p/(1+p)}\right)^{1/p}},\ k = 1, ..., m;$$

$$f_k = \sigma_k^2 (\alpha \circ \mathrm{y})^T K_k (\alpha \circ \mathrm{y}).$$



DeFIMKL

- Pinar, Havens, Anderson et al. [FUZZ-IEEE 2015]
- Decision-level based FI-based MKL

Algorithm 2: DeFIMKL Classifier Training

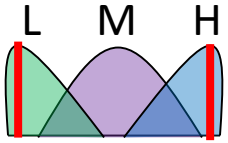
Data: (x_i, y_i) - feature vector and label pairs; K_k - kernel matrices
Result: u - Lexicographically ordered g vector
for *each kernel matrix* **do**
  Compute the kernel SVM classifier decision values, η_k, as in Equation (6).
  Remap the decision values onto the interval $[-1, +1]$ as f_k using Equation (7). ← Sigmoid scaling
Solve the minimization problem in[11] for u (i.e., g). ← Quadratic programming

Algorithm 3: DeFIMKL Classifier Testing

Compute the SVM decision values $f_k(\mathrm{x})$ by using Equations (6) and (7).
Apply the Choquet integral at Equation (5) with respect to the learned g.
Compute the class label by $\mathrm{sgn}\{f_g(\mathrm{x})\}$.

$$\eta_k(\mathbf{x}) = \sum_{i=1}^{n} \alpha_{ik} y_i \kappa_k(\mathbf{x}_i, \mathbf{x}) - b_k$$

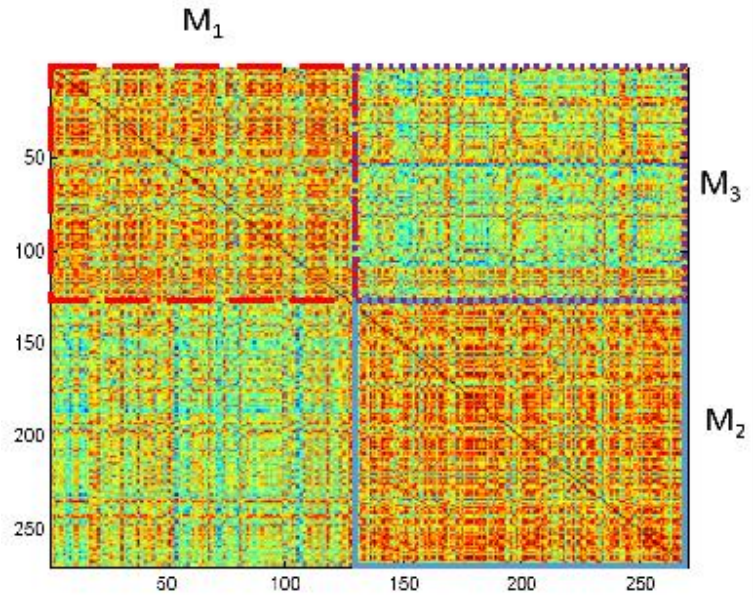
$$f_k(\mathbf{x}) = \frac{\eta_k(\mathbf{x})}{\sqrt{1 + \eta_k^2(\mathbf{x})}}$$

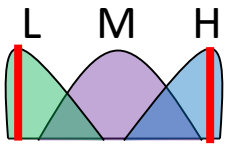


$$f_g(\mathbf{x}_i) = \sum_{k=1}^{m} f_{\pi(k)}(\mathbf{x}_i)\,[g(A_k) - g(A_{k-1})]$$

MKL: Heuristic approach

- Price, Anderson, Havens, et al. [SPIE 2015/16]



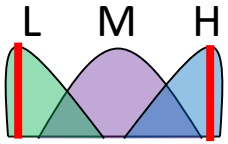


Algorithm 4: MK Weight Assignment Algorithm from Kernel Matrices

Data: (y_i) - feature vector labels and K_k - kernel matrices
Result: σ_k - kernel weights
for k=*1 to K* **do**
Extract submatrix from K_k for (class 1, class 1) data as an unordered set of individual values, M_1
Extract submatrix from K_k for (class 2, class 2) data as an unordered set of individual values, M_2
Extract submatrix from K_k for (class 1, class 2) data as an unordered set of individual values, M_3
%%Extract means and standard deviations (std)
$\mu_1 = \frac{1}{|M_1|}\sum_{i=1}^{|M_1|} M_{1,i}$ %% mean of submatrix M_1
$\tau_1 = \sqrt{\frac{1}{|M_1|}\sum_{i=1}^{|M_1|}(M_{1,i}-\mu_1)^2}$ %% std of submatrix M_1
$\mu_2 = \frac{1}{|M_2|}\sum_{i=1}^{|M_1|} M_{2,i}$ %% mean of submatrix M_2
$\tau_2 = \sqrt{\frac{1}{|M_2|}\sum_{i=1}^{|M_2|}(M_{2,i}-\mu_2)^2}$ %% std of submatrix M_2
$\mu_3 = \frac{1}{|M_3|}\sum_{i=1}^{|M_1|} M_{3,i}$ %% mean of submatrix M_3
$\tau_3 = \sqrt{\frac{1}{|M_3|}\sum_{i=1}^{|M_3|}(M_{3,i}-\mu_3)^2}$ %% std of submatrix M_3
%%Compute the Bhattacharya distance for two normal distributions
$b_1 = \frac{(\mu_1-\mu_3)^2}{4(\tau_1^2+\tau_3^2)} + 0.5\ln\left(\frac{\tau_1^2+\tau_3^2}{2\sqrt{\tau_1^2\tau_3^2}}\right)$
$b_2 = \frac{(\mu_2-\mu_3)^2}{4(\tau_2^2+\tau_3^2)} + 0.5\ln\left(\frac{\tau_2^2+\tau_3^2}{2\sqrt{\tau_2^2\tau_3^2}}\right)$
%%Compute the kernel weight
$\sigma_k = \frac{b_1+b_2}{b_1+b_2+\tau_1+\tau_2+\tau_3}$
%%Weight normalization
$\sigma_k = \sigma_k / \sum_{i=1}^{K}\sigma_i$

art of MKL

- Extremely prone to overfitting
  - 100% performance on training and not hold up on testing
    - Cross-validation, regularization, etc.
  - Need to take into consideration model complexity
- Some look to MKL for feature selection …
  - Kernel over specification
- In practice
  - Most try a homogeneous combination of RBFs
    - Typically, one sigma is good … however, sometimes different sigmas for different subsets pays out
  - Really need to find the "correct kernel configuration"
    - Not an easy problem whatsoever

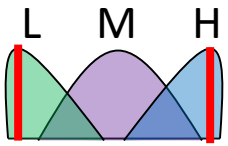


Heterogeneous kernels and normalization

- Might need one RBF, two parts poly, one dot, etc.
- Different scaling's
- MKL SHOULD be able to deal with this … but …
- Sometimes higher values win out
- Can pre-process: 0 mean and 1 variance in RHKS

$$\frac{1}{n}\sum_{i=1}^{n}\kappa_k(\mathbf{x}_i,\mathbf{x}_i)+\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\kappa_k(\mathbf{x}_i,\mathbf{x}_j)=1$$

$$\kappa(\mathbf{x},\bar{\mathbf{x}})\rightarrow\frac{\kappa(\mathbf{x},\bar{\mathbf{x}})}{\frac{1}{n}\sum_{i=1}^{n}\kappa(\mathbf{x}_i,\mathbf{x}_i)+\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\kappa(\mathbf{x}_i,\mathbf{x}_j)}$$



Benchmark datasets

TABLE III: RBF Kernel Parameter Ranges

Data Set					
Sonar	**Dermatology**	**Wine**	**Ionosphere**	**Ecoli**	**Glass**
[-2.2, -0.5]	[-2.3, 0.2]	[-2.5, 2]	[-2.1, 1.2]	[-3, 3]	[-2, 2]

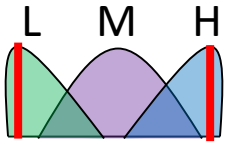
TABLE II: UCI Benchmark Data Sets

	Data Set					
	Sonar	**Dermatology**	**Wine**	**Ionosphere**	**Ecoli**	**Glass**
No. of Objects	208	366	178	351	336	214
No. of Features	60	33	13	34	7	9
Binary Classes	{1} vs. {2}	{1,2,3} vs. {4,5,6}	{1} vs. {2,3}	{0} vs. {1}	{1,2,3,4} vs. {5,6,7,8}	{1,2,3} vs. {4,5,6}

TABLE IV: Classification Accuracy Results on Benchmark Data Sets*

	Data Set					
Algorithm	Sonar	Derm	Wine	Ionosphere	Ecoli	Glass
$MKLGL_1$	83.0 (5.81)	**97.3** (1.99)	**99.6** (0.97)	**95.2** (2.36)	**97.1** (1.71)	**94.5** (3.29)
$MKLGL_2$	84.6 (5.11)	**97.2** (1.60)	**99.6** (1.02)	**95.5** (2.40)	**97.2** (1.80)	**94.0** (3.53)
$GAMKL_1$	84.0 (6.00)	**97.1** (1.70)	99.4 (1.16)	94.8 (2.59)	**97.1** (1.93)	**94.0** (3.87)
$GAMKL_1^*$	84.6 (5.67)	**97.3** (1.75)	**99.6** (1.00)	94.8 (2.53)	96.9 (1.86)	93.3 (3.99)
$GAMKL_2$	**86.0** (5.64)	**97.1** (1.55)	**99.5** (1.10)	**95.1** (2.29)	**97.5** (1.60)	**94.0** (3.24)
$GAMKL_2^*$	**86.4** (5.62)	**96.8** (1.84)	99.4 (1.16)	**95.7** (2.39)	**97.4** (1.68)	**94.2** (3.49)
DeFIMKL	78.9 (5.66)	93.2 (3.07)	99.4 (1.17)	92.3 (7.13)	**97.3** (1.77)	91.2 (3.78)
$DeFIMKL_1$	**84.9** (6.03) $\lambda = 2$	84.2 (4.12) $\lambda = 0.5$	**99.5** (1.10) $\lambda = 0.5$	88.8 (3.26) $\lambda = 4$	91.8 (3.00) $\lambda = 3$	78.1 (6.20) $\lambda = 0.5$
$DeFIMKL_2$	84.4 (6.82) $\lambda = 1$	87.6 (3.80) $\lambda = 0.5$	**99.7** (0.87) $\lambda = 1.5$	90.0 (3.35) $\lambda = 0.5$	91.9 (3.26) $\lambda = 2.5$	83.1 (4.83) $\lambda = 0.5$

*Bold indicates best result according to a two-valued t-test at a 5% significance level.



