6.869 Advances in Computer Vision Spring 2010

http://people.csail.mit.edu/torralba/courses/6.869/6.869. computervision.htm

Lecture 18
Bayes

Project

5-10 minutes presentation

6 pages report

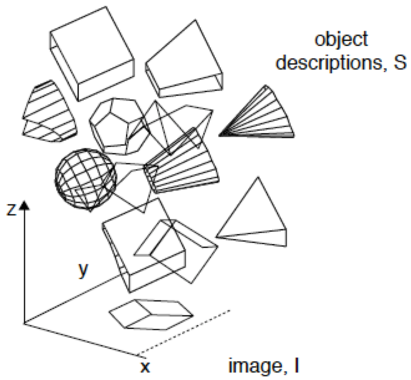
Regularization

- Many problems have multiple solutions.
- Or there is insufficient data

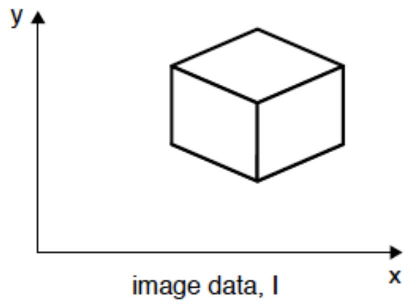
We need to constraint the solutions

Bayesian models of object perception

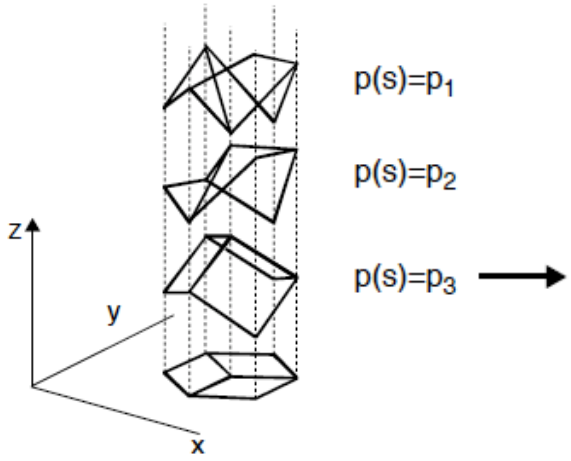
Daniel Kersten[*] and Alan Yuille[†]



Possible interpretations

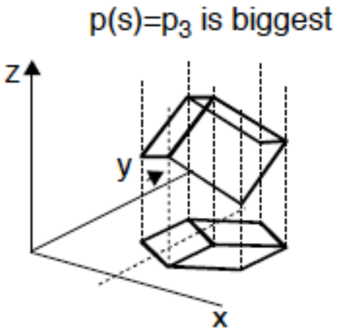


Input

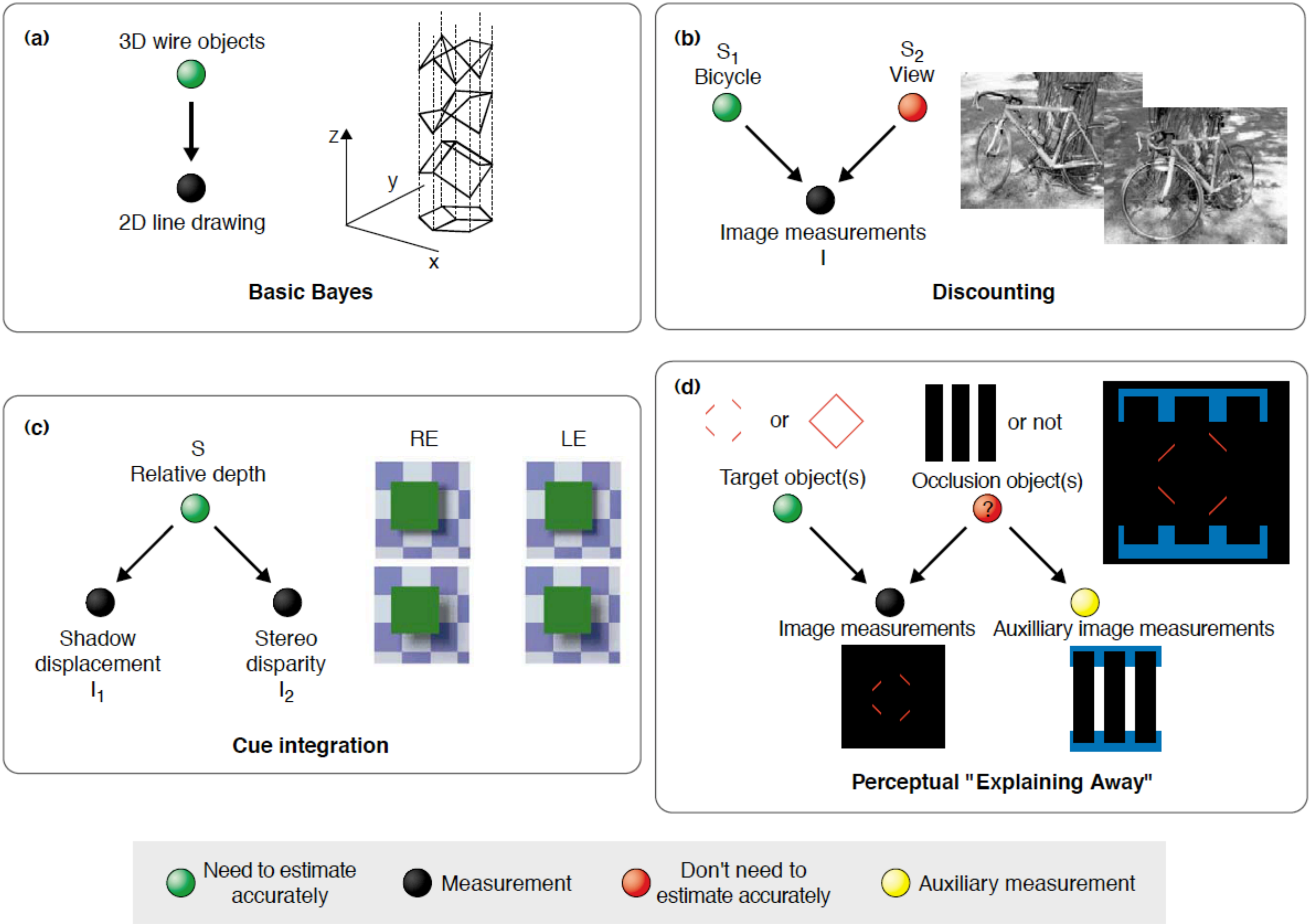


likelihood, p(IIS), narrows selection consistent with projection

Likelihood



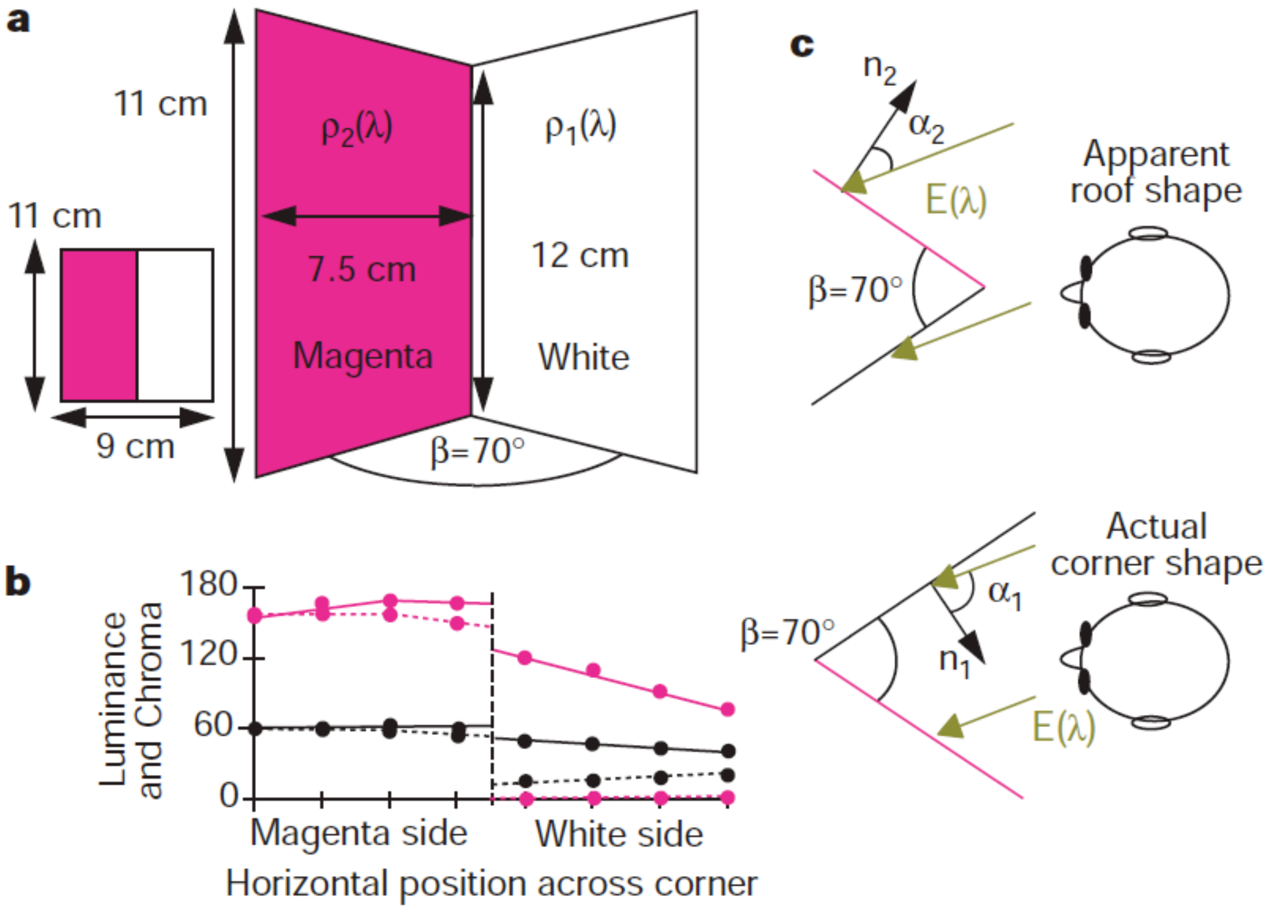
Prior

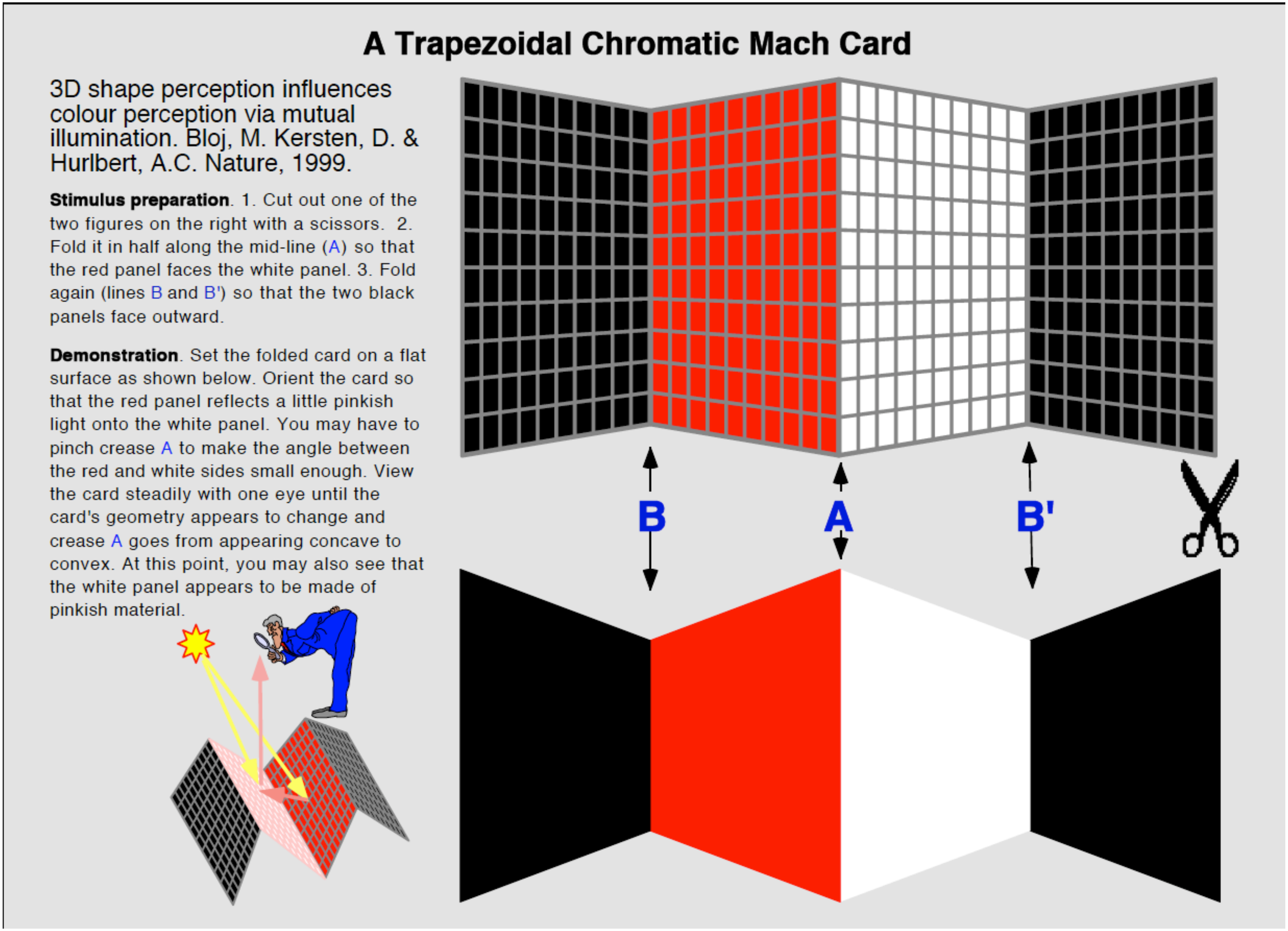

(a)
3D wire objects
2D line drawing
z
y
x
Basic Bayes
(b)
S_1
Bicycle
S_2
View
Image measurements
I
Discounting
(c)
S
Relative depth
Shadow
displacement
I_1
Stereo
disparity
I_2
RE
LE
Cue integration
(d)
or
or not
Target object(s)
Occlusion object(s)
?
Image measurements
Auxilliary image measurements
Perceptual "Explaining Away"
Need to estimate accurately
Measurement
Don't need to estimate accurately
Auxilliary measurement

Perception of three-dimensional shape influences colour perception through mutual illumination

M. G. Bloj*, D. Kersten† & A. C. Hurlbert*

* Physiological Sciences, Medical School, Newcastle upon Tyne, NE2 4HH, UK
† Psychology Department, University of Minnesota, 75 East River Road,
Minneapolis, Minnesota 55455, USA





A Trapezoidal Chromatic Mach Card

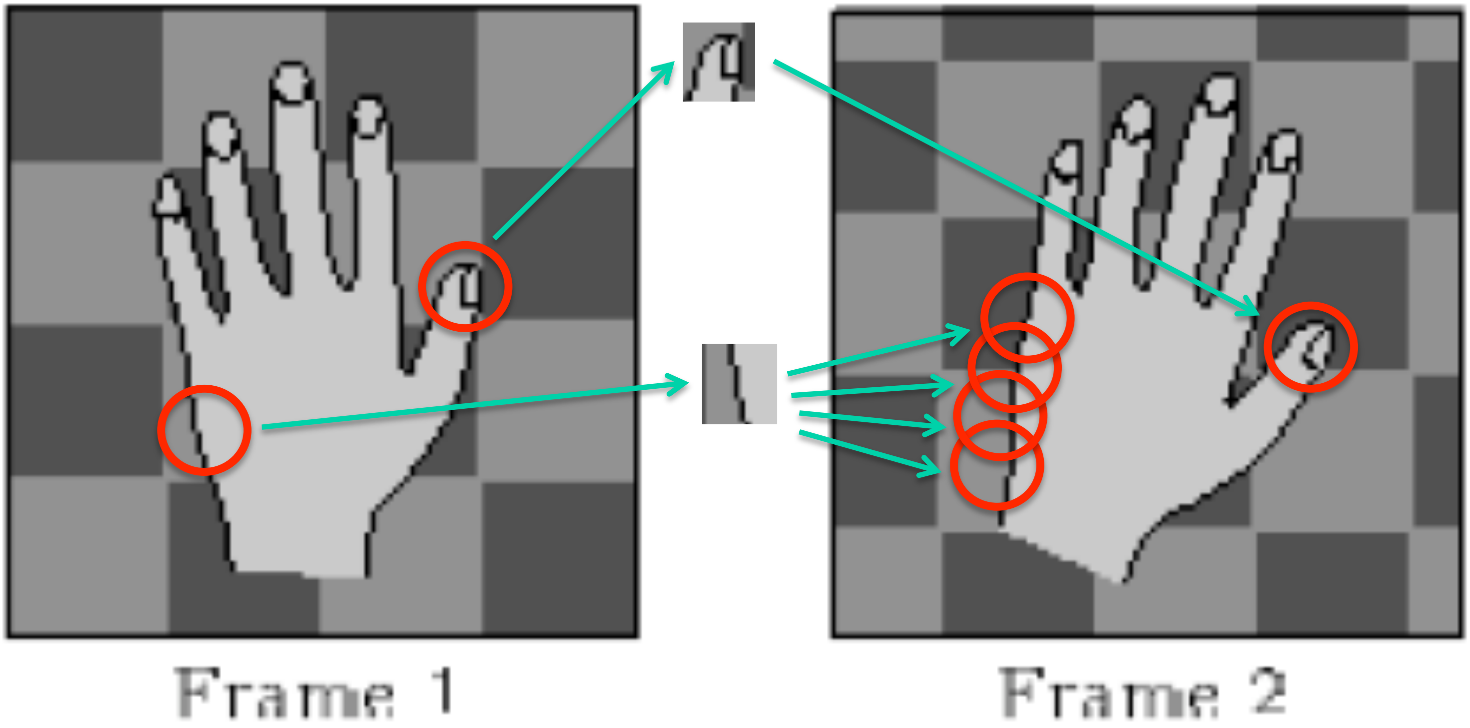
3D shape perception influences colour perception via mutual illumination. Bloj, M. Kersten, D. & Hurlbert, A.C. Nature, 1999.

Stimulus preparation. 1. Cut out one of the two figures on the right with a scissors. 2. Fold it in half along the mid-line (A) so that the red panel faces the white panel. 3. Fold again (lines B and B') so that the two black panels face outward.

Demonstration. Set the folded card on a flat surface as shown below. Orient the card so that the red panel reflects a little pinkish light onto the white panel. You may have to pinch crease A to make the angle between the red and white sides small enough. View the card steadily with one eye until the card's geometry appears to change and crease A goes from appearing concave to convex. At this point, you may also see that the white panel appears to be made of pinkish material.

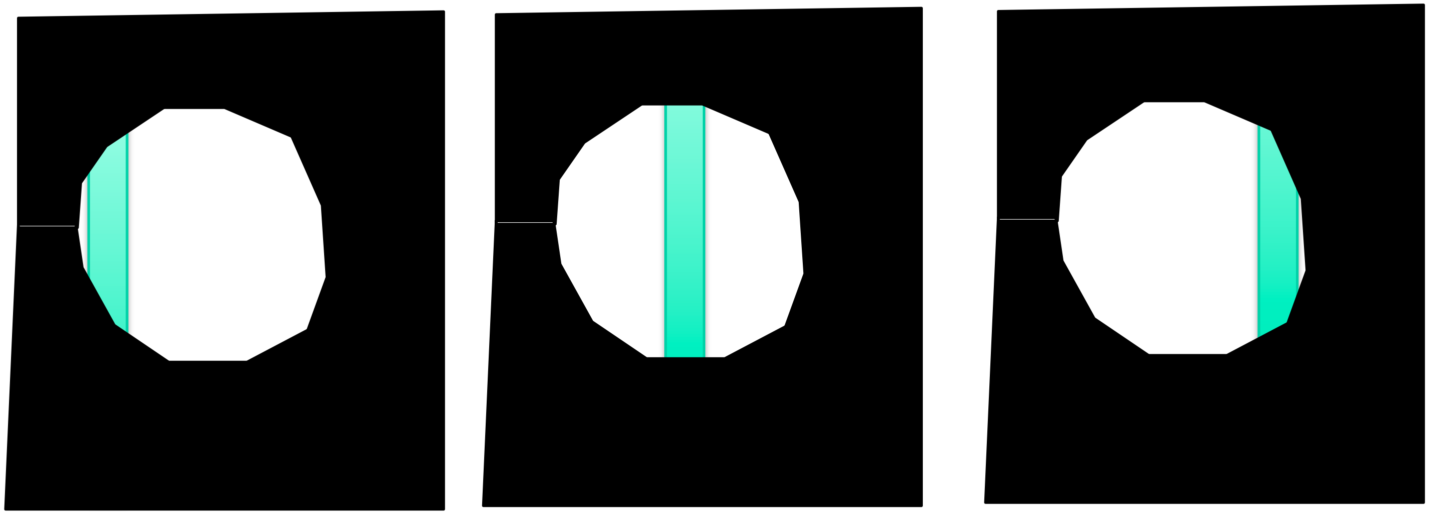
http://vision.psych.umn.edu/users/kersten/kersten-lab/Mutual_illumination/BlojKerstenHurlbertDemo99.pdf http://vision.psych.umn.edu/users/kersten/kersten-lab/demos/MatteOrShiny.html

Example: motion estimation



Example: motion estimation

Aperture problem

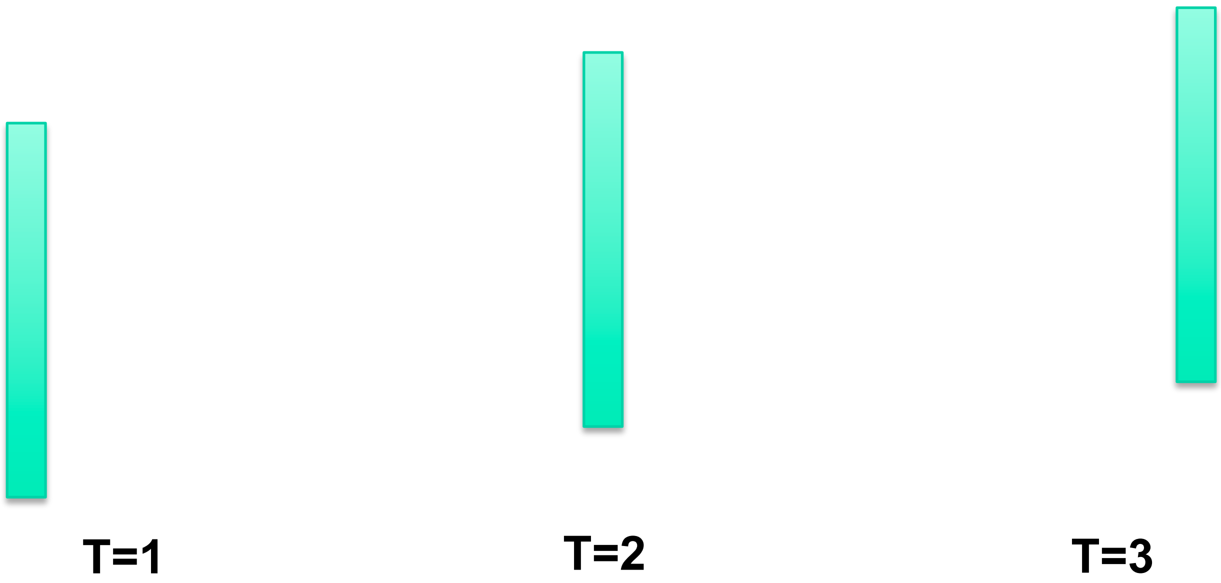


T=1 **T=2** **T=3**

http://web.mit.edu/persci/demos/Motion&Form/demos/one-square/one-square.html

Example: motion estimation

Aperture problem

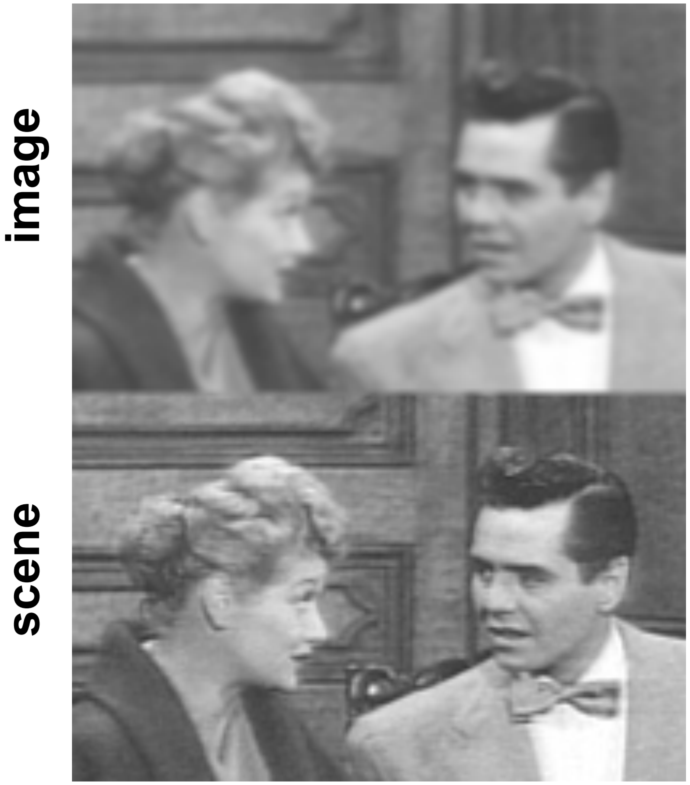


http://web.mit.edu/persci/demos/Motion&Form/demos/one-square/one-square.html

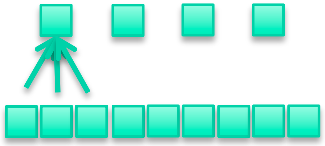
Super-resolution

- Image: low resolution image
- Scene: high resolution image

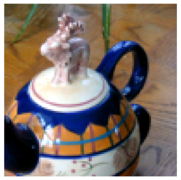
ultimate goal...

