CS 696i, Spring 2005, part III

Computer Vision

LAST LECTURE

Kobus Barnard

Computer Science, University of Arizona

Assignment Two Due Today !!

Assignments recieved before tommorow might get bonus marks

Grace period until Thursday

After Thursday, penalty wil be 2*N% / day
where N is the number of people in your group

Computer vision in 100 easy minutes

30 bonus minutes for recognition

Tools for Recognition

Everything discussed in the 100 minutes is useful for the main problem in vision:

Recognizing objects and Understanding Scene Semantics

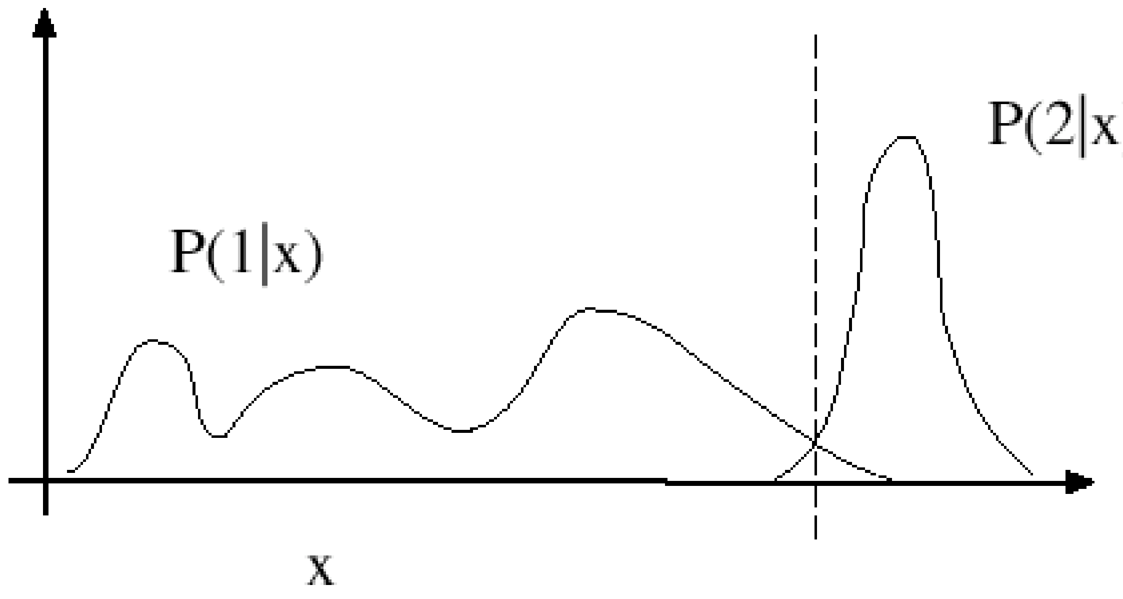
Some examples:

Illumination preprocessing, tracking, depth from multiple views, low level feature detection, texture segmentation, fitting models (for atomic structures).

Recognition by finding patterns

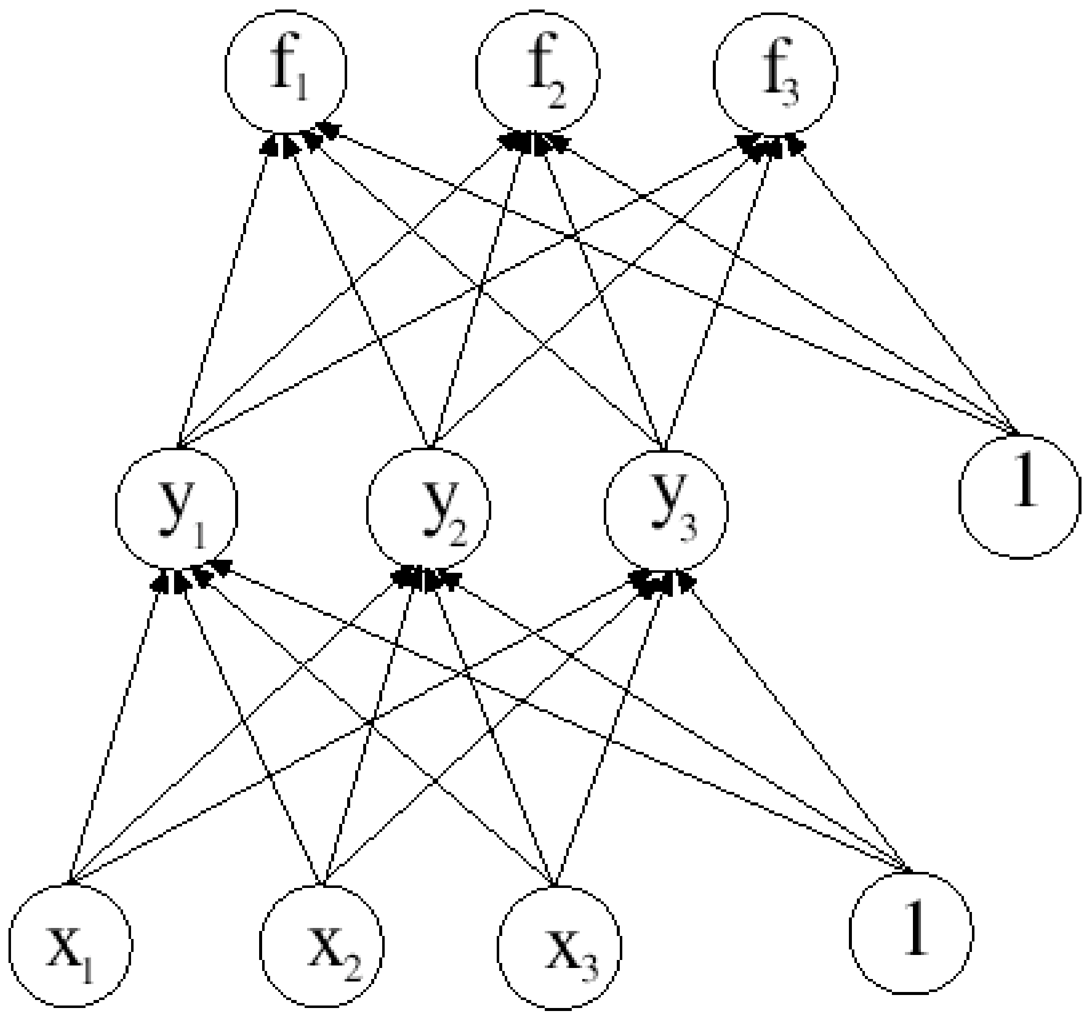
- We have seen very simple template matching (filters)
- Some objects behave like quite simple templates
  - Frontal faces
- Strategy:
  - Find image windows
  - Correct lighting
  - Pass them to a statistical test (the classifier) that accepts faces and rejects non-faces
  - The classifier is typically trained on a database of examples
  - Example classifiers:
    - LDA, nearest neighbor, neural network, naive Bayes, SVM