[Pinar, Havens, Anderson, Hu, FUZZ-IEEE 2015]

Regularized solutions

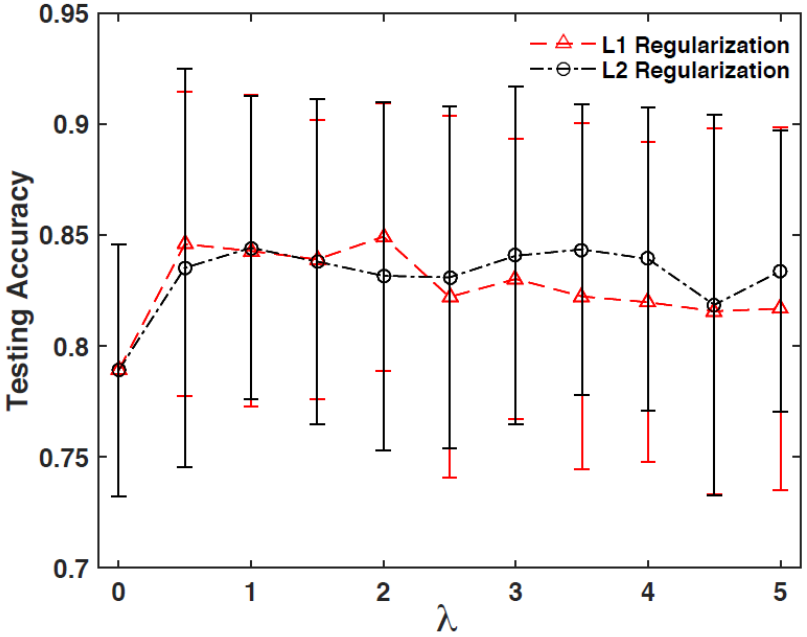


Fig. 2: DeFIMKL performance using regularization on Sonar data. Error bars indicate ± one standard deviation.

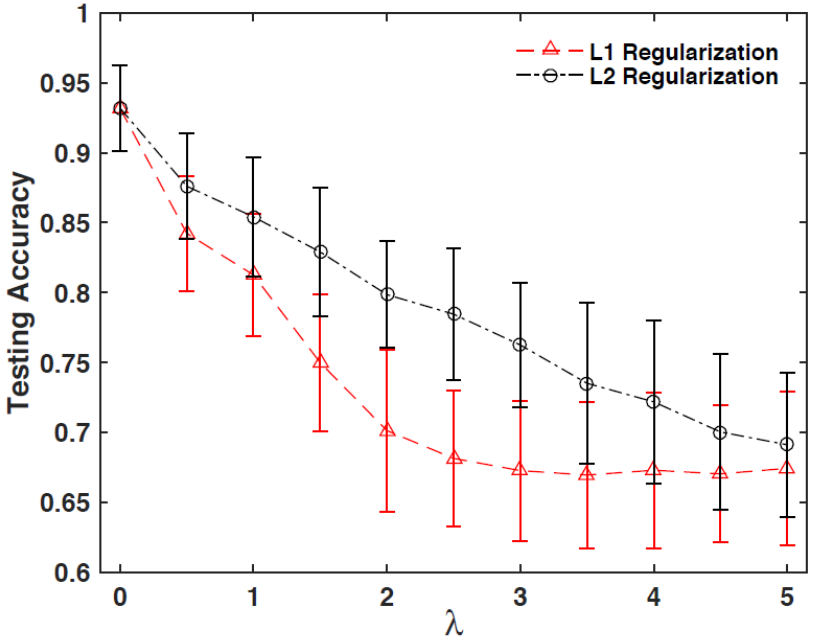
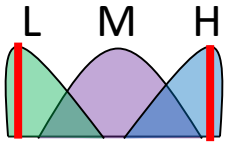
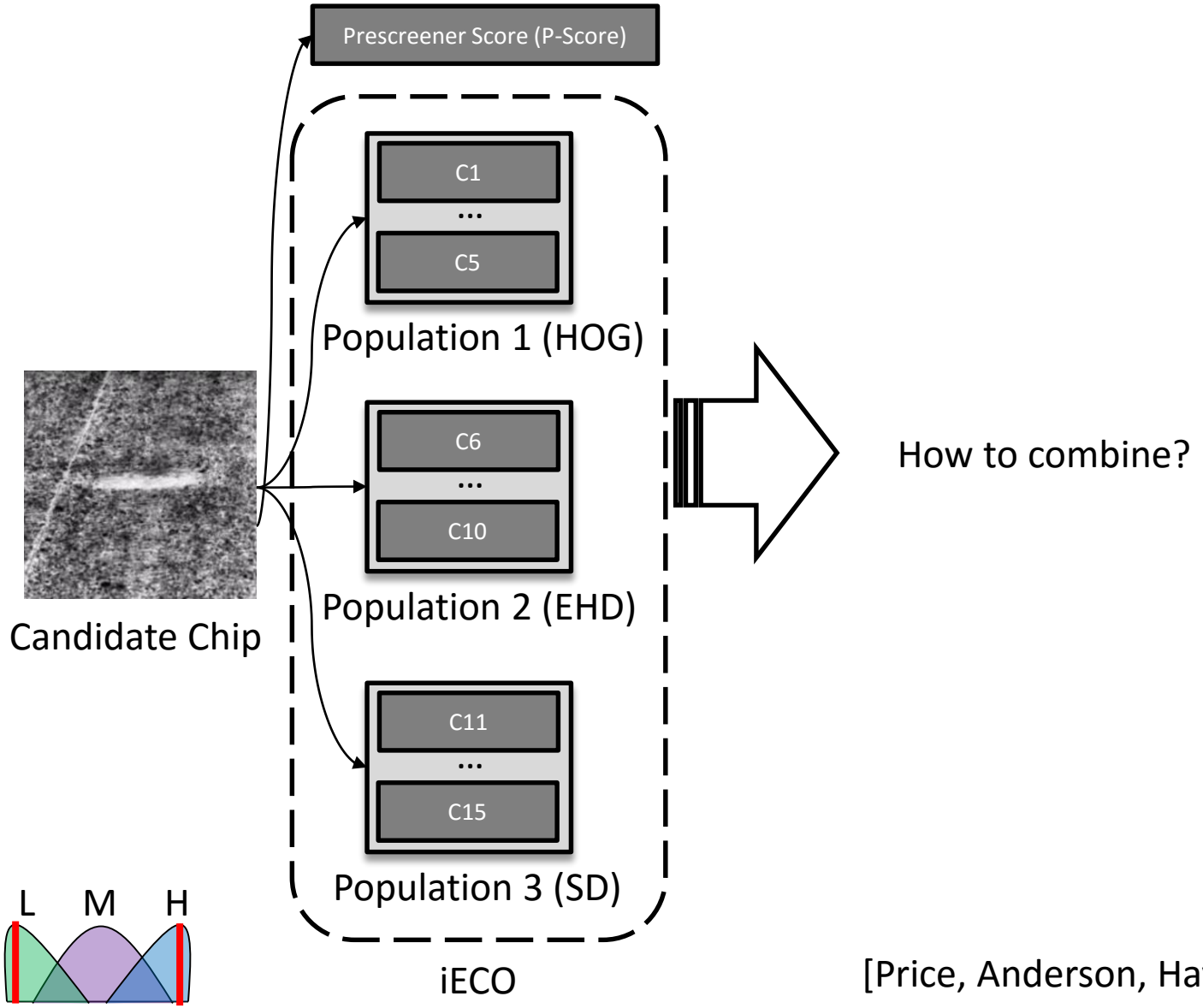


Fig. 3: DeFIMKL performance using regularization on Dermatology data—classes $\{1, 2, 3\}$ versus $\{4, 5, 6\}$. Error bars indicate ± one standard deviation.



[Pinar, Havens, Anderson, Hu, FUZZ-IEEE 2015]

MKL fusion of iECO features



[Price, Anderson, Havens, SPIE 2015 & 2016]

MKL fusion of iECO features

Candidate Chip

Prescreener Score (P-Score)

C1
...
C5

Population 1 (HOG)

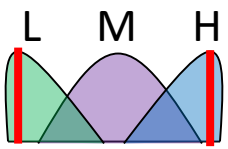
C6
...
C10

Population 2 (EHD)

C11
...
C15

Population 3 (SD)

iECO

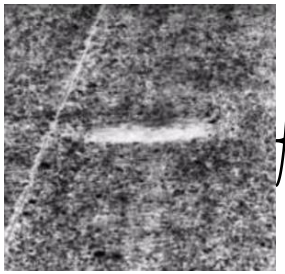


C1 C2 ... C15 P-Score → SK SVM

Feature vector concatenation and SKSVM

[Price, Anderson, Havens, SPIE 2015 & 2016]

MKL fusion of iECO features



Candidate Chip

Prescreener Score (P-Score)

C1
...
C5

Population 1 (HOG)

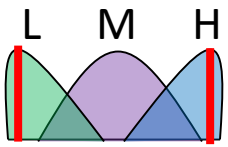
C6
...
C10

Population 2 (EHD)

C11
...
C15

Population 3 (SD)

iECO



C1
C2
...
C5
C6
C7
...
C10
...
C15
P-Score

MKL

MK SVM

Each feature

[Price, Anderson, Havens, SPIE 2015 & 2016]

MKL fusion of iECO features

Candidate Chip

Prescreener Score (P-Score)

C1 … C5

Population 1 (HOG)

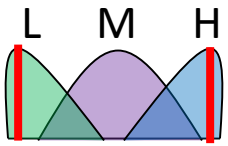
C6 … C10

Population 2 (EHD)

C11 … C15

Population 3 (SD)

iECO



C1 C2 … C5

C6 C7 … C10

C11 C12 … C15

P-Score

MKL

MK SVM

Feature-level fusion and grouping

[Price, Anderson, Havens, SPIE 2015 & 2016]

MKL fusion of iECO features

Candidate Chip

Prescreener Score (P-Score)

C1 … C5

Population 1 (HOG)

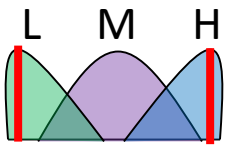
C6 … C10

Population 2 (EHD)