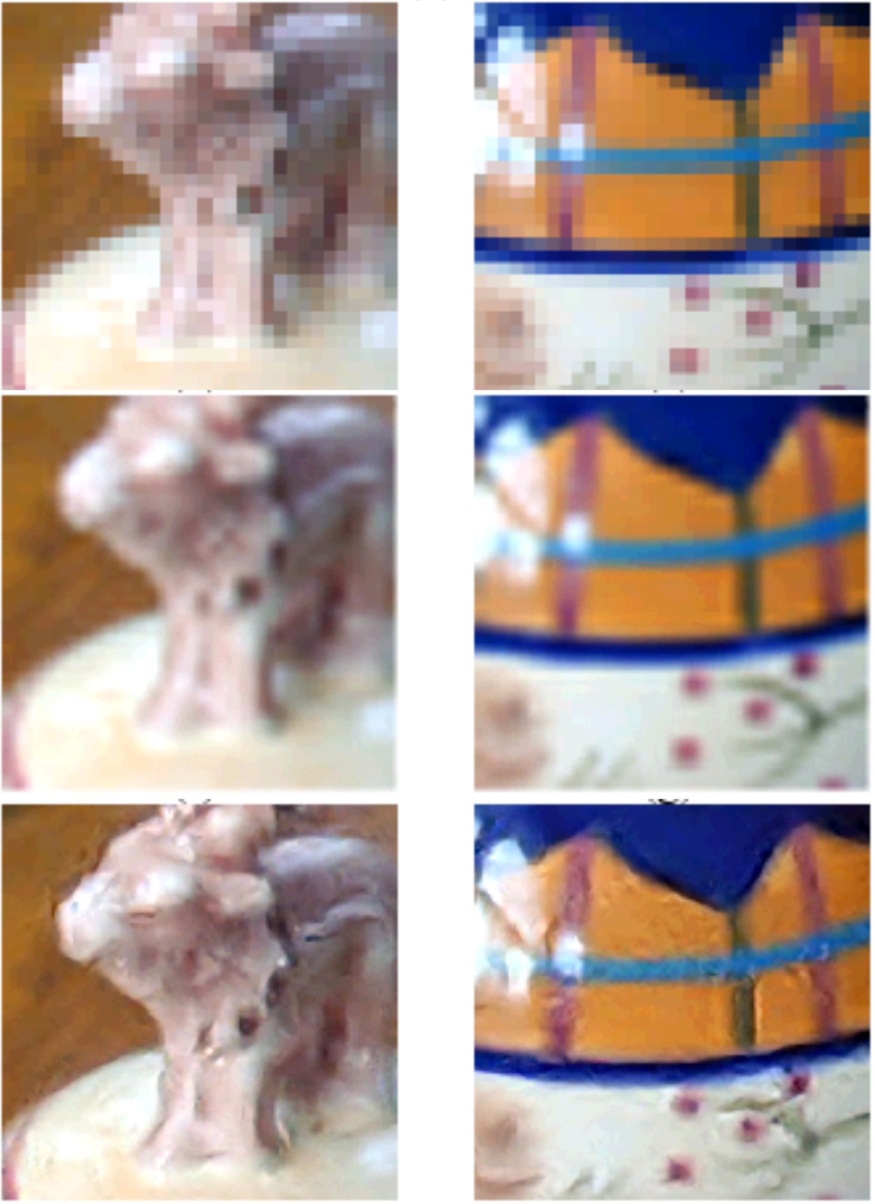


Observation model



Pixel-based images are not resolution independent

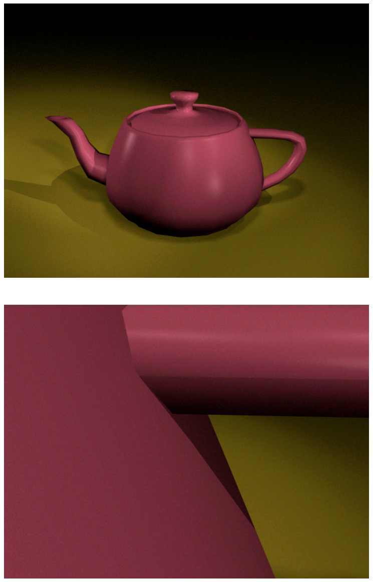




Pixel replication

Cubic spline, sharpened

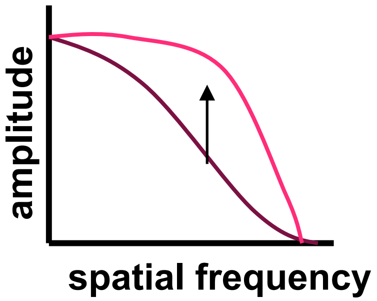
Training-based super -resolution



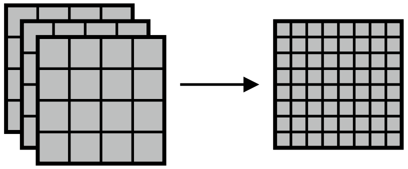
Polygon-based graphics images are resolution independent

3 approaches to perceptual sharpening

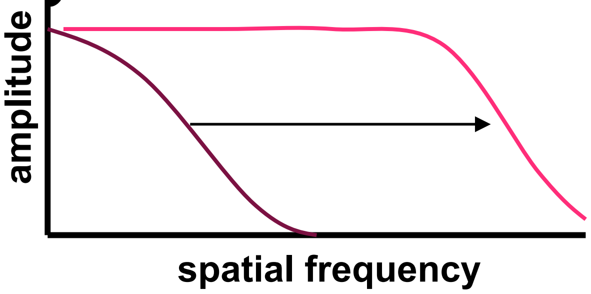
(1) Sharpening; boost existing high frequencies.



(2) Use multiple frames to obtain higher sampling rate in a still frame.



(3) Estimate high frequencies not present in image, although implicitly defined.



Here, we focus on (3), which we'll call "super-resolution".

Super-resolution: other approaches

- Schultz and Stevenson, 1994
- Pentland and Horowitz, 1993
- fractal image compression (Polvere, 1998; Iterated Systems)
- astronomical image processing (eg. Gull and Daniell, 1978; "pixons" http://casswww.ucsd.edu/puetter.html)

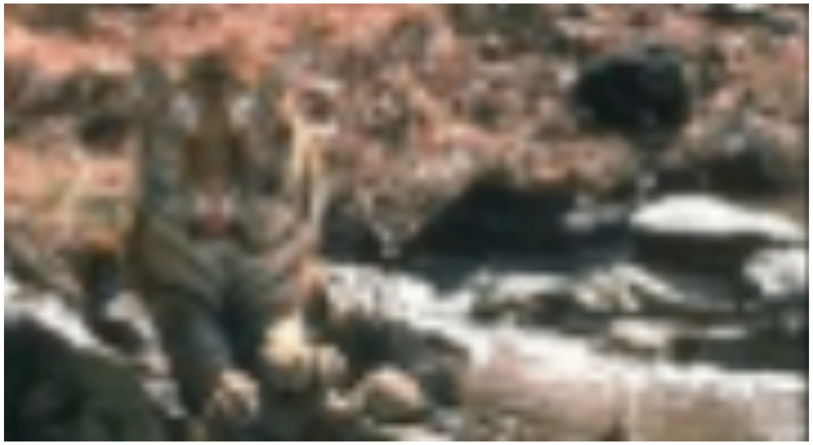
Training images, **~100,000 image/scene patch pairs**

Images from two Corel database categories: “giraffes” and “urban skyline”.

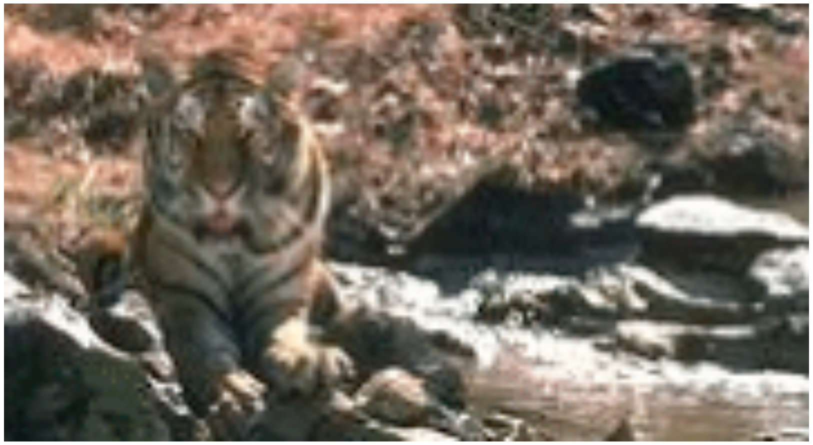
Do a first interpolation

Zoomed low-resolution

Low-resolution



Zoomed low-resolution

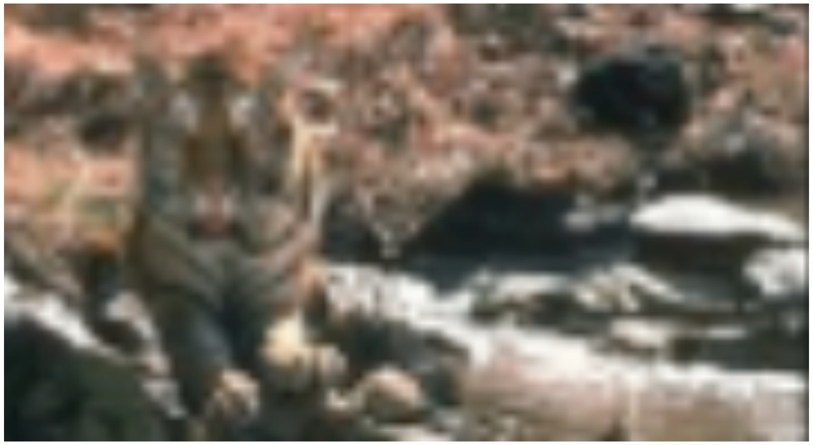


Full frequency original

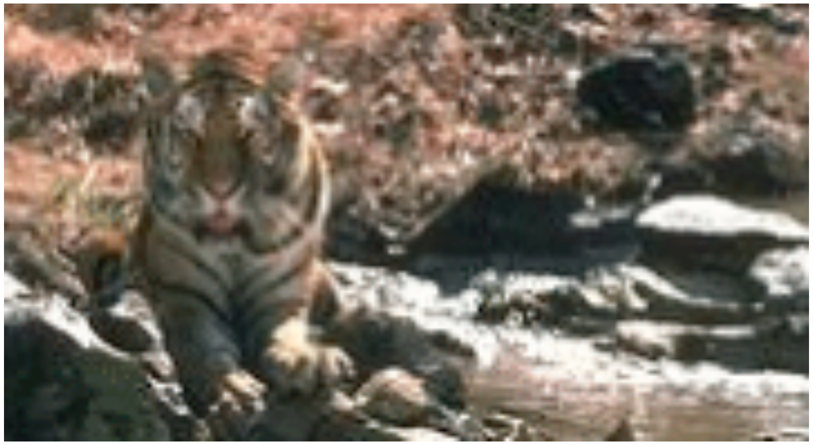
Low-resolution

Representation

Zoomed low-freq.

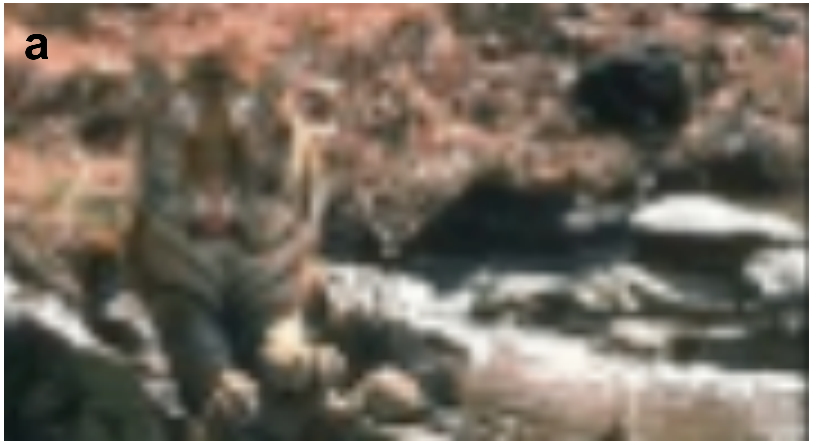


Full freq. original

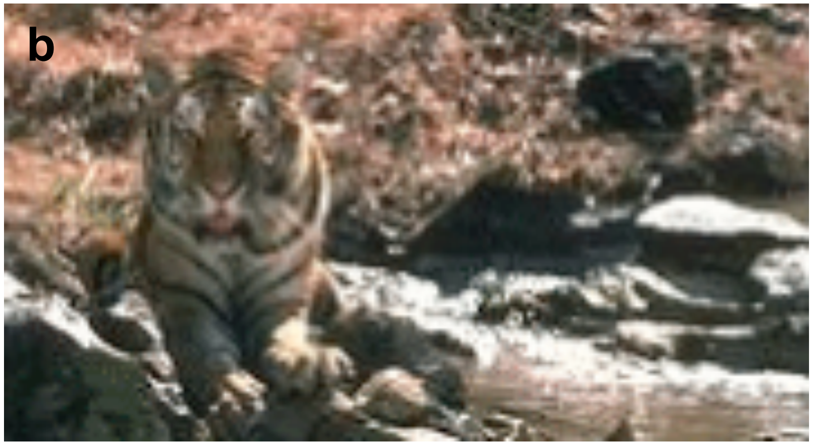


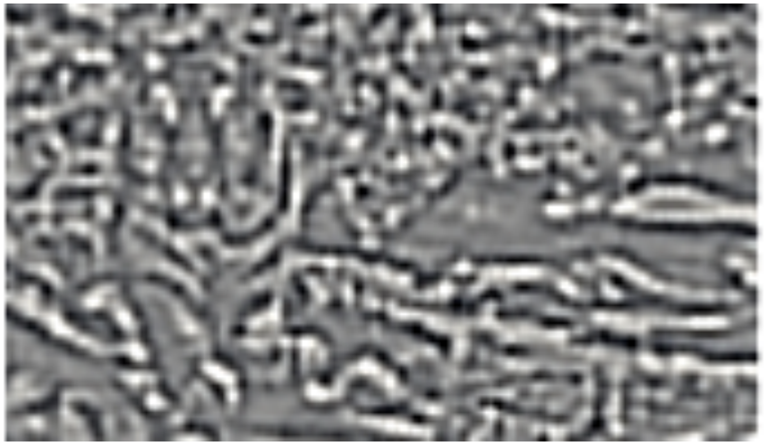
Representation

Zoomed low-freq.

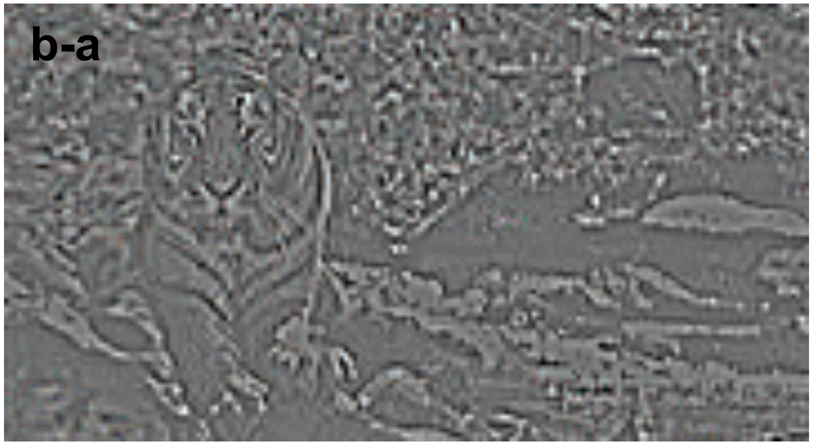


Full freq. original





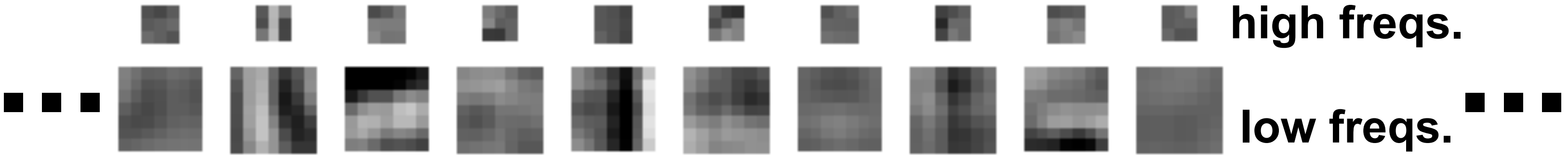
Low-band input (contrast normalized, PCA fitted)



True high freqs

(to minimize the complexity of the relationships we have to learn,
we remove the lowest frequencies from the input image,
and normalize the local contrast level).

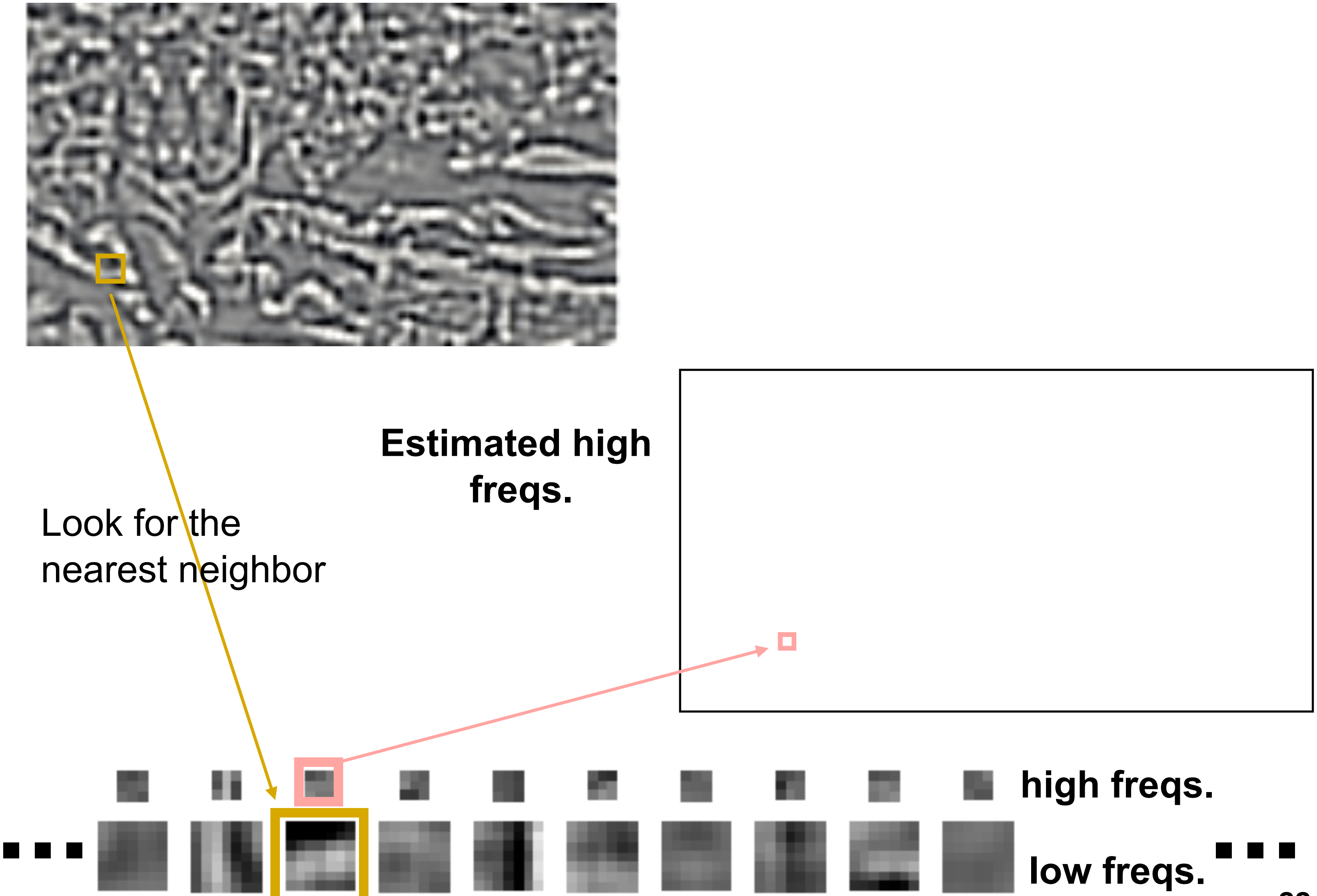
Gather ~100,000 patches



Training data samples (magnified)

Nearest neighbor estimate

Input low freqs.