Finding a decision boundary is not the same as modeling a conditional density.



Neural networks

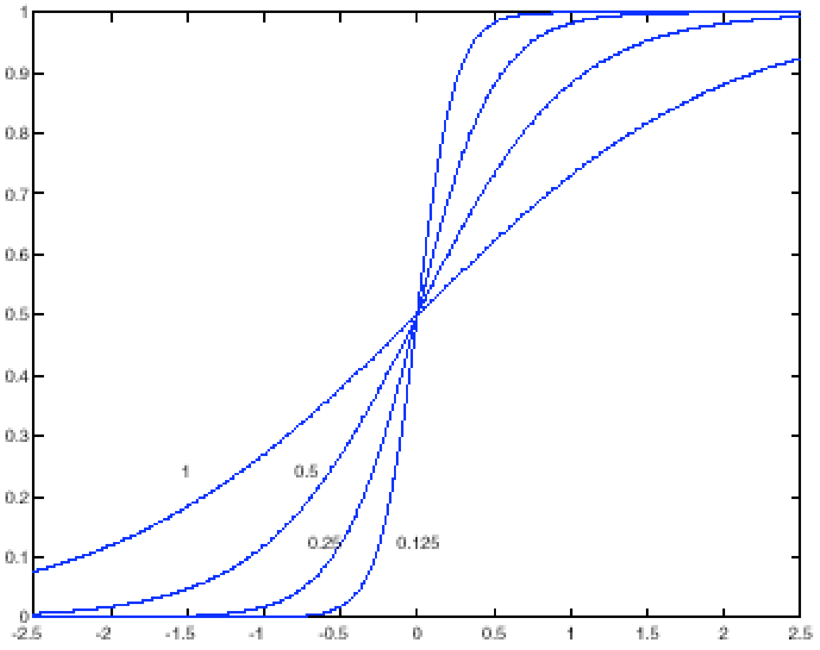
- Linear decision boundaries are useful
  - but often not powerful enough
  - we seek an easy way to get more complex boundaries
- Compose linear decision boundaries from simpler ones
  - i.e. have several linear classifiers, and apply a classifier to their output
  - a nuisance, because sign(ax+by+cz) etc. isn’t differentiable.
  - use a smooth “squashing function” in place of sign.



$$\boldsymbol{g}(\boldsymbol{x}) \approx \boldsymbol{f}(\boldsymbol{x}) = [\phi(\boldsymbol{w}_{21} \cdot \boldsymbol{y}), \phi(\boldsymbol{w}_{22} \cdot \boldsymbol{y}), \ldots \phi(\boldsymbol{w}_{2n} \cdot \boldsymbol{y})]$$

$$\boldsymbol{y}(\boldsymbol{z}) = [\phi(\boldsymbol{w}_{11} \cdot \boldsymbol{z}), \phi(\boldsymbol{w}_{12} \cdot \boldsymbol{z}), \ldots \phi(\boldsymbol{w}_{1m} \cdot \boldsymbol{z}), 1]$$