C11 … C15

Population 3 (SD)

iECO



C1 C2 … C5

C6 C7 … C10

C11 C12 … C15

P-Score

SKSVM

SKSVM

SKSVM

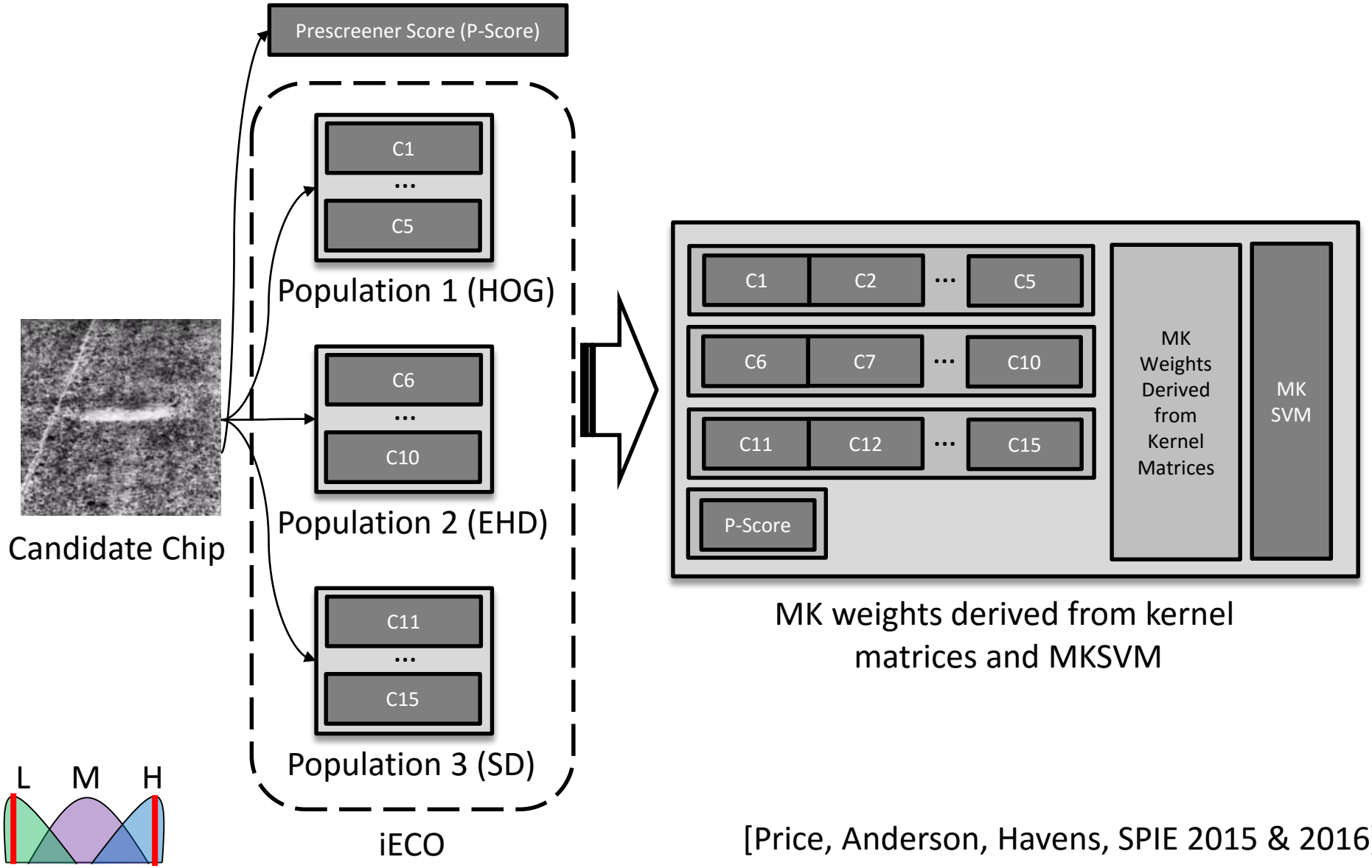
SKSVM

DeFIMKL

DeFIMKL and SKSVM (decision level fusion)

[Price, Anderson, Havens, SPIE 2015 & 2016]

MKL fusion of iECO features



[Price, Anderson, Havens, SPIE 2015 & 2016]

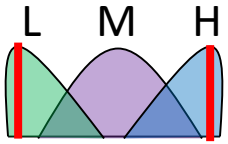
MKL fusion of iECO features: Experiments

- U.S. Army test site
- Multiple target and clutter types, burial depths and times of day
- Varied in burial depth and metal content

Table 1. Data collection summary for each lane.

Lane	Number of Targets	Area (m^2)	Metal Shallow	Metal Deep	Non-Metal Shallow	Non-Metal Deep
A	44	3626.9	21	3	11	9
B	50	4212.7	22	4	14	10
C	79	3944.8	31	15	21	12

- Lane-based cross validation results
- Receiver operating characteristic (ROC) curves
- Many MKL combinations, report the top performers



[Price, Anderson, Havens, SPIE 2015 & 2016]

Kernels per-group

- 4 RBF kernels
  - One for each group of iECO descriptors (top five individuals each)
  - One for the pre-screener score

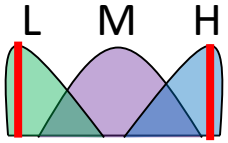
NAUC scores

Learning Strategy	Weight Assignment	Fold-1	Fold-2	Fold-3
Fixed-Rule	Uniform	0.328	0.608	0.610
Heuristic	Kernel Matrix-Based	0.330	0.611	0.610
	Normalized Base Learner Acc.	0.306	0.587	0.597
Optimization Function	MKLGL	0.318	0.595	0.578
	MKLGL/Kernel Matrix-Based Seed	0.319	0.595	0.579
	DeFIMKL	0.317	0.607	0.614

We hypothesized that iECO features were all relatively important (versus some are duds)
Trained with diversity in mind and all seem to find a way to look at the problem differently
Heuristic weights were not uniform (but close)

Table 3. Resultant kernel weights for MKLGL and weight assignment method put forth herein

Technique	Chromosome 1 (HOG)	Chromosome 2 (EHD)	Chromosome 3 (SD)	Prescreener score
MKLGL (Fold-1)	0.9877	0.0009	0.0000	0.0114
MKLGL (Fold-2)	0.9759	0.0008	0.0000	0.0233
MKLGL (Fold-3)	0.0050	0.9608	0.0000	0.0341
MK weights (Fold-1)	0.2640	0.2715	0.2810	0.1835
MK weights (Fold-2)	0.2658	0.2636	0.2923	0.1782
MK weights (Fold-3)	0.2600	0.2566	0.2839	0.1995



Take note of performance of uniform relative to optimization function
Overall, different problems demand different solutions (fusions)

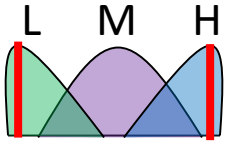
Kernels per-feature

- 16 RBF kernels
  - One for each iECO descriptor's top five individuals (thus 15 inputs)
  - One for the pre-screener score

Learning Strategy	Weight Assignment	Fold-1	Fold-2	Fold-3
Fixed-Rule	Uniform	0.333	0.612	0.615
Heuristic	Kernel Matrix-Based	0.335	0.616	0.617
	Normalized Base Learner Acc.	0.332	0.613	0.615
Optimization Function	MKLGL	0.317	0.5826	0.5991
	MKLGL/Kernel Matrix-Based Seed	0.327	0.579	0.596
	DeFIMKL	Too many inputs to solve for		

previous slide

Learning Strategy	Weight Assignment	Fold-1	Fold-2	Fold-3
Fixed-Rule	Uniform	0.328	0.608	0.610
Heuristic	Kernel Matrix-Based	0.330	0.611	0.610
	Normalized Base Learner Acc.	0.306	0.587	0.597
Optimization Function	MKLGL	0.318	0.595	0.578
	MKLGL/Kernel Matrix-Based Seed	0.319	0.595	0.579
	DeFIMKL	0.317	0.607	0.614



[Price, Anderson, Havens, SPIE 2015 & 2016]

Another example: diversity of inputs to MKL

- Hyperspectral image processing
- Band grouping
- Different proximity metrics

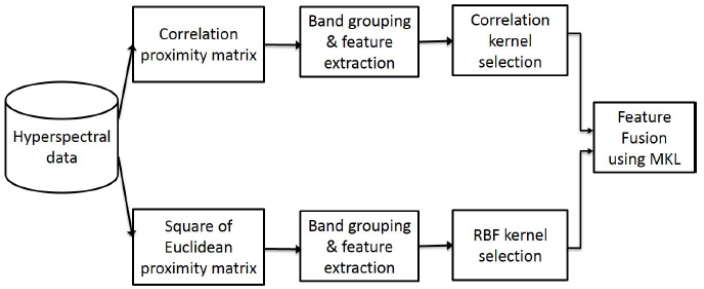
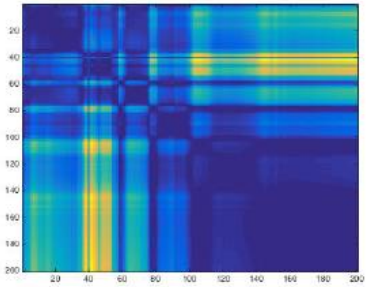


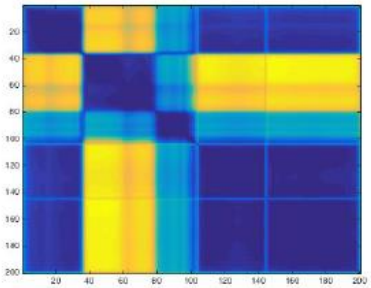
Table 1. Producer's accuracies for ℓ_p-norm MKL based fusion.

