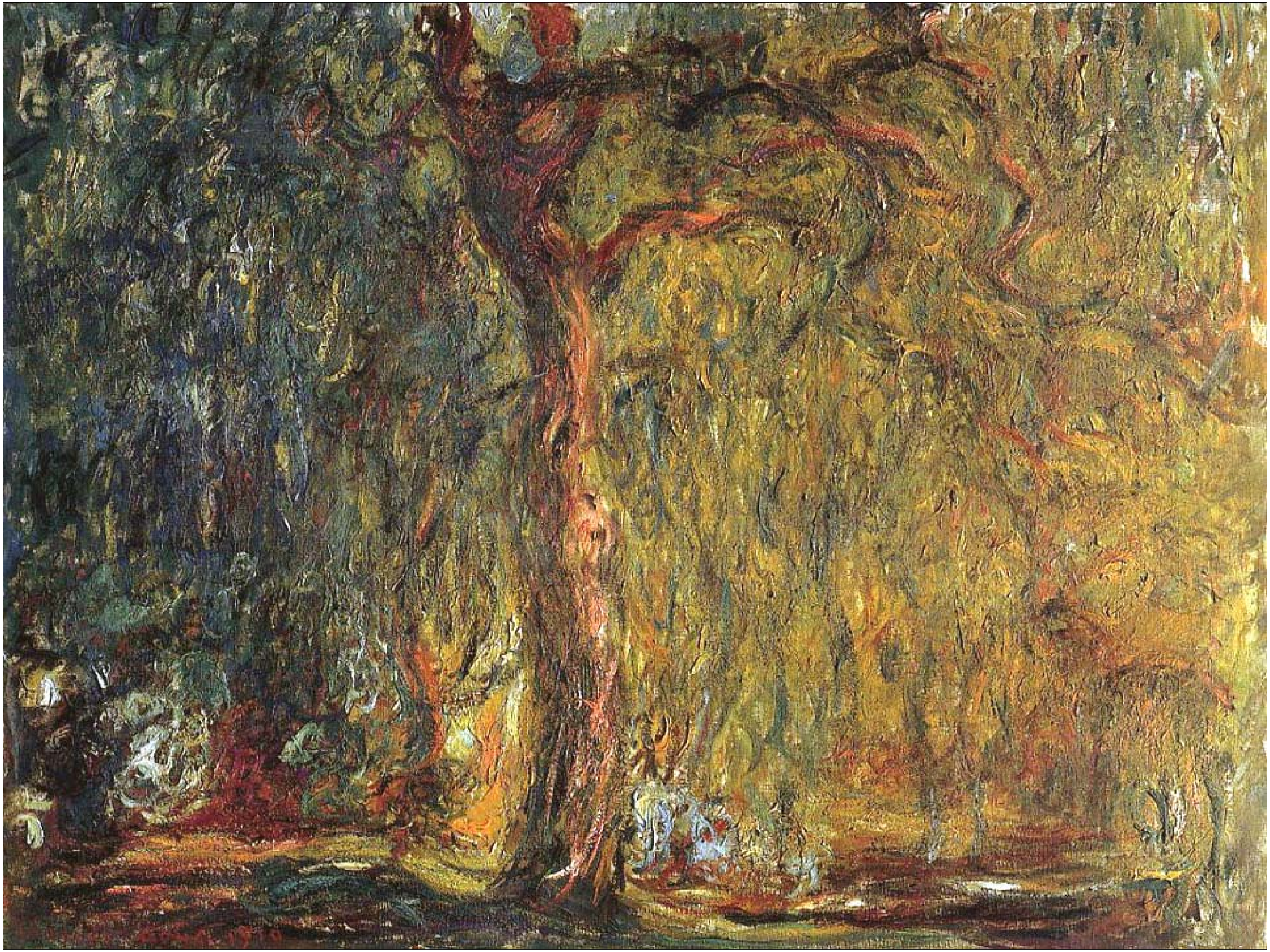
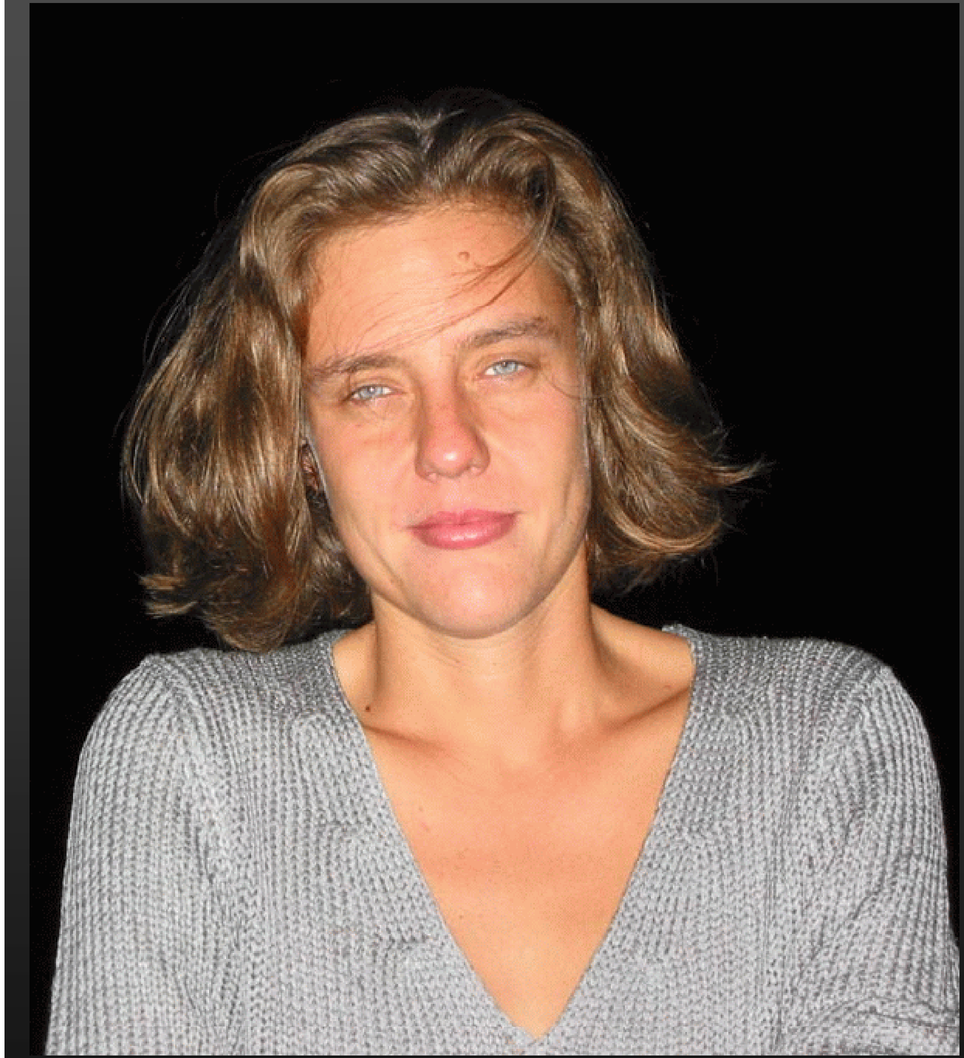


Reconnaissance d'objets et vision artificielle

Jean Ponce (ponce@di.ens.fr)
http://www.di.ens.fr/~ponce
Equipe-projet WILLOW
ENS/INRIA/CNRS UMR 8548
Laboratoire d'Informatique
Ecole Normale Supérieure, Paris

Cordelia Schmid



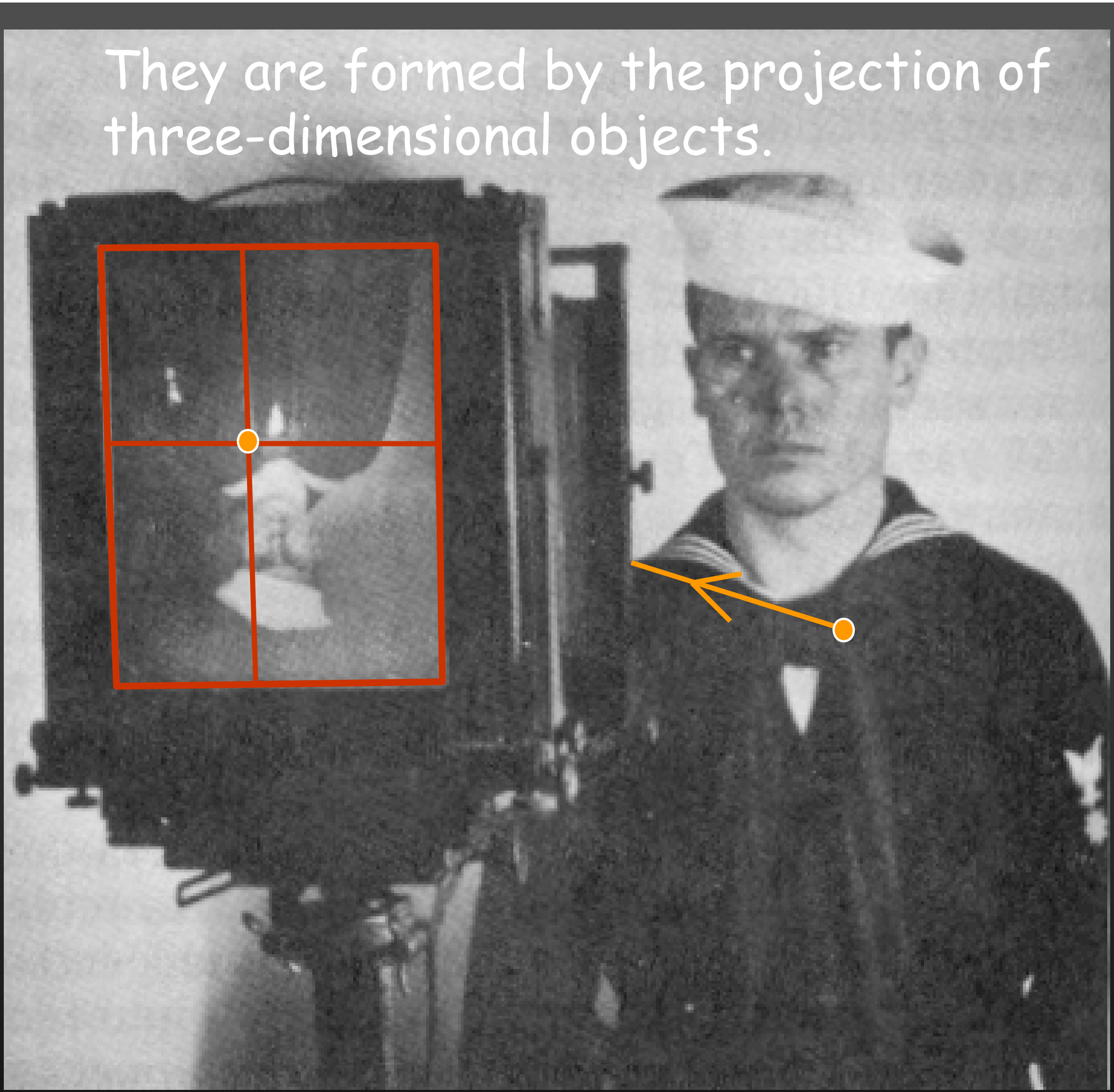
http://lear.inrialpes.fr/~schmid/

Josef Sivic

http://www.di.ens.fr/~josef/

Outline

- What computer vision is about
- What this class is about
- A brief history of visual recognition
- Alignment methods



They are formed by the projection of three-dimensional objects.

Images are brightness/color patterns drawn in a plane.

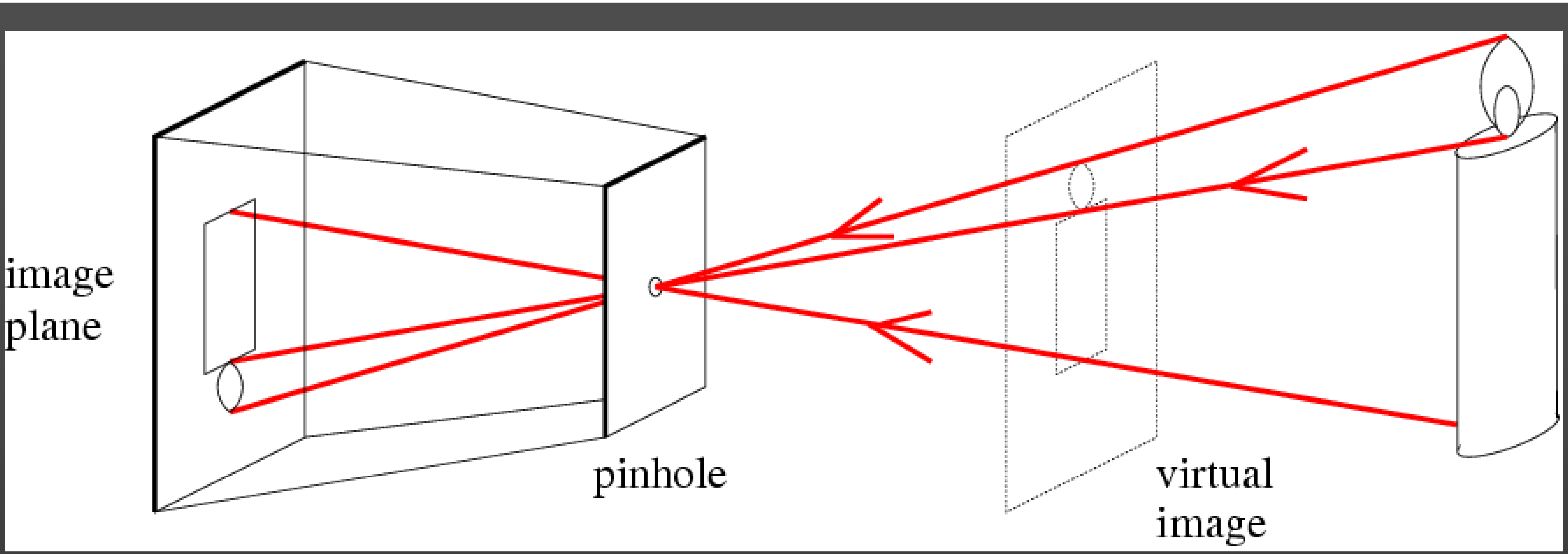
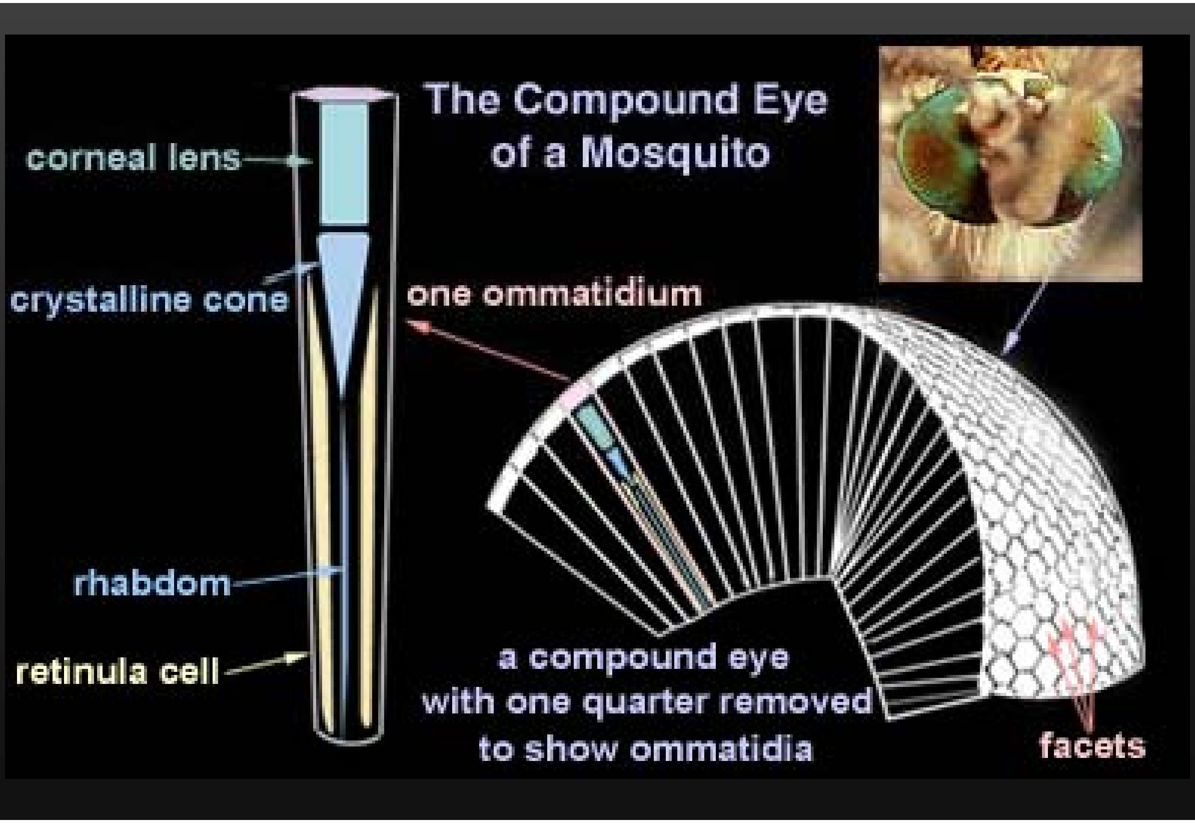
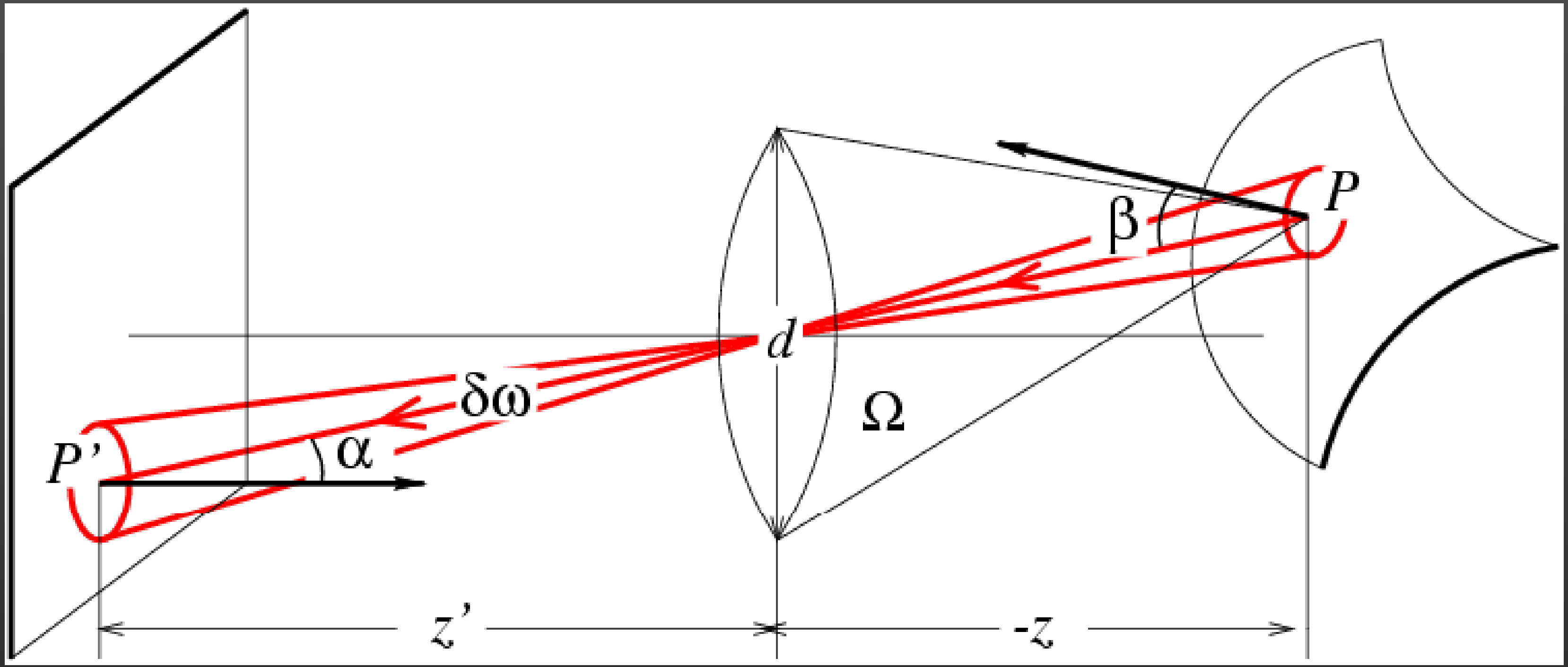