Estimated high freqs.

Look for the nearest neighbor

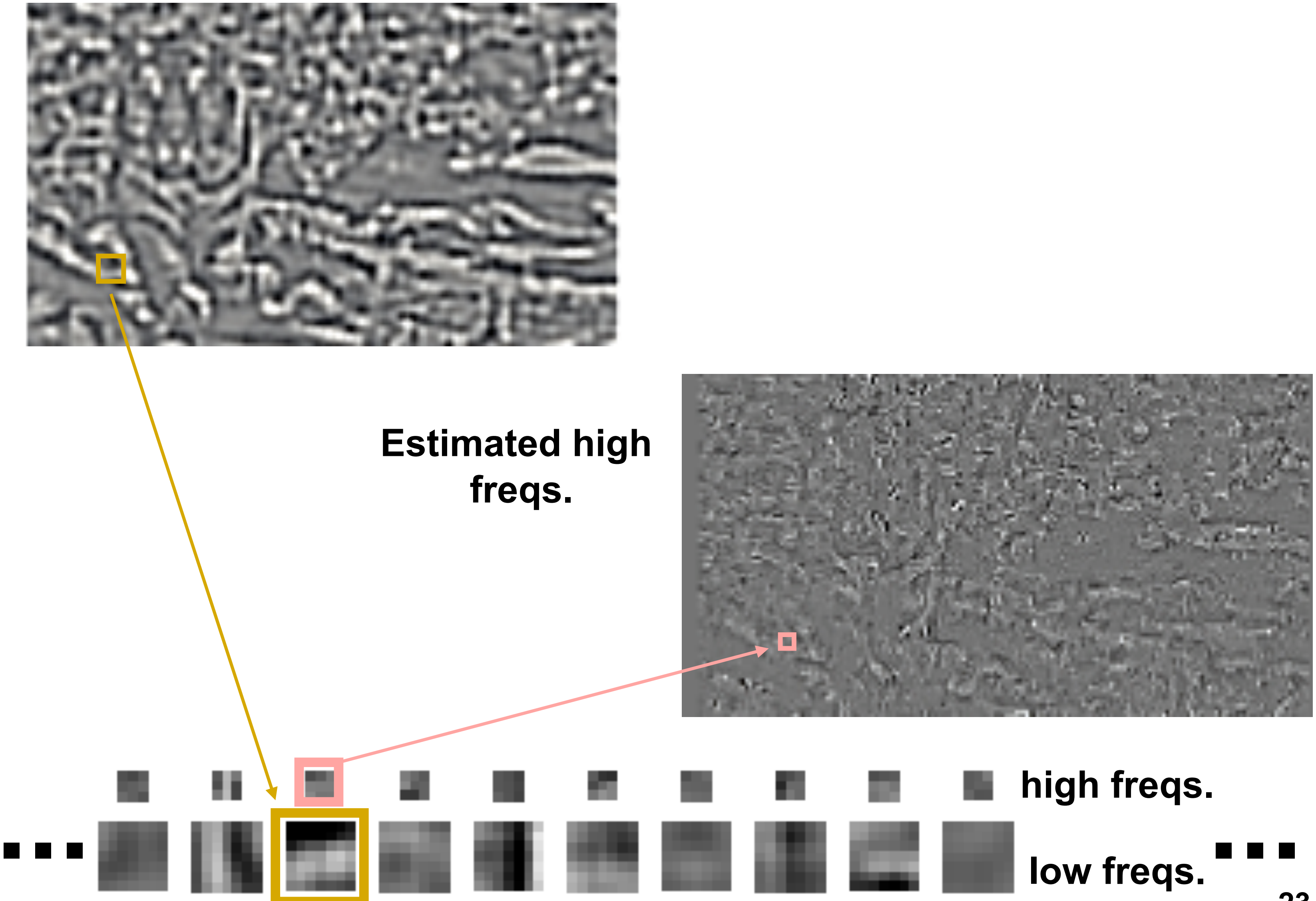
high freqs.

low freqs.

Training data samples (magnified)

Nearest neighbor estimate

Input low freqs.



Estimated high freqs.

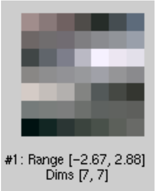
high freqs.

low freqs.

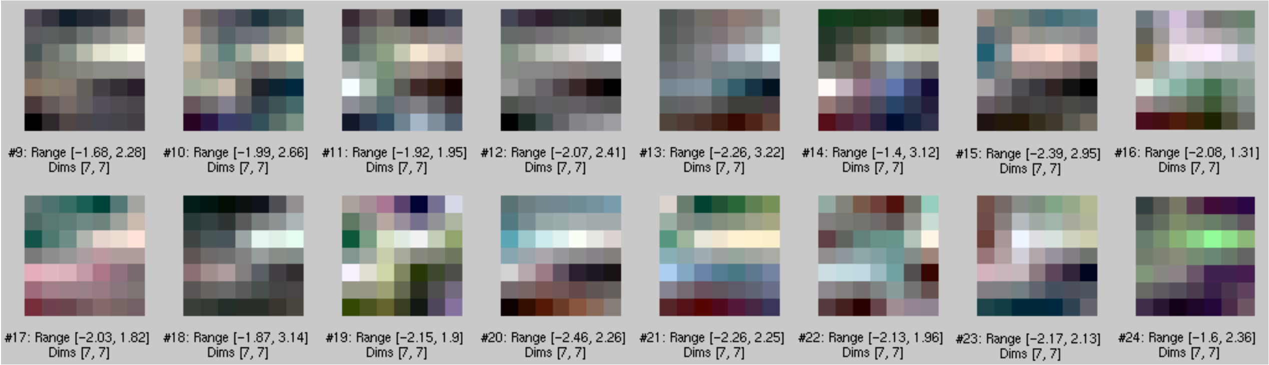
Training data samples (magnified)

But why to use only one match?
There are many candidates

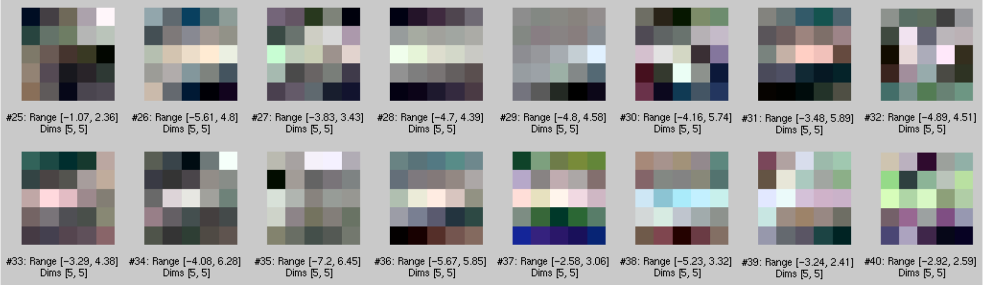
Input patch



Closest image patches from database



Corresponding high-resolution patches from database



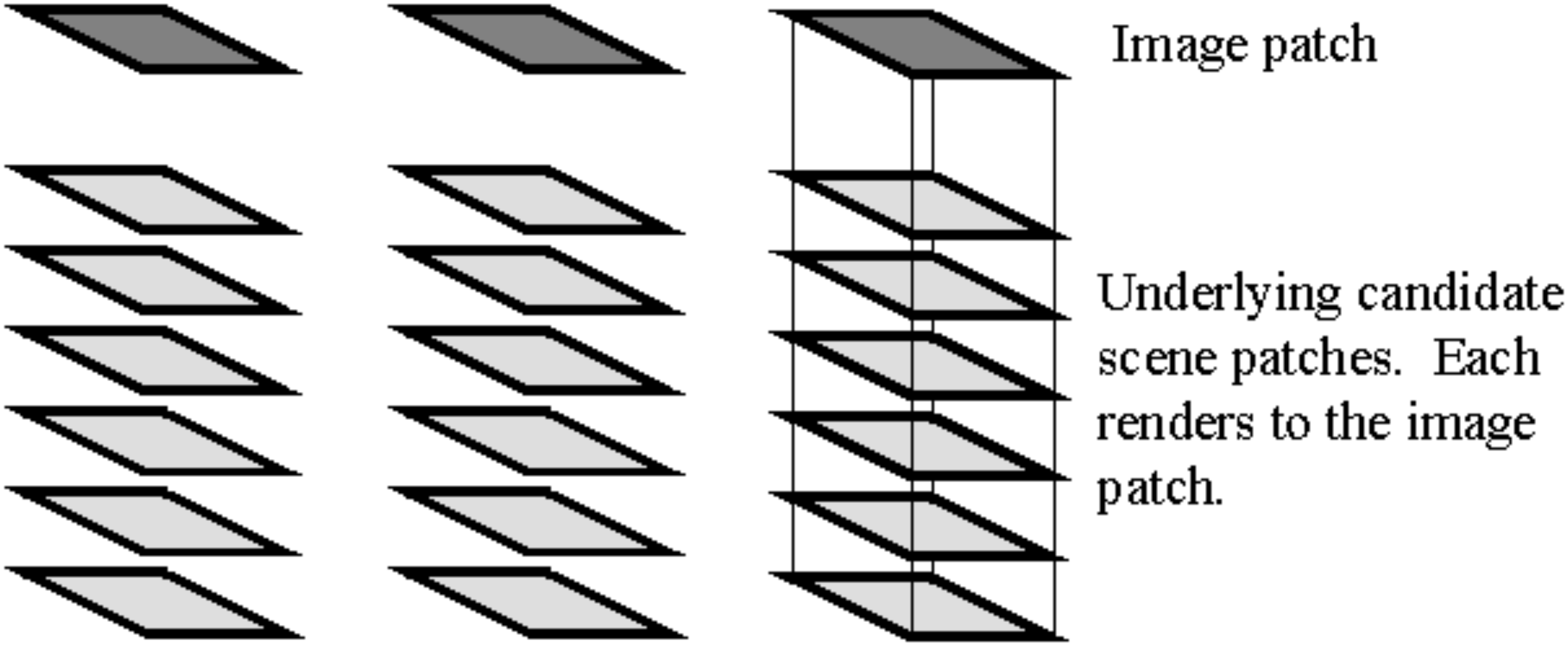
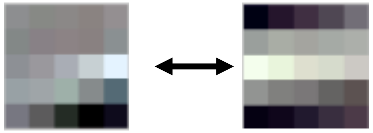


Image patch
Underlying candidate scene patches. Each renders to the image patch.

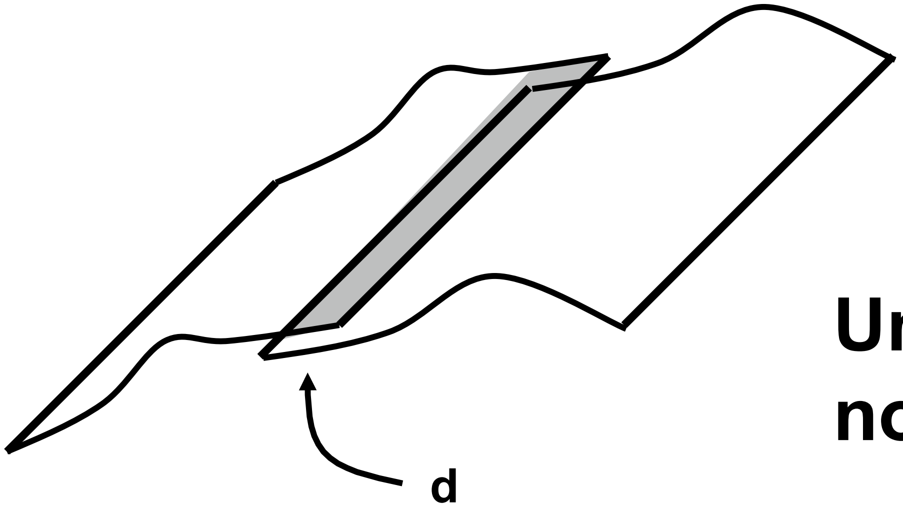
We can add an additional constraint

Scene-scene compatibility function, $\Psi(x_i, x_j)$



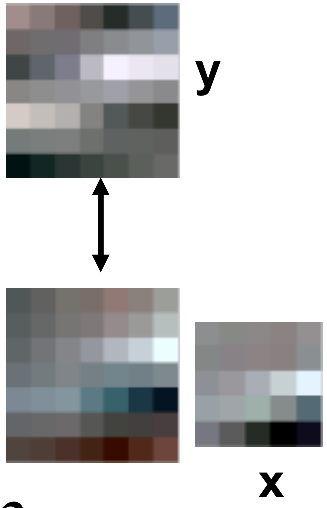
Assume overlapped regions, d, of hi-res. patches differ by Gaussian observation noise:

$$\Psi(x_i, x_j) = \exp^{-|d_i - d_j|^2/2\sigma^2}$$



Uniqueness constraint, not smoothness.

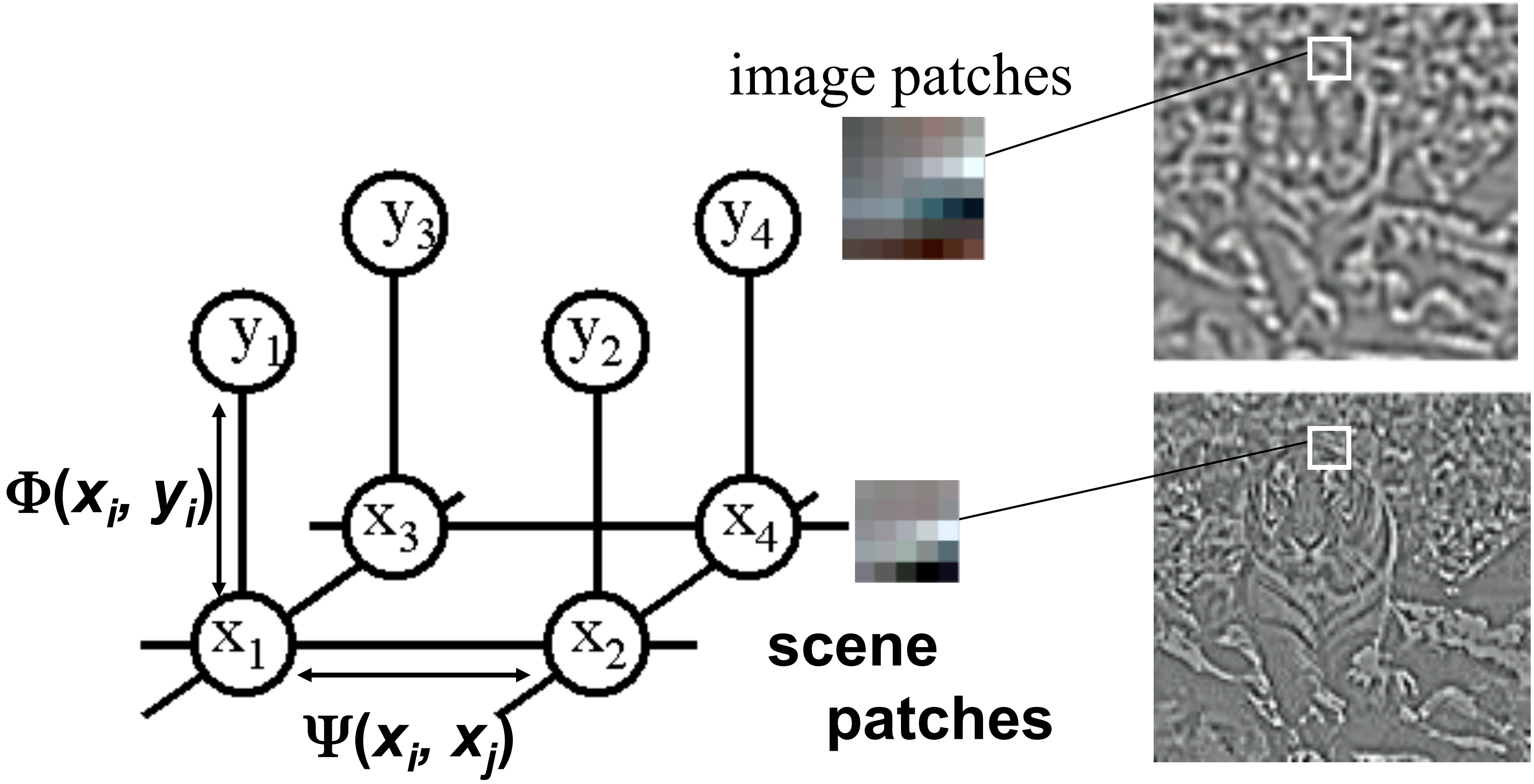
Image-scene compatibility function, $\Phi(x_i, y_i)$



Assume Gaussian noise takes you from observed image patch to synthetic sample:

$$\Phi(x_i, y_i) = \exp^{-|y_i - y(x_i)|^2/2\sigma^2}$$

Markov network



$\Phi(x_i, y_i)$

$\Psi(x_i, x_j)$

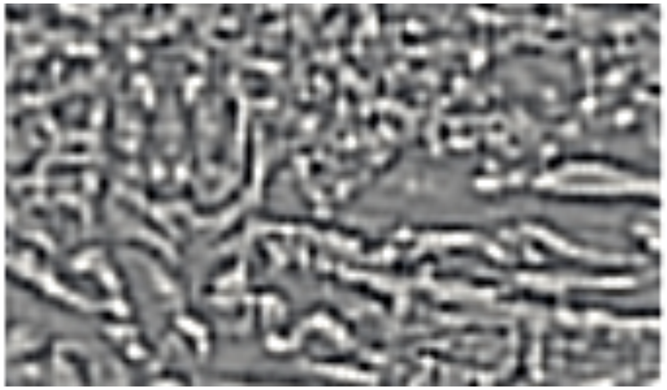
image patches

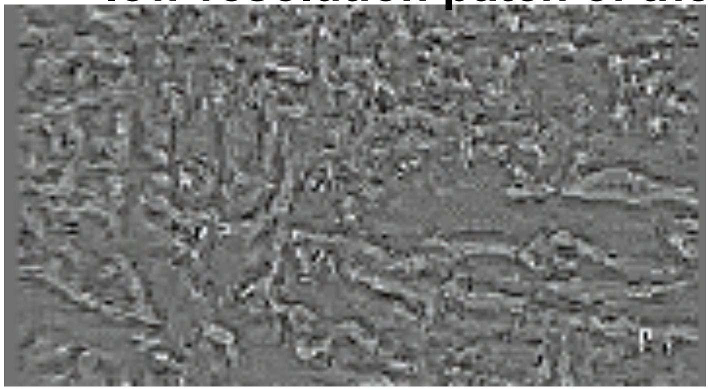
scene patches

Belief Propagation

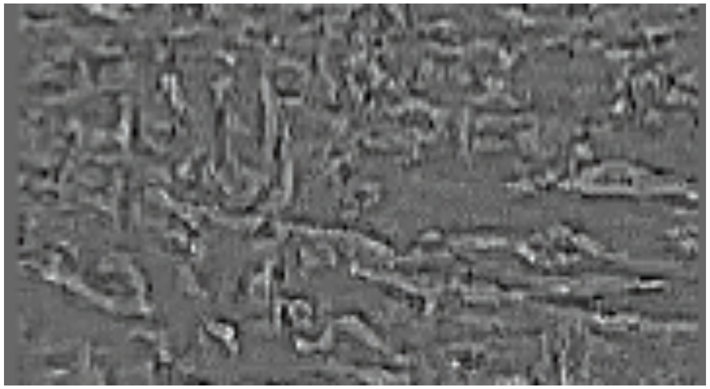
After a few iterations of belief propagation, the algorithm selects spatially consistent high resolution interpretations for each low-resolution patch of the input image.