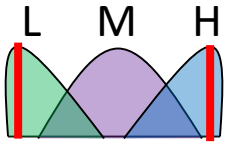
ℓ_p-norm	Method (SqE = Square of Euclidean, Corr = Correlation)	# of kernels	corn (notill)	corn (min)	grass (pasture)	grass (trees)	hay (windowed)	soybeans (notill)	soybeans (min)	soybeans (clean)	woods
NA	SqE	1	64.52	46.18	75.57	92.80	97.19	63.82	78.57	28.31	97.78
	Corr	1	62.42	34.93	81.11	85.09	96.16	62.02	66.46	30.55	97.49
$p = 1.1$	Fusion of SqE & Corr	1 + 1	68.35	46.48	79.60	90.79	96.16	64.99	79.08	45.42	**97.87**
	SqE	2	65.91	52.62	87.41	92.29	97.44	64.34	80.09	37.27	97.78
	Corr	2	62.77	36.13	82.12	87.27	96.16	62.02	66.41	30.55	97.49
	Fusion of SqE & Corr	2 + 2	68.70	52.02	88.41	93.13	96.42	64.99	80.45	46.64	97.87
$p = 2$	Fusion of SqE & Corr	1 + 1	69.92	53.37	82.87	91.96	96.16	69.77	80.14	52.55	97.20
	SqE	2	69.14	57.12	88.66	92.80	97.70	69.51	81.81	46.64	97.78
	Corr	2	65.82	39.28	87.15	88.11	97.19	63.70	66.67	37.07	97.49
	Fusion of SqE & Corr	2 + 2	73.50	62.82	91.18	93.97	97.70	72.22	83.23	60.08	97.10
$p = 100$	Fusion of SqE & Corr	1 + 1	71.49	60.57	83.63	93.63	96.93	72.87	81.16	60.29	96.14
	SqE	2	72.97	64.32	89.67	94.47	97.70	73.00	83.13	55.80	97.29
	Corr	2	68.88	46.03	89.92	89.61	97.44	66.93	67.68	45.42	97.39
	Fusion of SqE & Corr	2 + 2	**77.24**	**69.42**	**92.44**	**94.97**	**97.95**	**76.49**	**85.11**	**66.40**	95.75

Sq of Euclidean



Correlation

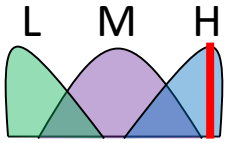




[Muhammad, Anderson, Ball, Younan, IGARSS and WHISPERS 2016]

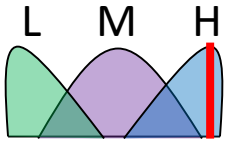
Linguistic summarization

- Motivation
  - **People** are often the recipient of our CV solutions
  - Don't know about you, but I do not want 1,000,000 decisions (e.g., detections) for 2 hours of video
  - Wouldn't it be wonderful if machines could do what they do and go a step further and convert their information into a **"representation"** that we find particularly useful?
  - Furthermore, can the machine go ahead and reduce the often massive amount of spatio-temporal redundancy?
    - Maybe 2 hours of video, but just one event!!!!
  - While we are at it, can we go further and compute with these quantities? [our linguistic summarizations versus the "raw signal/image data"]



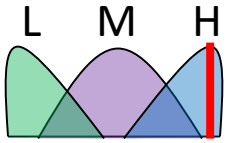
Linguistic summarization

- Different approaches exist, e.g.,
  - Anderson, Luke and Keller
    - Linguistic summarization of human behavior in spatio-temporal video data (specifically voxel space)
    - "Linguistic descriptions" (structured linguistic information)
  - Keller et al.
    - Linguistic summarization of time series data
    - "Linguistic protoforms" (structured linguistic information)



Linguistic summarization

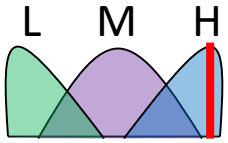
- Video *monitoring* scenario
  - 15fps for 8am-noon => 216,000 frames!!!
    - Information overload
    - Most is spatially and temporally redundant at that
- Well-being monitoring
  - Help elders live longer healthier independent lives
  - Any adverse events (e.g., falls)?
  - Where were they, for how long, what context?
  - We want events
  - Was one morning like another?
    - 1 event morning (sleeping on couch) [vs. 216,000!!!!]



[Anderson, Jim, et al., TFS, CVIU, WCCI, FUZZ-IEEE, etc.]

Linguistic summarization

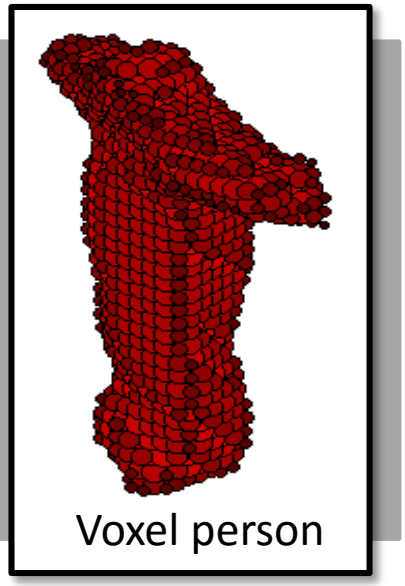
- Automatic summarization of video
  - Data reduction tool
  - Idea situation (for both experts – nurses - and pattern recognition algorithms)
    - "Jim was eating in the kitchen for a brief time in the early morning"
    - "Jim napped in the livingroom for a long time"
    - …
- Produce linguistic summarizations

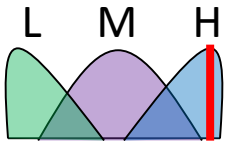


[Anderson, Jim, et al., TFS, CVIU, WCCI, FUZZ-IEEE, etc.]

Linguistic summarization

- What is an activity?
  - Uncertainty inherent in activity definitions
    - What is standing, jumping, fallen, squatting, …
- Can be thought of as the *compatibility* between a human and a set of activities
  - e.g., rule-based definitions and fuzzy logic

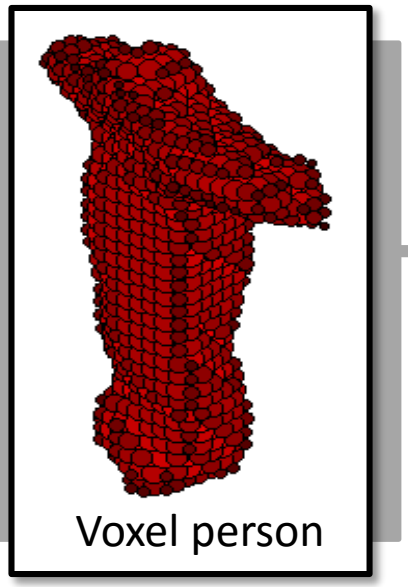

Voxel person



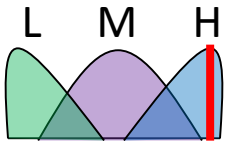
[Anderson, Jim, et al., TFS, CVIU, WCCI, FUZZ-IEEE, etc.]

Linguistic summarization

- What is an activity?
  - Uncertainty inherent in activity definitions
    - What is standing, jumping, fallen, squatting, …
- Can be thought of as the *compatibility* between a human and a set of activities
  - e.g., rule-based definitions and fuzzy logic



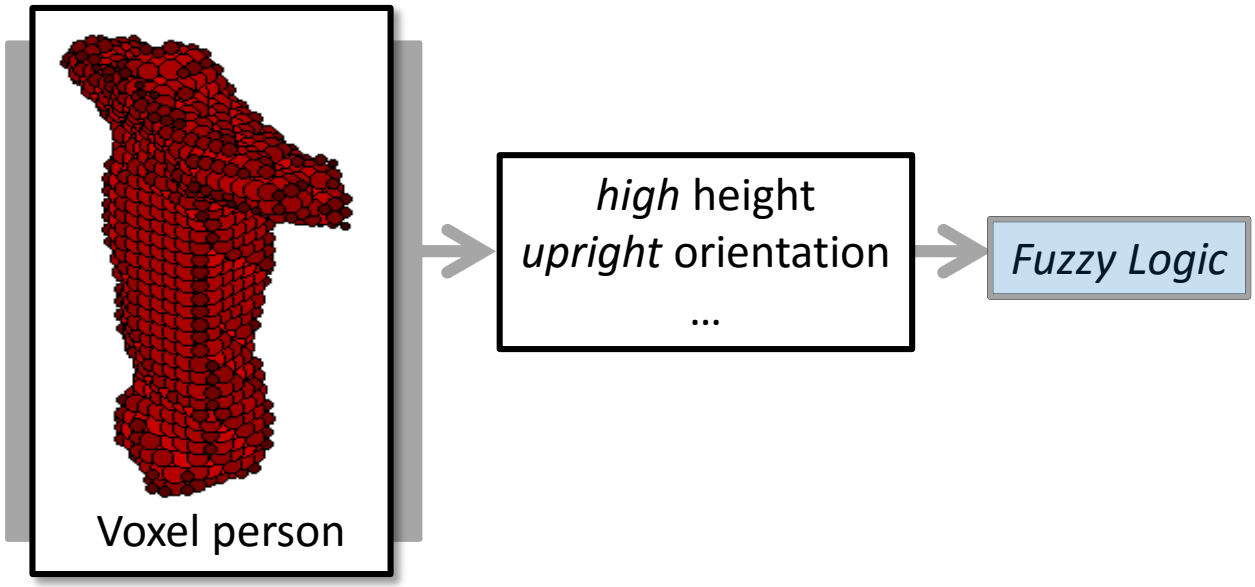
high height
upright orientation
…

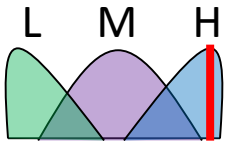


[Anderson, Jim, et al., TFS, CVIU, WCCI, FUZZ-IEEE, etc.]