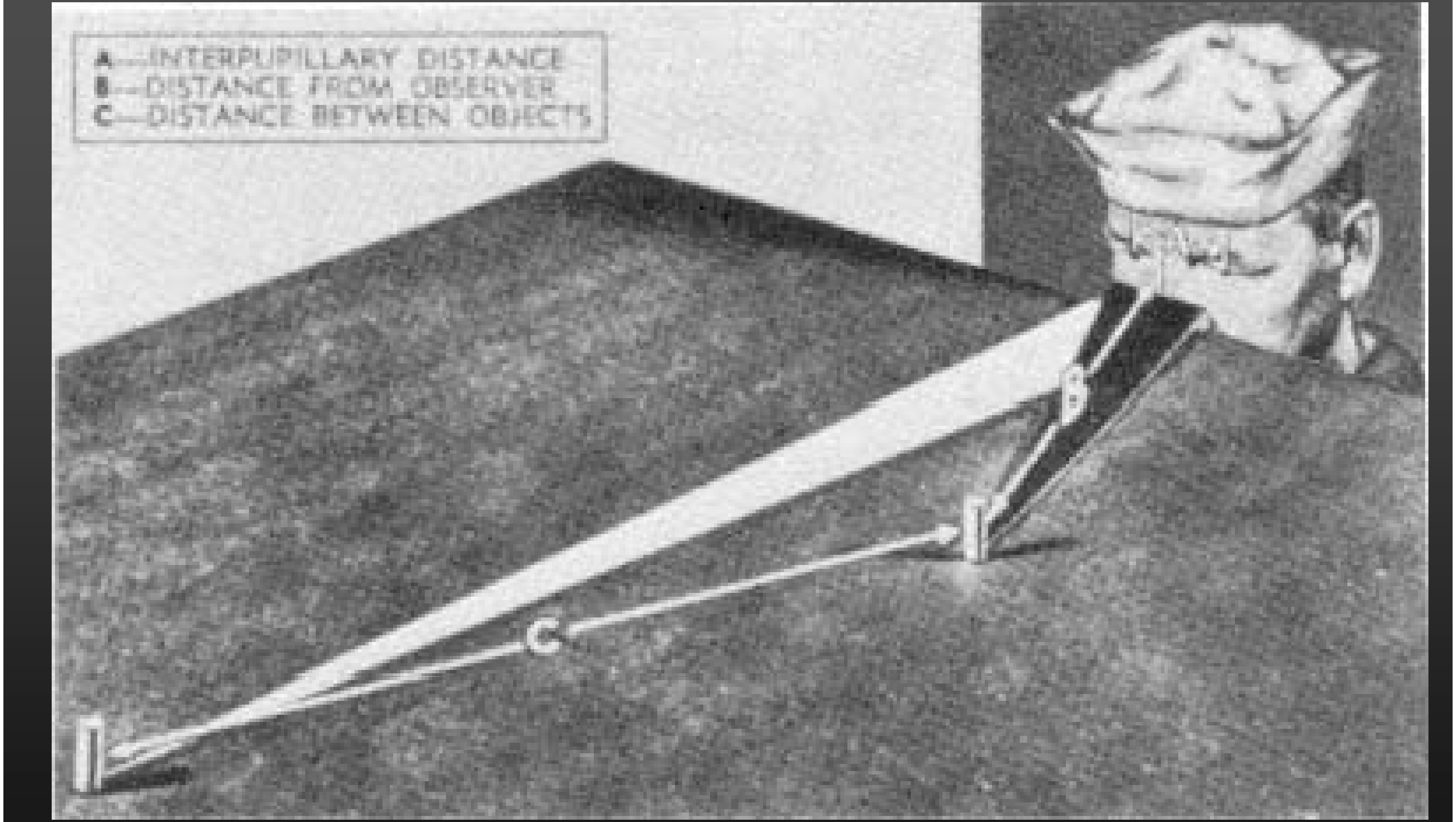
image plane
pinhole
virtual image


The Compound Eye of a Mosquito
corneal lens
crystalline cone
rhabdom
retinula cell
one ommatidium
a compound eye with one quarter removed to show ommatidia
facets



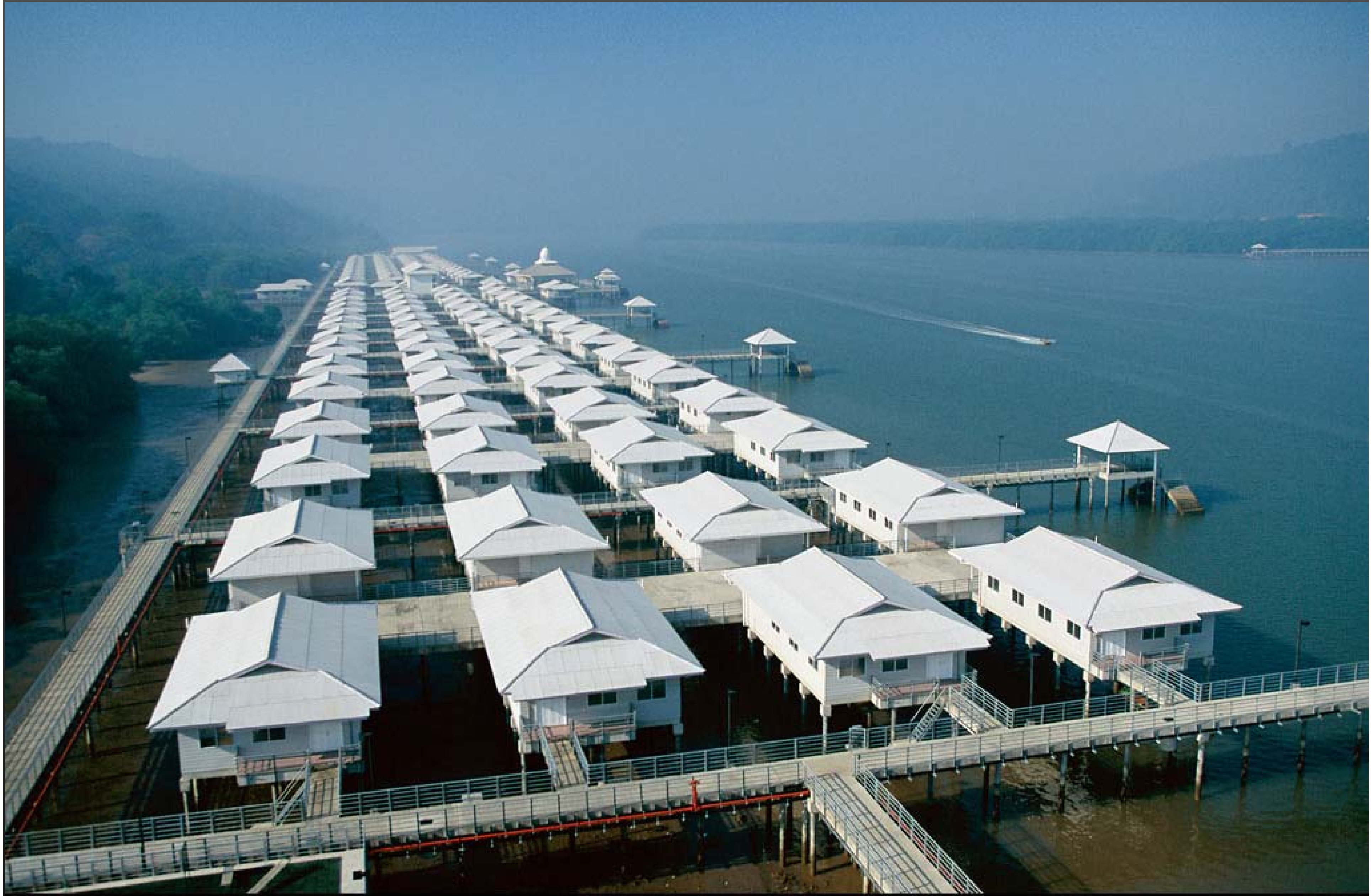
$$E=(\Pi/4)\left[\,(d/z')^2\cos^4\alpha\,\right]L$$

Question : how do we see "in 3D" ?

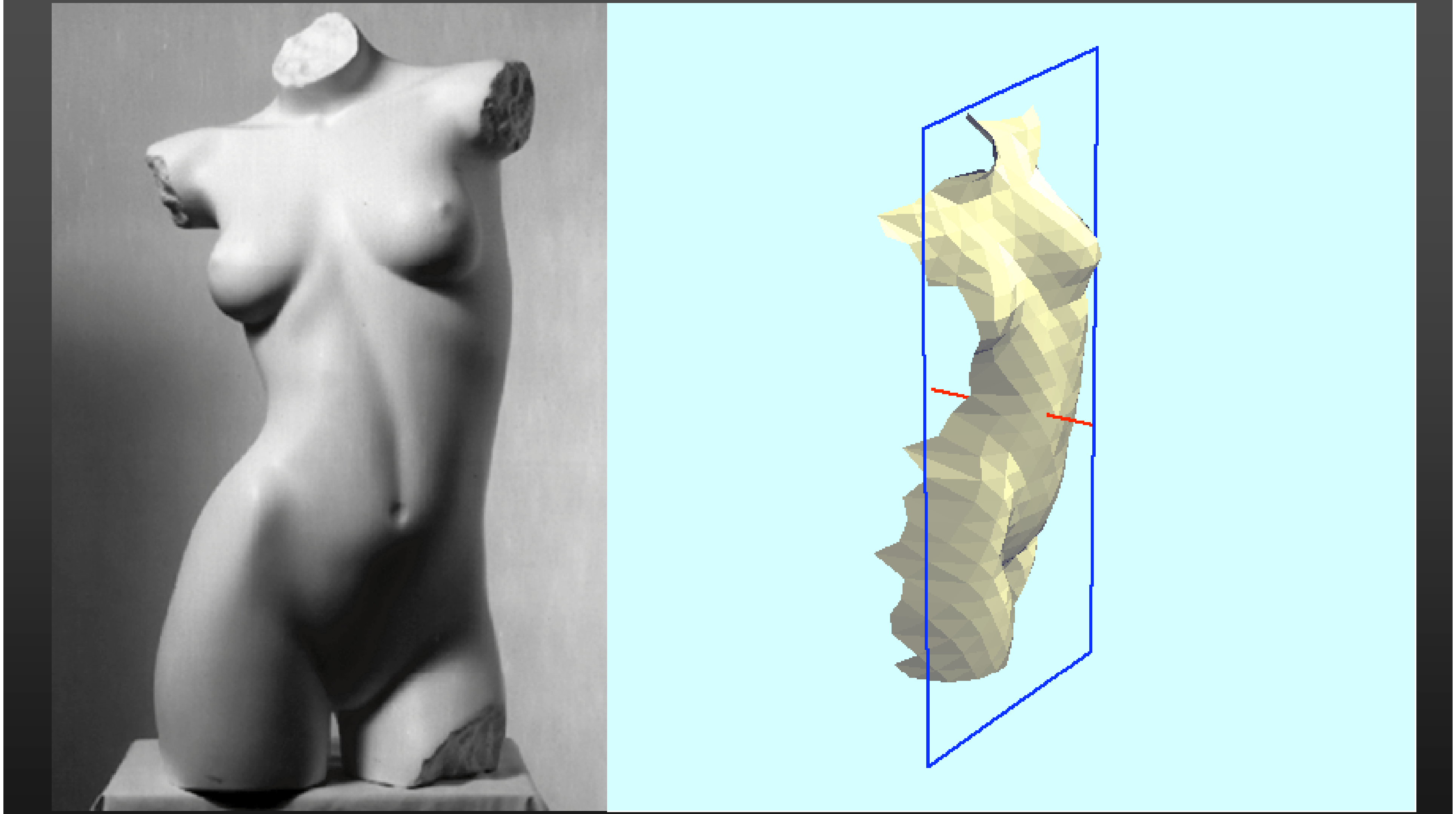


(First-order) answer: with our two eyes.

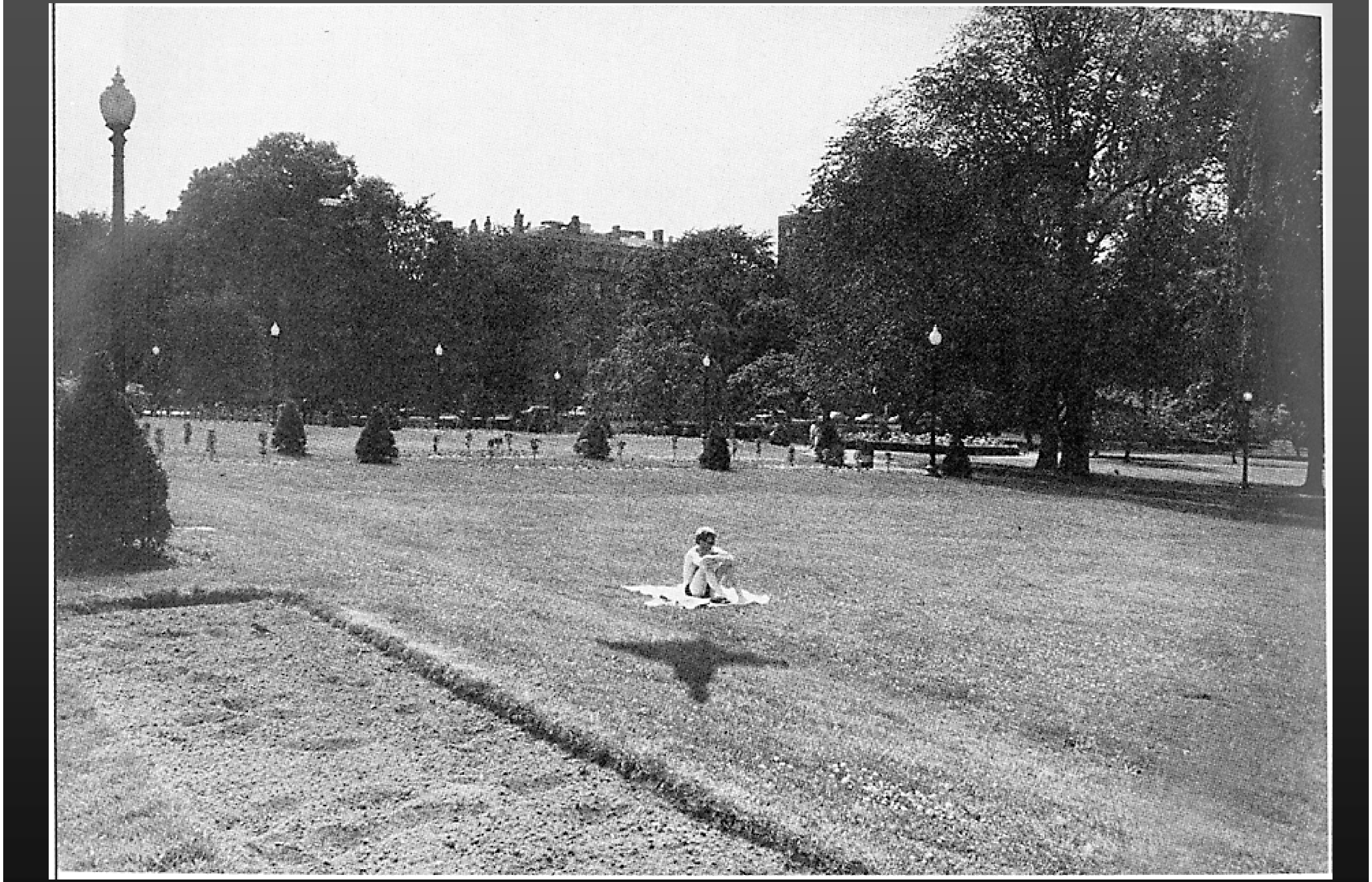
But there are other cues..