Input

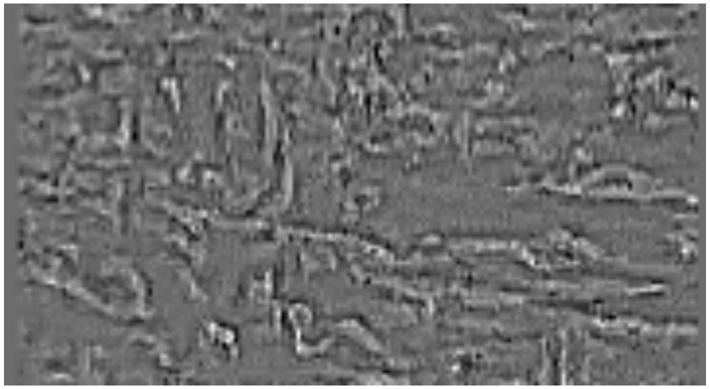




Iter. 0

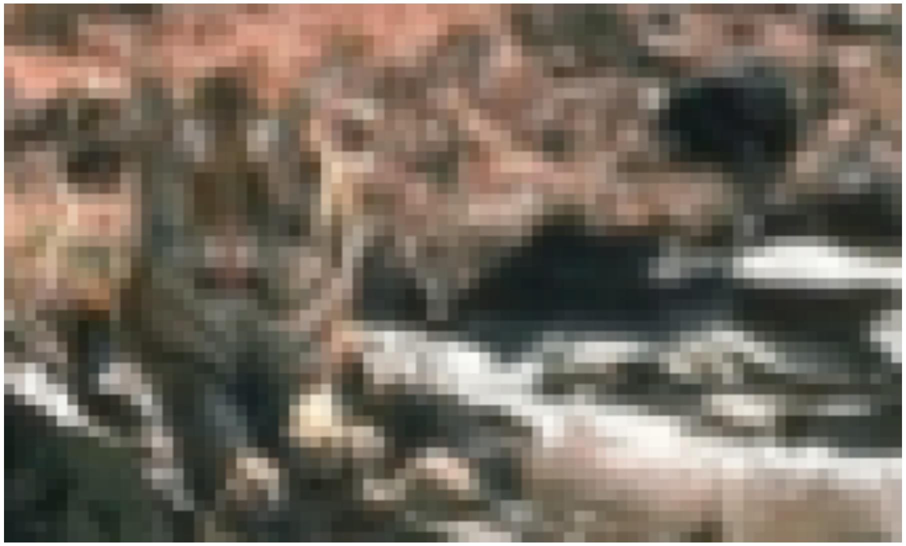


Iter. 1



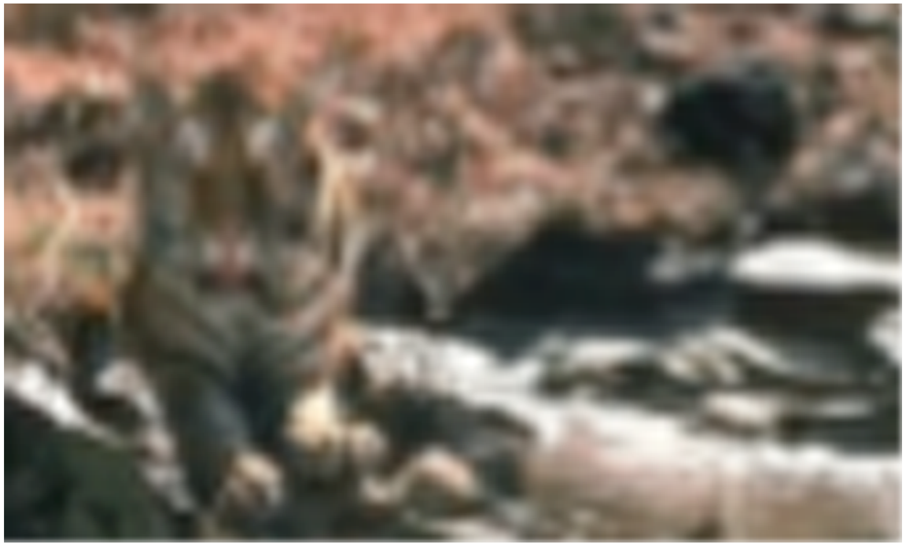
Iter. 3

Zooming 2 octaves

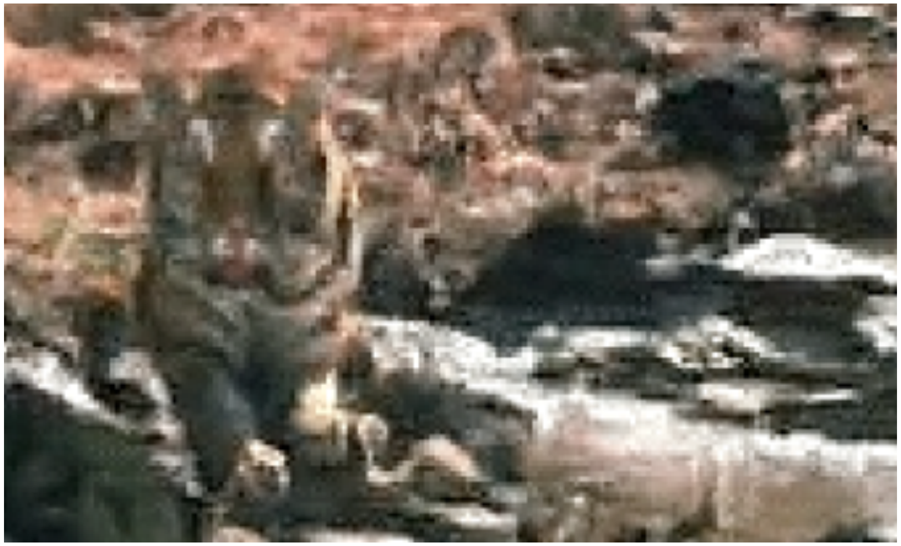


85 x 51 input

We apply the super-resolution algorithm recursively, zooming up 2 powers of 2, or a factor of 4 in each dimension.



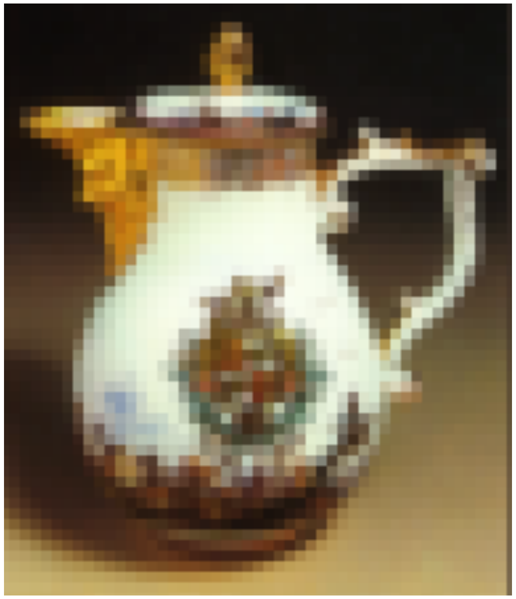
Cubic spline zoom to 340x204



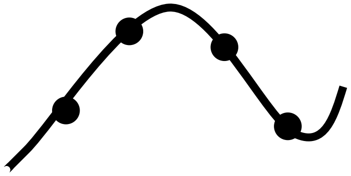
Max. likelihood zoom to 340x204

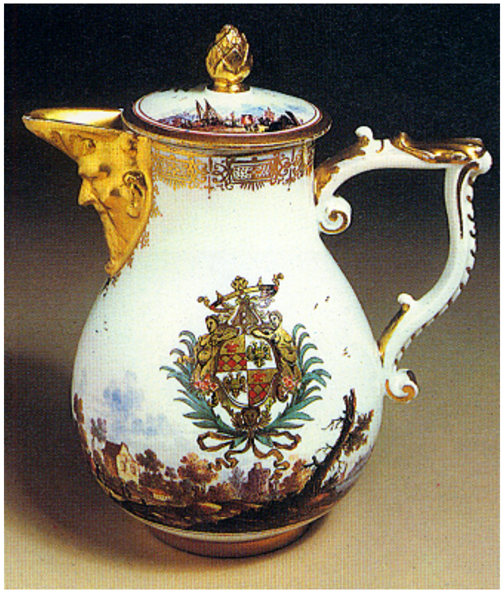
Now we examine the effect of the prior assumptions made about images on the high resolution reconstruction.
First, cubic spline interpolation.

Original
50x58



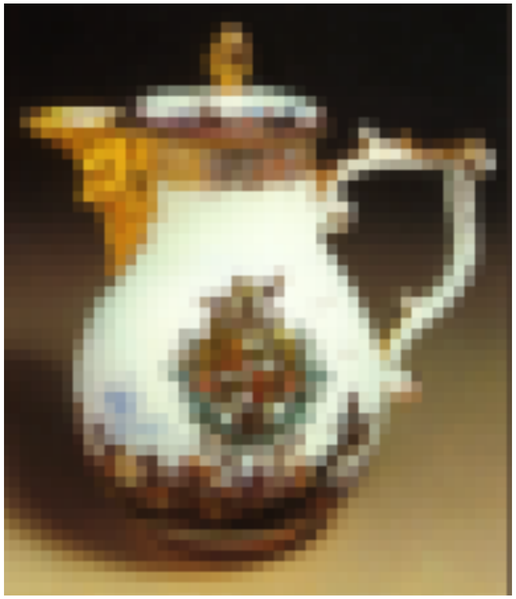
(cubic spline implies thin plate prior)



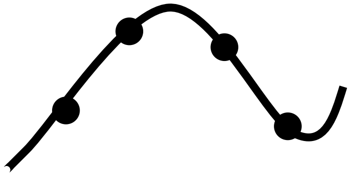


True
200x232

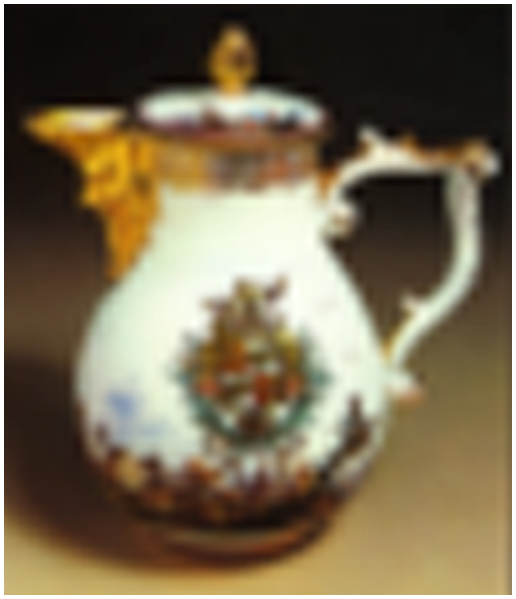
Original
50x58

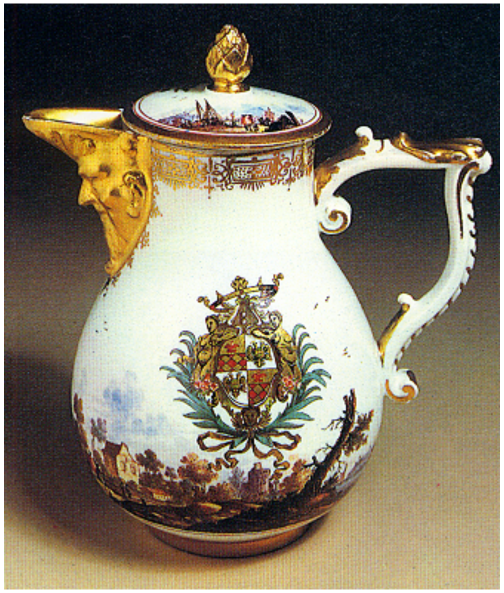


(cubic spline implies
thin plate prior)



Cubic spline

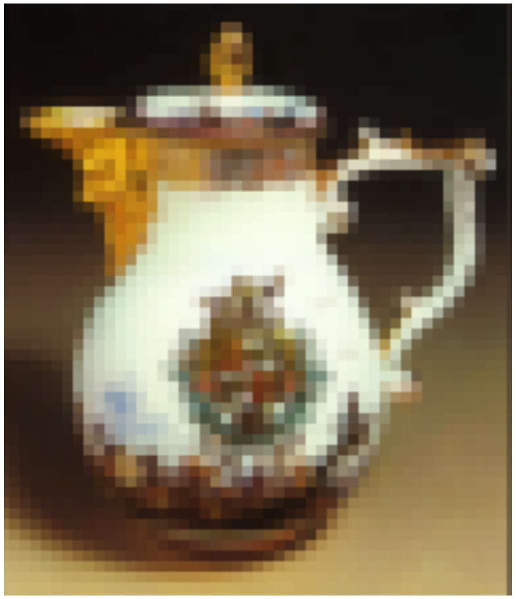


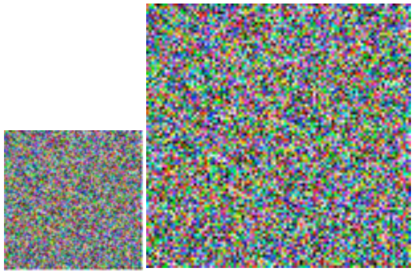


True
200x232

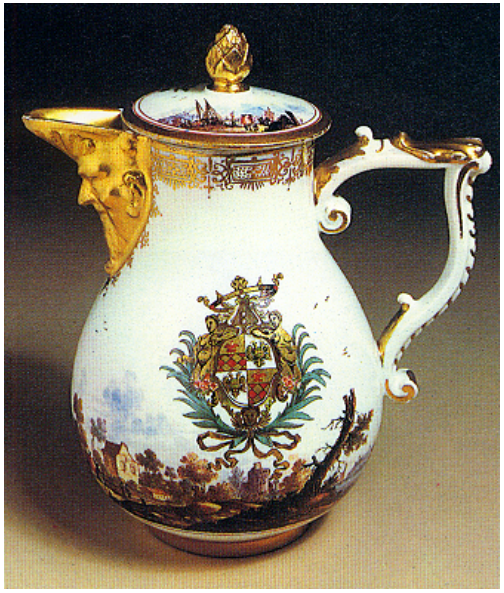
Next, train the Markov network algorithm on a world of random noise images.

Original
50x58





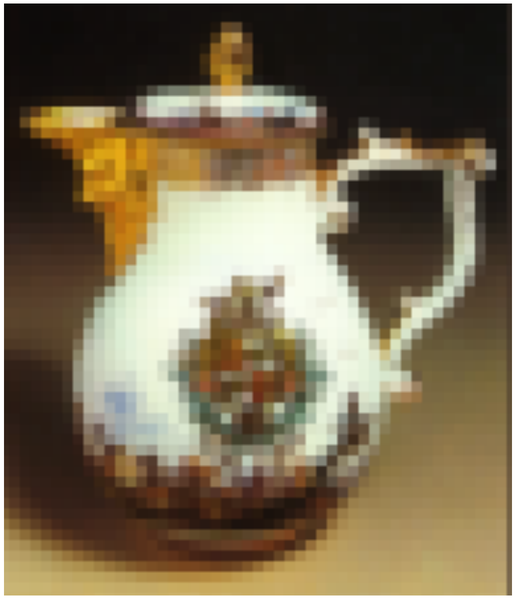
Training images

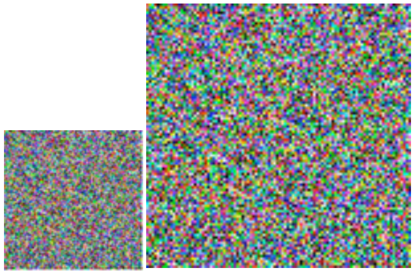


True

The algorithm learns that, in such a world, we add random noise when zoom to a higher resolution.

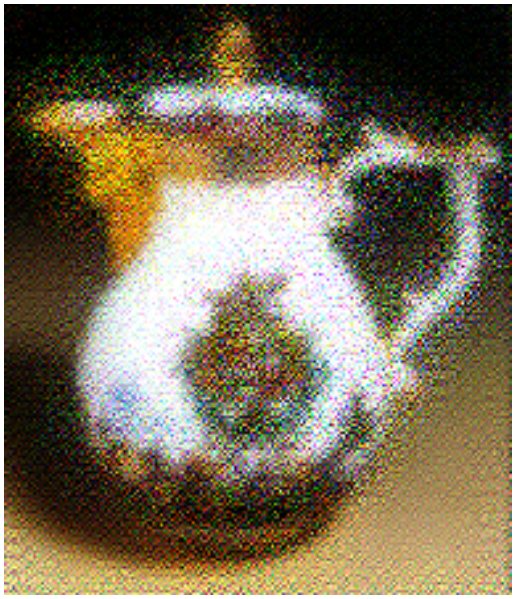
Original 50x58

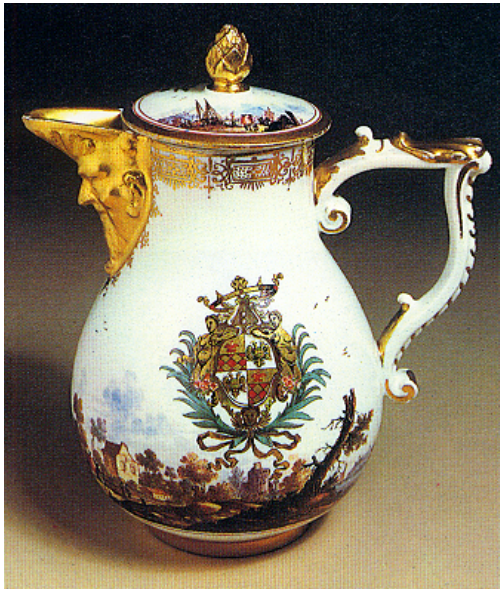




Training images

Markov network

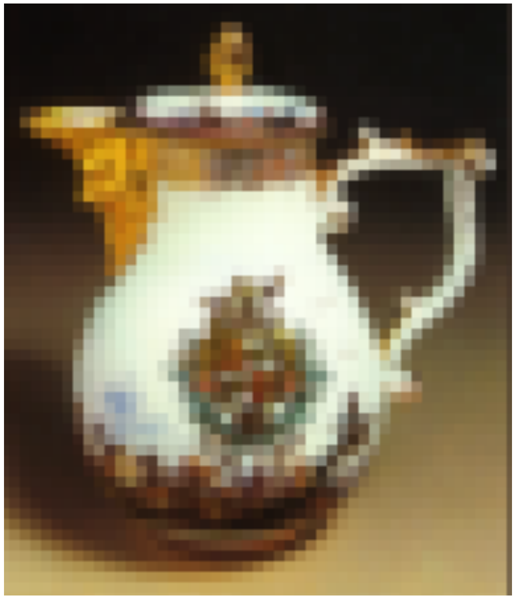


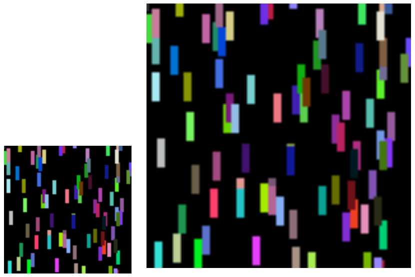


True

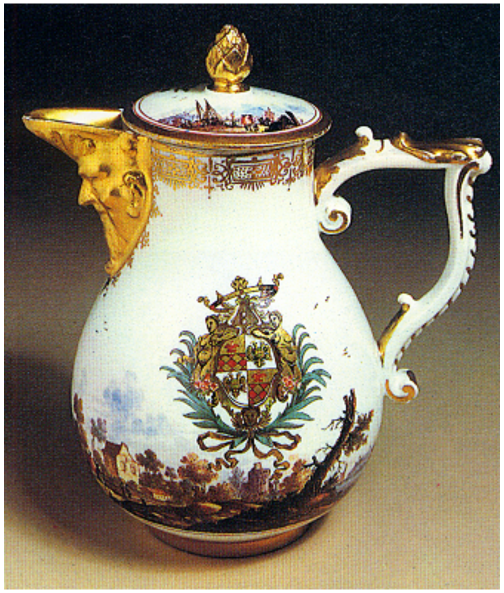
Next, train on a world of vertically oriented rectangles.

Original
50x58





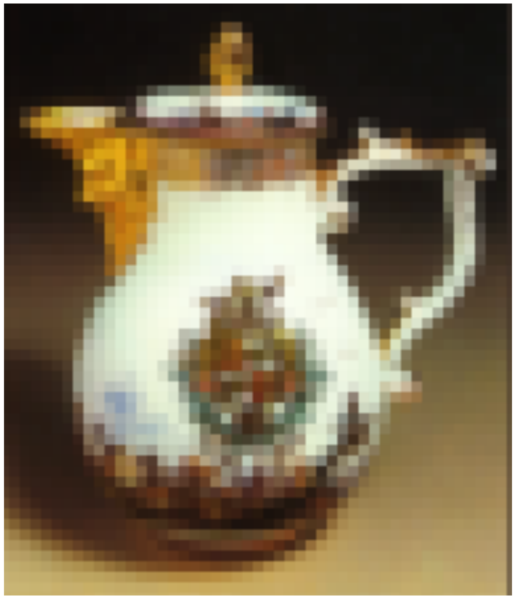
Training images

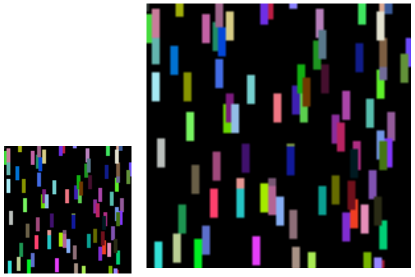


True

The Markov network algorithm hallucinates those vertical rectangles that it was trained on.

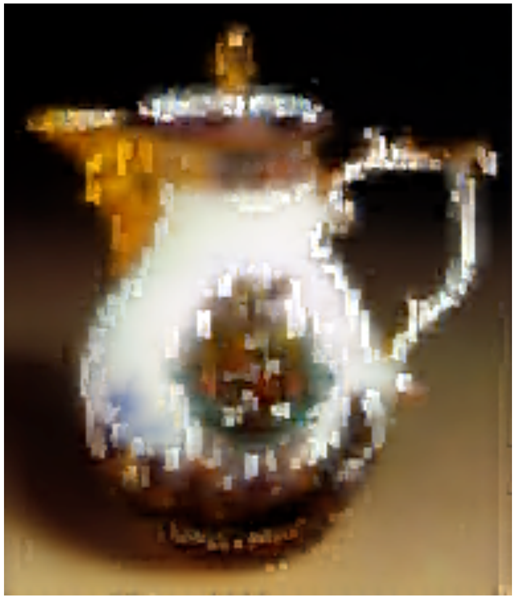
Original
50x58

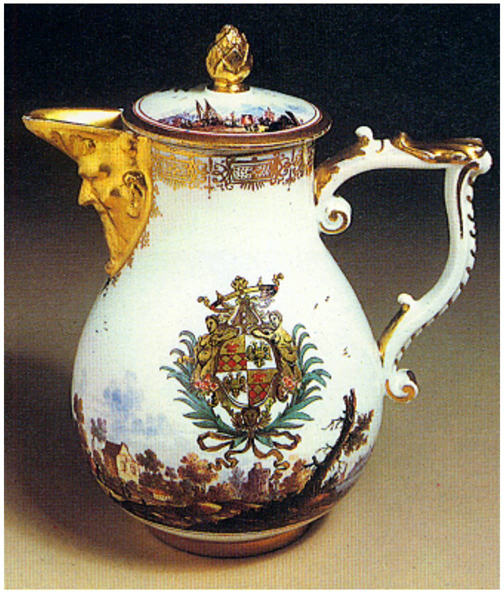




Training images

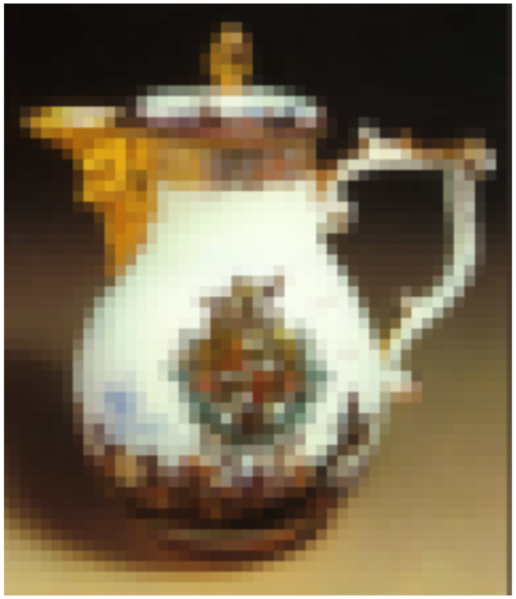
Markov
network





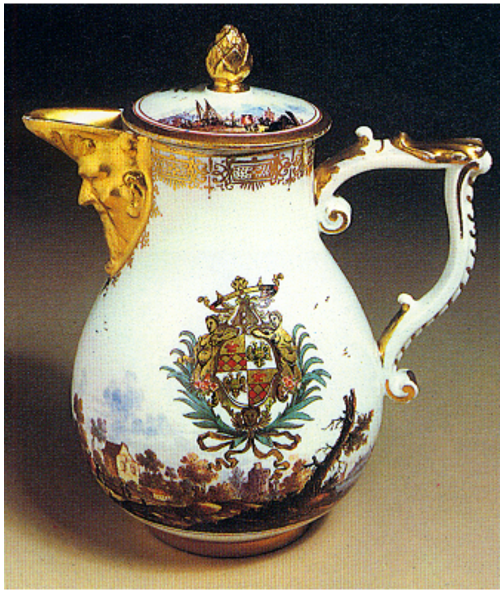
True

Original
50x58



Now train on a generic collection of images.