Linguistic summarization

- What is an activity?
  - Uncertainty inherent in activity definitions
    - What is standing, jumping, fallen, squatting, …
- Can be thought of as the *compatibility* between a human and a set of activities
  - e.g., rule-based definitions and fuzzy logic

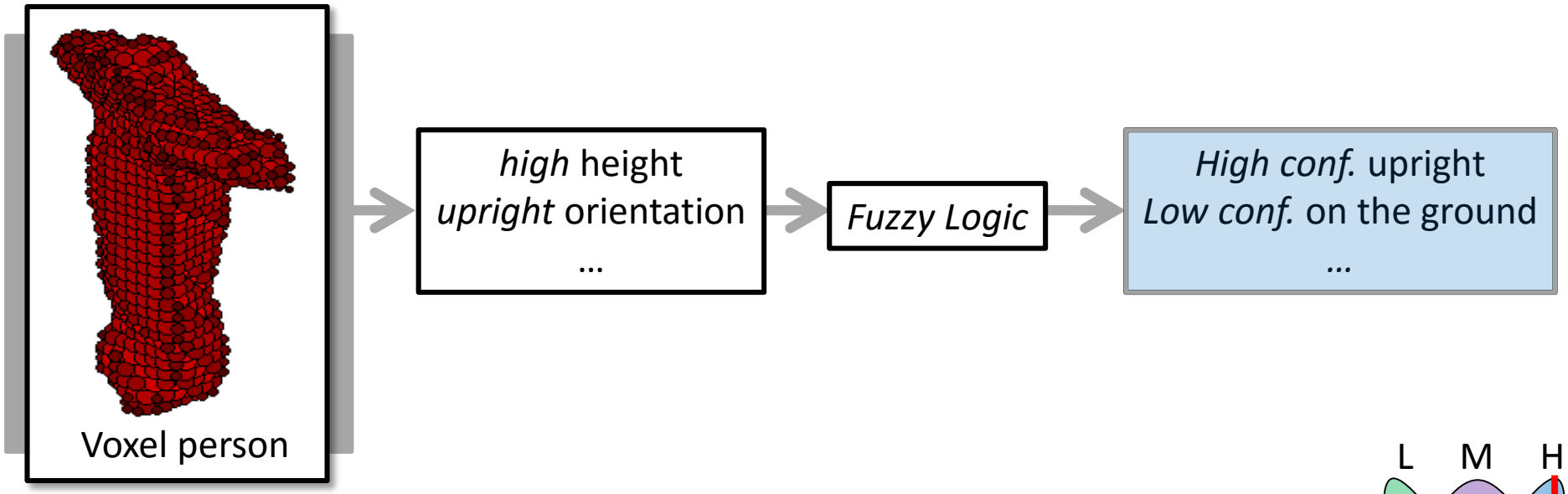


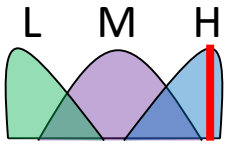


[Anderson, Jim, et al., TFS, CVIU, WCCI, FUZZ-IEEE, etc.]

Linguistic summarization

- What is an activity?
  - Uncertainty inherent in activity definitions
    - What is standing, jumping, fallen, squatting, …
- Can be thought of as the *compatibility* between a human and a set of activities
  - e.g., rule-based definitions and fuzzy logic

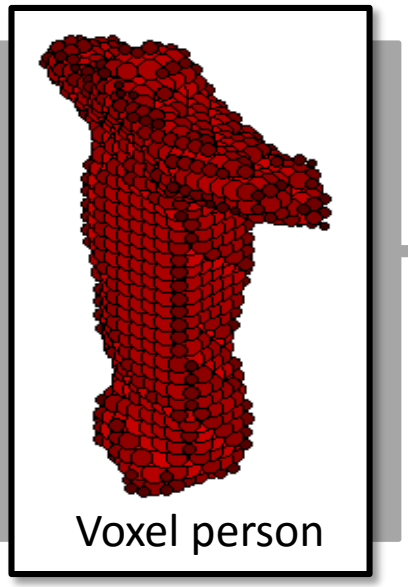




[Anderson, Jim, et al., TFS, CVIU, WCCI, FUZZ-IEEE, etc.]

Linguistic summarization

- What is an activity?
  - Uncertainty inherent in activity definitions
    - What is standing, jumping, fallen, squatting, …
- Can be thought of as the *compatibility* between a human and a set of activities
  - e.g., rule-based definitions and fuzzy logic

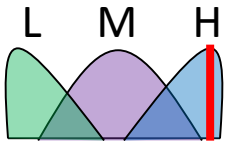


high height
upright orientation
…

Fuzzy Logic

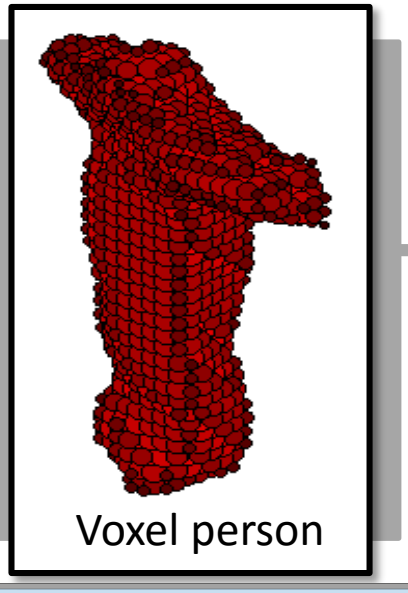
High conf. upright
Low conf. on the ground
…

[Human] is *[Activity]* in [Location] for *[Time]*



Linguistic summarization

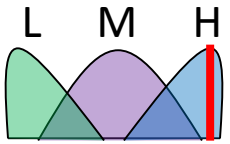
- What is an activity?
  - Uncertainty inherent in activity definitions
    - What is standing, jumping, fallen, squatting, …
- Can be thought of as the *compatibility* between a human and a set of activities
  - e.g., rule-based definitions and fuzzy logic



high height
upright orientation
…

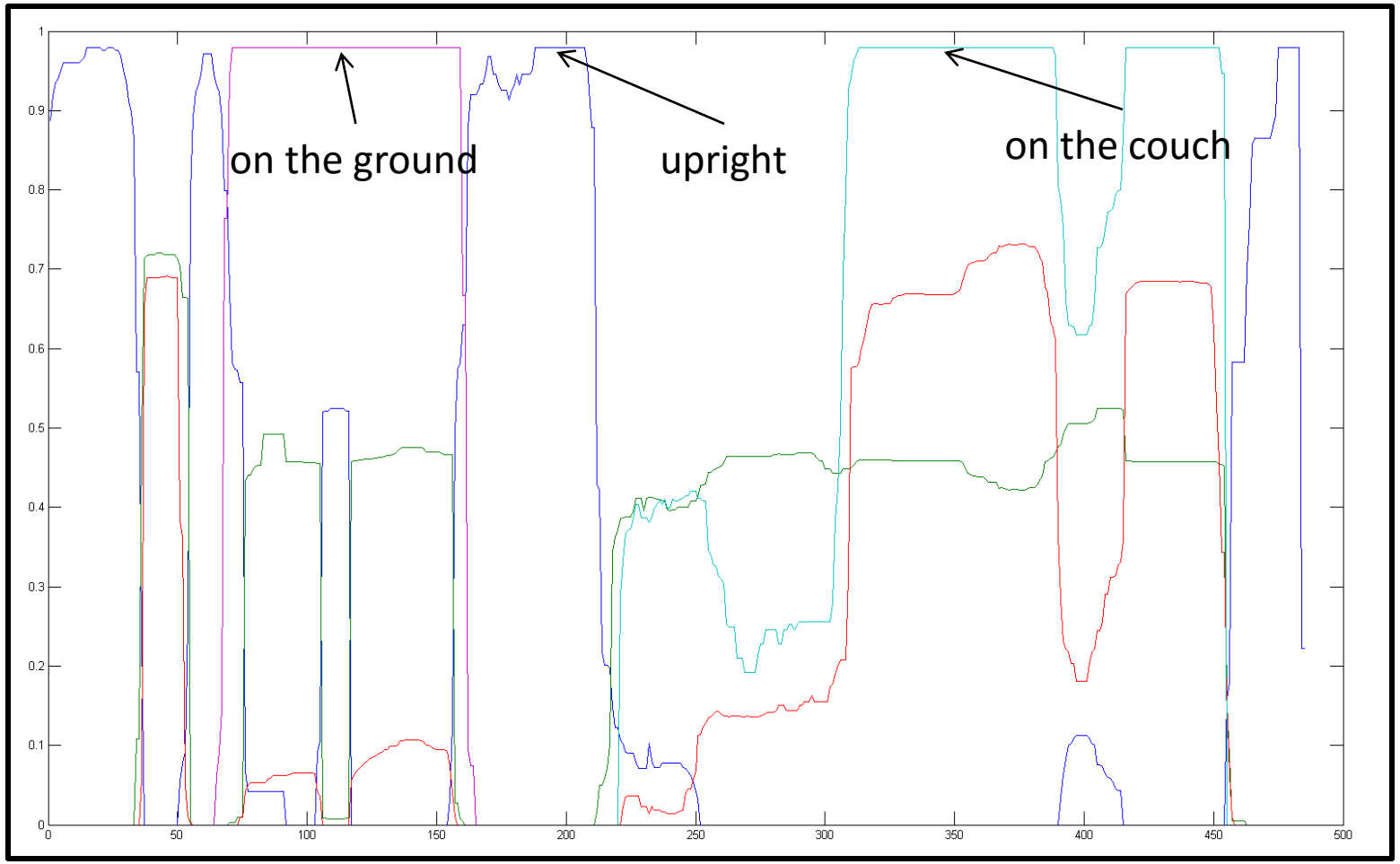
Fuzzy Logic

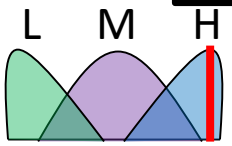
High conf. upright
Low conf. on the ground
…



Keller is *walking with a limp* in the living room for *a long time* in the *early morning*