NATIONALGEOGRAPHIC.COM

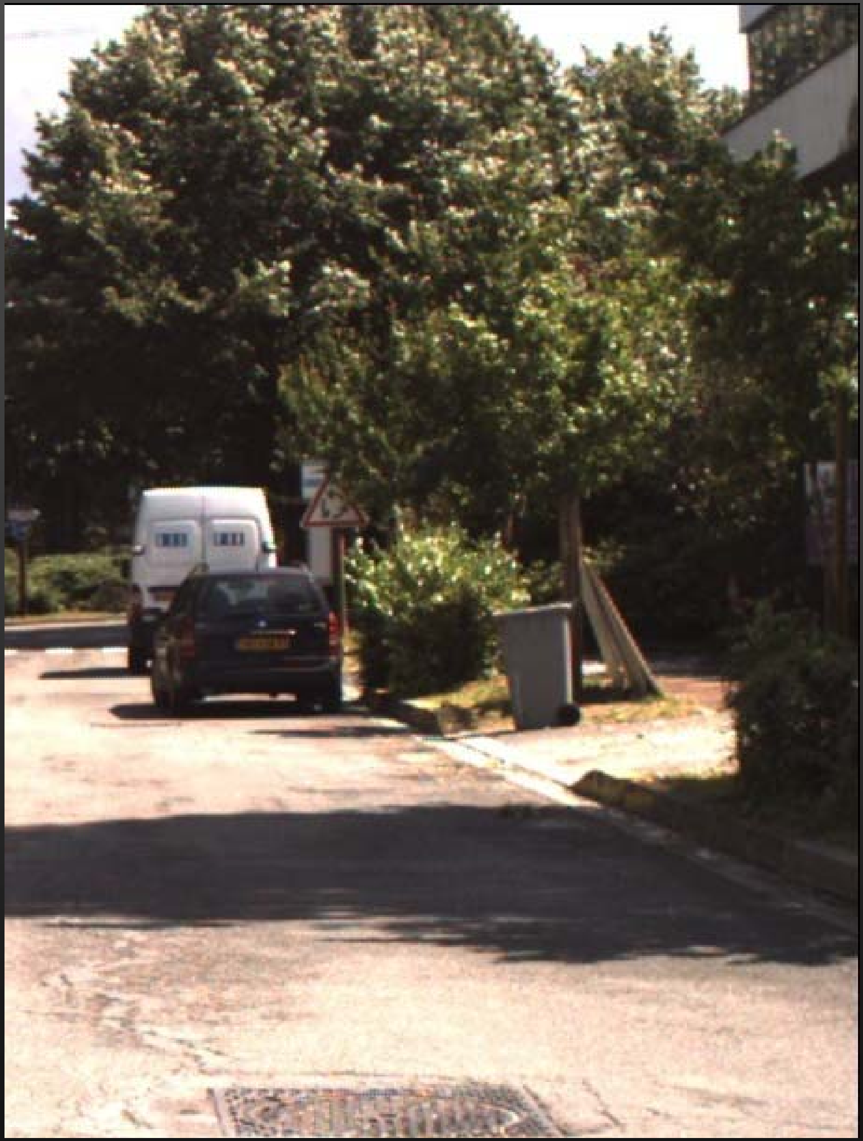


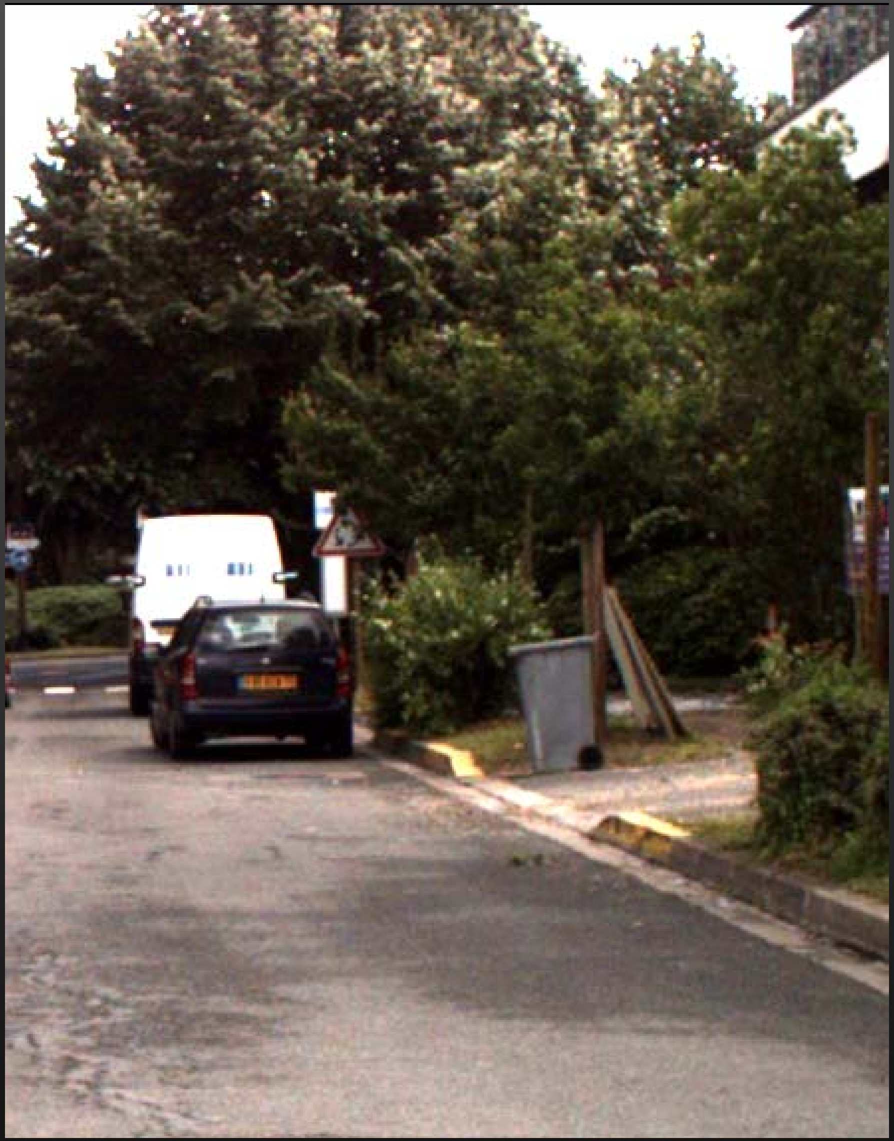
Source: J. Koenderink



Source: J. Koenderink

What is happening with the shadows?





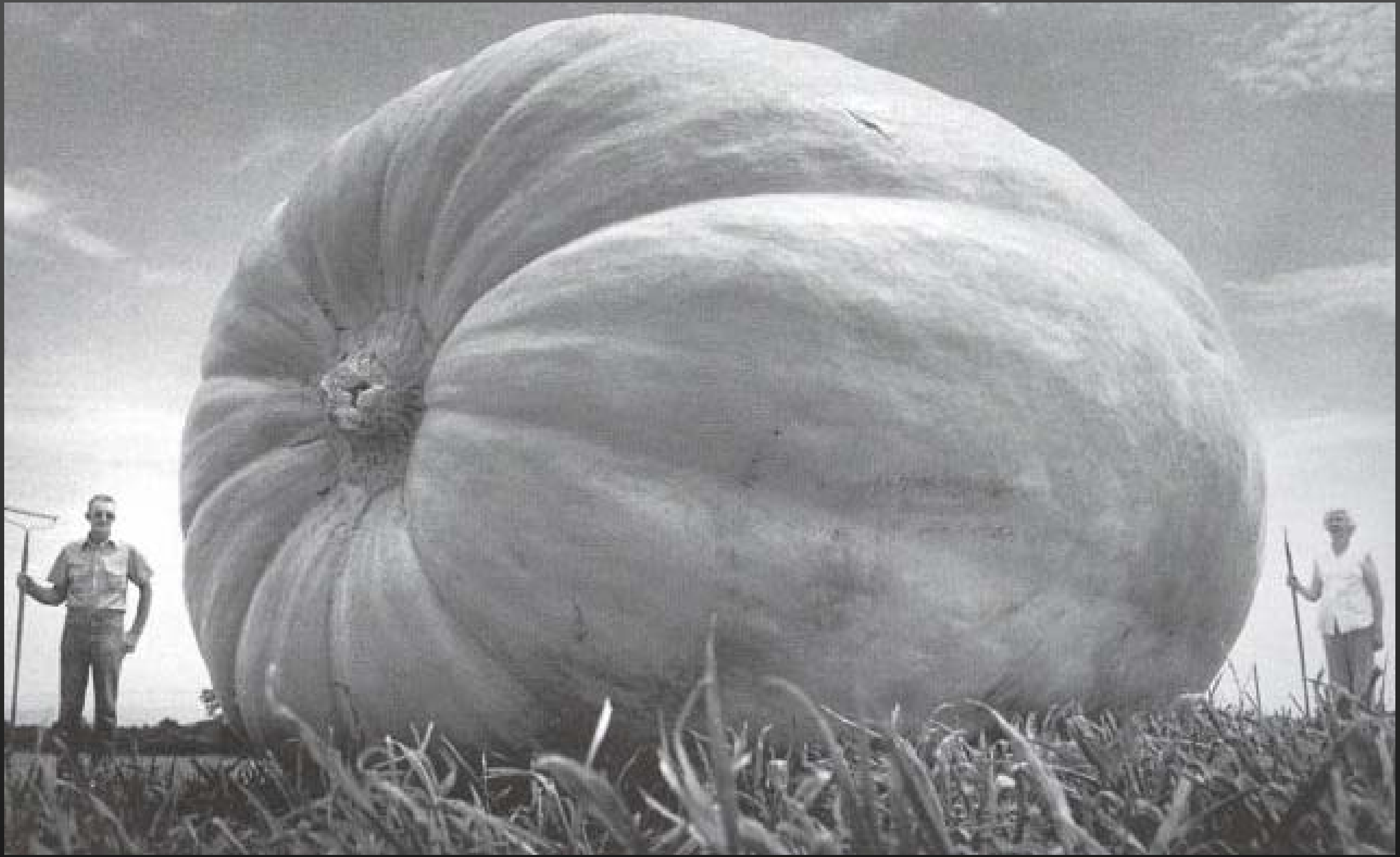


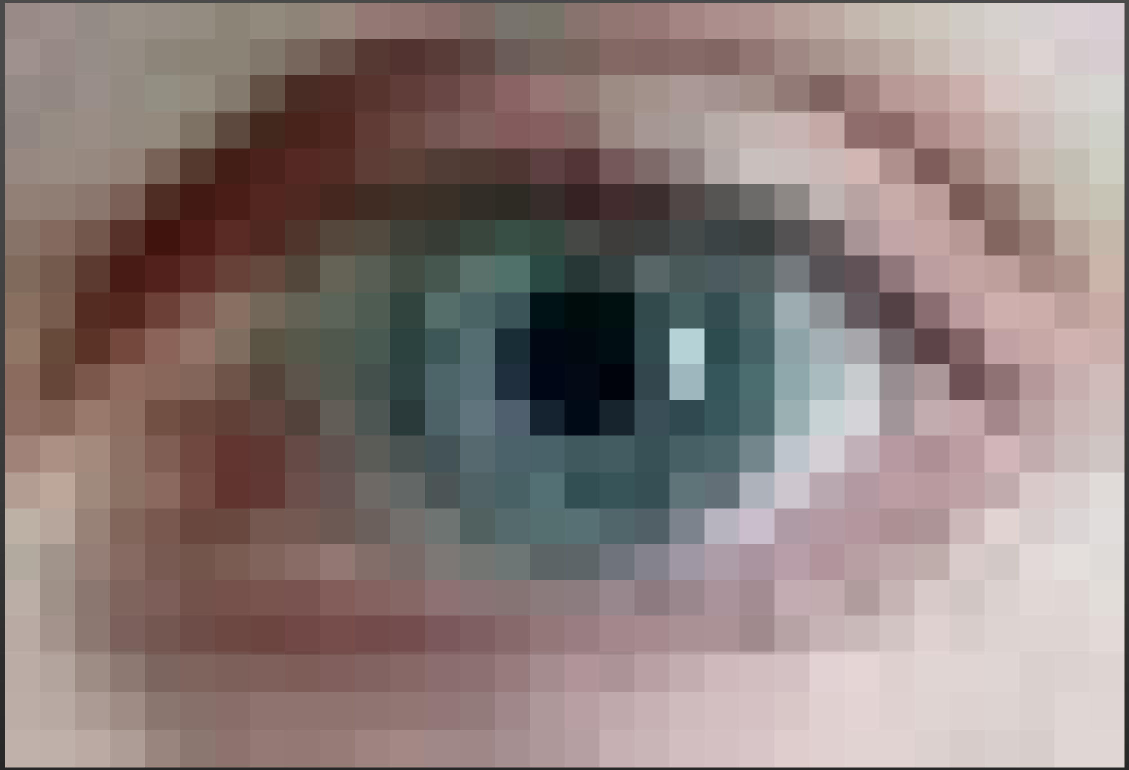
Image source: F. Durand

Challenges or opportunities?

Image source: J. Koenderink

- Images are confusing, but they also reveal the structure of the world through numerous cues.
- Our job is to interpret the cues!

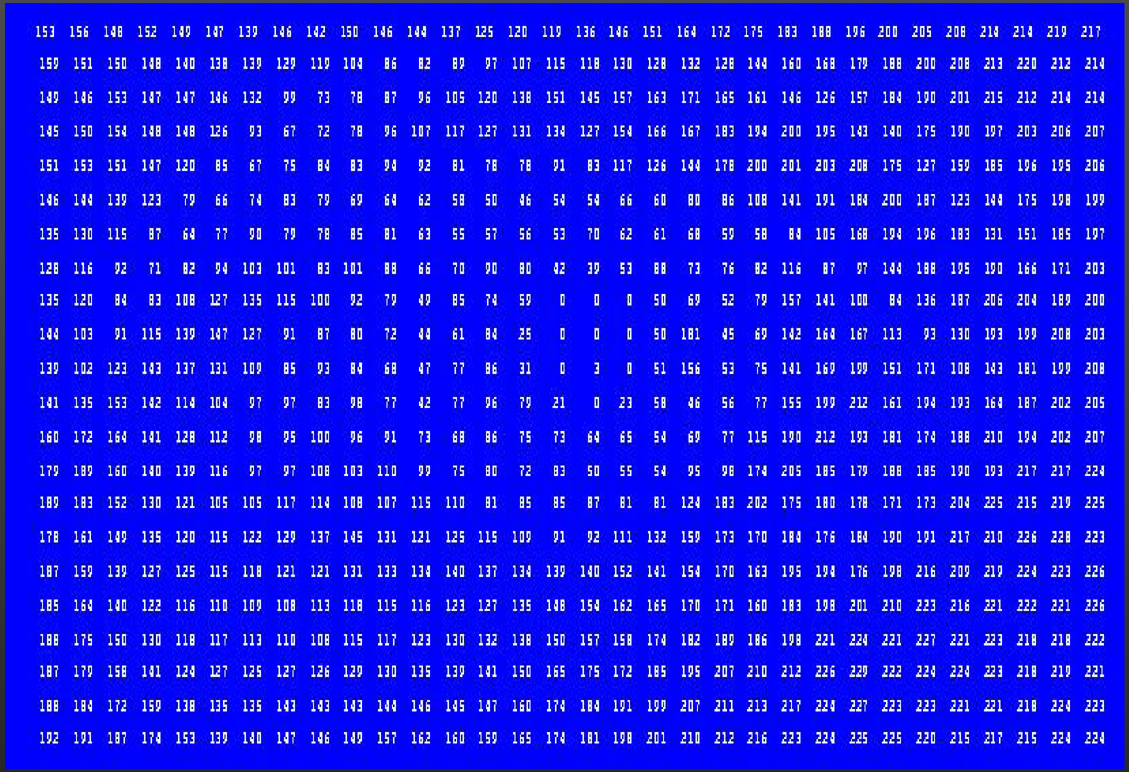
The goal of computer vision



To perceive the "world behind the picture", e.g.,

- as a metric measurement device
- as a device for measuring "semantic" information

The goal of computer vision

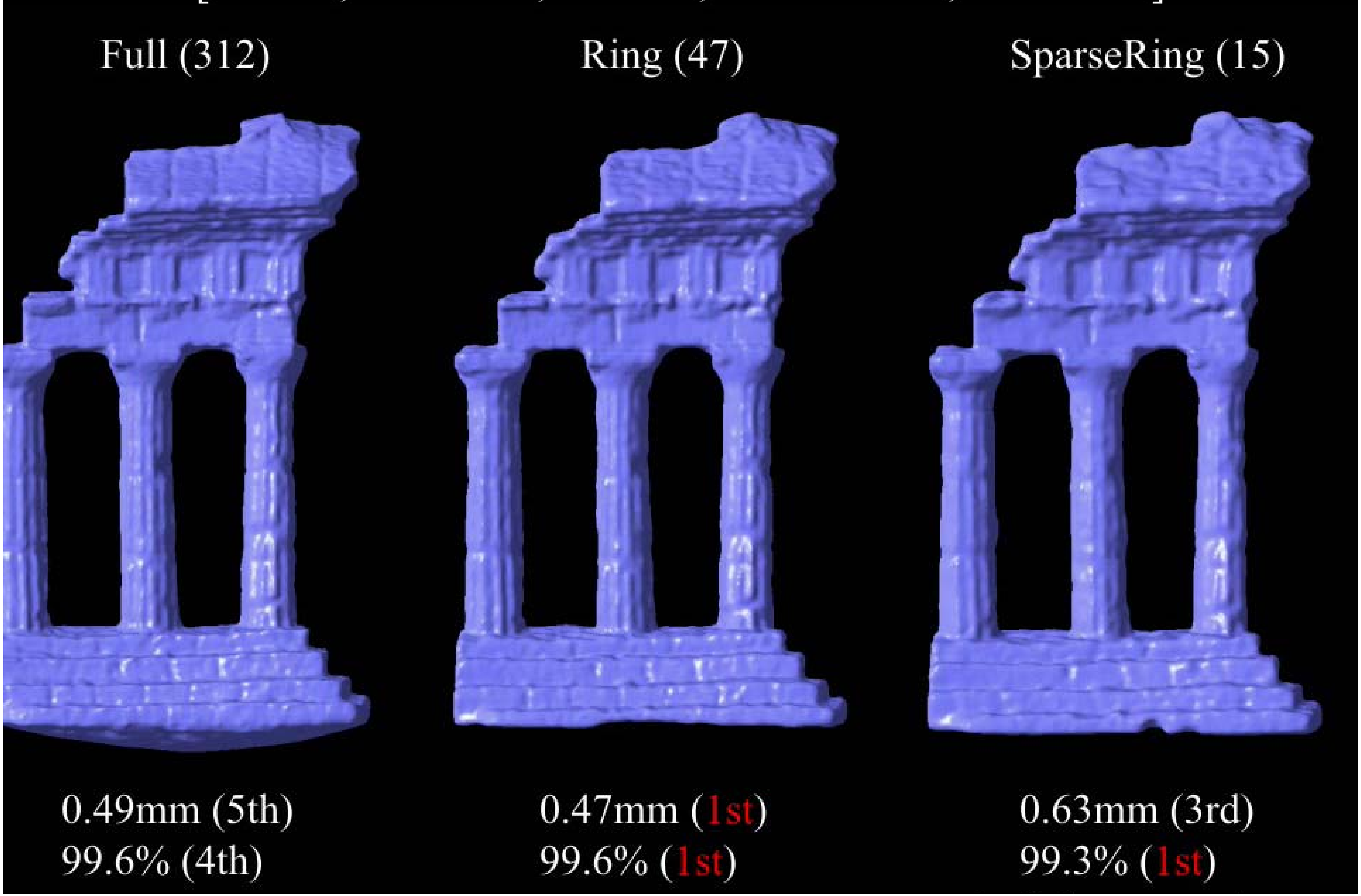


153	156	148	152	149	147	139	146	142	150	146	144	137	125	120	119	136	146	151	164	172	175	183	188	196	200	205	208	214	214	219	217
159	151	150	148	140	138	139	129	119	104	86	82	89	97	107	115	118	130	128	132	128	144	160	168	179	188	200	208	213	220	212	214
149	146	153	147	147	146	132	99	73	78	87	96	105	120	138	151	145	157	163	171	165	161	146	126	157	184	190	201	215	212	214	214
145	150	154	148	148	126	93	67	72	78	96	107	117	127	131	134	127	154	166	167	183	194	200	195	143	140	175	190	197	203	206	207
151	153	151	147	120	85	67	75	84	83	94	92	81	78	78	91	83	117	126	144	178	200	201	203	208	175	127	159	185	196	195	206
146	144	139	123	79	66	74	83	79	69	64	62	58	50	46	54	54	66	60	80	86	108	141	191	184	200	187	123	144	175	198	199
135	130	115	87	64	77	90	79	78	85	81	63	55	57	56	53	70	62	61	68	59	58	84	105	168	194	196	183	131	151	185	197
128	116	92	71	82	94	103	101	83	101	88	66	70	90	80	42	39	53	88	73	76	82	116	87	97	144	188	195	190	166	171	203
135	120	84	83	108	127	135	115	100	92	79	49	85	74	59	0	0	0	50	69	52	79	157	141	100	84	136	187	206	204	189	200
144	103	91	115	139	147	127	91	87	80	72	44	61	84	25	0	0	0	50	181	45	69	142	164	167	113	93	130	193	199	208	203
139	102	123	143	137	131	109	85	93	84	68	47	77	86	31	0	3	0	51	156	53	75	141	169	199	151	171	108	143	181	199	208
141	135	153	142	114	104	97	97	83	98	77	42	77	96	79	21	0	23	58	46	56	77	155	199	212	161	194	193	164	187	202	205
160	172	164	141	128	112	98	95	100	96	91	73	68	86	75	73	64	65	54	69	77	115	190	212	193	181	174	188	210	194	202	207
179	189	160	140	139	116	97	97	108	103	110	99	75	80	72	83	50	55	54	95	98	174	205	185	179	188	185	190	193	217	217	224
189	183	152	130	121	105	105	117	114	108	107	115	110	81	85	85	87	81	81	124	183	202	175	180	178	171	173	204	225	215	219	225
178	161	149	135	120	115	122	129	137	145	131	121	125	115	109	91	92	111	132	159	173	170	184	176	184	190	191	217	210	226	228	223
187	159	139	127	125	115	118	121	121	131	133	134	140	137	134	139	140	152	141	154	170	163	195	194	176	198	216	209	219	224	223	226
185	164	140	122	116	110	109	108	113	118	115	116	123	127	135	148	154	162	165	170	171	160	183	198	201	210	223	216	221	222	221	226
188	175	150	130	118	117	113	110	108	115	117	123	130	132	138	150	157	158	174	182	189	186	198	221	224	221	227	221	223	218	218	222
187	179	158	141	124	127	125	127	126	129	130	135	139	141	150	165	175	172	185	195	207	210	212	226	229	222	224	224	223	218	219	221
188	184	172	159	138	135	135	143	143	143	144	146	145	147	160	174	184	191	199	207	211	213	217	224	227	223	223	221	221	218	224	223
192	191	187	174	153	139	140	147	146	149	157	162	160	159	165	174	181	198	201	210	212	216	223	224	225	225	220	215	217	215	224	224