$$\boldsymbol{z}(\boldsymbol{x}) = [x_1, x_2, \ldots, x_p, 1]$$



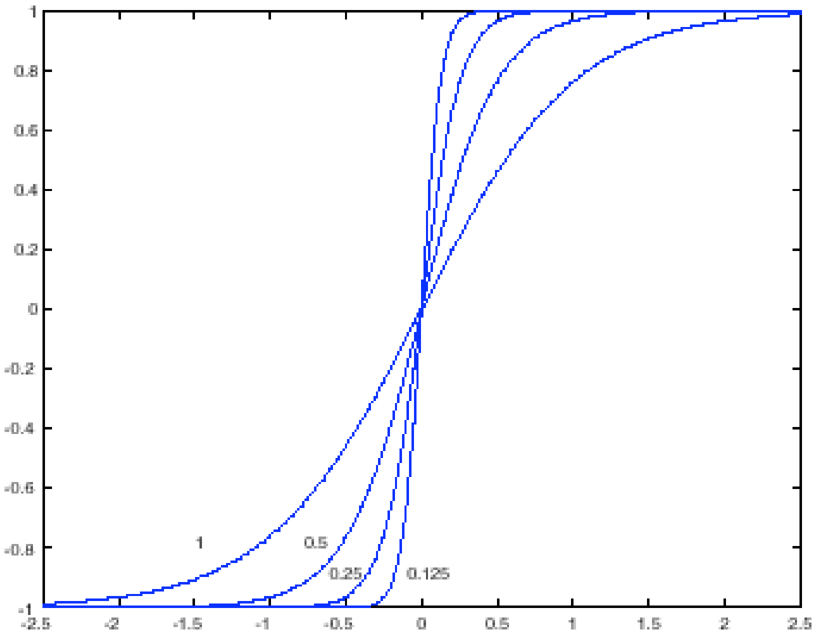
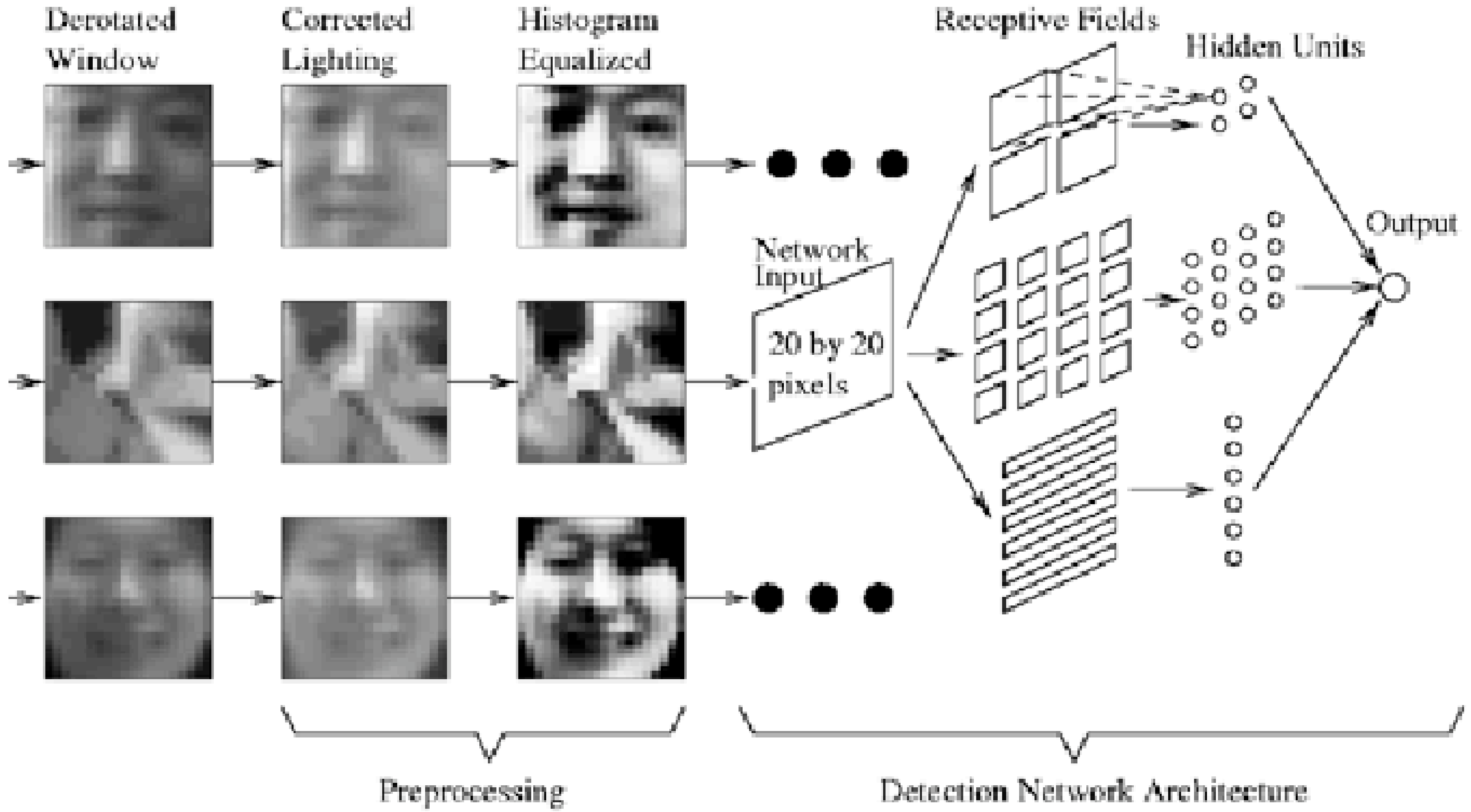


Figure 22.14. On the **left**, a series of squashing functions obtained using $\phi(x;\nu) = \frac{e^{x/\nu}}{1+e^{x/\nu}}$, for different values of ν indicated on the figure. On the **right**, a series of squashing functions obtained using $\phi(x;\nu,A) = A\tanh(x/\nu)$ for different values of ν indicated on the figure. Generally, for x close to the center of the range, the squashing function is linear; for x small or large, it is strongly non-linear.



The vertical face-finding part of Rowley, Baluja and Kanade's system

Figure from "Rotation invariant neural-network based face detection," H.A. Rowley, S. Baluja and T. Kanade, Proc. Computer Vision and Pattern Recognition, 1998, copyright 1998, IEEE