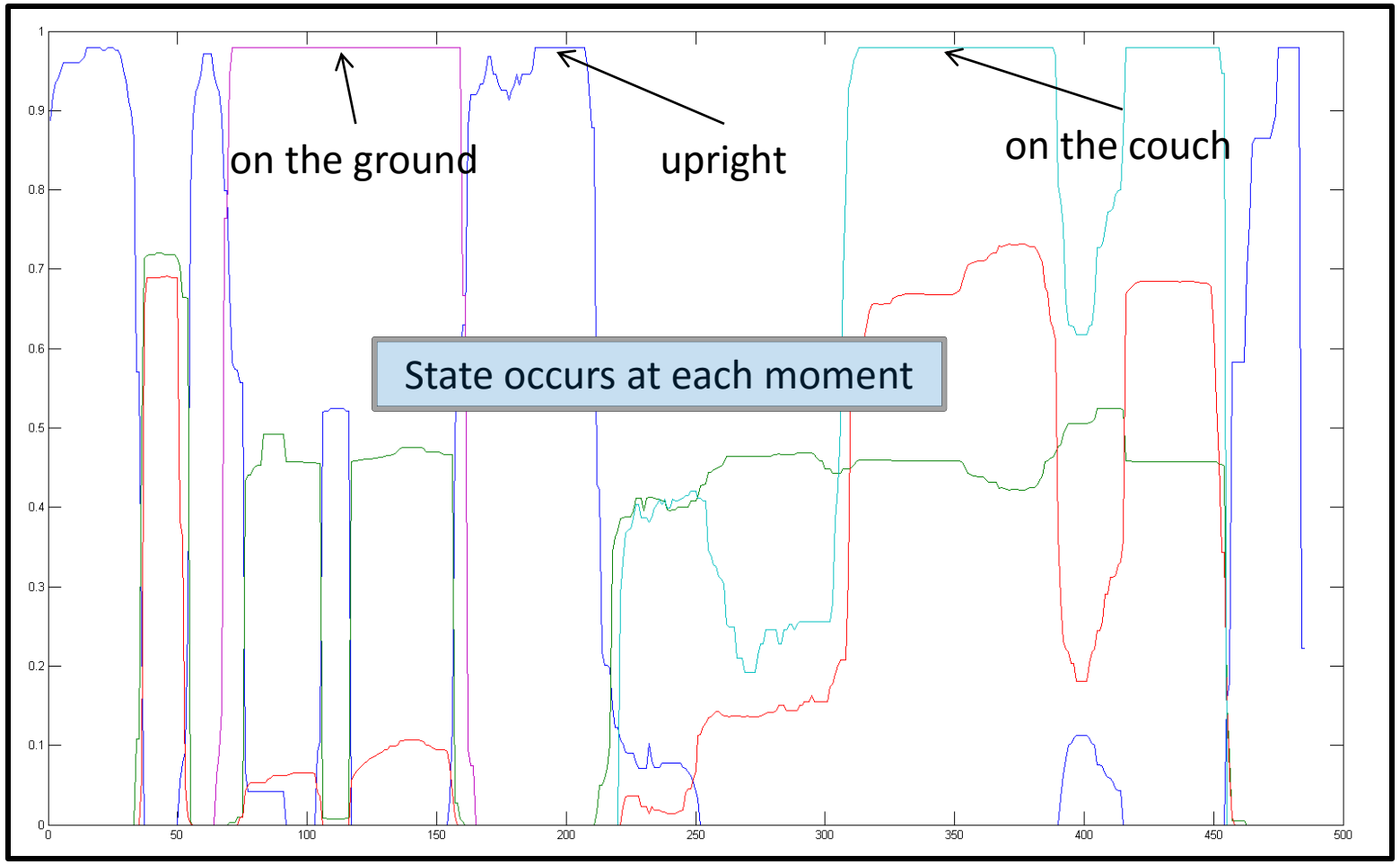
Linguistic summarization

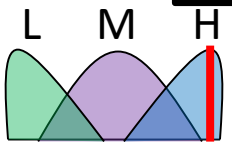




[Anderson, Jim, et al., TFS, CVIU, WCCI, FUZZ-IEEE, etc.]

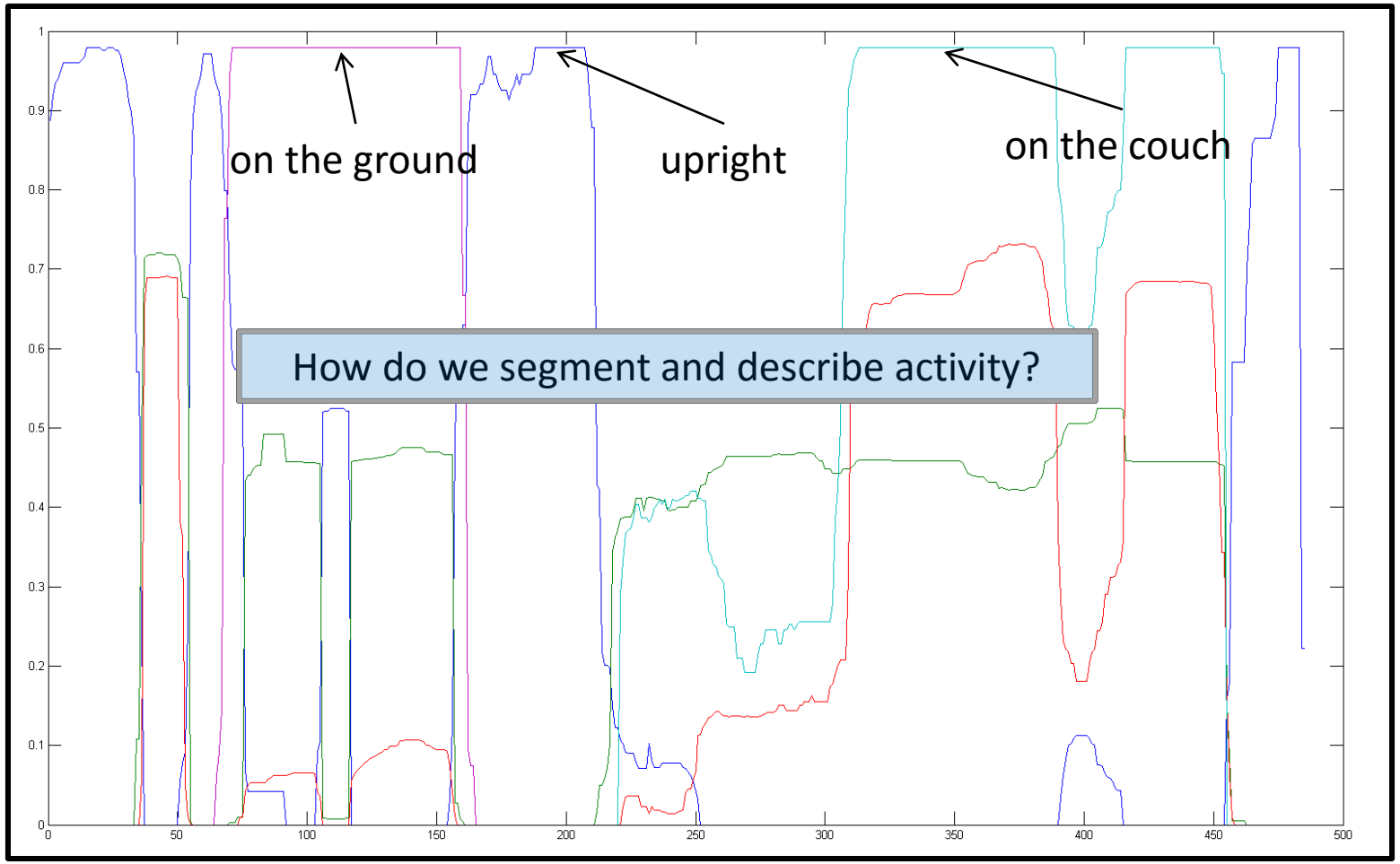
Linguistic summarization

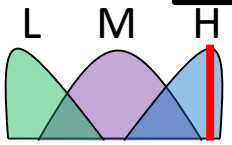




[Anderson, Jim, et al., TFS, CVIU, WCCI, FUZZ-IEEE, etc.]

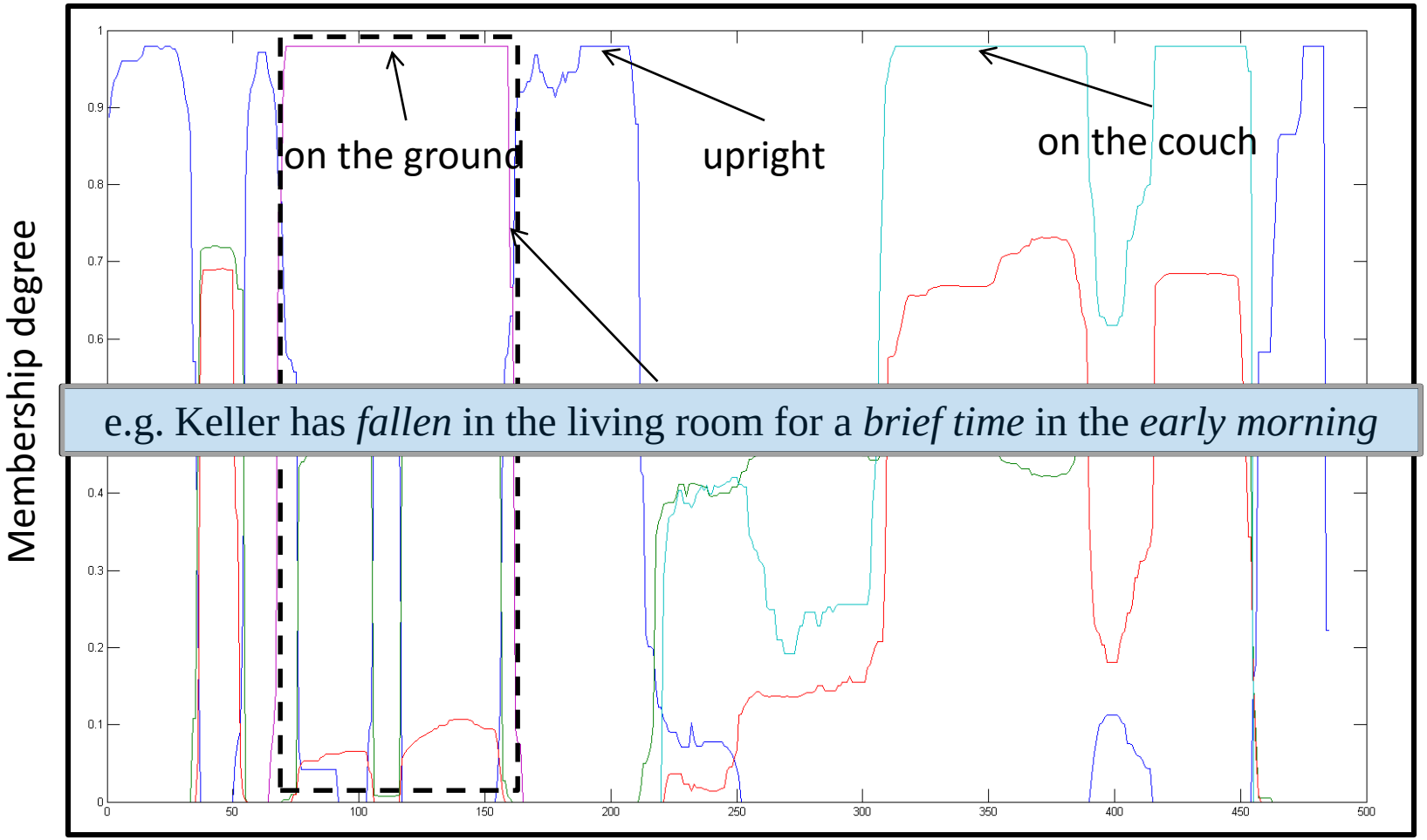
Linguistic summarization

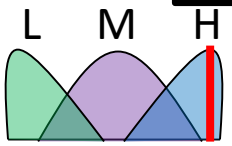




[Anderson, Jim, et al., TFS, CVIU, WCCI, FUZZ-IEEE, etc.]