To perceive the “world behind the picture”, e.g.,

- as a metric measurement device
- as a device for “measuring” semantic information

Vision as metric measurement device: Furukawa & Ponce (CVPR'07)
(cf also Keriven's class "Vision et reconstruction 3D)

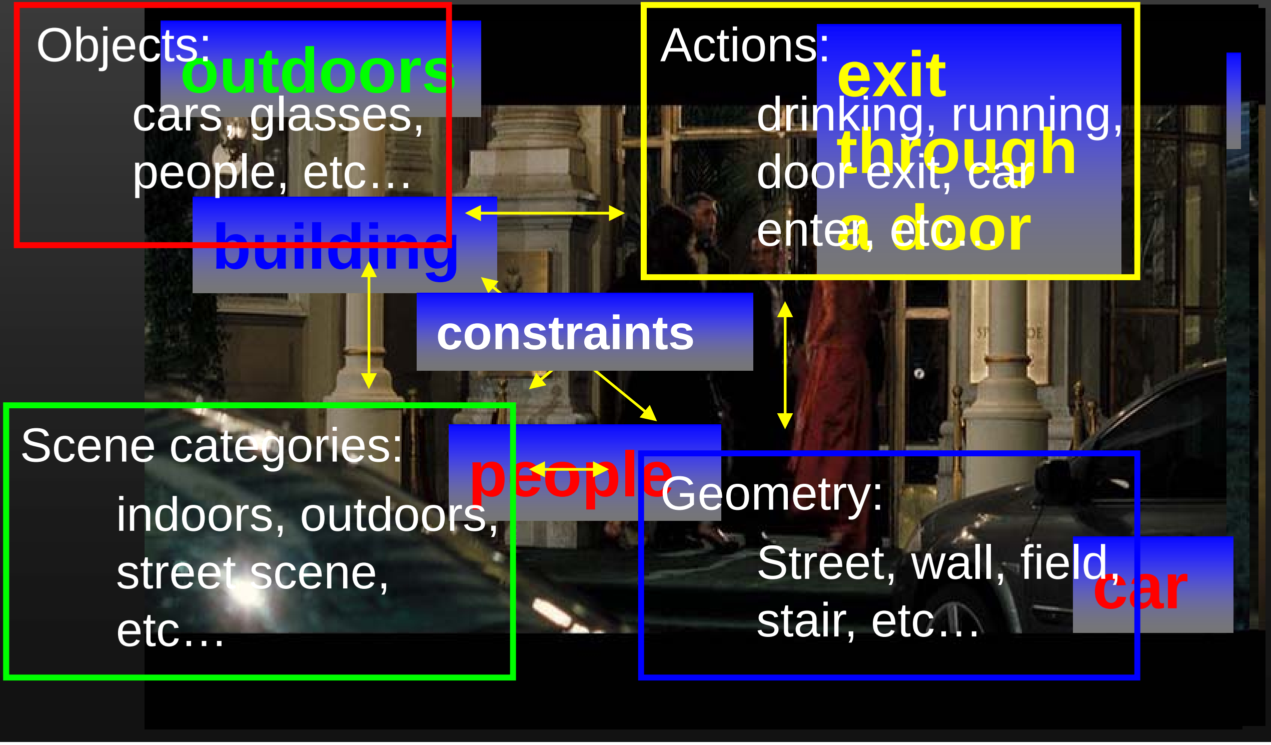


Visual scene analysis

(Courtesy Ivan Laptev, VISTA)

Visual scene analysis

(Courtesy Ivan Laptev, VISTA)



Objects:

cars, glasses, people, etc…

Actions:

drinking, running, door exit, car enter, etc…

Scene categories:

indoors, outdoors, street scene, etc…

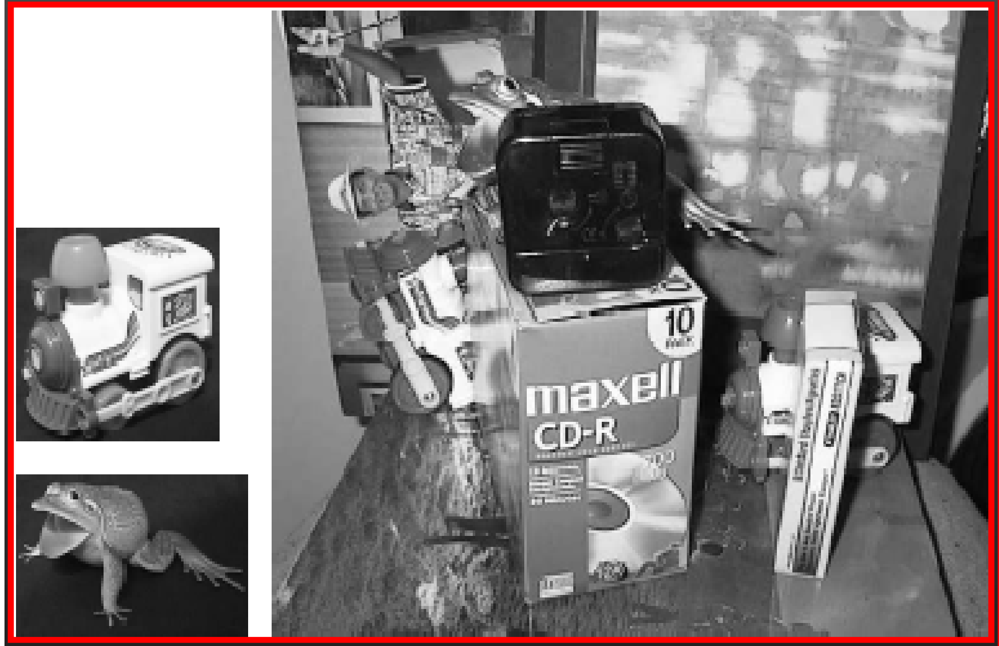
Geometry:

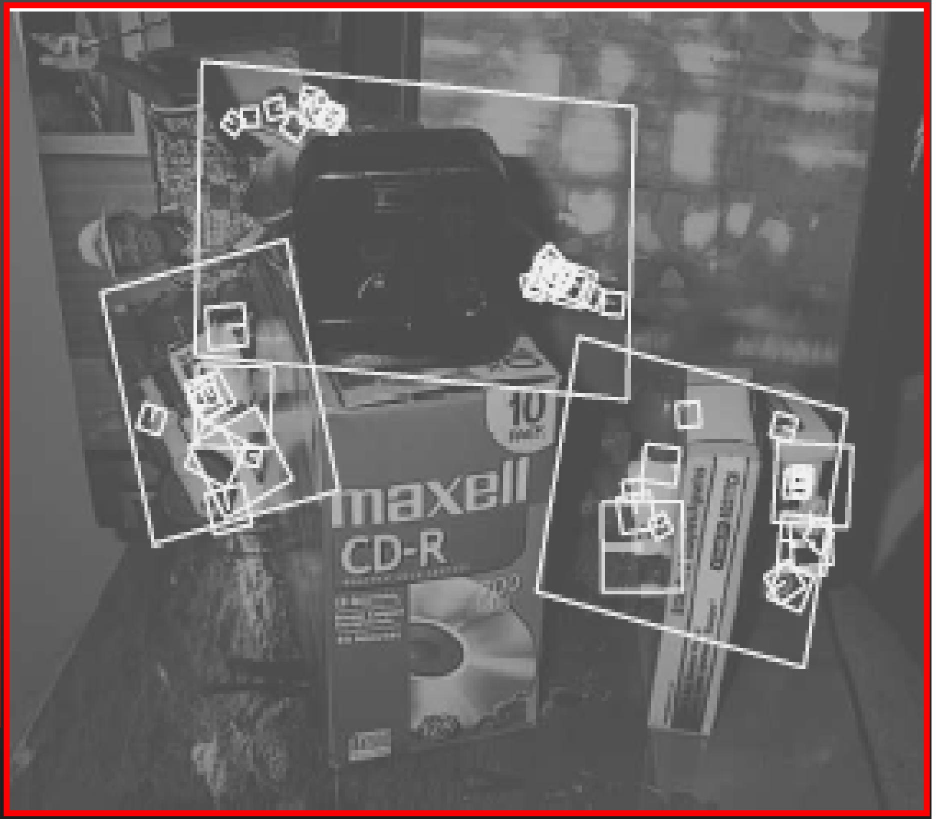
Street, wall, field, stair, etc…

Outline

- What computer vision is about
- What this class is about
- A brief history of visual recognition
- Alignment methods

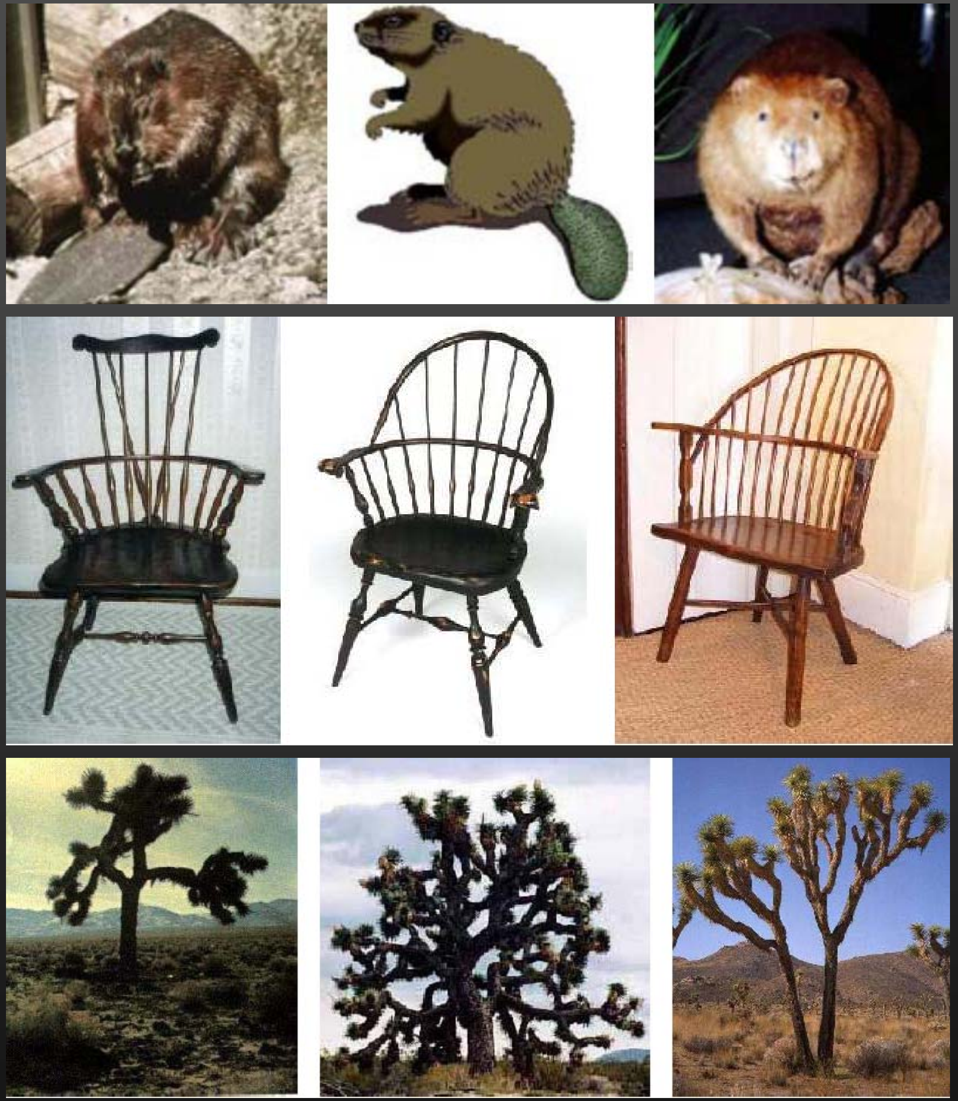
Specific object detection





(Lowe, 2004)