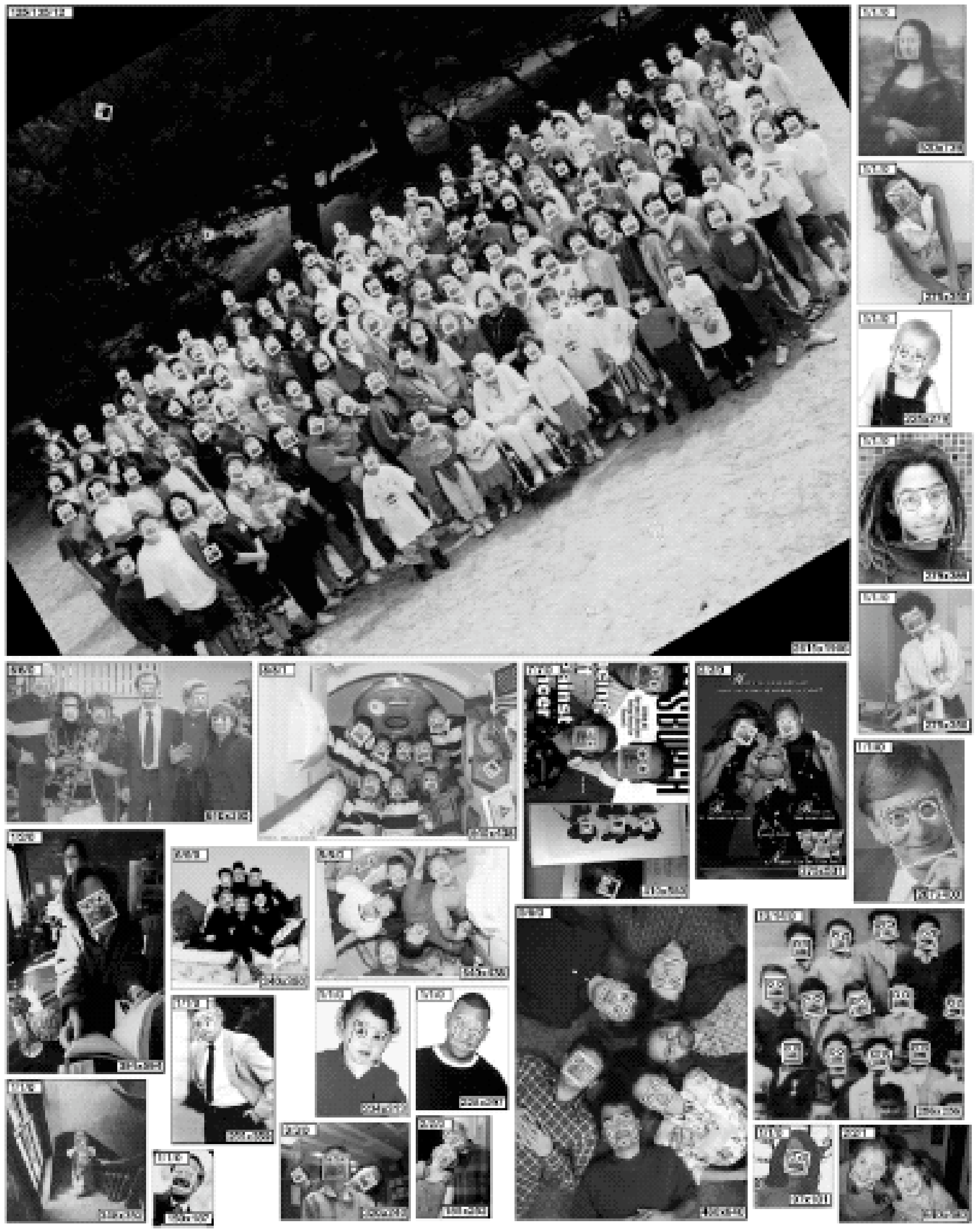
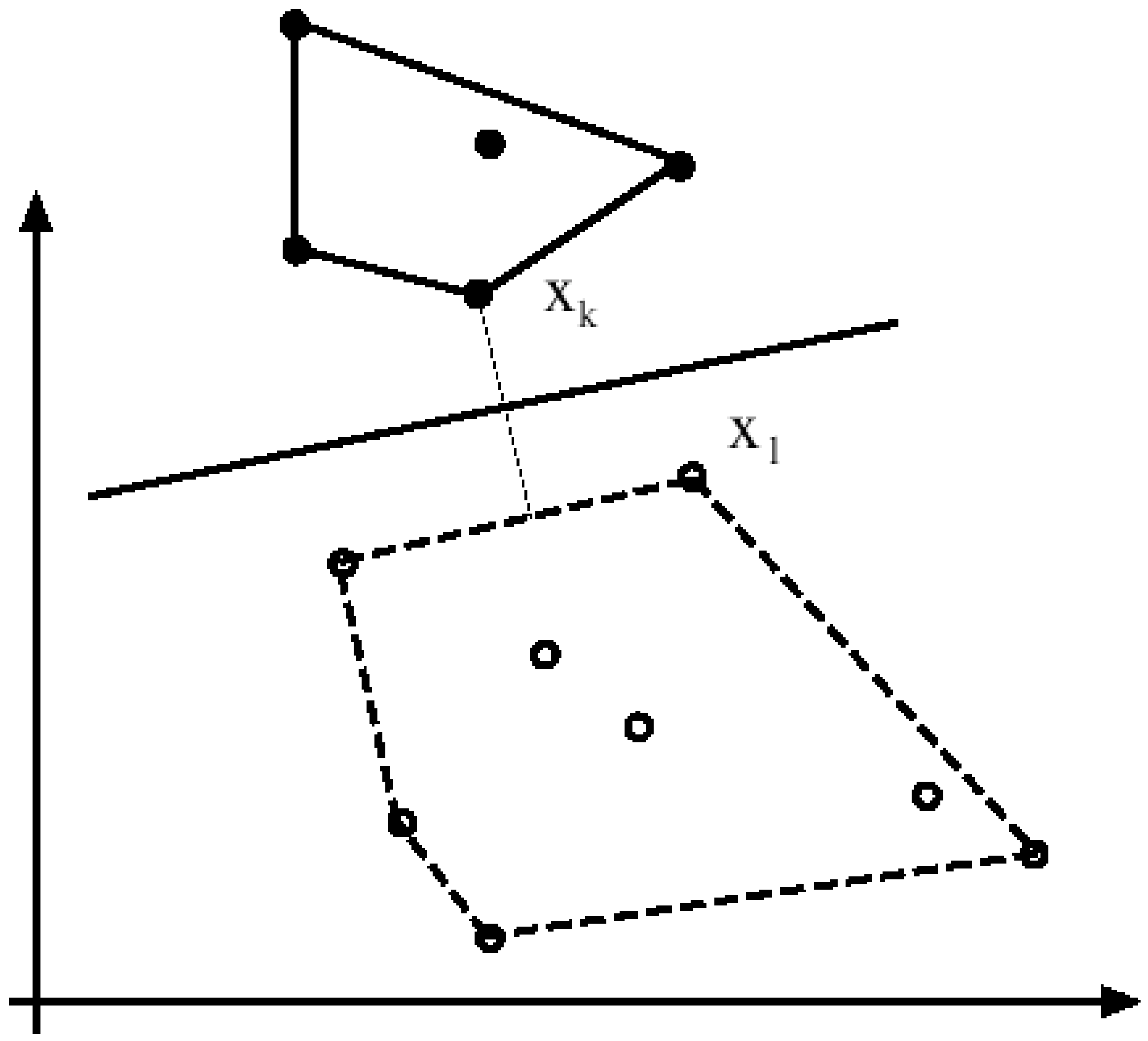