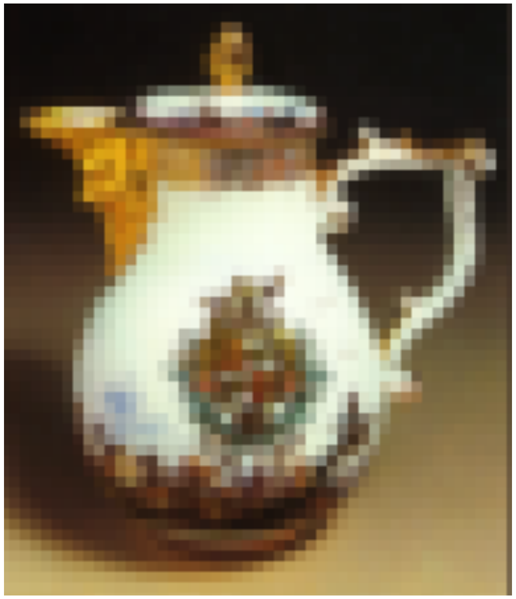
Training images



True

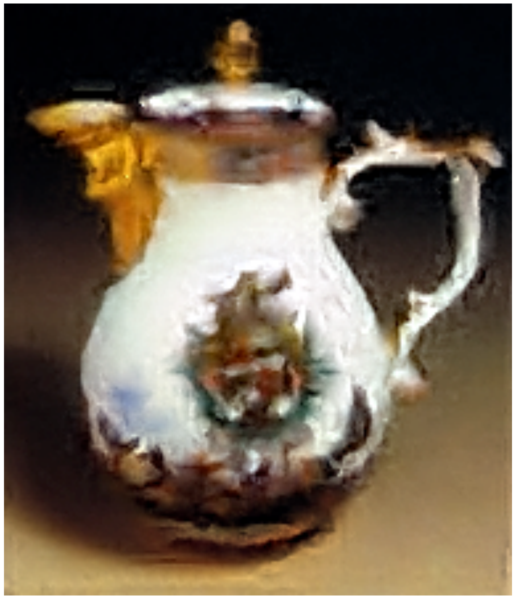
The algorithm makes a reasonable guess at the high resolution image, based on its training images.

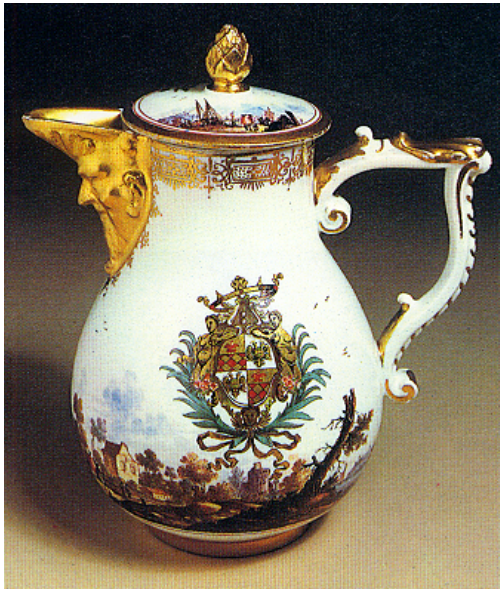
Original 50x58



Training images

Markov network



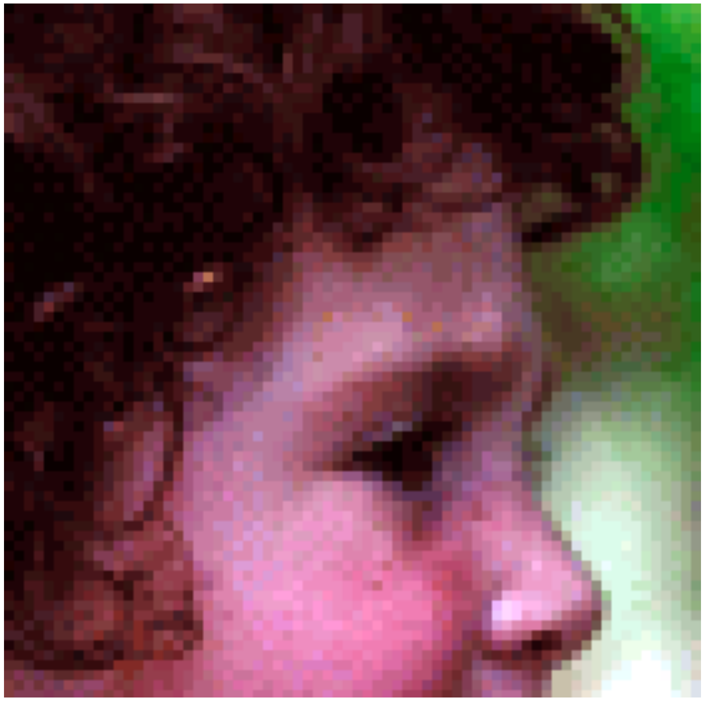


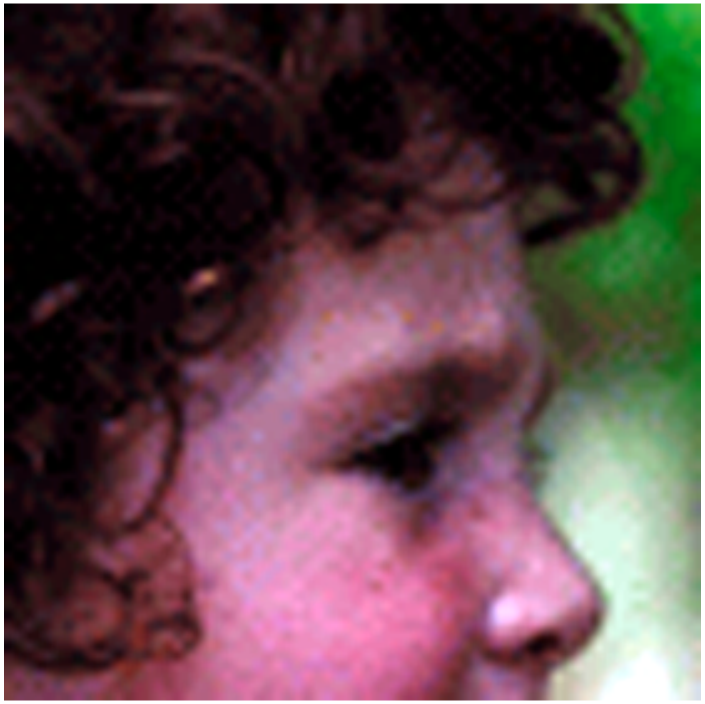
True

Generic training images

Next, train on a generic set of training images. Using the same camera as for the test image, but a random collection of photographs.

Original
70x70

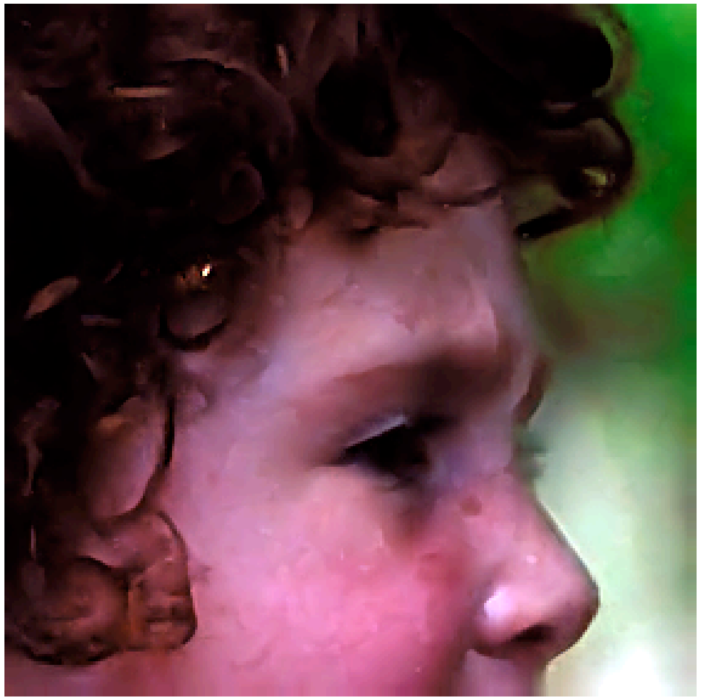


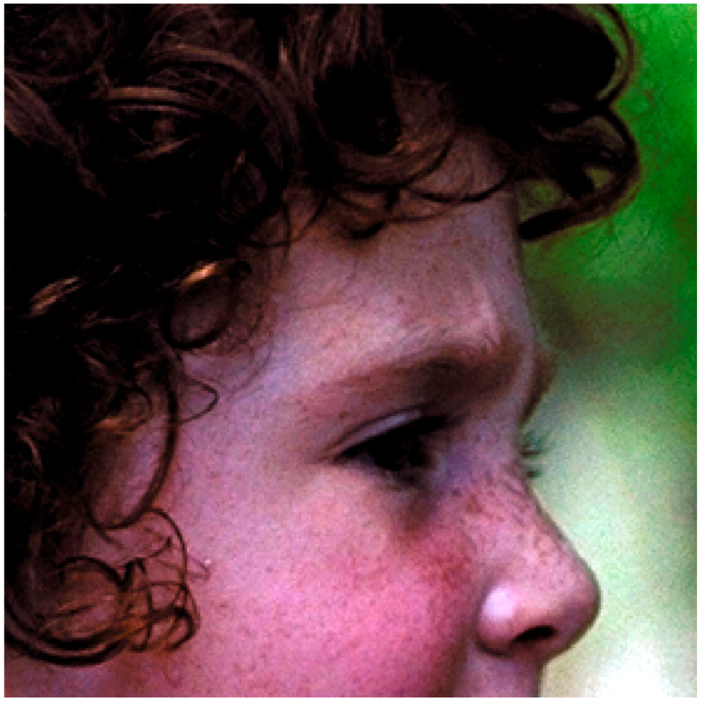


Cubic
Spline

Markov
net,

training:
generic





True
280x2
80

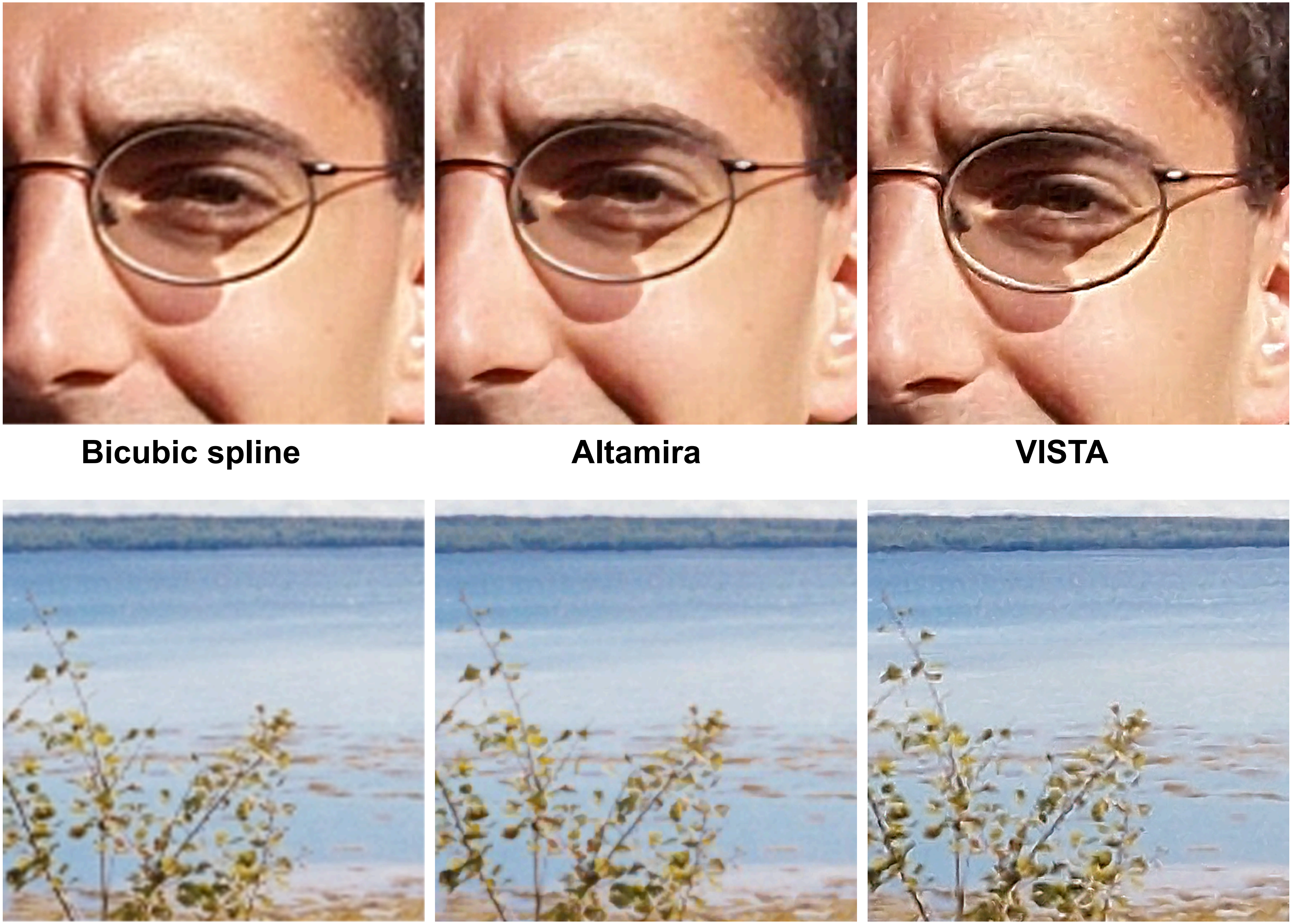
Kodak Imaging Science Technology Lab test.

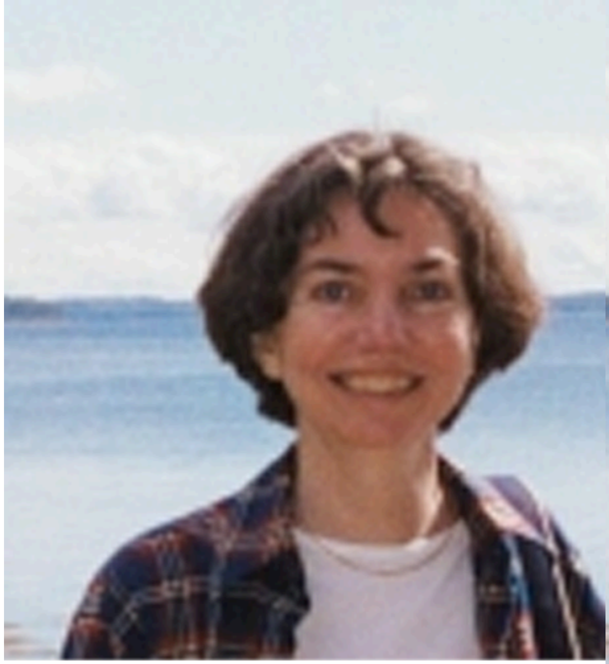
3 test images, 640x480, to be zoomed up by 4 in each dimension.

8 judges, making 2-alternative, forced-choice comparisons.

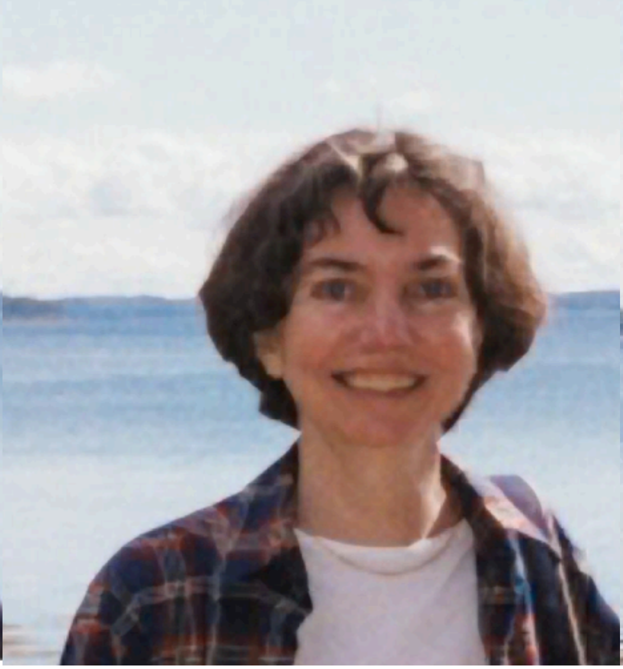
Algorithms compared

- **Bicubic Interpolation**
- **Mitra's Directional Filter**
- **Fuzzy Logic Filter**
- **Vector Quantization**
- **VISTA**