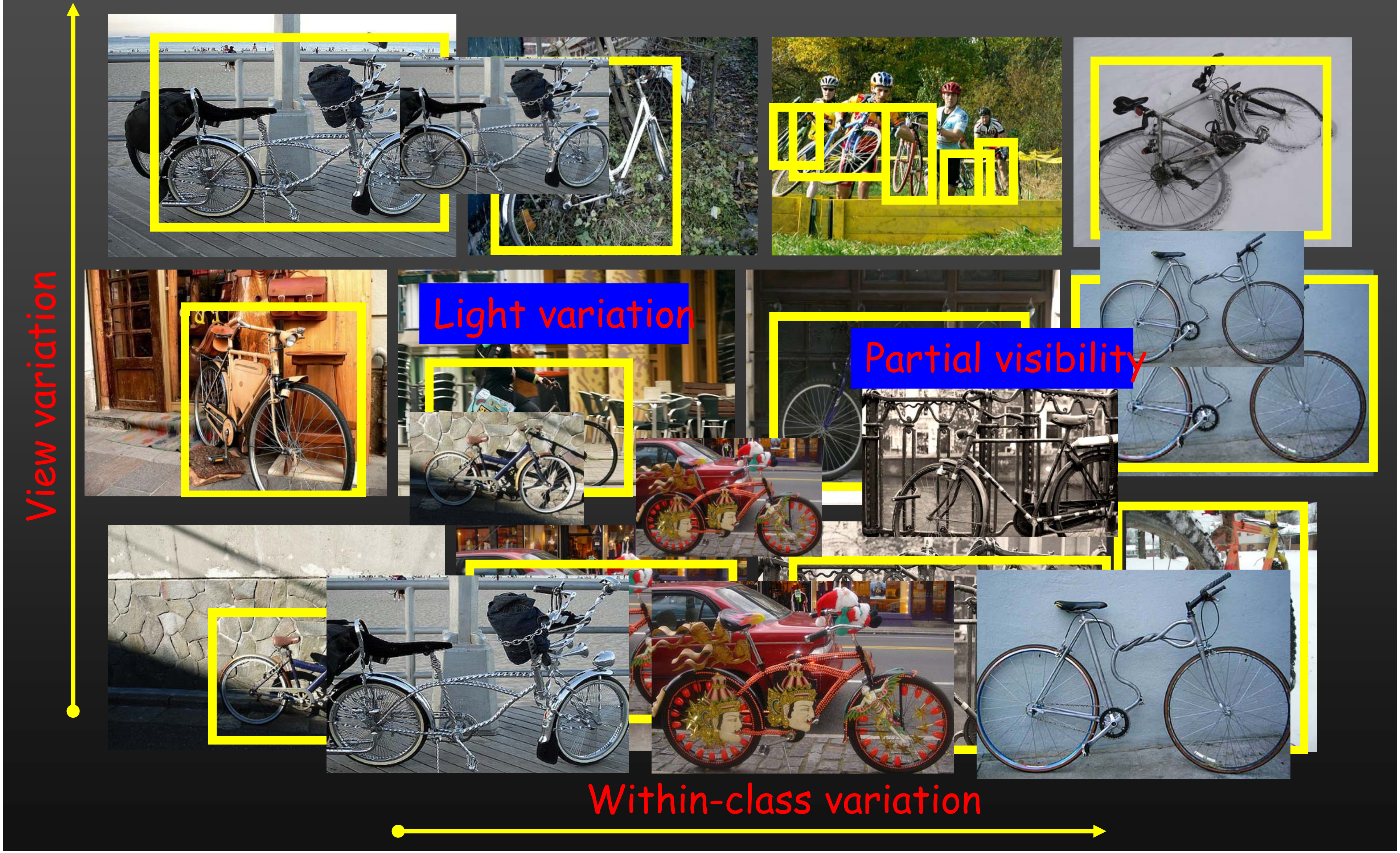
Image classification



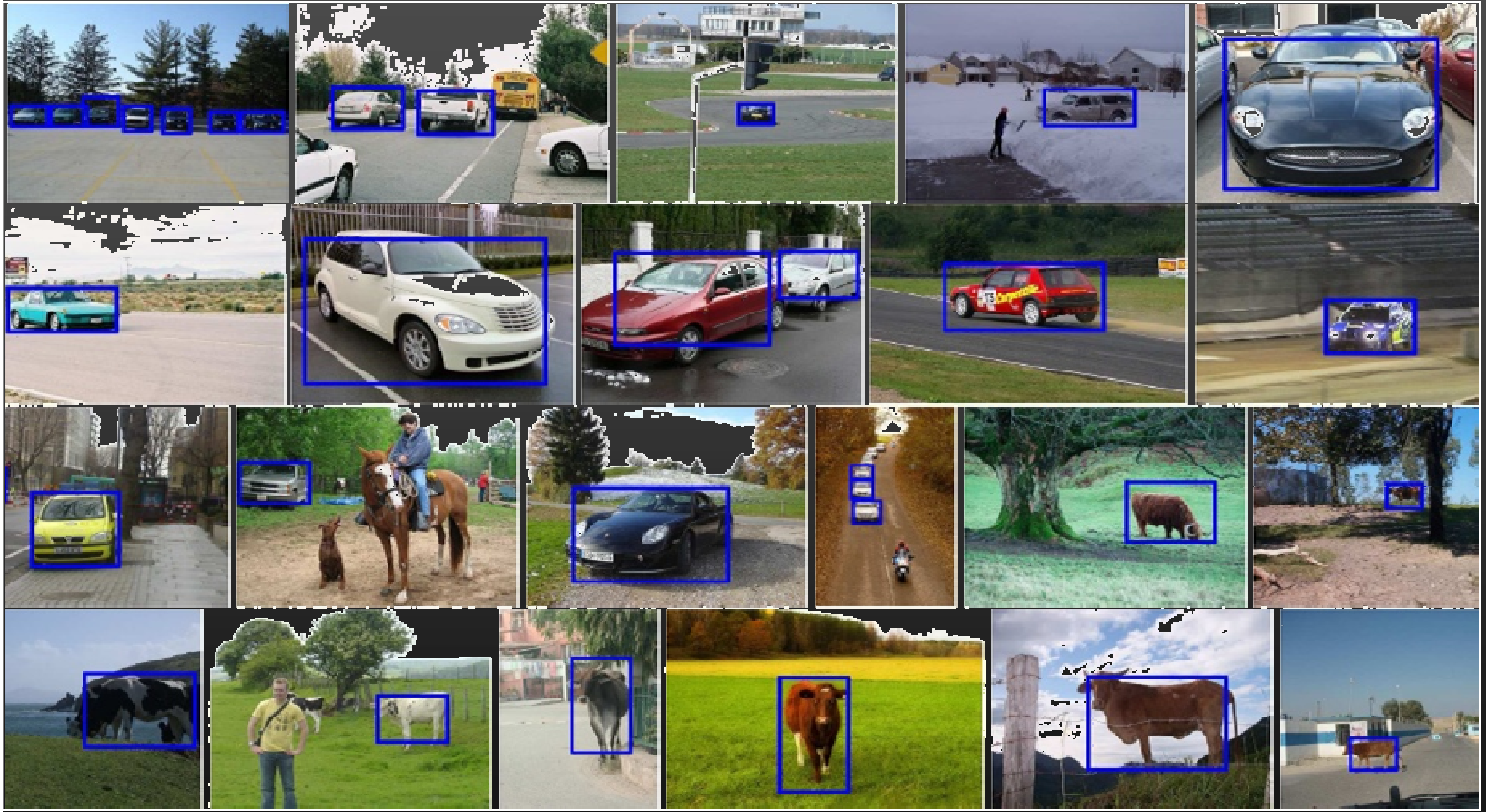
Caltech 101 : http://www.vision.caltech.edu/Image_Datasets/Caltech101/

Object category detection

(Courtesy Ivan Laptev)



Model ≡ locally rigid assembly of parts
Part ≡ locally rigid assembly of features



Qualitative experiments on Pascal VOC'07 (Kushal, Schmid, Ponce, 2008)

Scene understanding

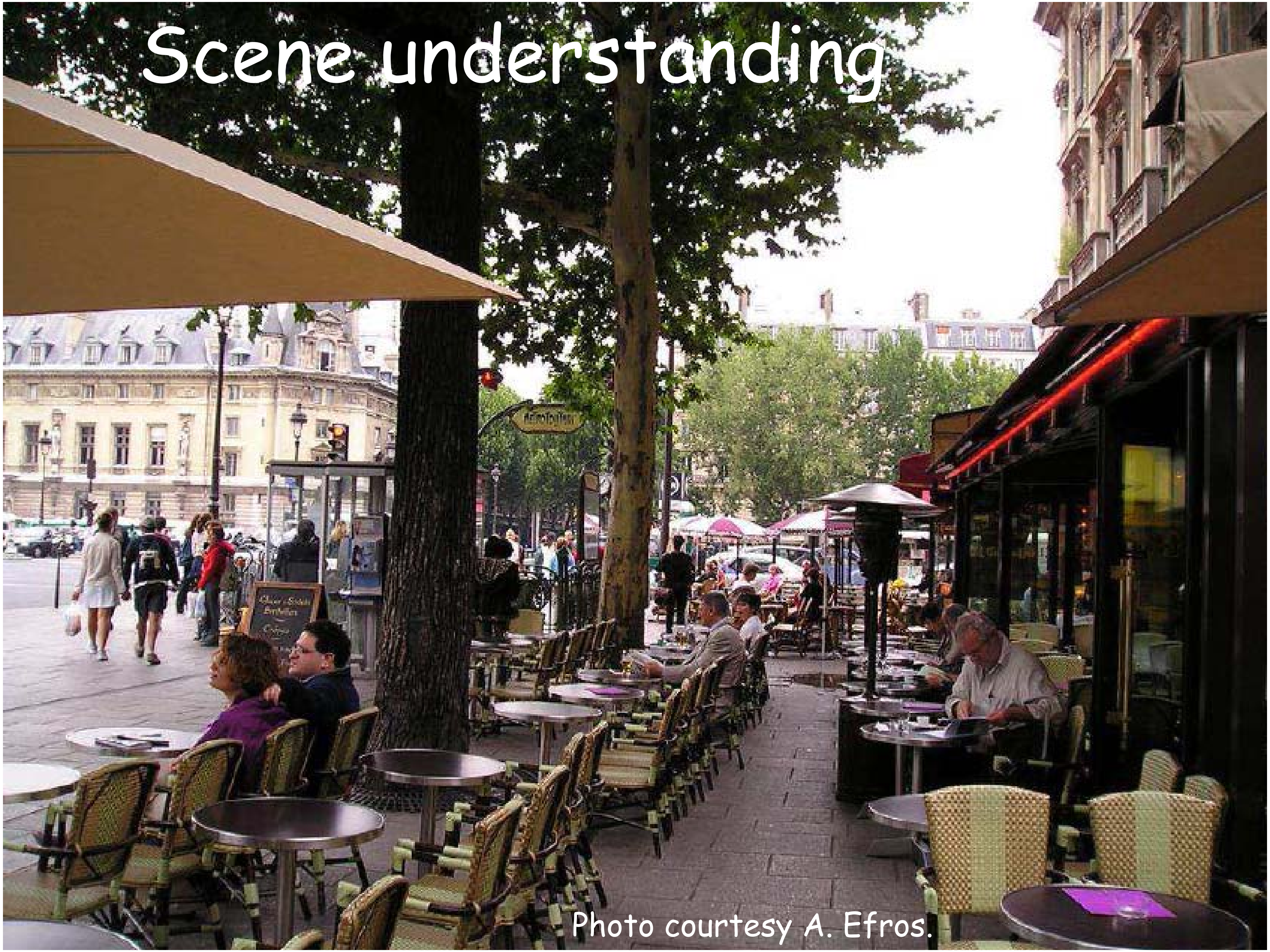
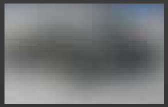
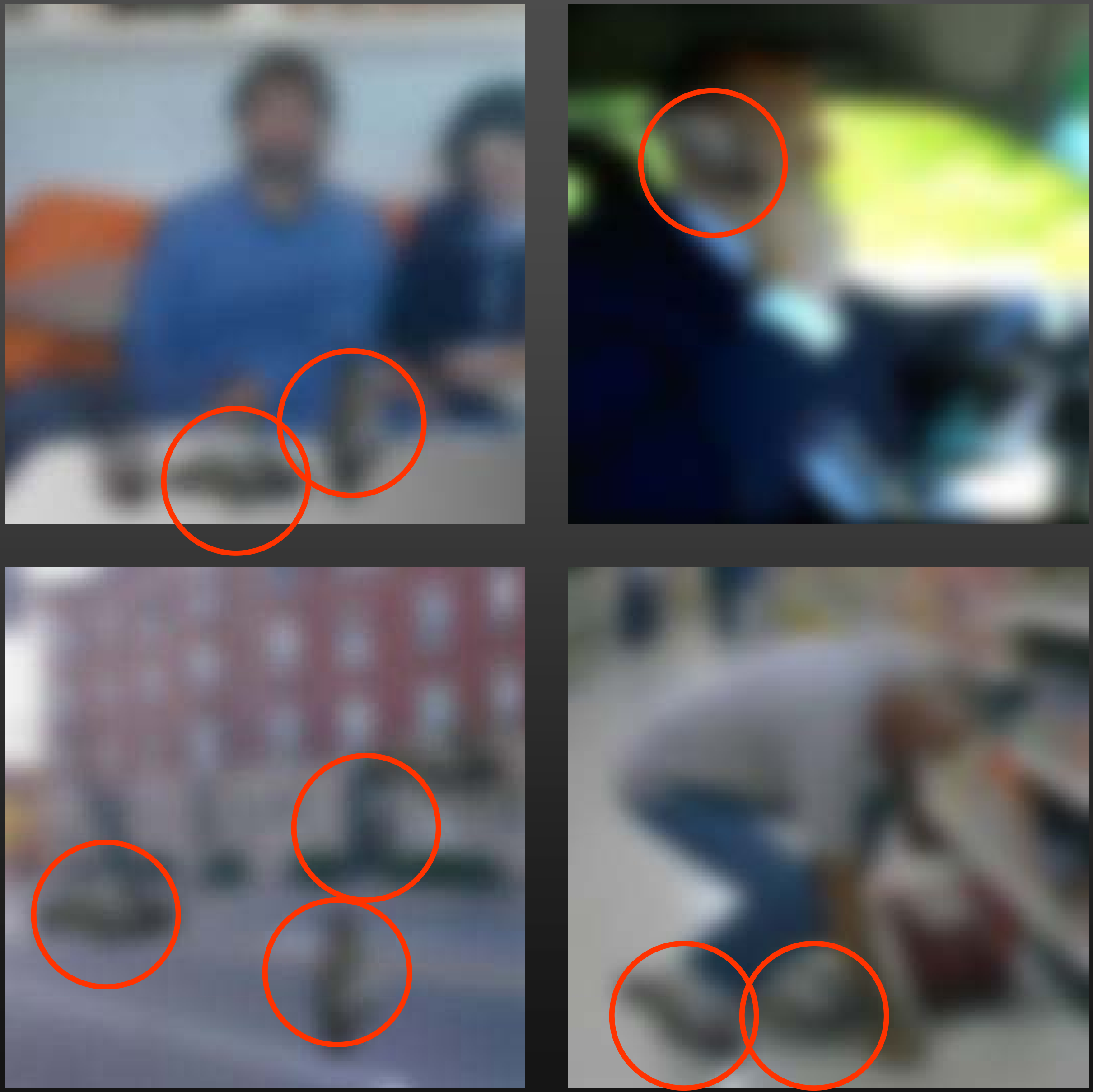


Photo courtesy A. Efros.

Local ambiguity and global scene interpretation





slide credit: Fei-Fei, Fergus & Torralba

Reconnaissance d'objets et vision artificielle

(Jean Ponce, Cordelia Schmid, Josef Sivic)

La reconnaissance automatique des objets –et de manière plus générale, l'interprétation de la scène– figurant dans une photographie ou une vidéo est le plus grand défi de la vision artificielle. Ce cours présente les modèles d'images, d'objets, et de scènes, ainsi que les méthodes et algorithmes utilisés aujourd'hui pour affronter ce défi.

Plan du cours :

- Caractéristiques visuelles : points d'intérêt, régions affines, invariants, descripteurs Sift.
- Détection d'objets et de classes spécifiques : alignement 2D et 3D, méthodes de votes, détection de visages et Adaboost.
- Classification d'images : sacs de caractéristiques visuelles et machines à vecteurs de support, grilles et pyramides, réseaux convolutionnels.
- Détection de catégories d'objets : constellations de caractéristiques visuelles, assemblages de fragments, méthodes de fenêtre glissantes, apprentissage faiblement supervisé de modèles.
- Aller plus loin : analyse de scène, analyse des activités dans les vidéos.

Bibliographie :

- D.A. Forsyth and J. Ponce, "Computer Vision: A Modern Approach", Prentice-Hall, 2003.
- J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, "Toward Category-Level Object Recognition", Lecture Notes in Computer Science 4170, Springer-Verlag, 2007.

Other notable computer vision books

- O. Faugeras, Q.T. Luong, and T. Papadopoulo, “Geometry of Multiple Images,” MIT Press, 2001.

- R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”, Cambridge University Press, 2004.

- J. Koenderink, “Solid Shape”, MIT Press, 1990.

Slides

After classes:

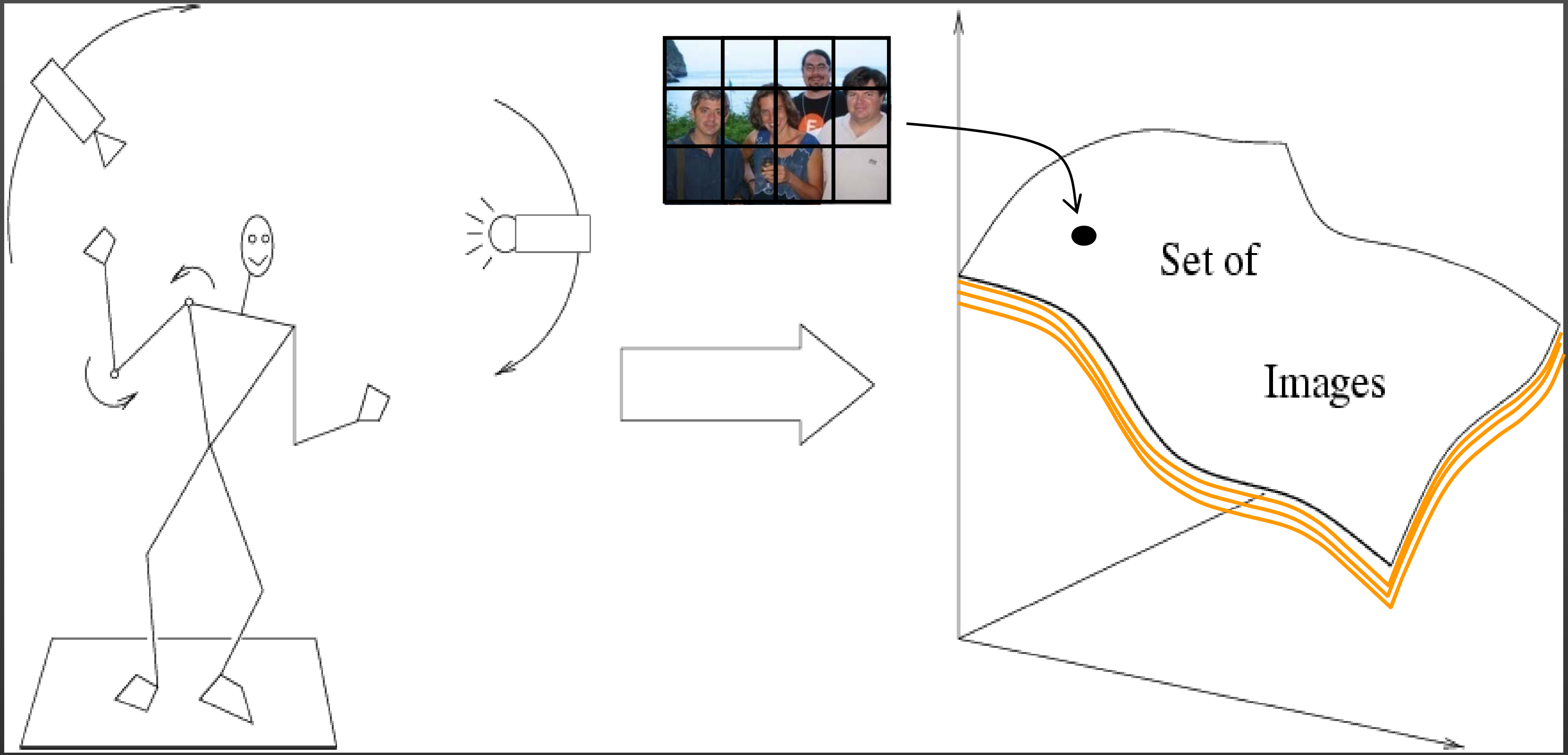
http://www.di.ens.fr/~ponce/recvis/lecture1.ppt

http://www.di.ens.fr/~ponce/recvis/lecture1.pdf

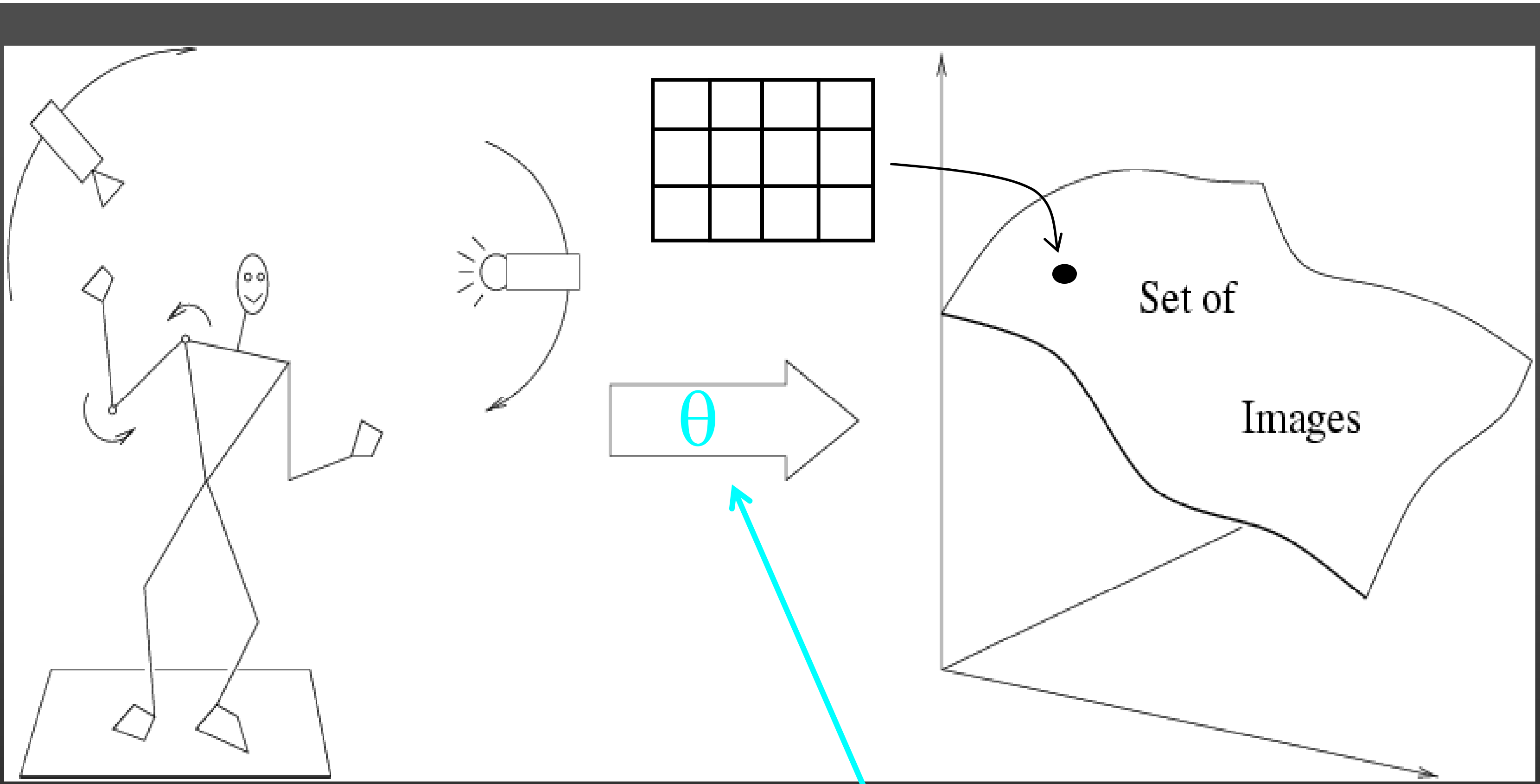
Note: Much of the material used in this lecture is courtesy of Svetlana Lazebnik:, http://www.cs.unc.edu/~lazebnik/