Figure from "Rotation invariant neural-network based face detection," H.A. Rowley, S. Baluja and T. Kanade, Proc. Computer Vision and Pattern Recognition, 1998, copyright 1998, IEEE

Support Vector Machines

- Attempt to find the decision boundary directly
  - Cleaner than neural networks
  - Not all points affect the decision boundary
  - Can get more complex decision boundaries by using a non-linear “kernel” function.


x_k
x_l

Matching by relations

- Idea:
  - find bits, then say object is present if many bits are ok, and their relative position is roughly correct
- Advantage:
  - objects with complex configuration spaces don’t make good templates
    - internal degrees of freedom
    - aspect changes
    - (possibly) shading
    - variations in texture
    - etc.

Employ spatial relations

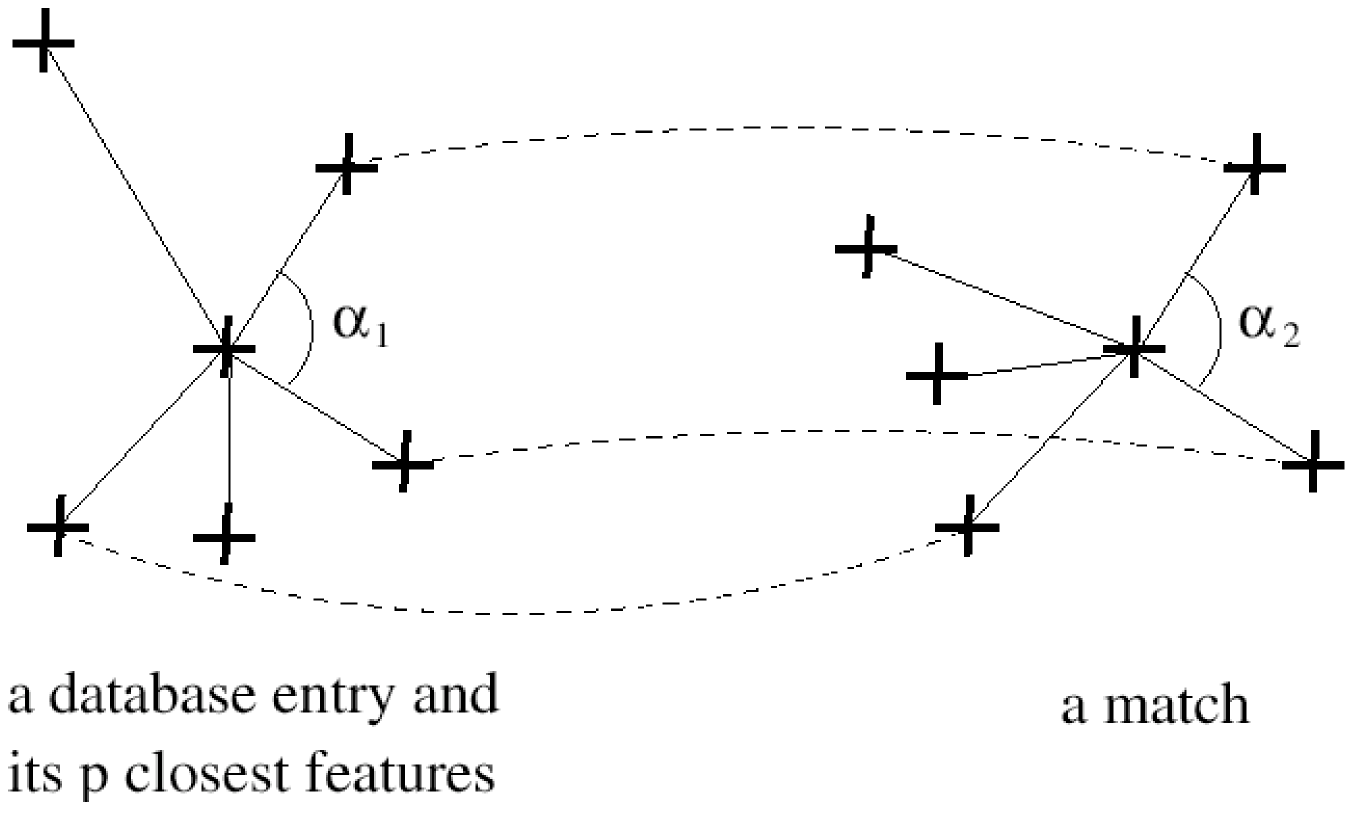


Figure from "Local gray value invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE