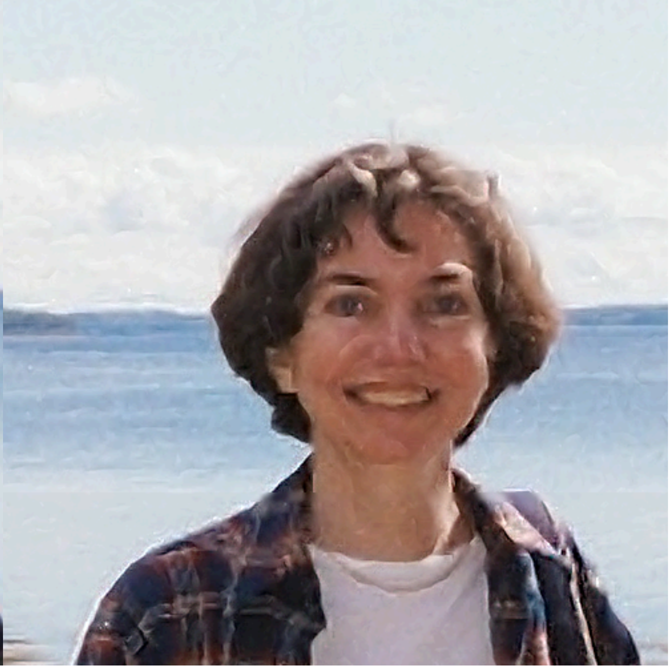

Bicubic spline
Altamira
VISTA



Bicubic spline

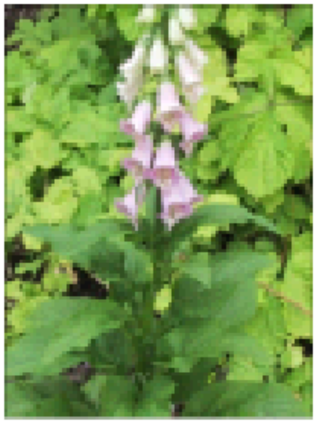


Altamira

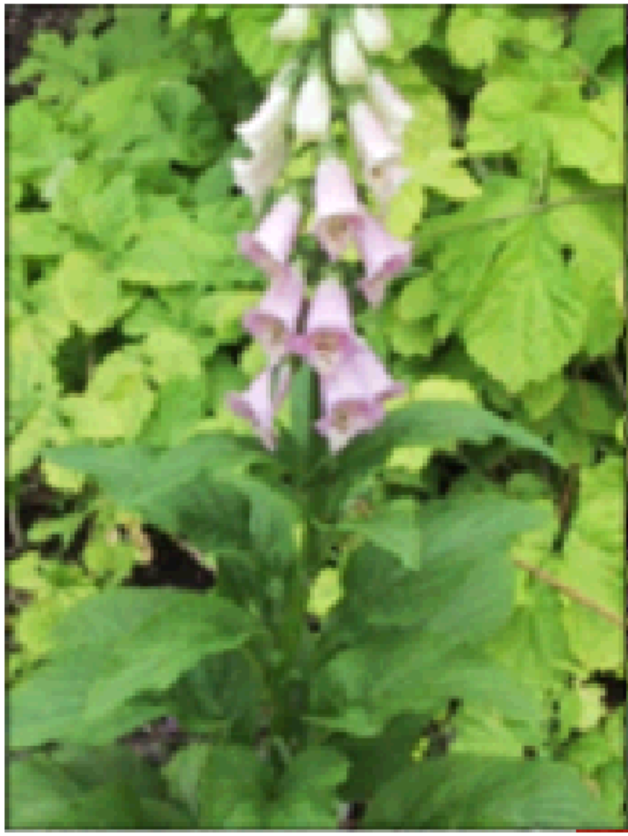


VISTA

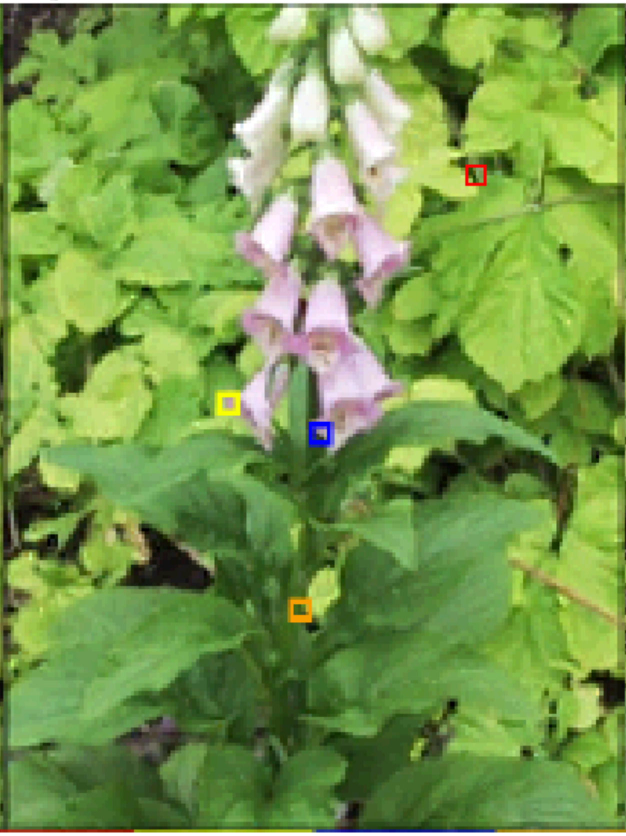
Input



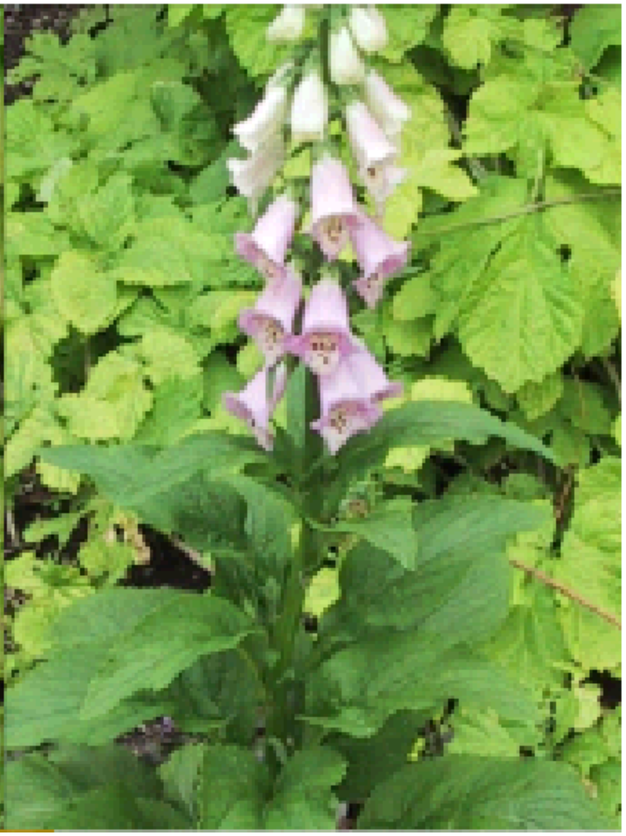
Cubic spline zoom



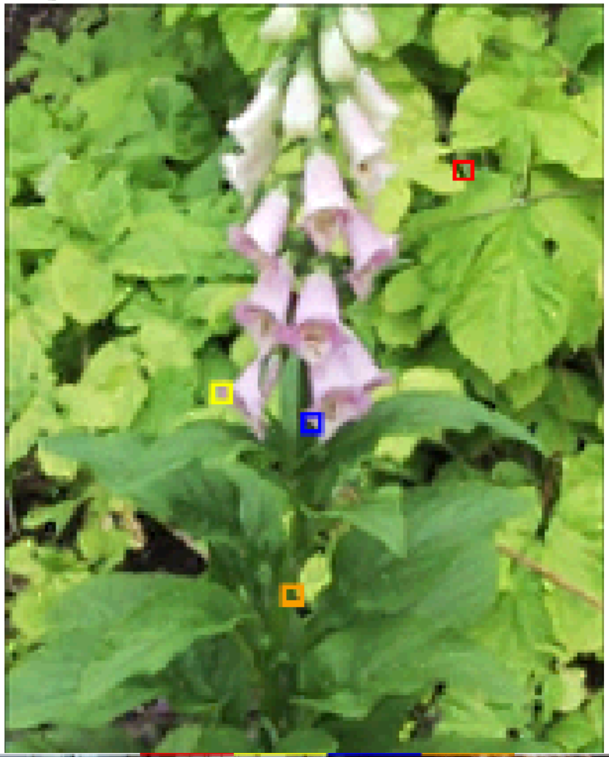
Super-resolution zoom



True high-resolution image



Super-resolution zoom



Source image patches

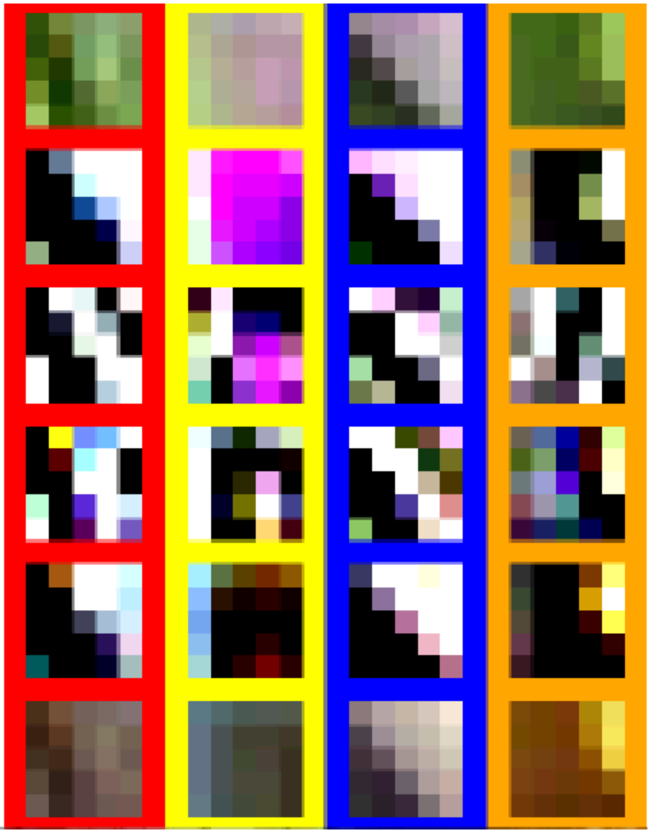
Bandpass filtered and contrast normalized

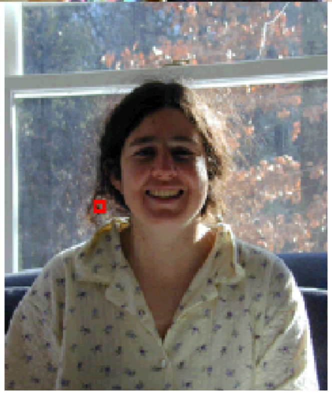
True high resolution pixels

High resolution pixels chosen by super-resolution

Bandpass filtered and contrast normalized best match patches from training data

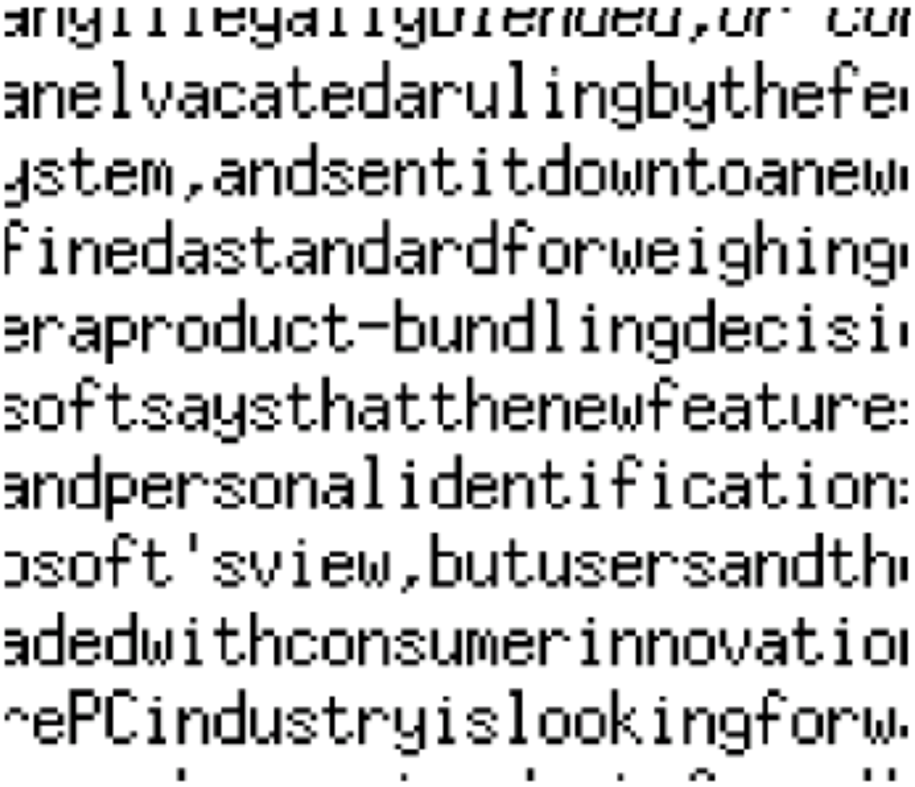
Best match patches from training data





Training images

Training image



anyillegallyblended,or cor

anelvacatedarulingbythefe

ystem,andsentitdowntoanew

finedastandardforweighing

eraproduct-bundlingdecisi

softsaysthatthenewfeature

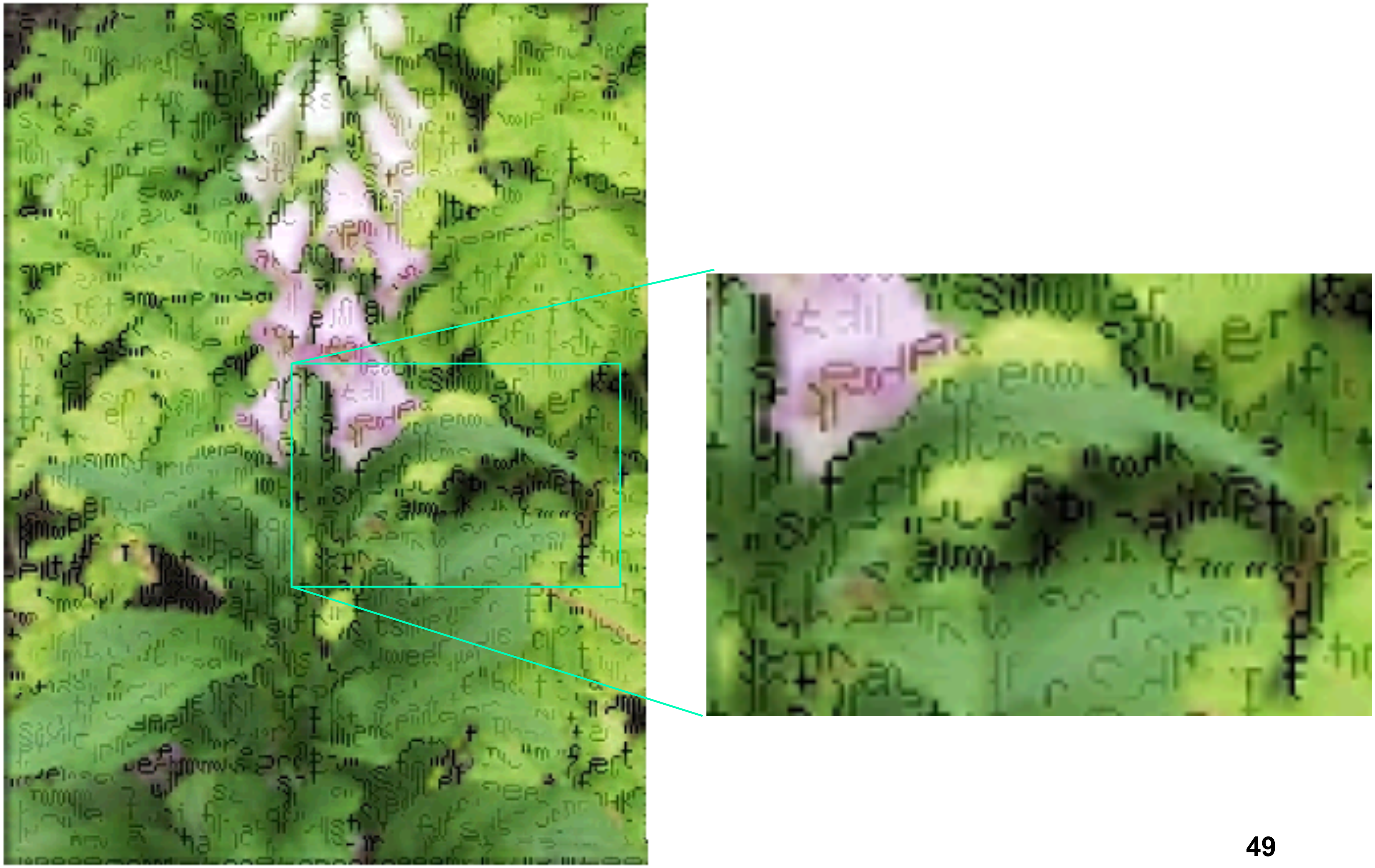
andpersonalidentification

osoft'sview,butusersandth

adedwithconsumerinnovatio

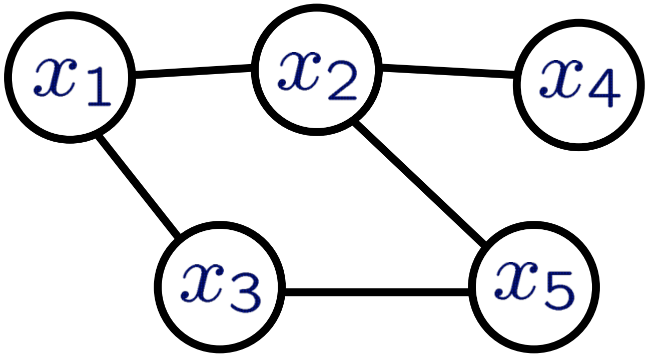
ePCindustryislookingforw

Processed image



Graphical Models

Pairwise Markov Random Fields

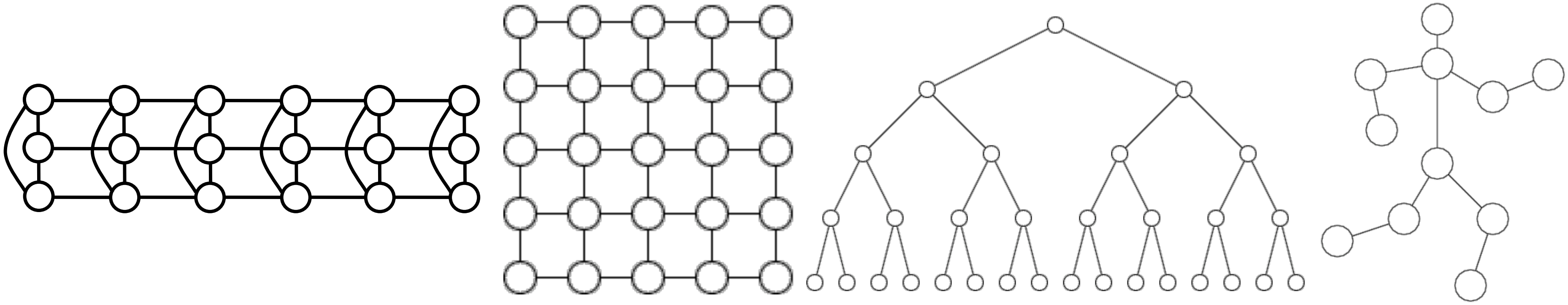


$\mathcal{V} \longrightarrow$ set of N nodes

$\mathcal{E} \longrightarrow$ edges (i, j) connecting nodes $i, j \in \mathcal{V}$

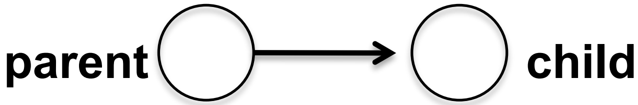
Nodes $i \in \mathcal{V}$ are associated with hidden variables x_i

Potential functions may depend on observations y



Directed graphical models

- An arc from A to B can be informally interpreted as indicating that A causes" B.1 Hence directed cycles are disallowedA directed, acyclic graph.
- Nodes are random variables. Can be scalars or vectors, continuous or discrete.
- The direction of the edge tells the parent-child-relation:

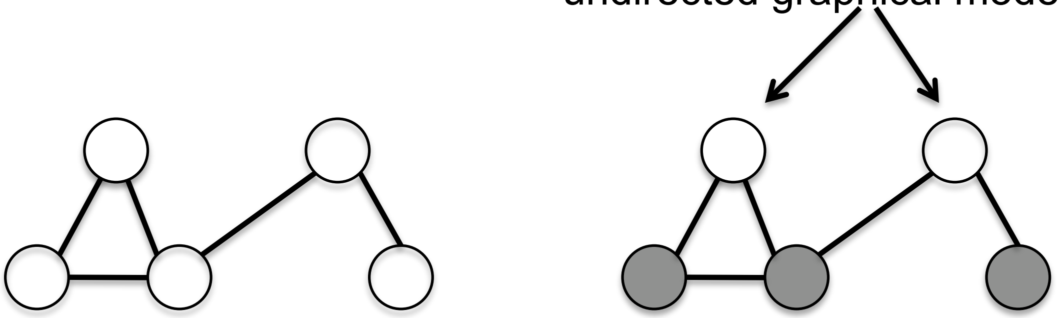


- With every node i is associated a conditional pdf defined by all the parent nodes π_i of node i. That conditional probability is $P_{x_i | x_{\pi_i}}$
- The joint distribution depicted by the graph is the product of all those conditional probabilities:

$$P_{x_1 \dots x_n} = \prod_{i=1}^{n} P_{x_i | x_{\pi_i}}$$

Undirected graphical models

- **A set of nodes joined by undirected edges.**
- **The graph makes conditional independencies explicit: If two nodes are not linked, and we condition on every other node in the graph, then those two nodes are conditionally independent.**

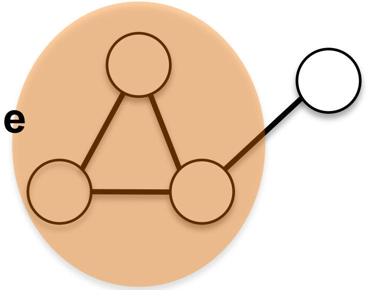


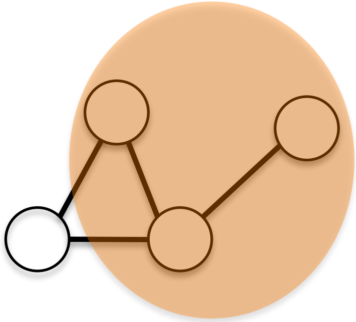
Conditionally independent, because are not connected by a line in the undirected graphical model

Undirected graphical models: cliques

- **Clique: a fully connected set of nodes**

clique

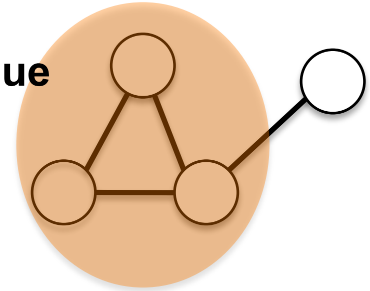




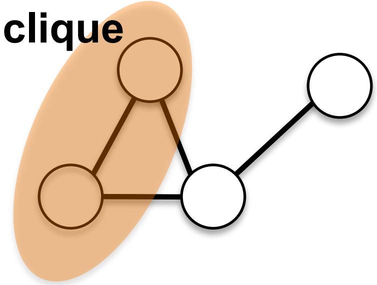
not a clique

- **A maximal clique is a clique that can't include more nodes of the graph w/o losing the clique property.**

Maximal clique



Non-maximal clique



Undirected graphical models: probability factorization

- **Hammersley-Clifford theorem addresses the pdf factorization implied by a graph: A distribution has the Markov structure implied by an undirected graph iff it can be represented in the factored form**

$$P_x = \frac{1}{Z} \prod_{c \in \xi} \Psi_{x_c}$$

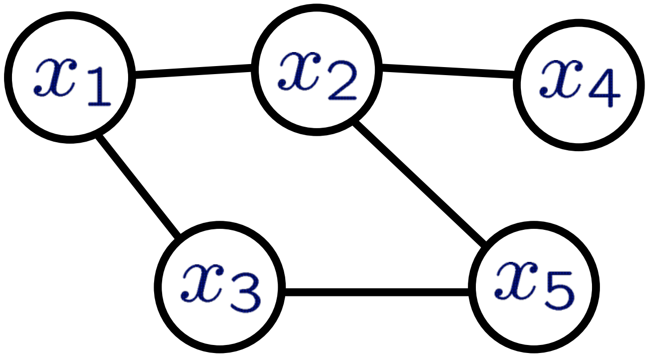
Potential functions of states of variables in maximal clique

Normalizing constant

set of maximal cliques

Graphical Models

Pairwise Markov Random Fields



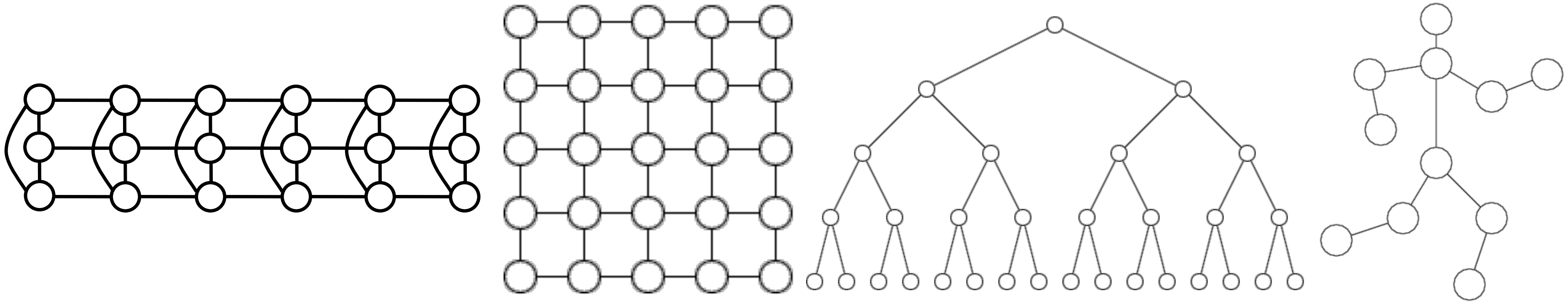
$\mathcal{V}$ ⟶ **set of** N **nodes**

$\mathcal{E}$ ⟶ **edges** (i,j) **connecting nodes** $i,j \in \mathcal{V}$

Nodes $i \in \mathcal{V}$ **are associated with hidden variables** x_i

$$p(x \mid y) \propto \prod_{(i,j)\in\mathcal{E}} \psi_{i,j}(x_i, x_j) \prod_{i\in\mathcal{V}} \psi_i(x_i, y)$$

Potential functions may depend on observations y



Making probability distributions modular, and therefore tractable:

Probabilistic graphical models

Vision is a problem involving the interactions of many variables: things can seem hopelessly complex. Everything is made tractable, or at least, simpler, if we modularize the problem. That's what probabilistic graphical models do, and let's examine that.

Readings: Jordan and Weiss intro article—fantastic!
Kevin Murphy web page—comprehensive and with
pointers to many advanced topics

A toy example

Suppose we have a system of 5 interacting variables, perhaps some are observed and some are not. There's some probabilistic relationship between the 5 variables, described by their joint probability,
P(x1, x2, x3, x4, x5).

If we want to find out what the likely state of variable x1 is (say, the position of the hand of some person we are observing), what can we do?

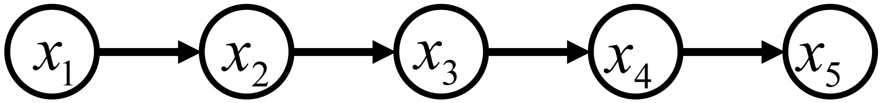
Two reasonable choices are: (a) find the value of x1 (and of all the other variables) that gives the maximum of P(x1, x2, x3, x4, x5); that's the MAP solution.
Or (b) marginalize over all the other variables and then take the mean or the maximum of the other variables. Marginalizing, then taking the mean, is equivalent to finding the MMSE solution. Marginalizing, then taking the max, is called the max marginal solution and sometimes a useful thing to do.