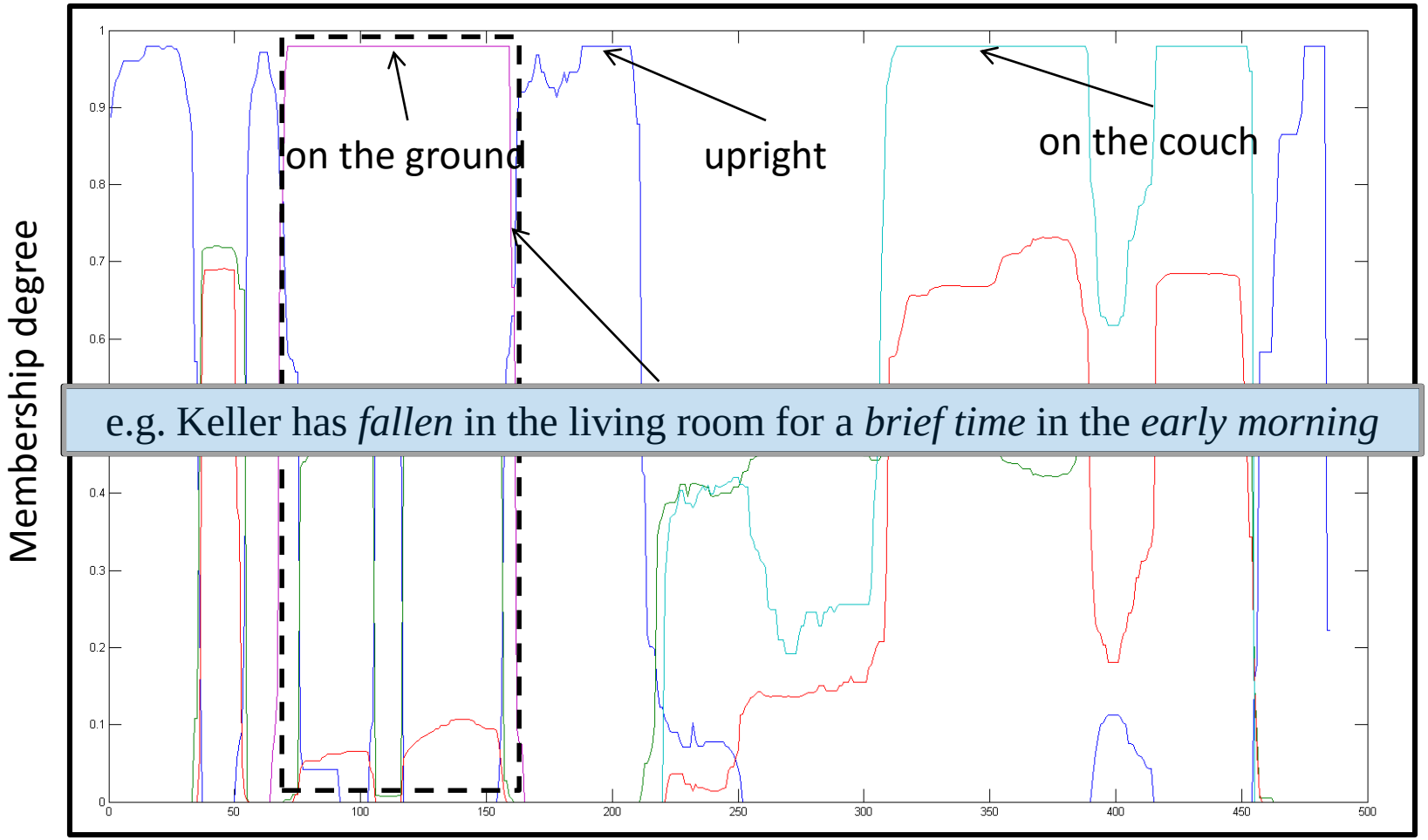
Linguistic summarization

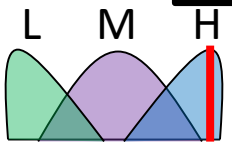




One idea: search for *crossings* in the membership functions (e.g., max at each step)

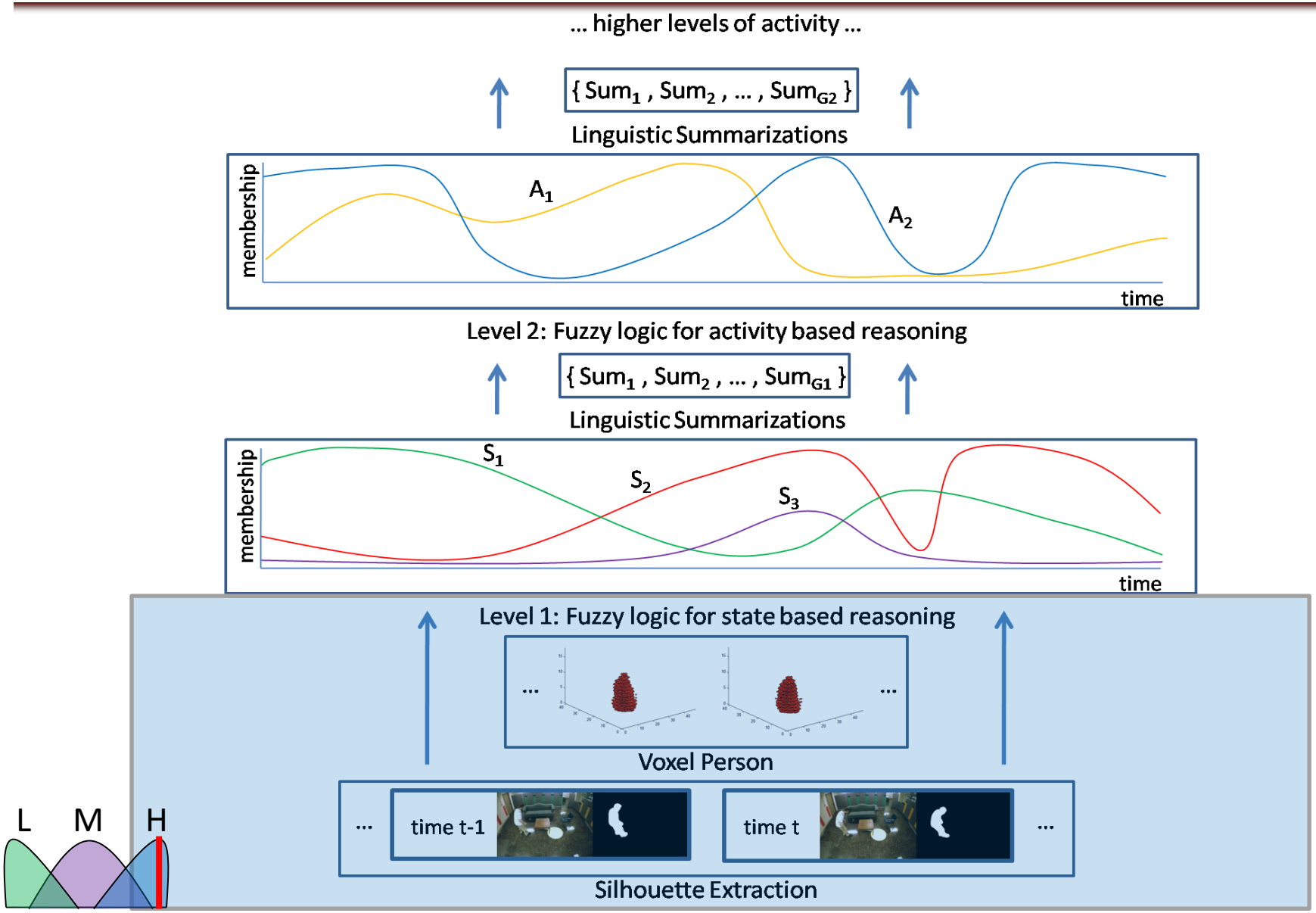
Linguistic summarization



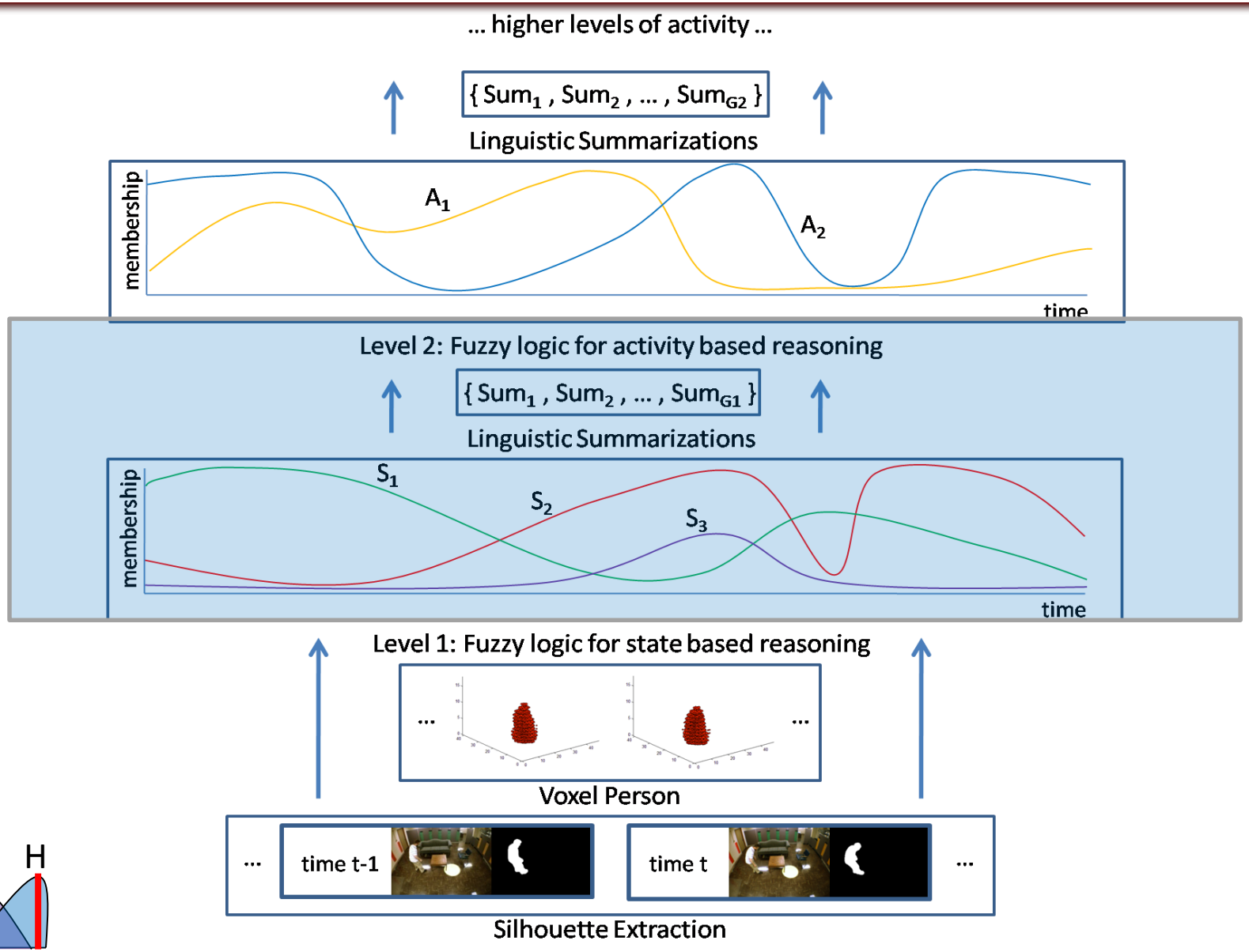


Could also treat as a signal processing task – or mine like a financial time series