Finding faces using relations

- Strategy:
  - Face is eyes, nose, mouth, etc. with appropriate relations between them
  - build a specialized detector for each of these (template matching) and look for groups with the right internal structure
  - Once we’ve found enough of a face, there is little uncertainty about where the other bits could be

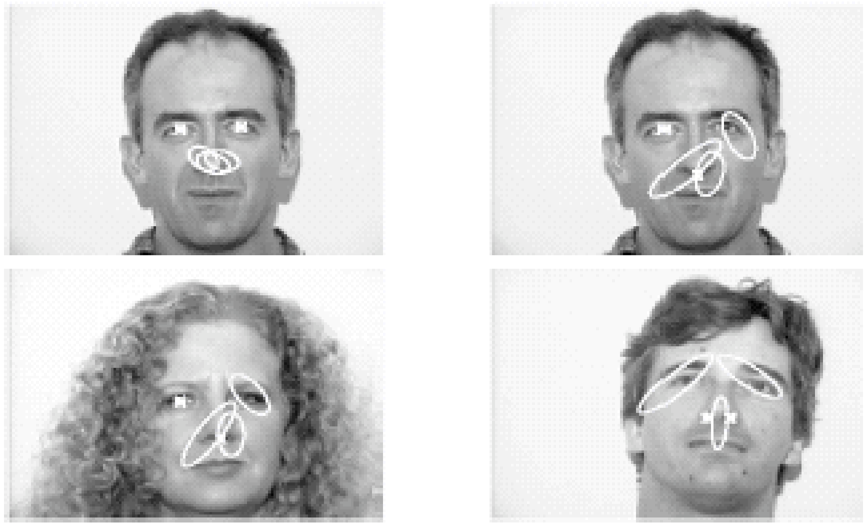
Finding faces using relations

- Strategy: compare

$$P(\text{one face at } \boldsymbol{F} | \boldsymbol{X}_{\text{le}} = \boldsymbol{x}_1, \boldsymbol{X}_{\text{re}} = \boldsymbol{x}_2, \boldsymbol{X}_{\text{m}} = \boldsymbol{x}_3, \boldsymbol{X}_{\text{n}} = \boldsymbol{x}_4, \text{all other responses})$$

with

$$P(\text{no face} | \boldsymbol{X}_{\text{le}} = \boldsymbol{x}_1, \boldsymbol{X}_{\text{re}} = \boldsymbol{x}_2, \boldsymbol{X}_{\text{m}} = \boldsymbol{x}_3, \boldsymbol{X}_{\text{n}} = \boldsymbol{x}_4, \text{all other responses})$$



Notice that once some facial features have been found, the position of the rest is quite strongly constrained.

Figure from, “Finding faces in cluttered scenes using random labeled graph matching,” by Leung, T. ;Burl, M and Perona, P., Proc. Int. Conf. on Computer Vision, 1995 copyright 1995, IEEE