To find the marginal probability at x1, we have to take this sum:

$$\sum_{x_2, x_3, x_4, x_5} P(x_1, x_2, x_3, x_4, x_5)$$

If the system really is high dimensional, that will quickly become intractable. But if there is some modularity in $P(x_1, x_2, x_3, x_4, x_5)$ then things become tractable again.

Suppose the variables form a Markov chain: x1 causes x2 which causes x3, etc. We might draw out this relationship as follows:



P(a,b) = P(b|a) P(a)

By the chain rule, for any probability distribution, we have:

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2, x_3, x_4, x_5 \mid x_1)$$

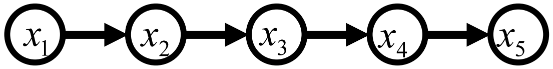
$$= P(x_1)P(x_2 \mid x_1)P(x_3, x_4, x_5 \mid x_1, x_2)$$

$$= P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1, x_2)P(x_4, x_5 \mid x_1, x_2, x_3)$$

$$= P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1, x_2)P(x_4 \mid x_1, x_2, x_3)P(x_5 \mid x_1, x_2, x_3, x_4)$$

But if we exploit the assumed modularity of the probability distribution over the 5 variables (in this case, the assumed Markov chain structure), then that expression simplifies:

$$= P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_2)P(x_4 \mid x_3)P(x_5 \mid x_4)$$



Now our marginalization summations distribute through those terms:

$$\sum_{x_2, x_3, x_4, x_5} P(x_1, x_2, x_3, x_4, x_5) = \quad P(x_1)\sum_{x_2} P(x_2 \mid x_1)\sum_{x_3} P(x_3 \mid x_2)\sum_{x_4} P(x_4 \mid x_3)\sum_{x_5} P(x_5 \mid x_4)$$

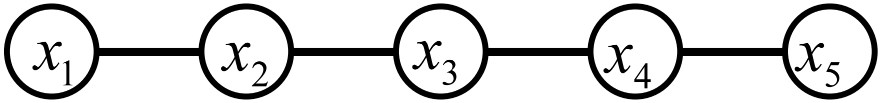
Belief propagation

Performing the marginalization by doing the partial sums is called "belief propagation".

$$\sum_{x_2, x_3, x_4, x_5} P(x_1, x_2, x_3, x_4, x_5) = \quad P(x_1) \sum_{x_2} P(x_2 \mid x_1) \sum_{x_3} P(x_3 \mid x_2) \sum_{x_4} P(x_4 \mid x_3) \sum_{x_5} P(x_5 \mid x_4)$$

In this example, it has saved us a lot of computation. Suppose each variable has 10 discrete states. Then, not knowing the special structure of P, we would have to perform 10000 additions (10^4) to marginalize over the four variables.
But doing the partial sums on the right hand side, we only need 40 additions (10*4) to perform the same marginalization!

Another modular probabilistic structure, more common in vision problems, is an undirected graph:



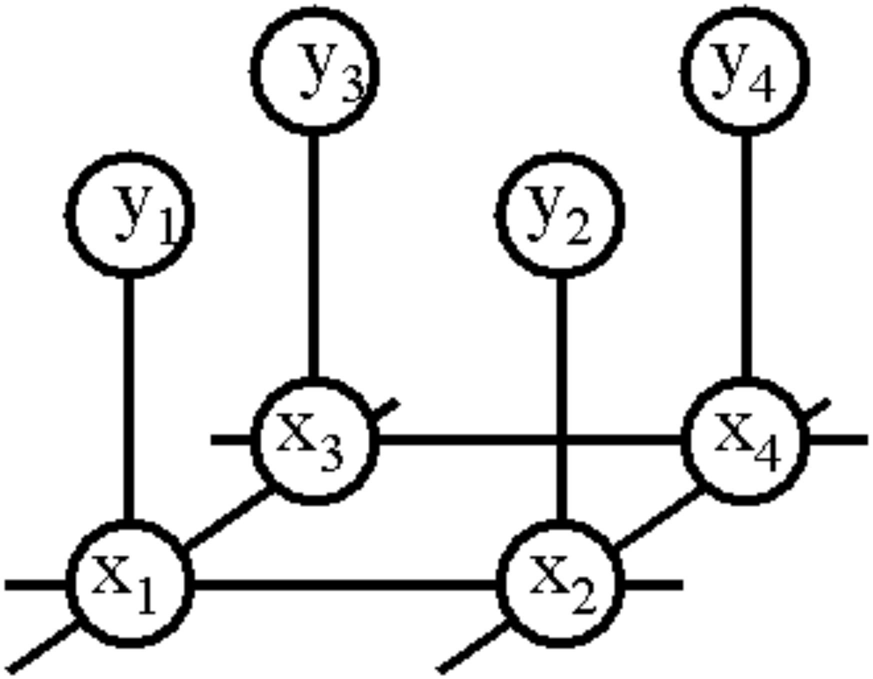
The joint probability for this graph is given by:

$$P(x_1, x_2, x_3, x_4, x_5) = \Phi(x_1, x_2)\Phi(x_2, x_3)\Phi(x_3, x_4)\Phi(x_4, x_5)$$

Where $\Phi(x_1, x_2)$ is called a “compatibility function”. We can define compatibility functions we result in the same joint probability as for the directed graph described in the previous slides; for that example, we could use either form.

Markov Random Fields

- Allows rich probabilistic models for images.
- But built in a local, modular way. Learn local relationships, get global effects out.



MRF nodes as pixels

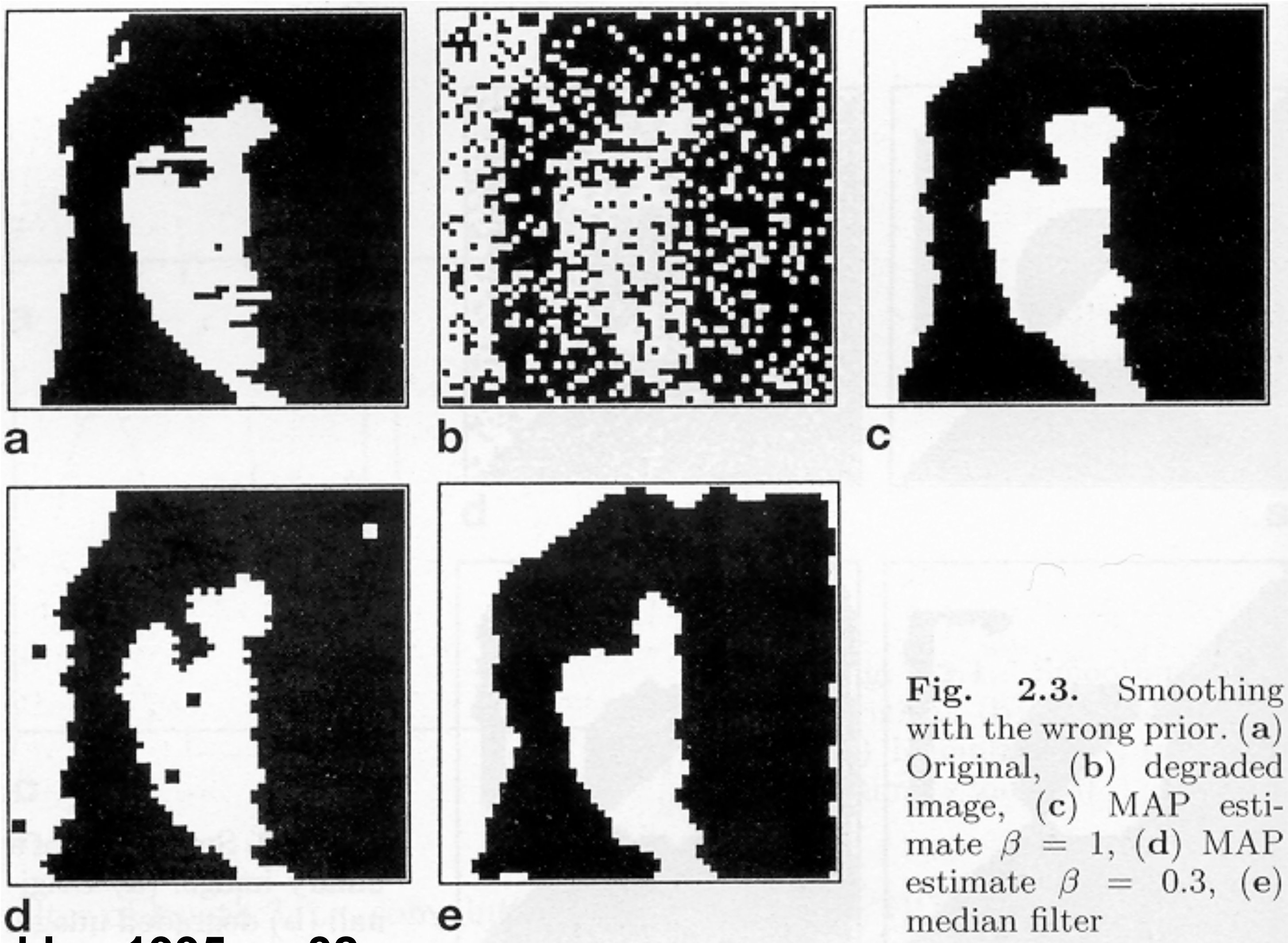


Fig. 2.3. Smoothing with the wrong prior. (**a**) Original, (**b**) degraded image, (**c**) MAP estimate $\beta = 1$, (**d**) MAP estimate $\beta = 0.3$, (**e**) median filter

Winkler, 1995, p. 32

MRF nodes as patches

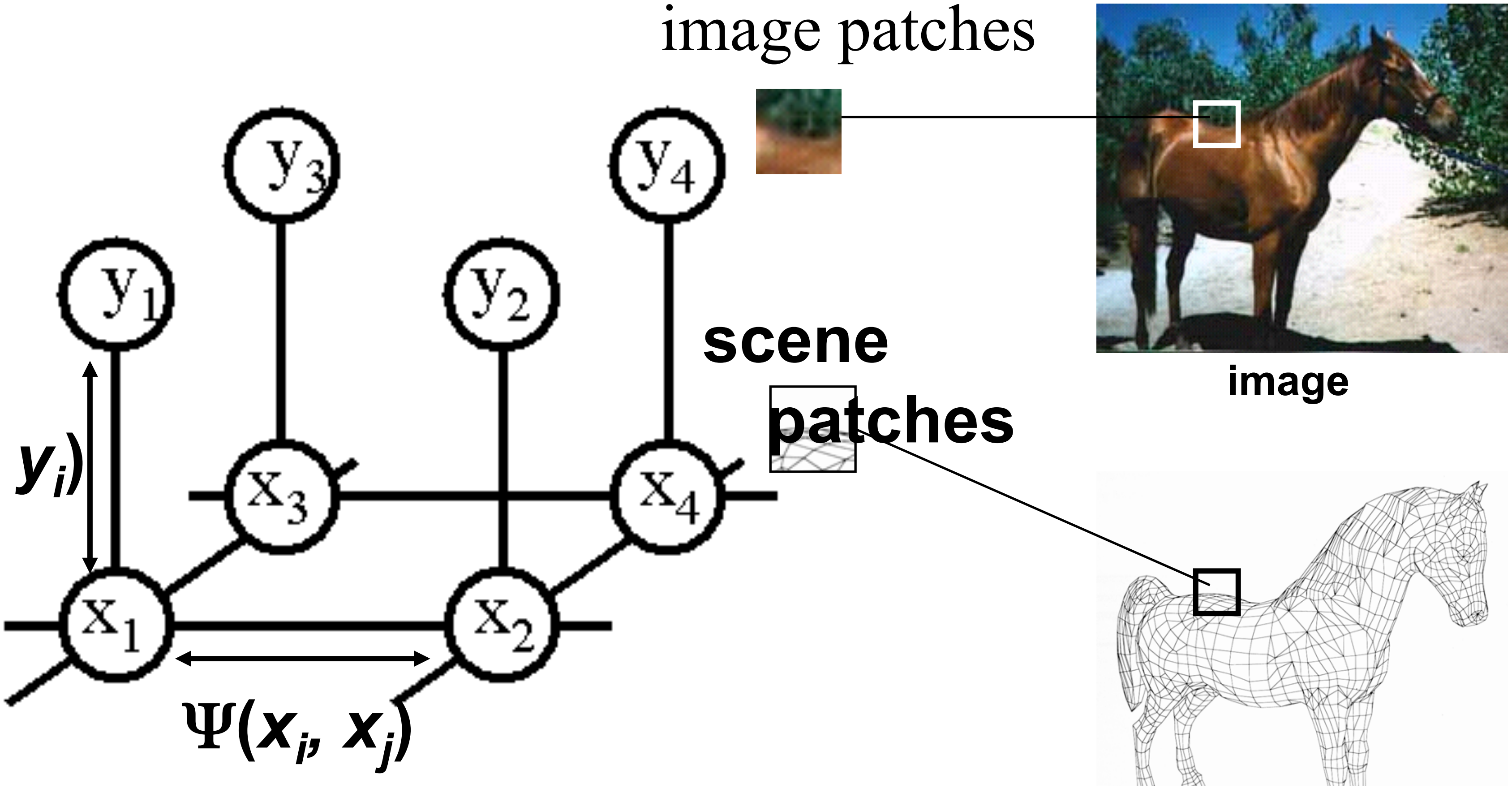


image patches

scene patches

$\Phi(x_i, y_i)$

$\Psi(x_i, x_j)$

image

scene

Network joint probability

$$P(x,y) = \frac{1}{Z}\prod_{i,j}\Psi(\underbrace{x_i, x_j}_{\text{neighboring scene nodes}})\prod_{i}\Phi(x_i, y_i)$$

scene

image

Scene-scene compatibility function

Image-scene compatibility function

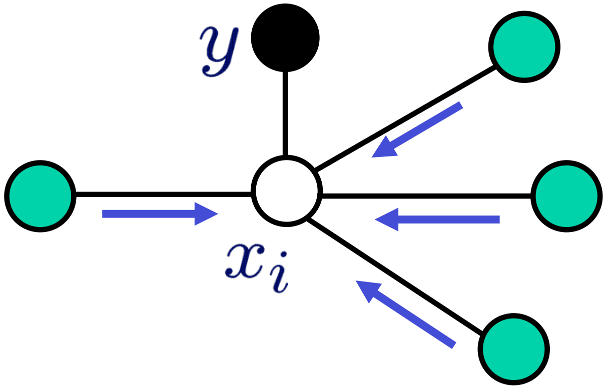
local observations

In order to use MRFs:

- Given observations y, and the parameters of the MRF, how infer the hidden variables, x?
- How learn the parameters of the MRF?

Belief Propagation

BELIEFS: **Approximate posterior marginal distributions**

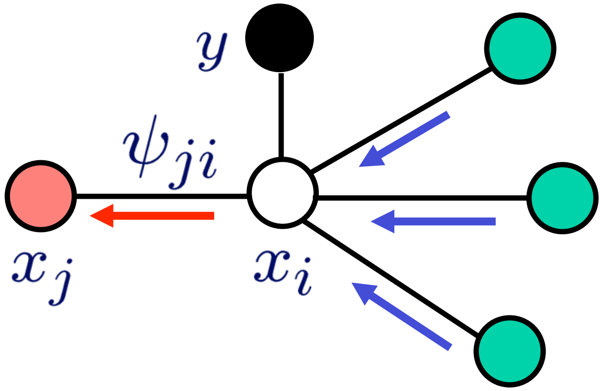


$$\widehat{p}(x_i \mid y) \propto \psi_i(x_i, y) \prod_{k \in \Gamma(i)} m_{ki}(x_i)$$

$\Gamma(i)$ ⟶ ***neighborhood*** **of node *i***

MESSAGES: **Approximate sufficient statistics**

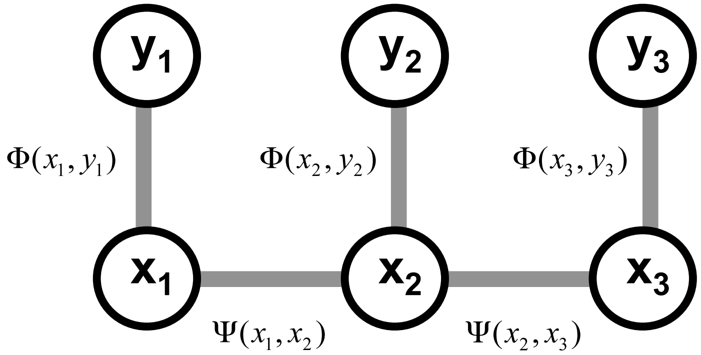
$$m_{ij}(x_j) \propto \int_{x_i} \psi_{j,i}(x_j, x_i) \psi_i(x_i, y) \prod_{k \in \Gamma(i) \setminus j} m_{ki}(x_i) \, dx_i$$



I. Belief Update (Message Product)

II. Message Propagation (Convolution)

Derivation of belief propagation



minimum mean square error (MMSE)

$$x_{1MMSE} = \underset{x_1}{\text{mean}} \; \underset{x_2}{\text{sum}} \; \underset{x_3}{\text{sum}} \; P(x_1, x_2, x_3, y_1, y_2, y_3)$$

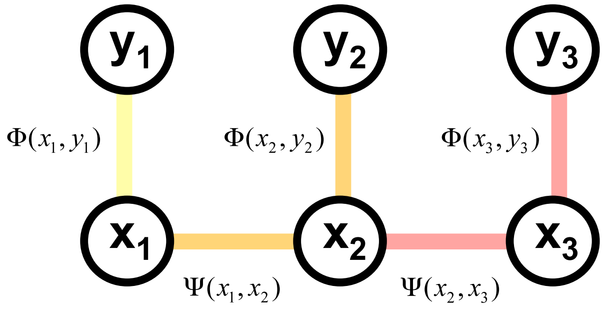
The posterior factorizes

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ P(x_1, x_2, x_3, y_1, y_2, y_3)$$

$$= \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ \Phi(x_1, y_1)$$

$$\Phi(x_2, y_2)\ \Psi(x_1, x_2)$$

$$\Phi(x_3, y_3)\ \Psi(x_2, x_3)$$

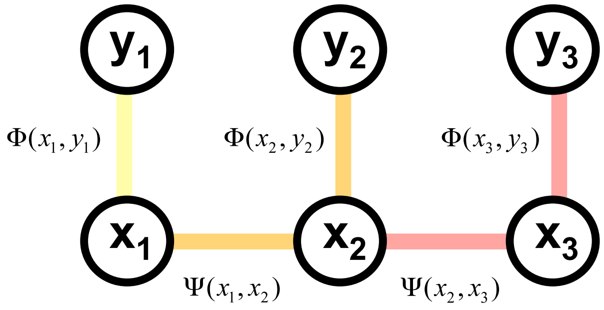


Propagation rules

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ P(x_1, x_2, x_3, y_1, y_2, y_3)$$

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ \Phi(x_1, y_1)$$
$$\Phi(x_2, y_2)\ \Psi(x_1, x_2)$$
$$\Phi(x_3, y_3)\ \Psi(x_2, x_3)$$

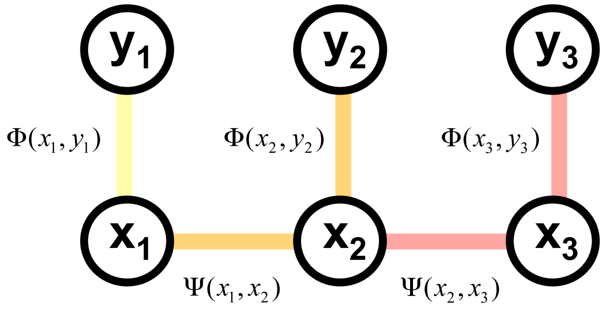
$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \Phi(x_1, y_1)$$
$$\underset{x_2}{\text{sum}}\ \Phi(x_2, y_2)\ \Psi(x_1, x_2)$$
$$\underset{x_3}{\text{sum}}\ \Phi(x_3, y_3)\ \Psi(x_2, x_3)$$



Propagation rules

$$x_{1MMSE} = \operatorname*{mean}_{x_1} \Phi(x_1, y_1) \operatorname*{sum}_{x_2} \Phi(x_2, y_2)\, \Psi(x_1, x_2) \operatorname*{sum}_{x_3} \Phi(x_3, y_3)\, \Psi(x_2, x_3)$$

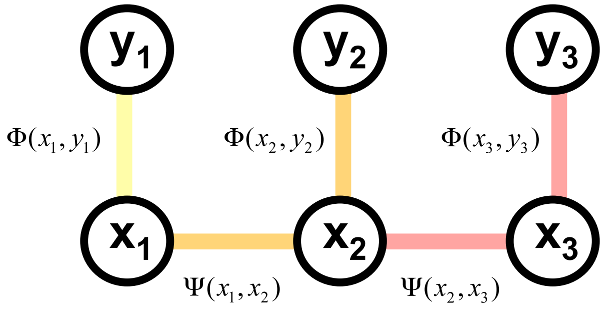
$$M_1^2(x_1) = \operatorname*{sum}_{x_2}\ \Psi(x_1, x_2)\ \ \Phi(x_2, y_2)\ \ M_2^3(x_2)$$



Propagation rules

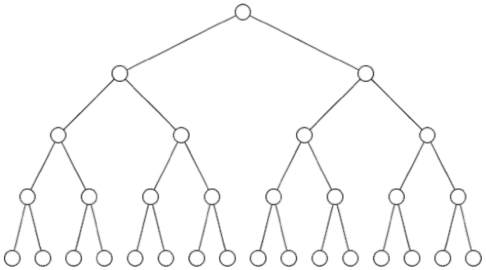
$$x_{1MMSE} = \operatorname*{mean}_{x_1} \Phi(x_1, y_1) \boxed{\operatorname*{sum}_{x_2} \Phi(x_2, y_2)\, \Psi(x_1, x_2)} \boxed{\operatorname*{sum}_{x_3} \Phi(x_3, y_3)\, \Psi(x_2, x_3)}$$

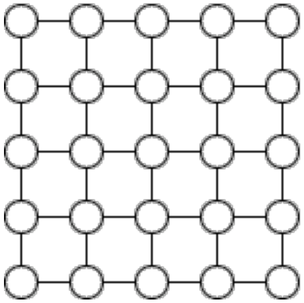
$$\boxed{M_1^2(x_1)} = \operatorname*{sum}_{x_2} \Psi(x_1, x_2)\ \Phi(x_2, y_2) \boxed{M_2^3(x_2)}$$



Justifications for BP

- **Gives *exact* marginals for trees**
  - → ***Optimal estimates***
  - → ***Confidence measures***





- For general graphs, *loopy BP* has excellent empirical performance in many applications
- Recent theory provides some guarantees:
  - Statisical physics: *variational method*
    (Yedidia, Freeman, & Weiss)
  - BP as reparameterization: *error bounds*
    (Wainwright, Jaakkola, & Willsky)
  - Many others…

Belief propagation: the nosey neighbor rule

“Given everything that I know, here’s what I think you should think”

(Given the probabilities of my being in different states, and how my states relate to your states, here’s what I think the probabilities of your states should be)

Optimal solution in a chain or tree: Belief Propagation

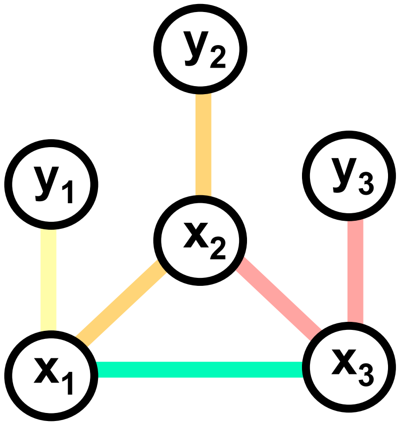
- “Do the right thing” Bayesian algorithm.
- For Gaussian random variables over time: Kalman filter.
- For hidden Markov models: forward/ backward algorithm (and MAP variant is Viterbi).