Outline

- What computer vision is about
- What this class is about
- A brief history of visual recognition
- Alignment methods



Variability: Camera position
Illumination
Internal parameters
Within-class variations

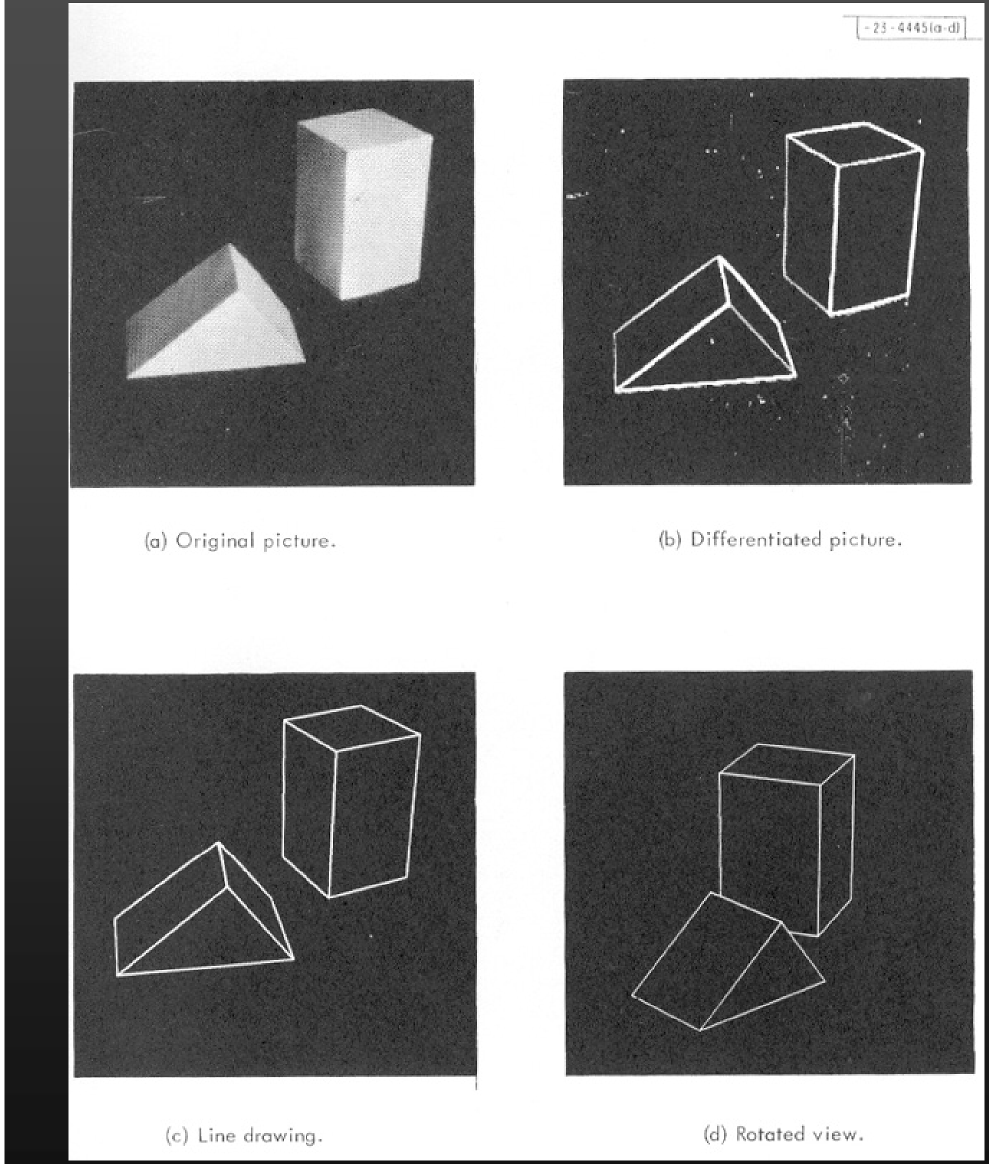


Variability:

Camera position
Illumination
Internal parameters

Roberts (1963); Lowe (1987); Faugeras & Hebert (1986); Grimson & Lozano-Perez (1986); Huttenlocher & Ullman (1987)

Origins of computer vision



(a) Original picture.

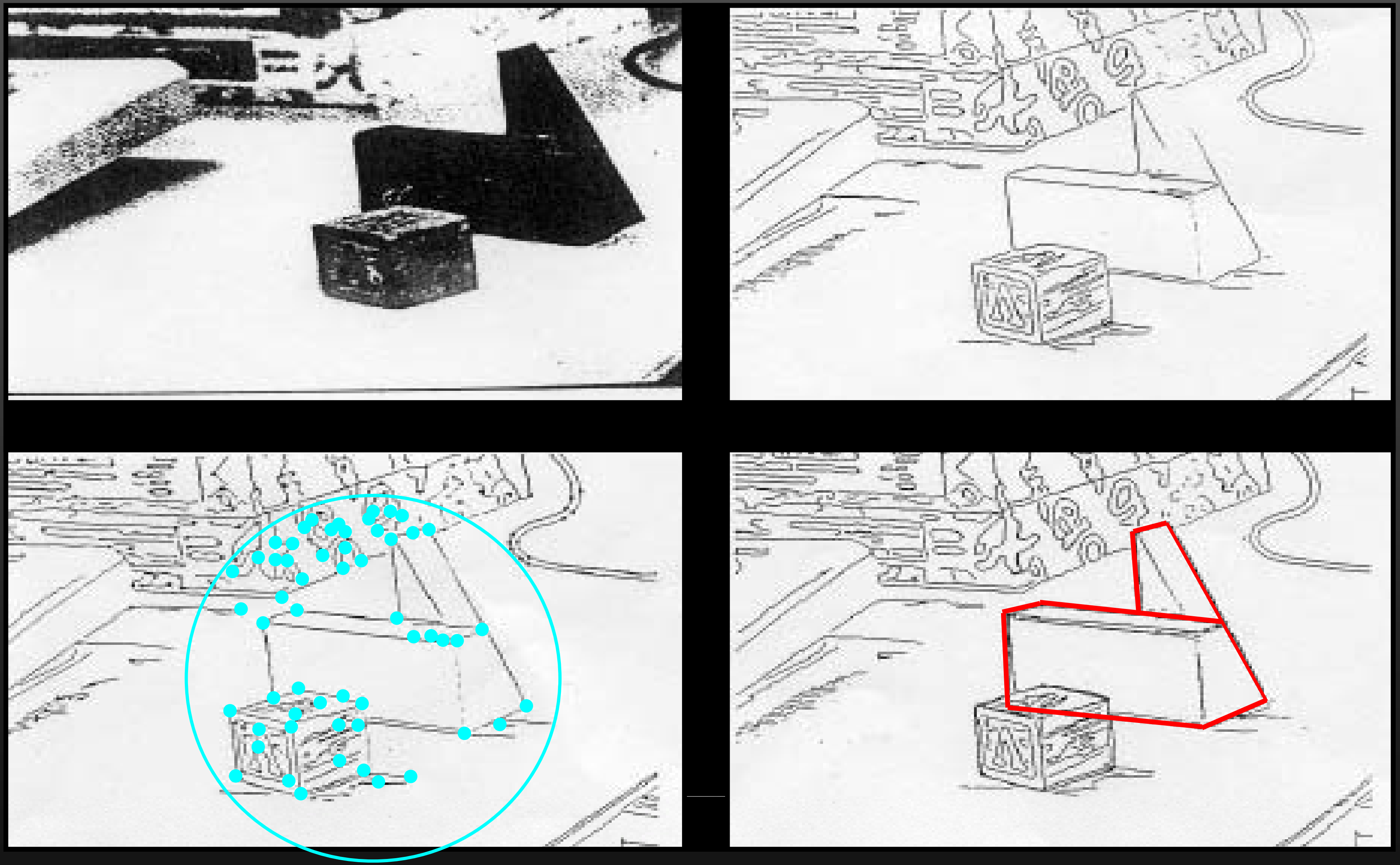
(b) Differentiated picture.

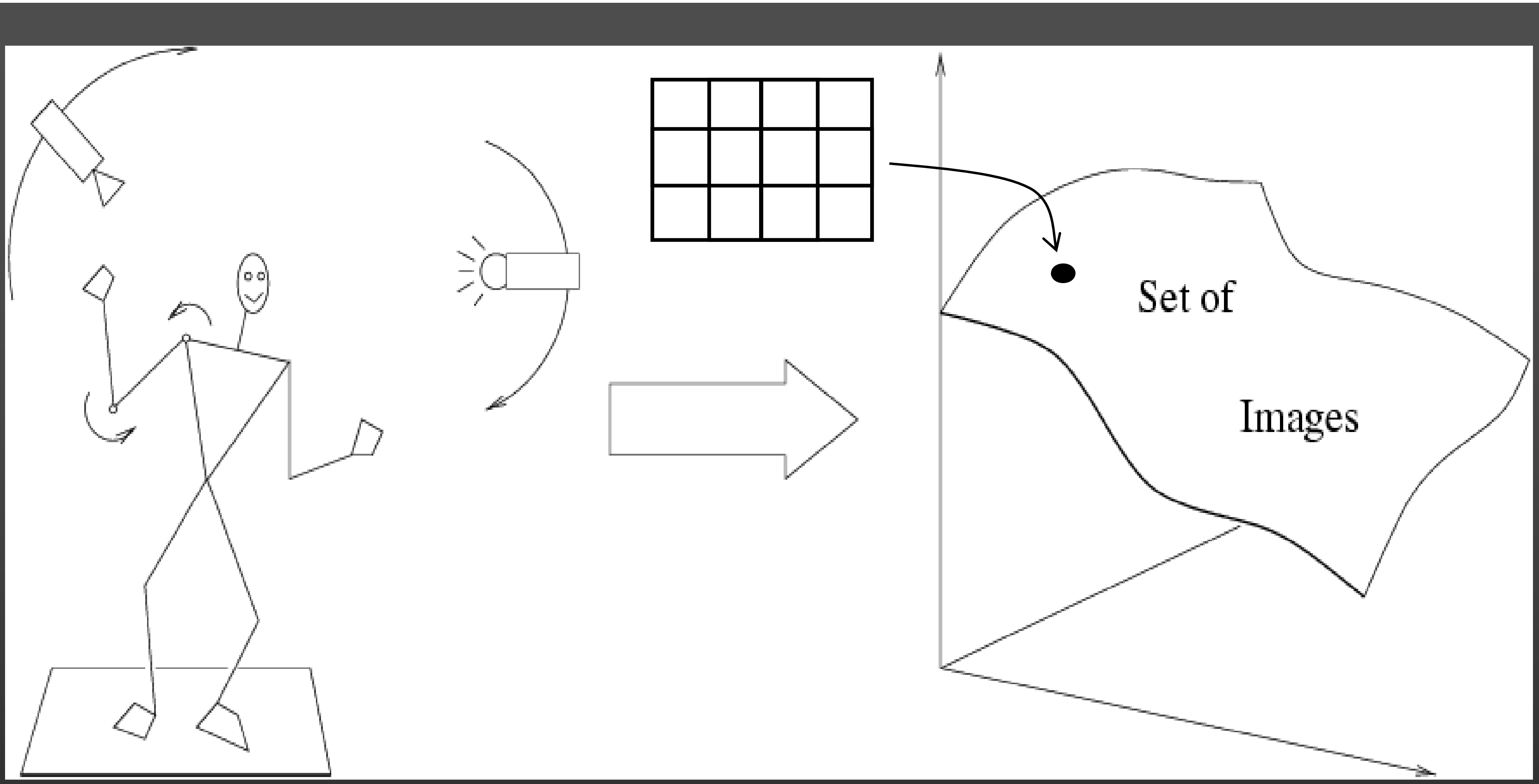
(c) Line drawing.

(d) Rotated view.

L. G. Roberts, *Machine Perception of Three Dimensional Solids,* Ph.D. thesis, MIT Department of Electrical Engineering, 1963.

Huttenlocher & Ullman (1987)

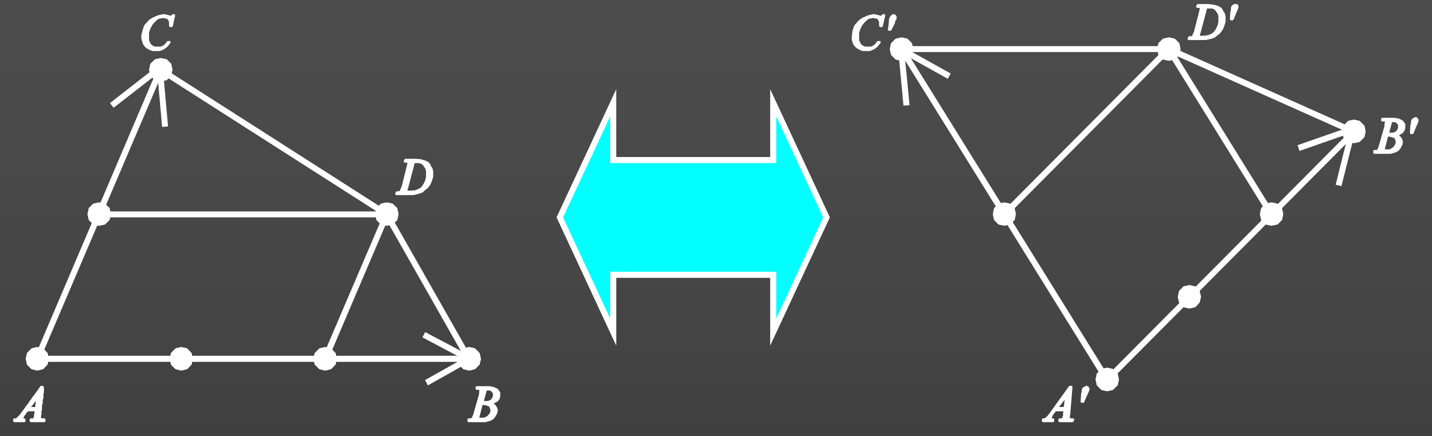




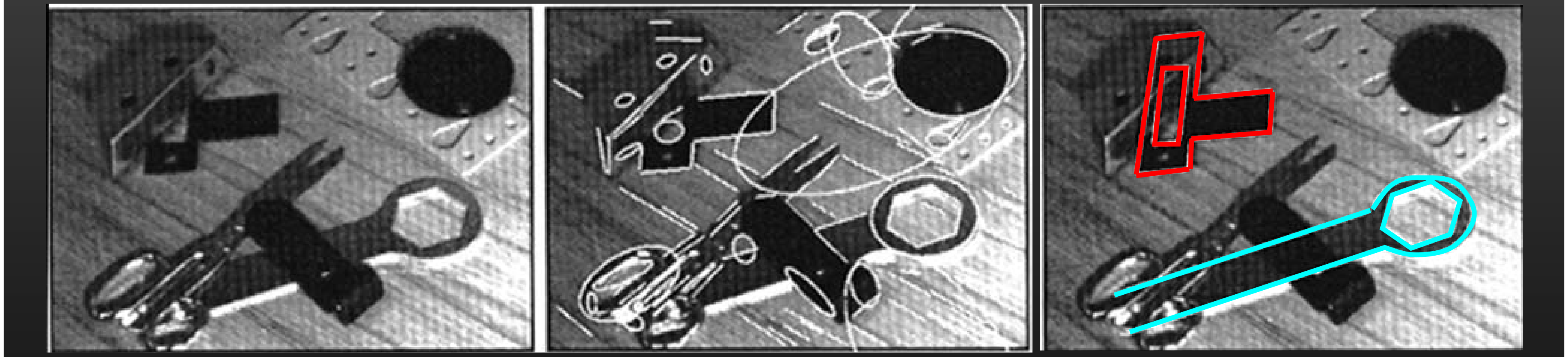
~~Variability~~ Invariance to: Camera position
Illumination
Internal parameters

Duda & Hart (1972); Weiss (1987); Mundy et al. (1992-94);
Rothwell et al. (1992); Burns et al. (1993)

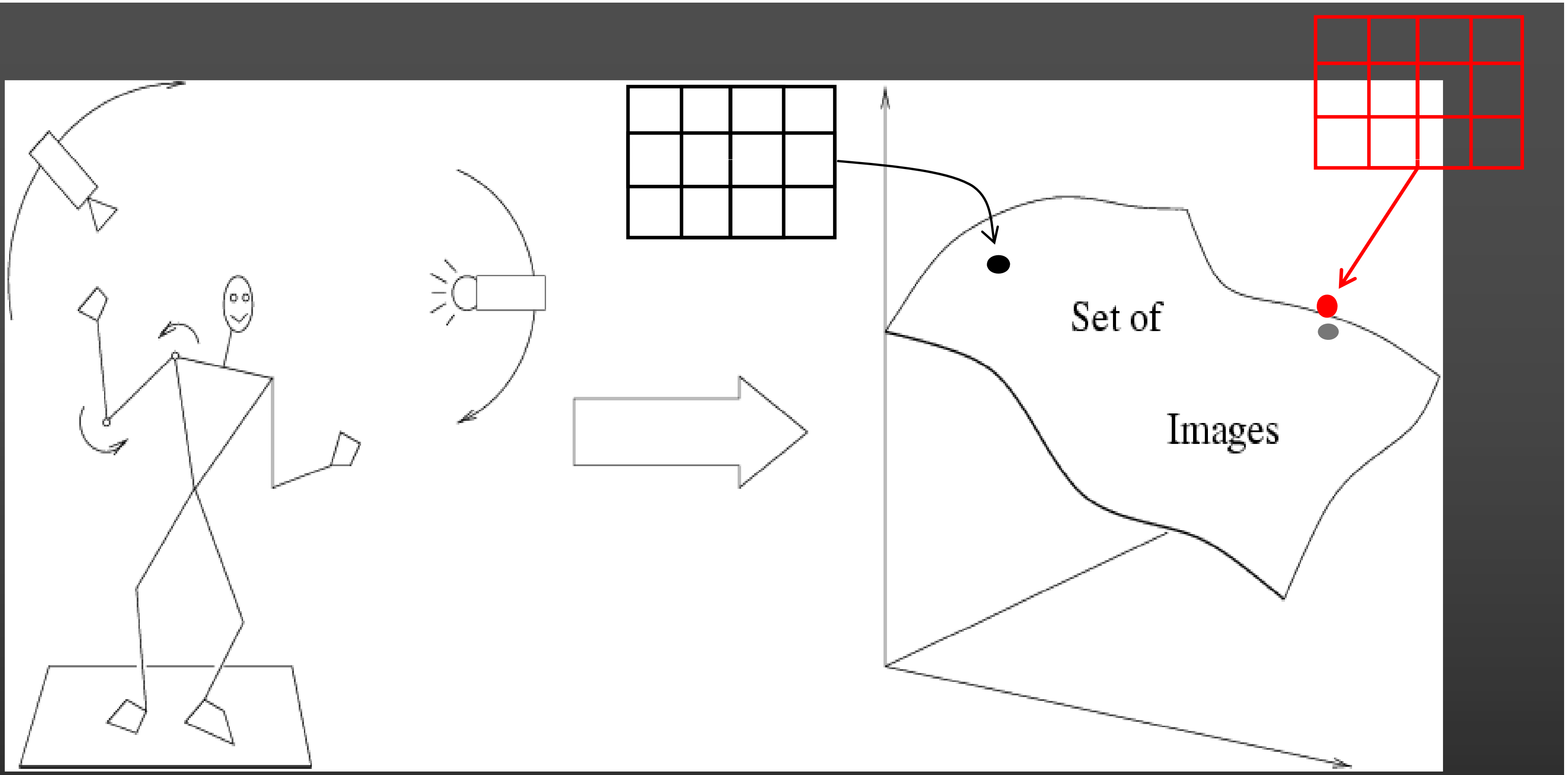
Example: affine invariants of coplanar points



Projective invariants (Rothwell et al., 1992):



BUT: True 3D objects do not admit monocular viewpoint invariants (Burns et al., 1993) !!