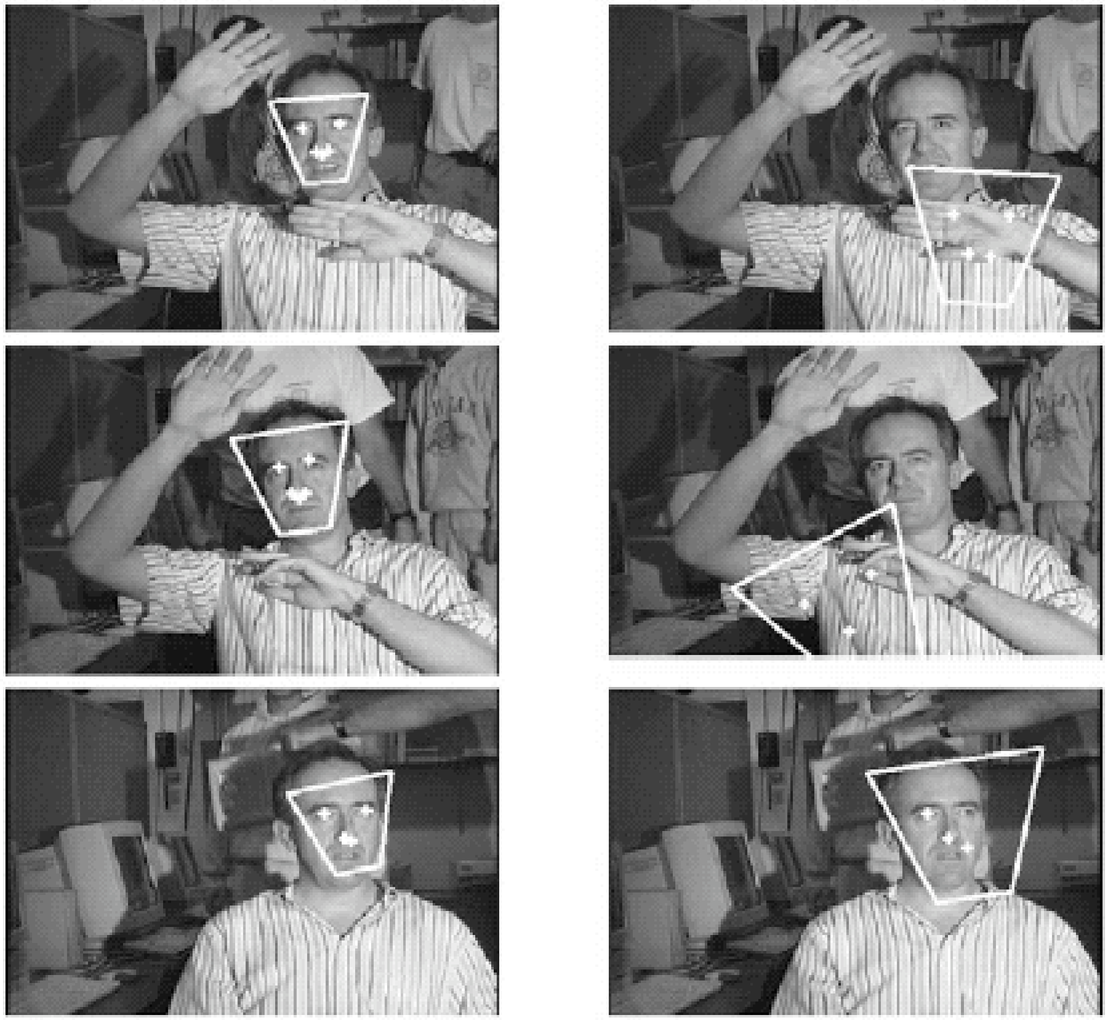
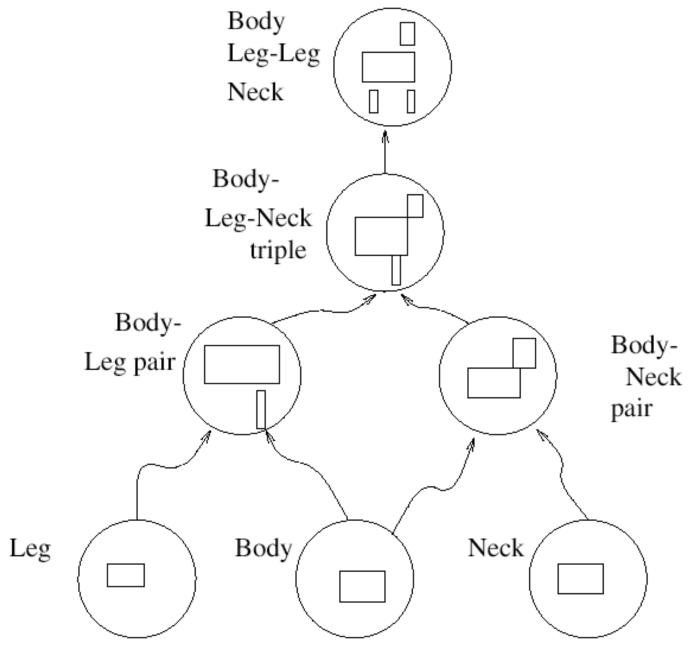
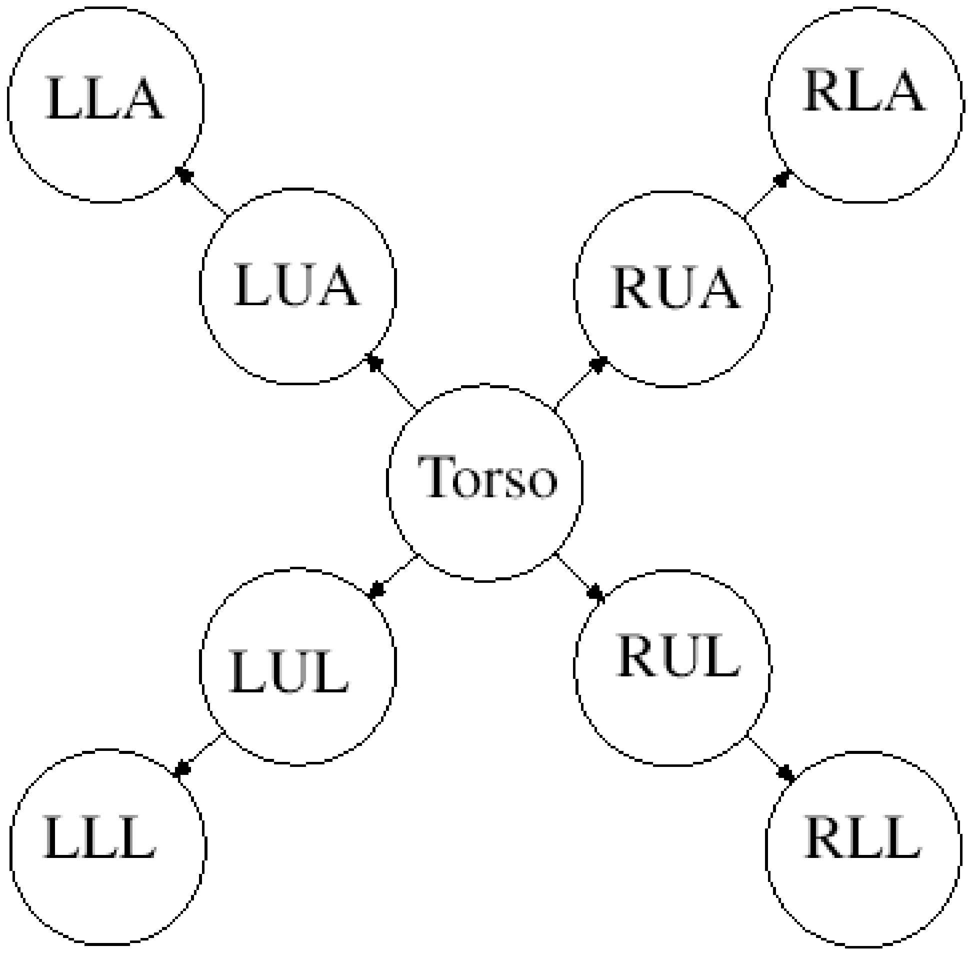


Figure from, “Finding faces in cluttered scenes using random labeled graph matching,” by Leung, T. ;Burl, M and Perona, P., Proc. Int. Conf. on Computer Vision, 1995 copyright 1995, IEEE