No factorization with loops!

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\, \Phi(x_1, y_1)$$

$$\underset{x_2}{\text{sum}}\, \Phi(x_2, y_2)\, \Psi(x_1, x_2)$$

$$\underset{x_3}{\text{sum}}\, \Phi(x_3, y_3)\, \Psi(x_2, x_3) \;\; \boldsymbol{\Psi(x_1, x_3)}$$



References on BP and GBP

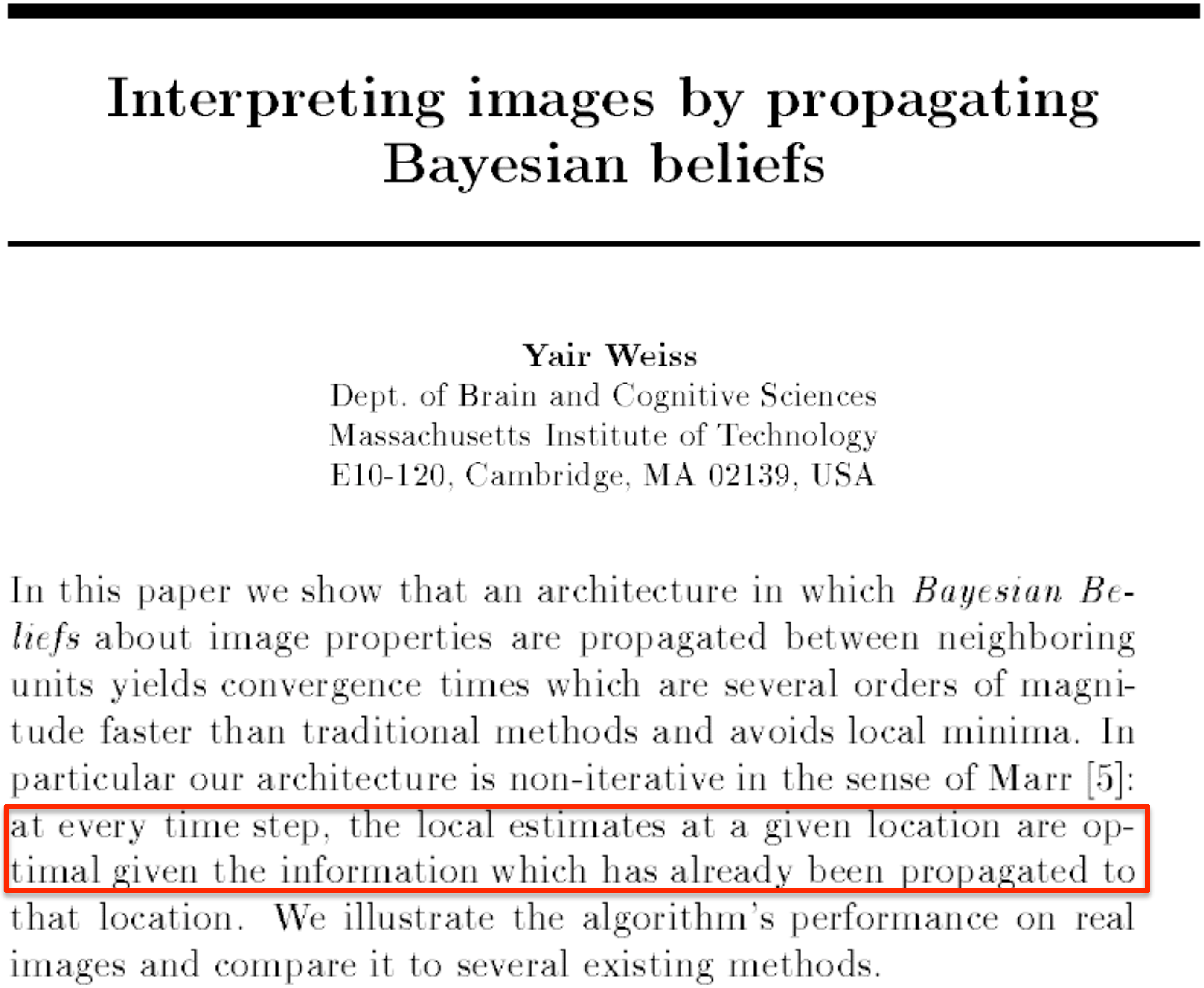
- J. Pearl, 1985
  - classic
- Y. Weiss, NIPS 1998
  - Inspires application of BP to vision
- W. Freeman et al learning low-level vision, IJCV 1999
  - Applications in super-resolution, motion, shading/paint discrimination
- H. Shum et al, ECCV 2002
  - Application to stereo
- M. Wainwright, T. Jaakkola, A. Willsky
  - Reparameterization version
- J. Yedidia, AAAI 2000
  - The clearest place to read about BP and GBP.

Vision applications of MRF's

- Stereo
- Motion estimation
- Labelling shading and reflectance
- Many others…

Vision applications of MRF’s

- Stereo
- Motion estimation
- Labelling shading and reflectance
- Segmentation
- Many more…



Interpreting images by propagating Bayesian beliefs

Yair Weiss
Dept. of Brain and Cognitive Sciences
Massachusetts Institute of Technology
E10-120, Cambridge, MA 02139, USA

In this paper we show that an architecture in which *Bayesian Beliefs* about image properties are propagated between neighboring units yields convergence times which are several orders of magnitude faster than traditional methods and avoids local minima. In particular our architecture is non-iterative in the sense of Marr [5]: at every time step, the local estimates at a given location are optimal given the information which has already been propagated to that location. We illustrate the algorithm's performance on real images and compare it to several existing methods.

$$J(Y) = \sum_k w_k (y_k - y_k^*)^2 + \lambda \sum_i (y_i - y_{i+1})^2$$

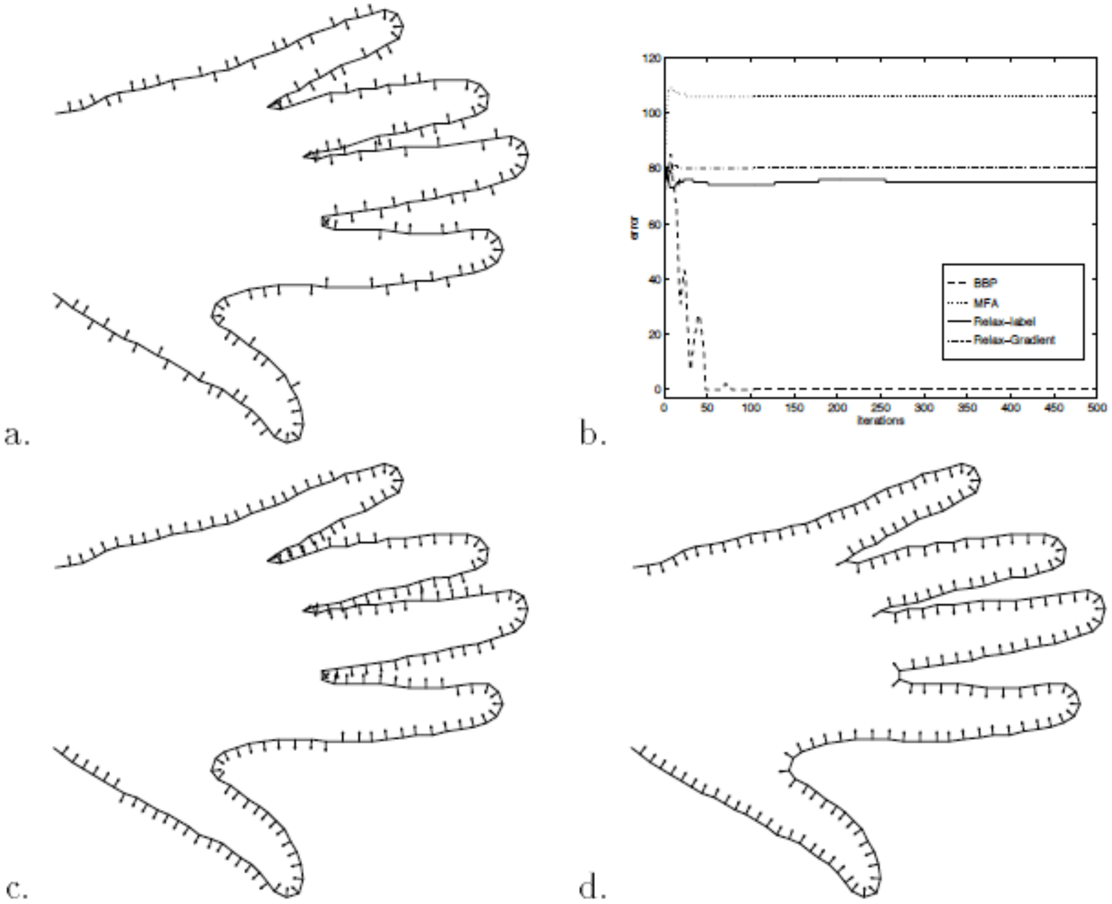


Figure 4: **a.** Local estimate of DOF along the contour. **b.** Performance of Hopfield,gradient descent, relaxation labeling and BBP as a function of time. BBP is the only method that converges to the global minimum. **c.** DOF estimate of Hopfield net after convergence. **d.** DOF estimate of BBP after convergence.

Vision applications of MRF's

- Stereo
- Motion estimation
- Labelling shading and reflectance
- Segmentation
- Many others…

Random Fields for segmentation

I = Image pixels (observed)
h = foreground/background labels (hidden) – one label per pixel
θ = Parameters

$$\underbrace{p(h \mid I, \theta)}_{\textbf{Posterior}}$$

1. **Generative approach models joint**
   → Markov random field (MRF)

2. **Discriminative approach models posterior directly**
   → Conditional random field (CRF)

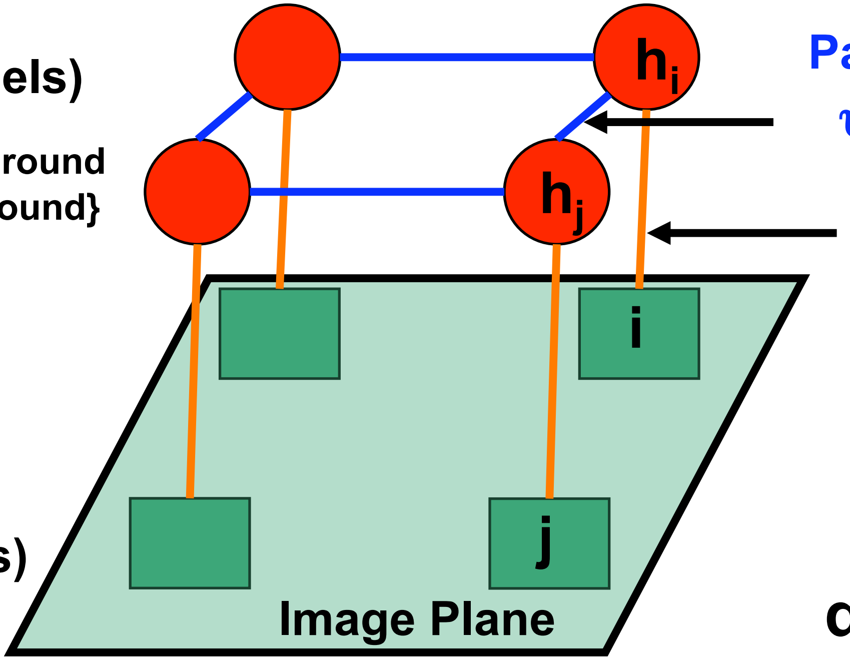
Generative Markov Random Field

$$p(h,I\,|\,\theta) = \boxed{p(I\,|\,h,\theta)}\boxed{p(h\,|\,\theta)}$$

$$= \frac{1}{Z(\theta)}\left[\prod_i \underbrace{\phi_i(I\,|\,h_i,\theta_i)}_{\text{Likelihood}}\prod_{ij}\underbrace{\psi_{ij}(h_i,h_j\,|\,\theta_{ij})}_{\text{MRF Prior}}\right]$$

h (labels)

∈{foreground ,background}



I (pixels)

Pairwise Potential (MRF)

$\psi_{ij}(h_i, h_j|\theta_{ij})$

Unary Potential

$\phi_i(I|h_i,\theta_i)$

Prior has no dependency on I

Conditional Random Field

Lafferty, McCallum and Pereira 2001

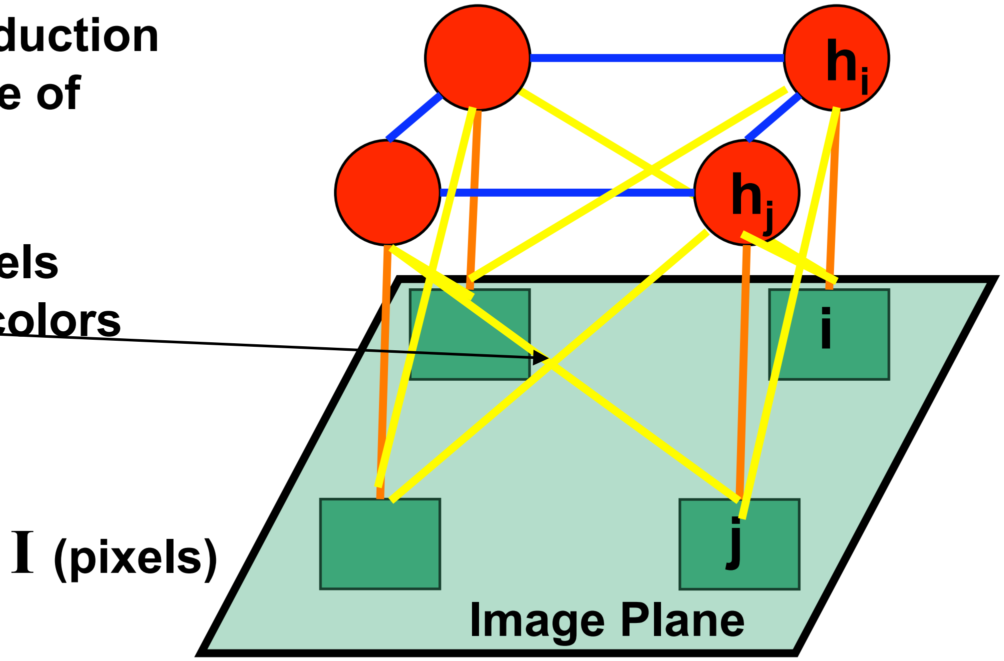
Discriminative approach

$$p(h \mid I,\theta) = \frac{1}{Z(I,\theta)} \left[\prod_i \underbrace{\phi_i(h_i, I \mid \theta_i)}_{\text{Unary}} \prod_{ij} \underbrace{\psi_{ij}(h_i, h_j, I \mid \theta_{ij})}_{\text{Pairwise}} \right]$$

- Dependency on I allows introduction of pairwise terms that make use of image.

- For example, neighboring labels should be similar only if pixel colors are similar → Contrast term

e.g Kumar and Hebert 2003



OBJCUT

Kumar, Torr & Zisserman 2005

$$p(h|\Omega,I,\theta) \propto \left[\prod_i \overbrace{\underbrace{\phi_i^1(I|h_i,\theta_i)}_{\text{Color Likelihood}} \cdot \underbrace{\phi_i^2(h_i|\Omega)}_{\text{Distance from }\Omega}}^{\text{Unary}} \prod_{ij} \overbrace{\underbrace{\psi_{ij}^1(h_i,h_j|\theta_{ij})}_{\text{Label smoothness}} \cdot \underbrace{\psi_{ij}^2(I|h_i,h_j,\theta_{ij})}_{\text{Contrast}}}^{\text{Pairwise}}\right]$$

- **Ω is a shape prior on the labels from a Layered Pictorial Structure (LPS) model**

- **Segmentation by:**

  - Match LPS model to image (get number of samples, each with a different pose

  -Marginalize over the samples using a single graph cut [Boykov & Jolly, 2001]

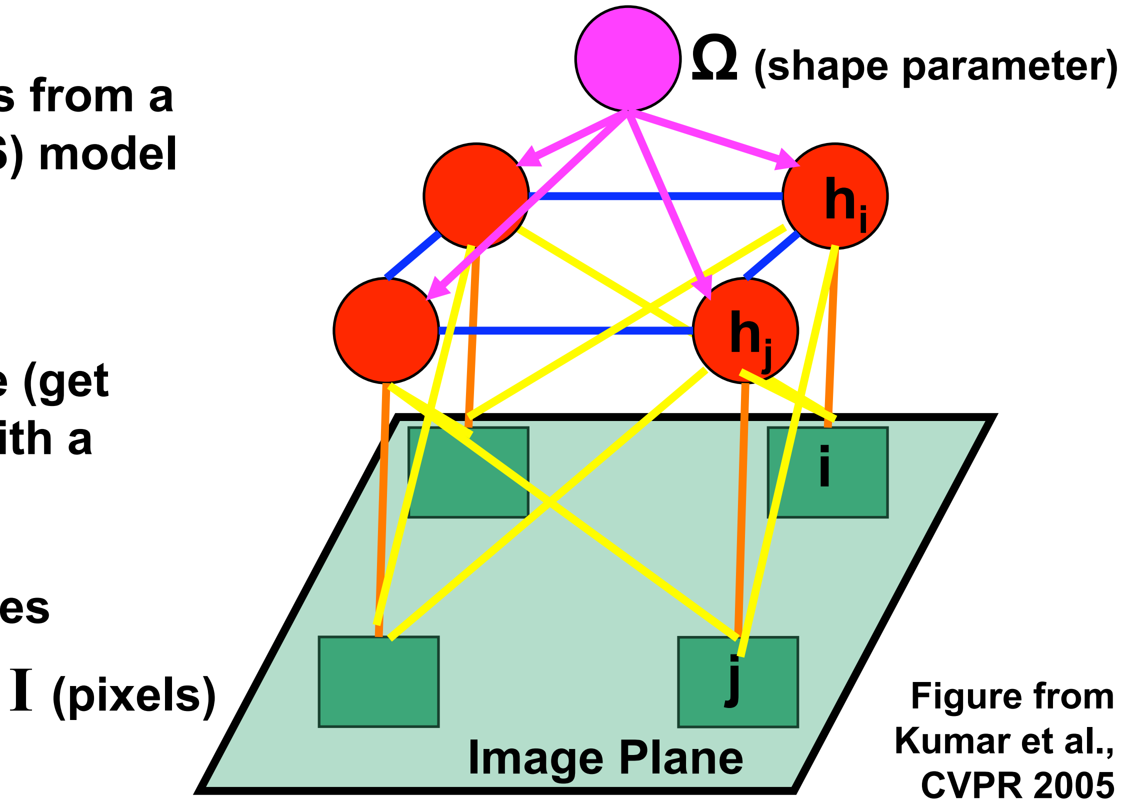
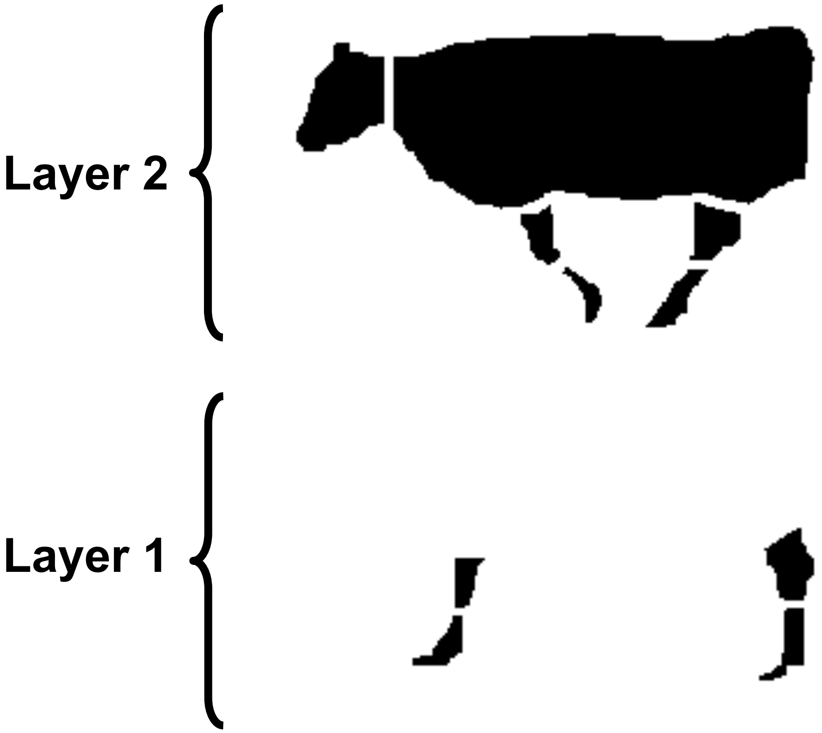
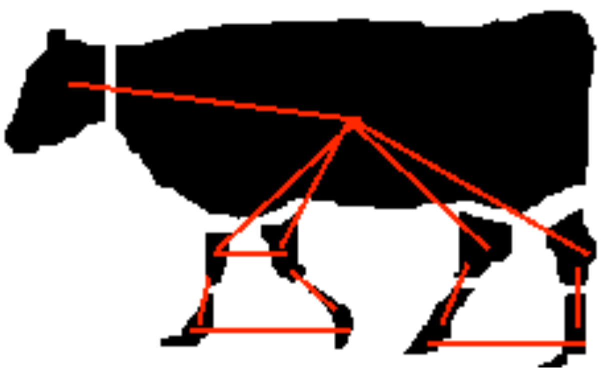


Figure from Kumar et al., CVPR 2005

OBJCUT:
Shape prior - **Ω** - Layered Pictorial Structures (LPS)

- Generative model
- Composition of parts + spatial layout





Spatial Layout
(Pairwise Configuration)

Parts in Layer 2 can occlude parts in Layer 1

Kumar, et al. 2004, 2005

OBJCUT: Results

Using LPS Model for Cow

In the absence of a clear boundary between object and background

Image

Segmentation