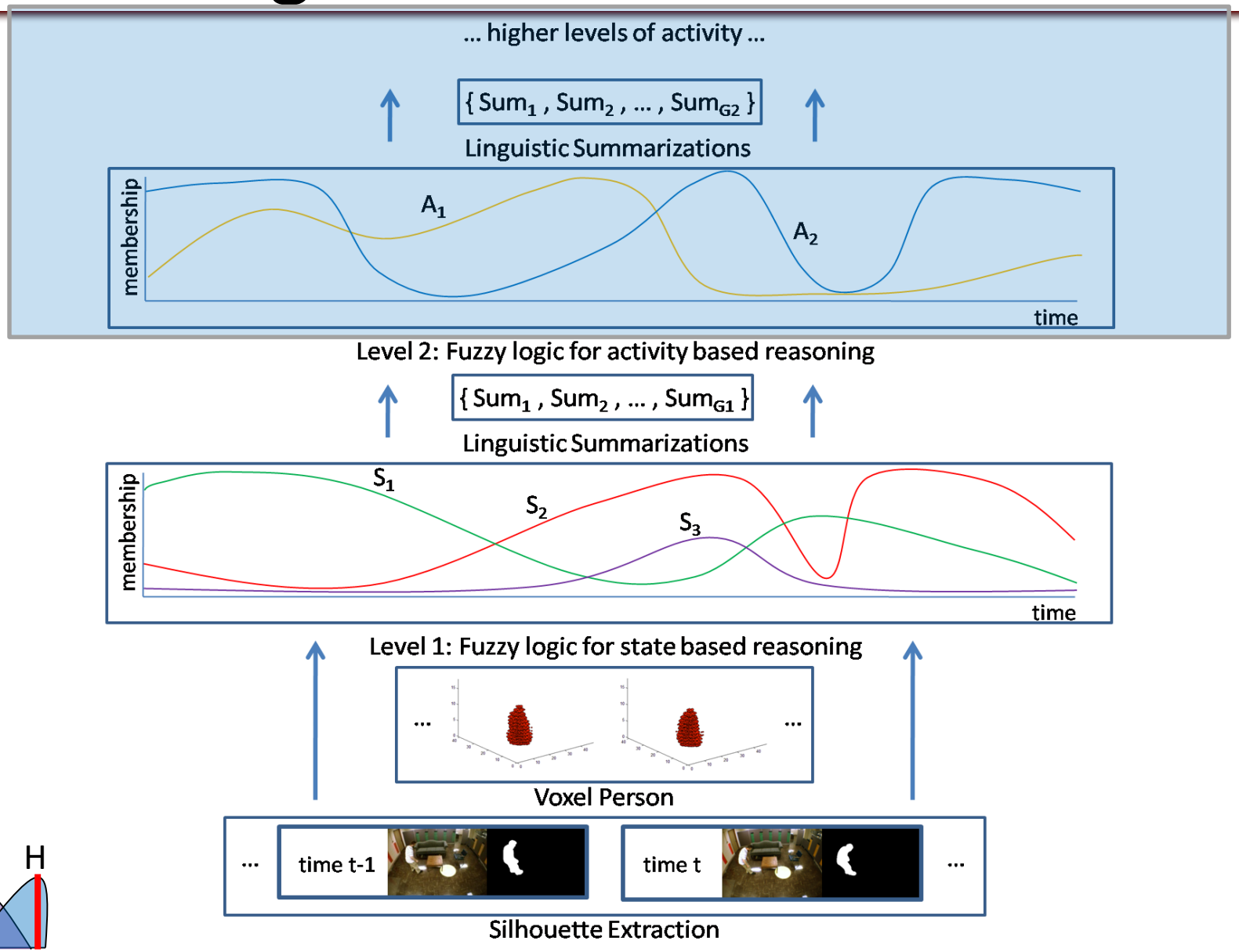
Linguistic summarization



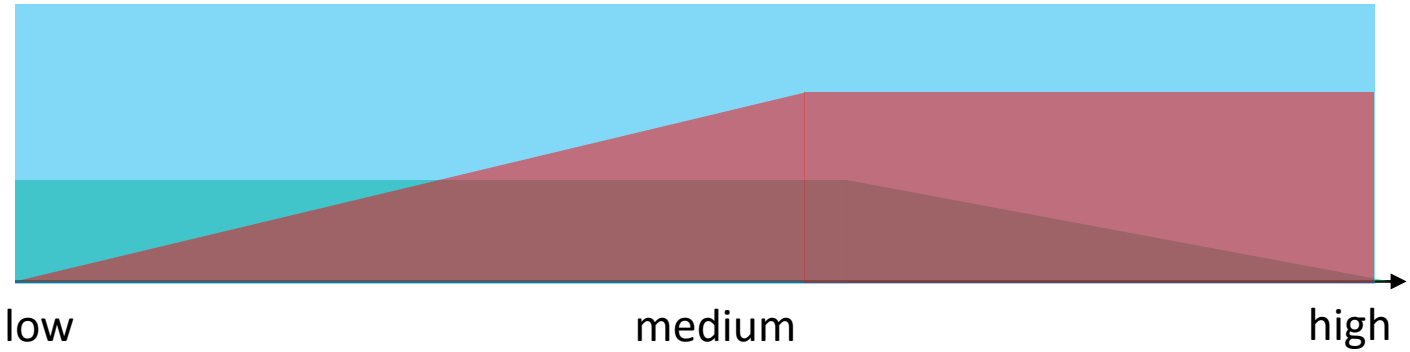
Linguistic summarization



Linguistic summarization



Where does (should) CI belong in CV?



- Low level
  - Can be difficult to press FS advantage, not a difficult sale at all for NNs and EA everywhere
- Medium
  - IF case can be made for modeling uncertainty w.r.t. FSs, likely NNs still, again EA
- High
  - Biggest payoff for FSs as closest to *human-like* operations (e.g., reasoning). NNs? EA still
- **What do you think?**

Suggestions for CI researchers

- CV is a **combination** of theory and application
  - In practice, seems to lean towards the latter …
- **Results** speak volume in the CV field
  - You MUST compare your techniques to LOTS of stuff
    - Known data sets, synthetic examples, others algorithms, …
  - Cannot compare FSs to FSs, compare to everything
    - E.g., compare to state-of-the-art "machine learning" methods
- Many *trends/fads* in the field (like any field)
  - Be careful to not repeat the past (**do your homework**!)
- Be a **jack of all trades**
  - For example, use/know NNs, EAs and FSs

Suggestions for CI researchers

- Like any research endeavor, you need to **ask yourself**
  - **Problem**: what *specific CV task* are you solving?
  - **Gap**: what is the limitation of current approaches?
  - **Idea**: clearly and succinctly, what is your approach? (leave out jargon)
  - **Novelty**: what specific gap(s) does your approach fill?
  - **Major contributions**: what are they specifically?
- **Heilmeier Catechism**
  - Can be understood by someone with a high school degree (vs. a "leading scientist")
  - What are you trying to do?
  - How is it done today and what are the limits of current practice?
  - What's new in your approach and why do you think it will be successful?
  - Who cares?
  - If you're successful, what difference will it make?
  - What are the risks and the payoffs?
  - How long will it take?
  - What are the midterm and final "exams" for your project (idea)?

Suggestions for CI researchers

- YES, focus on **"results"**
  - e.g., accuracy and variance
    - Don't play the 96.25% -> 96.26% == success game
  - Ask yourself, **what other benefits** do you get from using your technique?
    - Don't always play *their game*
    - E.g., possible that FSs are not as popular in CV because the metrics do not "favor" them (need to highlight their utility)
      - Do you get a more robust solution, increased ability to transfer to new domains, yield linguistic descriptions, better human-computer "interface", uncertainty to analyze, etc.

FA, EA and NN specific questions

- FS researchers
  - Uncertainty modeling, *computing* w.r.t. uncertainty and linguistic aspect
  - How do you **show** the **benefit** of FSs? [accuracy alone is shallow]
- EA researchers
  - Optimization
  - Why do we **NEED** an EA solution? Is there a closed form solution or something else that we should be using instead? [so rationalize]
  - Why are you **selecting a particular EA approach**, e.g., PSO vs. GA, etc.
  - Does CV really *change the problem/approach w.r.t. EA* … ?
- NN researchers
  - Universal function approximators
  - Can you avoid a **"black box"** solution?
  - Are you modeling a specific **human visual system process**?
  - Diversity and volume of data to **sufficiently approximate parameters**
  - Low and mid level, of course, but, can you **rationalize** high-level?

Datasets – For your viewing pleasure

- Will fill in before final presentation

That's all folks

Thanks for listening (to part 1!!!)

Questions/comments?

DisclaimAARRGGHHs: the views expressed in this presentation ARE fact and cannot be argued with, we made all of this up on the flight over here, pirates ARE real and NOT nice, by showing up you indirectly agreed to become a computer vision researcher.