Horses


Body
Leg-Leg
Neck
Body-
Leg-Neck
triple
Body-
Leg pair
Body-
Neck
pair
Leg
Body
Neck

People



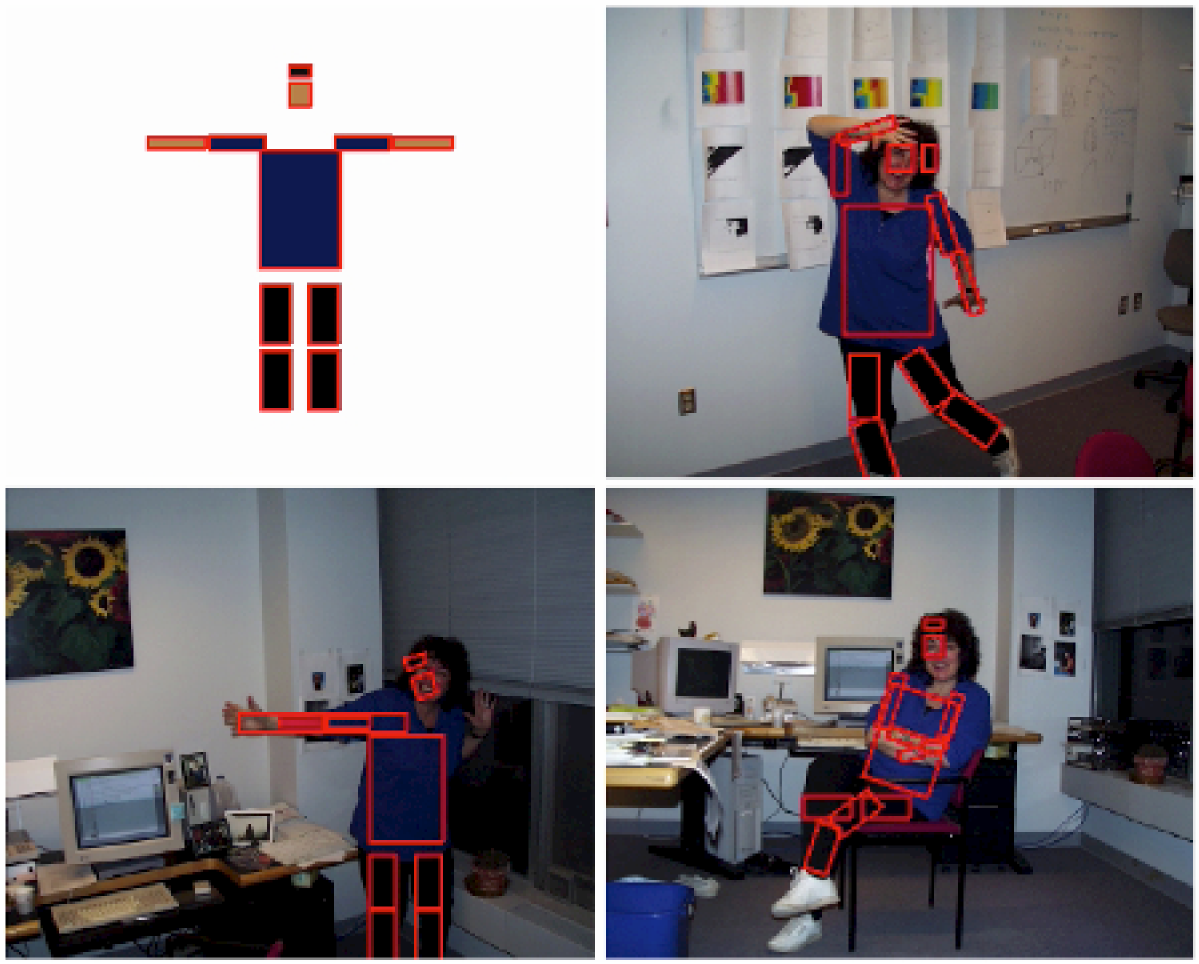


Figure from “Efficient Matching of Pictorial Structures,” P. Felzenszwalb and D.P. Huttenlocher, Proc. Computer Vision and Pattern Recognition2000, copyright 2000, IEEE

Computer vision in ~~100~~ 130 easy minutes

More minutes? CS 577 will be offered this spring.

Related course: CS 533 (graphics) will be offered this fall.

Open Problems in Recognition

Learning effective models for many objects

The form of the model / representation

Integrating non-visual knowledge (e.g., utility based recognition)

Things vision can (almost) do*

Given a sufficiently patterned object, determine its geometry from multiple views

Correct for uniform illumination at roughly human performance

Track moving objects

Recognize instances of objects and landmarks

Matching images under affine transformation

Frontal face recognition

*Under good conditions

But is this vision? Intelligence??

What part of intelligence do we expect to get out of vision?

Can we expect “real” recognition without the help of other kinds of intelligence?

If we need other intelligence (logic? language??), how does that fit in? When exactly is the vision system done? What are the desired “simple” modules and their outputs.

Should machine vision be in CS 696i at all?

Vision and Intelligence

Does the concept of “translation” help?

If we did it much better than assignment 2, would this be intelligence?

For language to work does it need to be grounded in the real world?

Is vision a good way to do this?

Final Comments

Our intelligence can be in the way when constructing and evaluating computational intelligent systems because we implicitly know so much (e.g. meanings of the tokens)

It is easy to be fooled (to what extent does your program "know" what a tiger is, if it puts "tiger" on the tiger 80% of the time)

It is easy to miss the fact that hard parts of the problem have implicitly been dealt by the problem setup / encoding

It is easy to put your intelligence into the "system" if you simply imagine how it might work

==> We build systems and test our ideas on real data

The Final Illusion