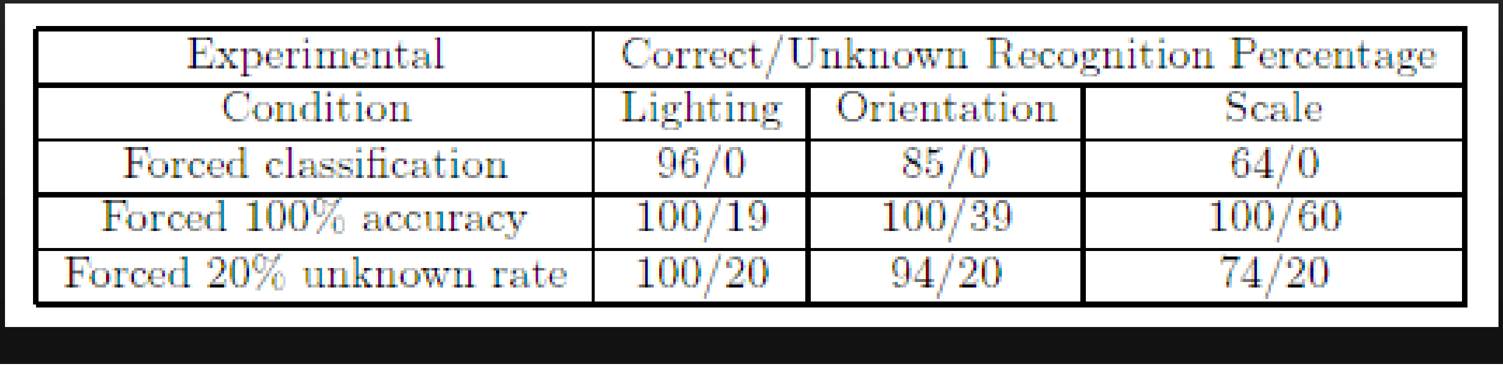


Empirical models of image variability:

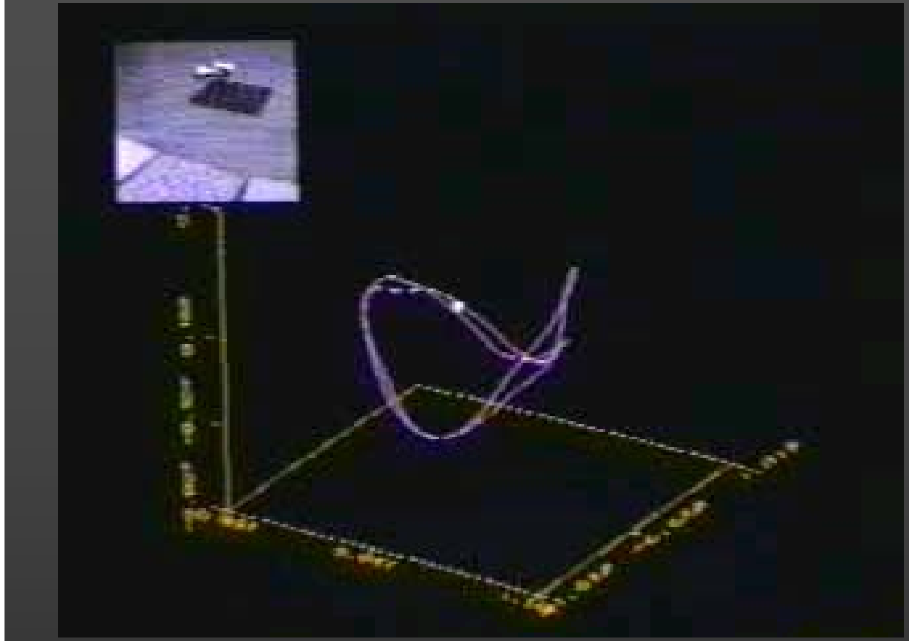
Appearance-based techniques

Turk & Pentland (1991); Murase & Nayar (1995); etc.

Eigenfaces (Turk & Pentland, 1991)

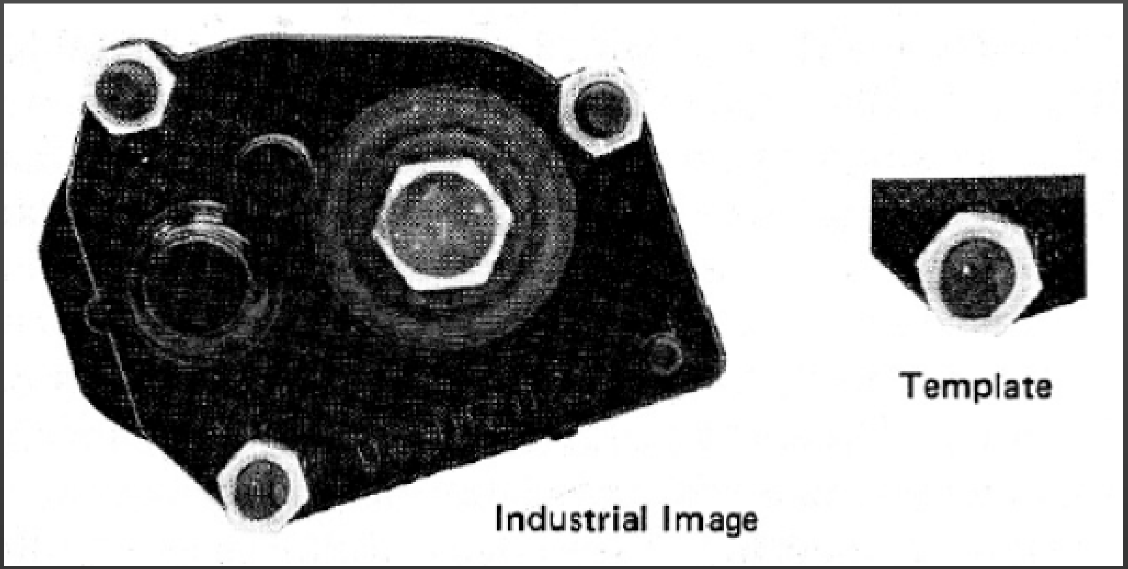


Experimental	Correct/Unknown Recognition Percentage		
Condition	Lighting	Orientation	Scale
Forced classification	96/0	85/0	64/0
Forced 100% accuracy	100/19	100/39	100/60
Forced 20% unknown rate	100/20	94/20	74/20



Appearance manifolds
(Murase & Nayar, 1995)

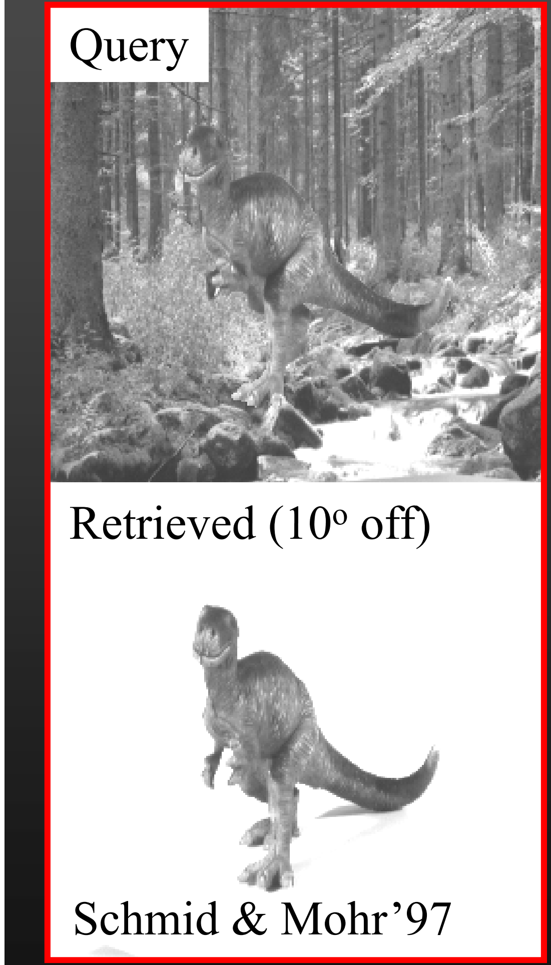
Correlation-based template matching (60s)

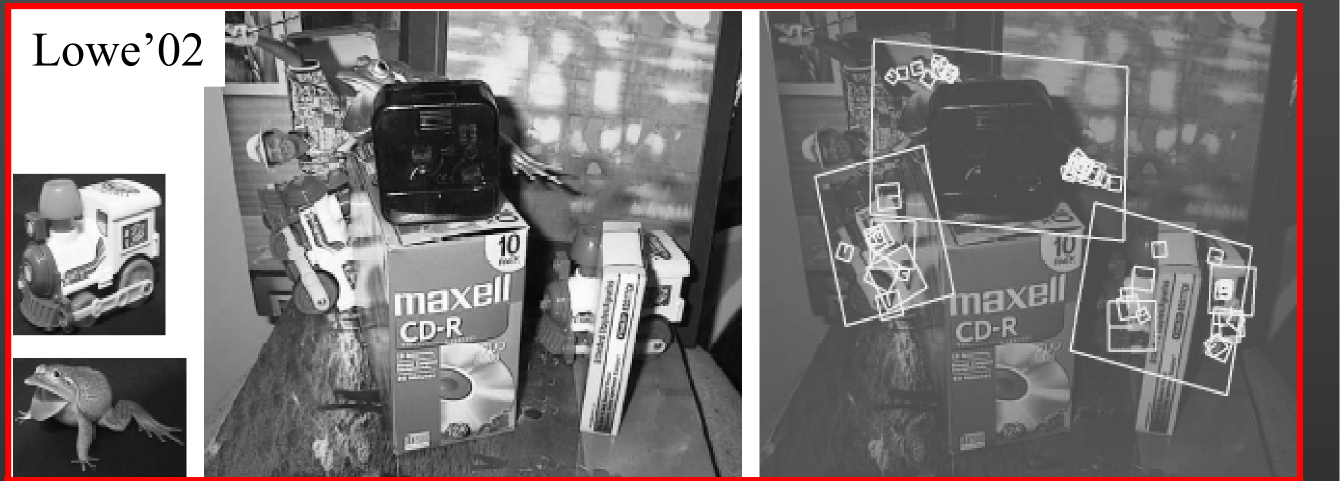


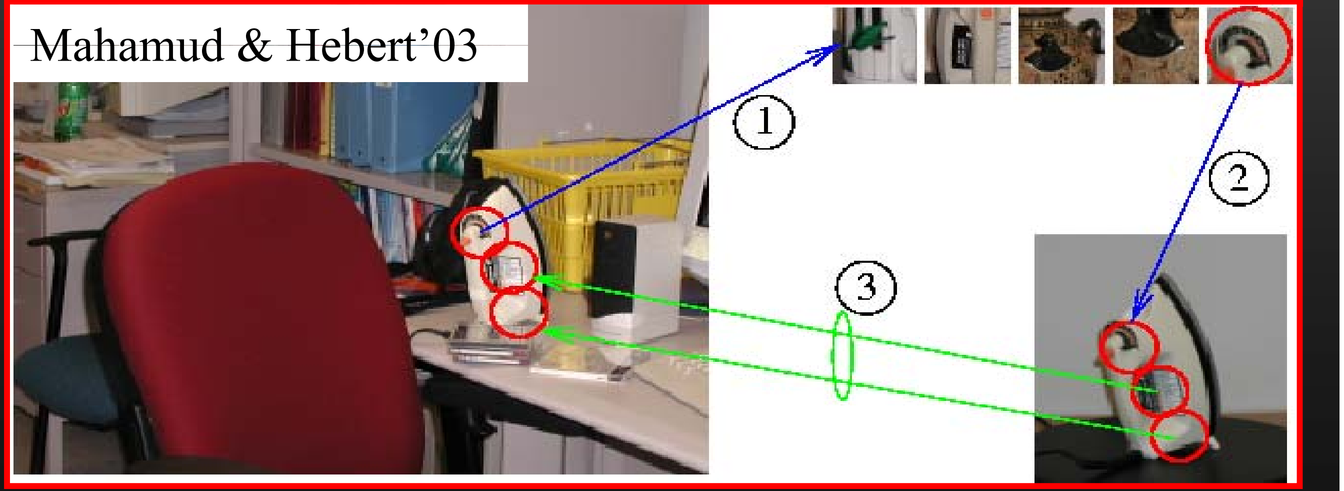
Ballard & Brown (1980, Fig. 3.3). Courtesy Bob Fisher and Ballard & Brown on-line.

- Automated target recognition
- Industrial inspection
- Optical character recognition
- Stereo matching
- Pattern recognition

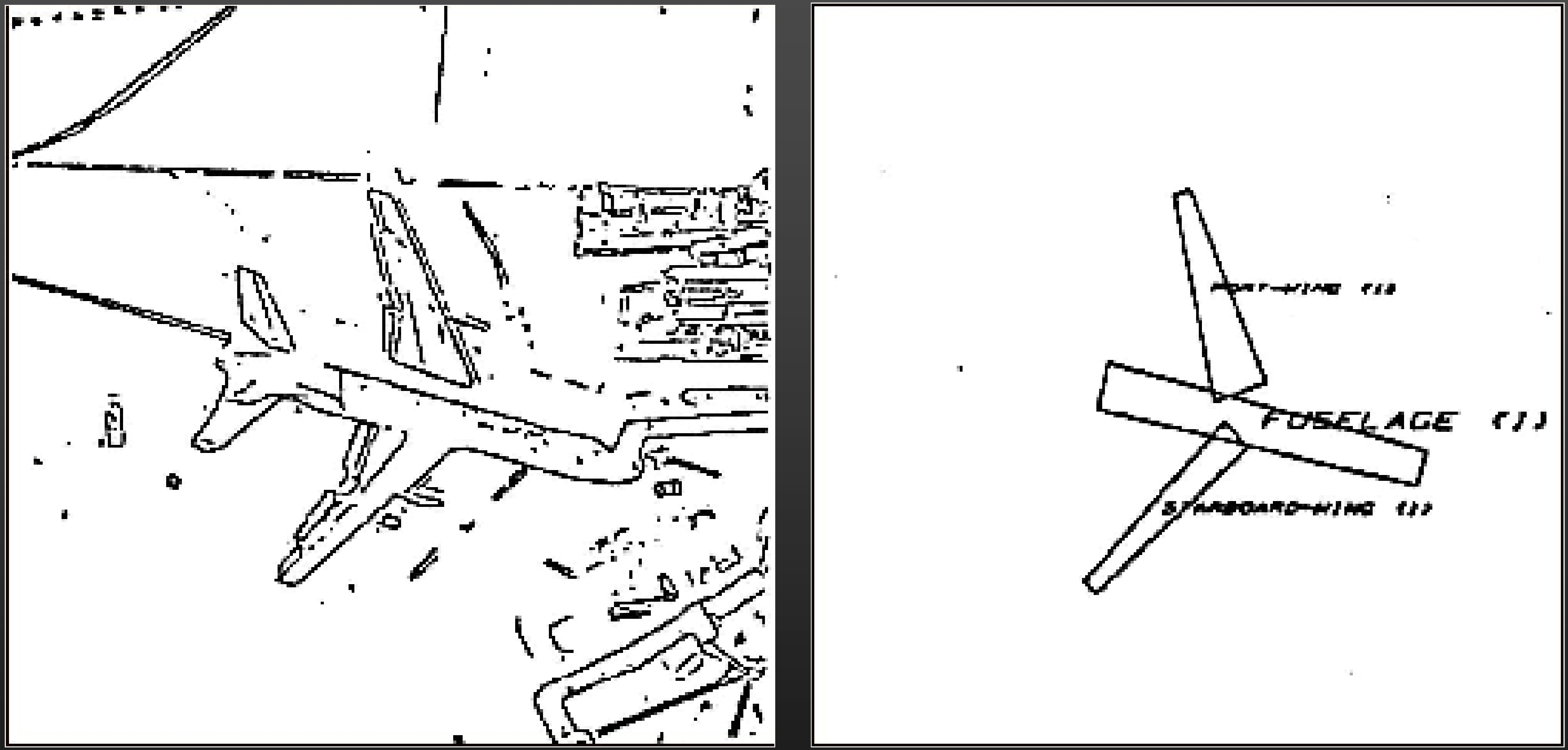
In the lates 1990s, a new approach emerges:
Combining *local* appearance, spatial constraints, invariants,
and classification techniques from machine learning.







Representing and recognizing object categories is harder

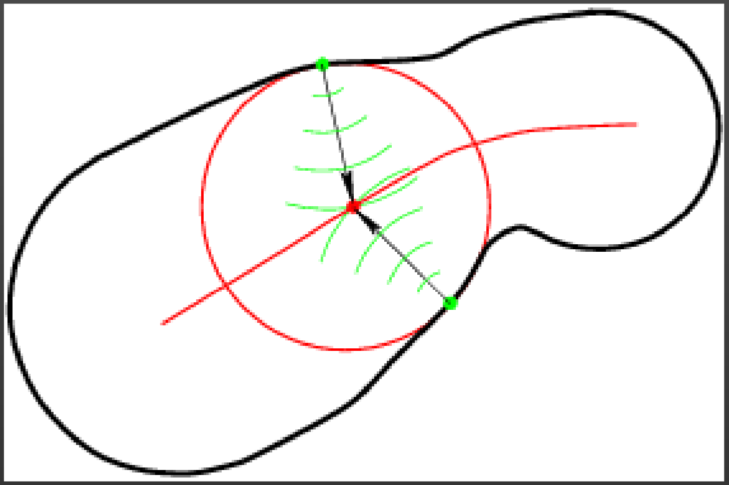


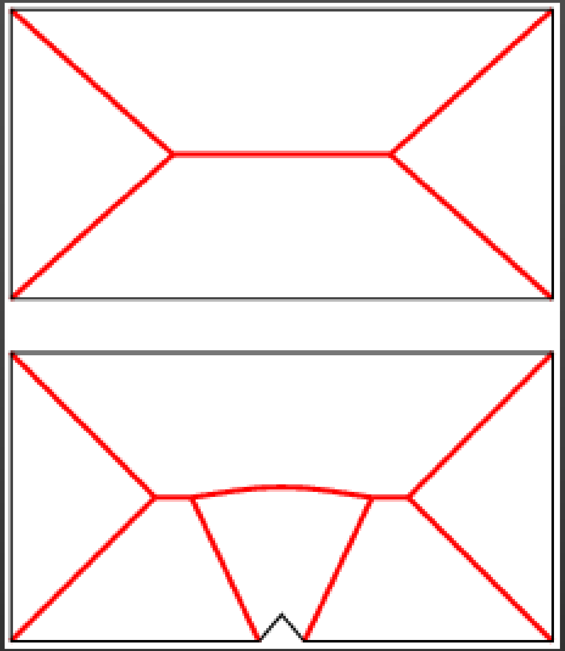
ACRONYM (Brooks and Binford, 1981)

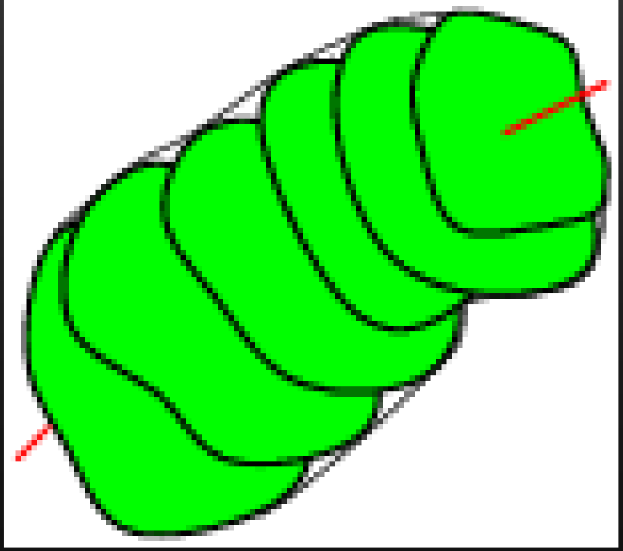
Binford (1971), Nevatia & Binford (1972), Marr & Nishihara (1978)

Parts and invariants

The Blum transform, 1967

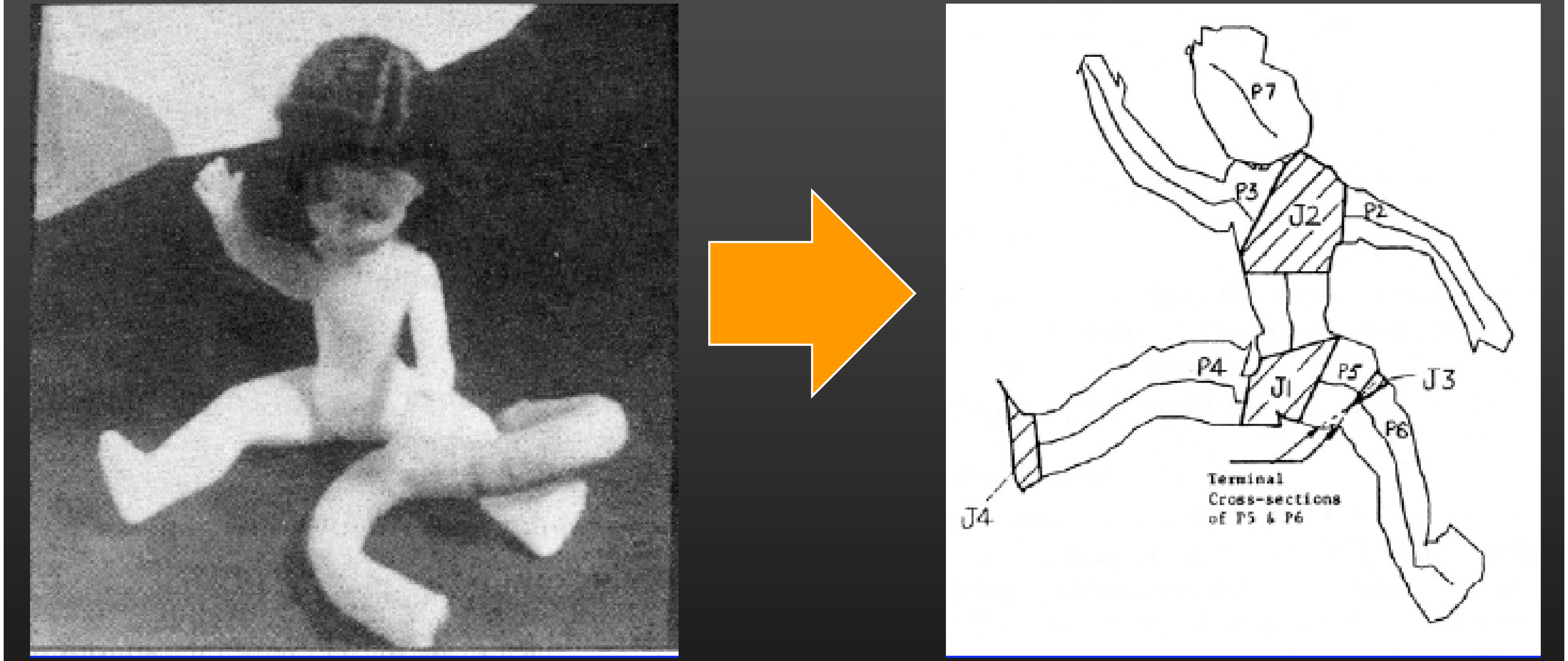






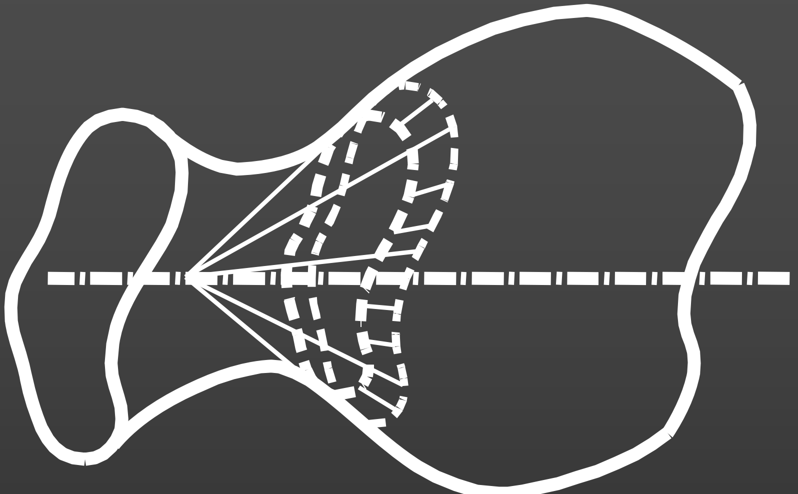
Generalized cylinders (Binford, 1971)

Generalized cylinders
(Binford, 1971; Marr & Nishihara, 1978)

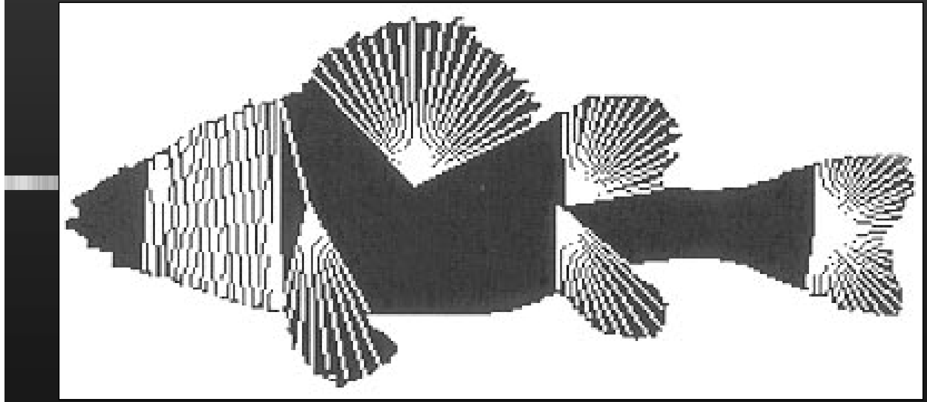


(Nevatia & Binford, 1972)

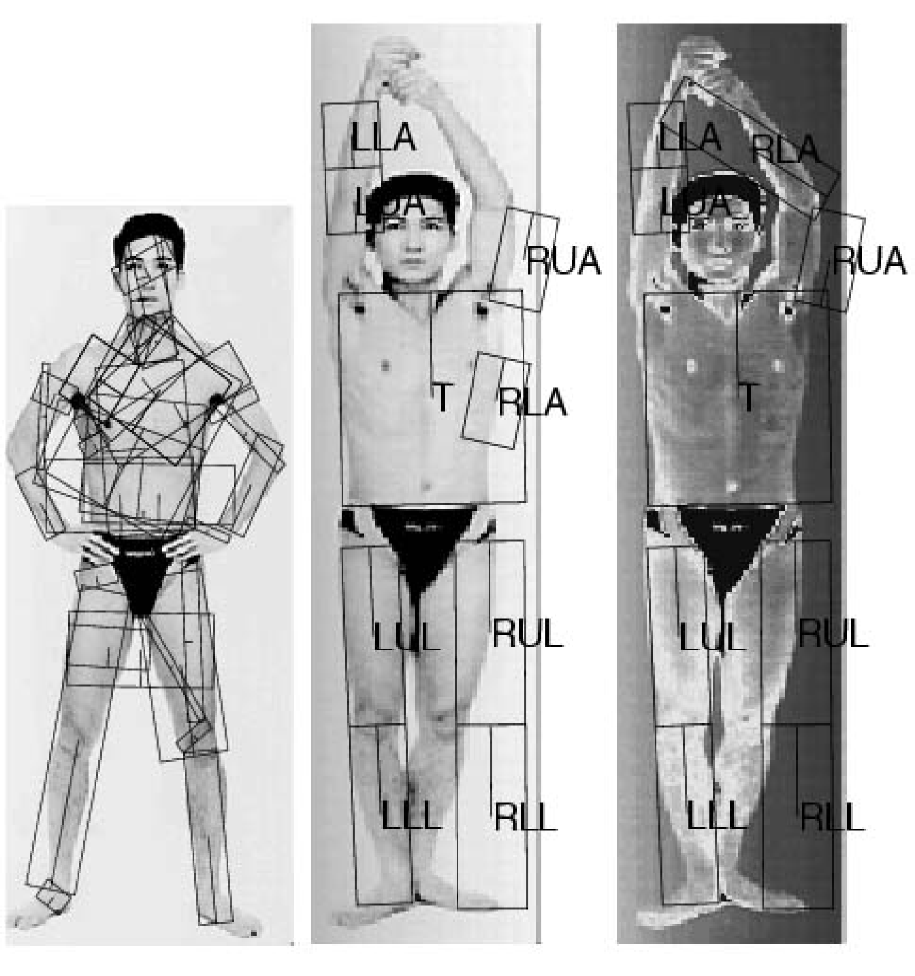
Parts and invariants II



Ponce et al. (1989)

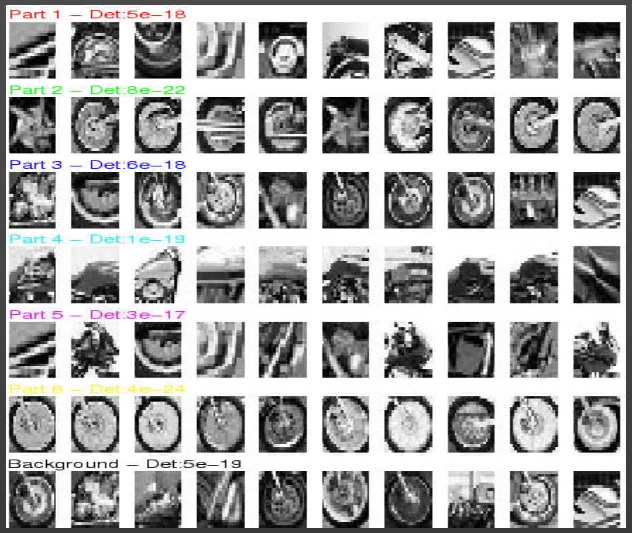


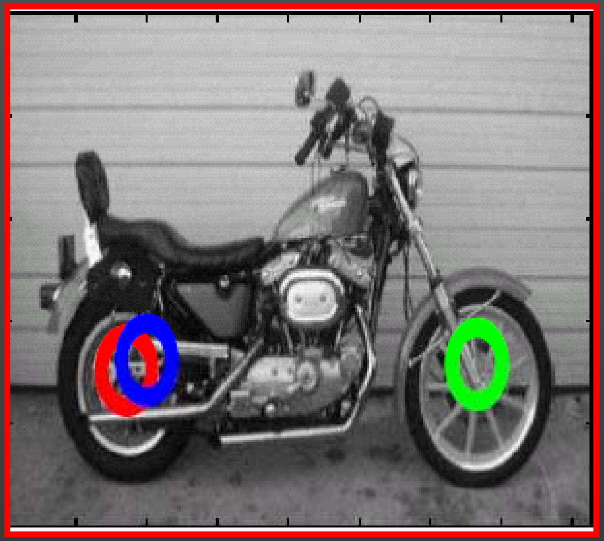
Zhu and Yuille (1996)

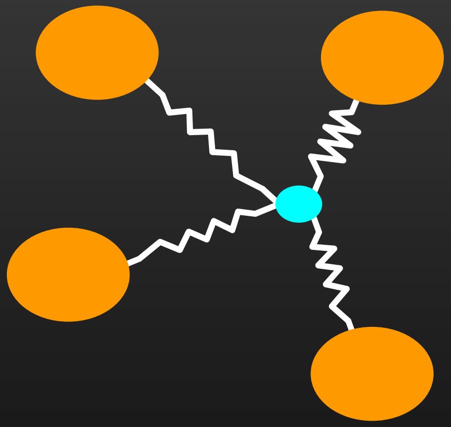


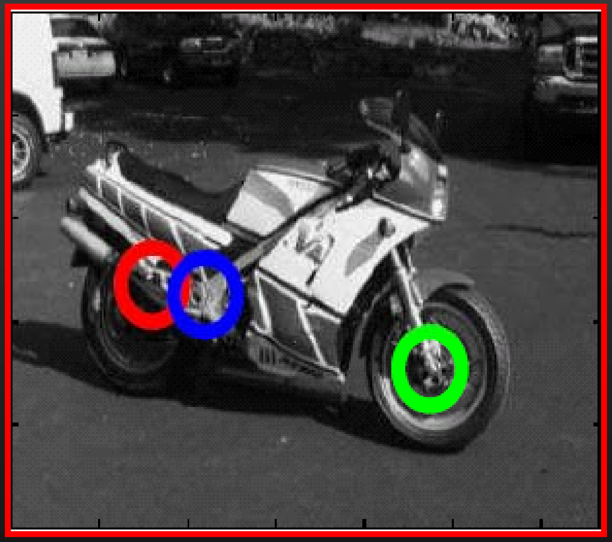
Ioffe and Forsyth (2000)

In the early 2000's, a new approach ?



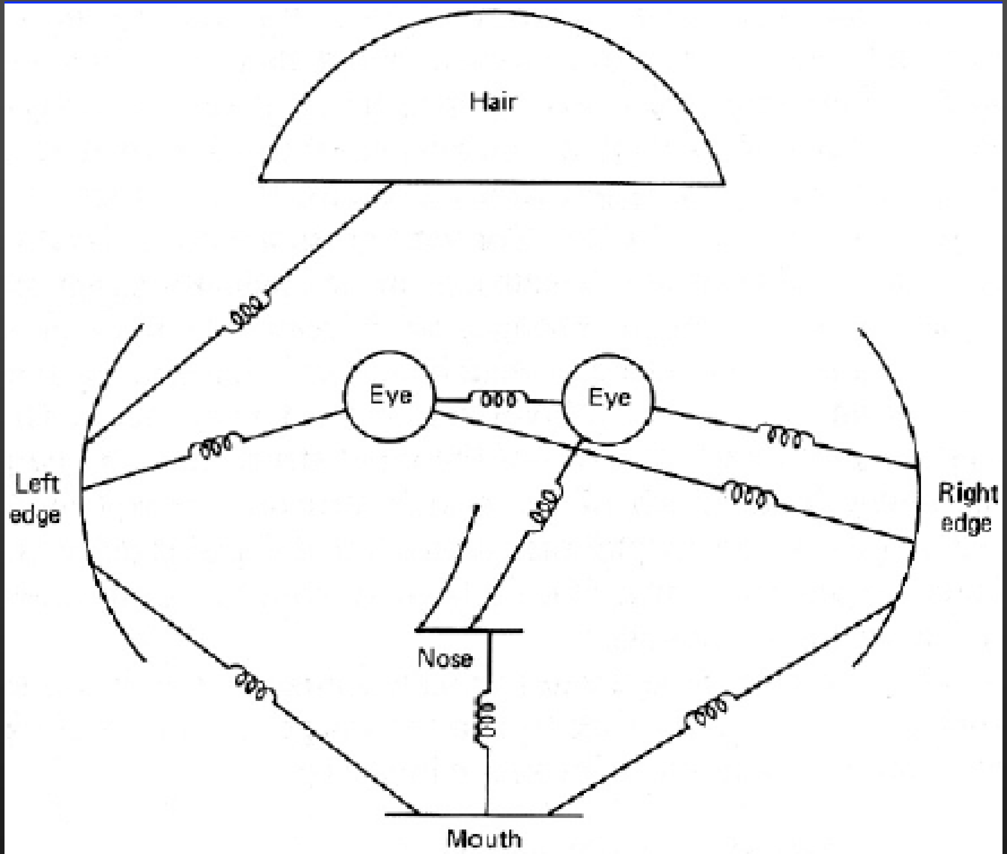




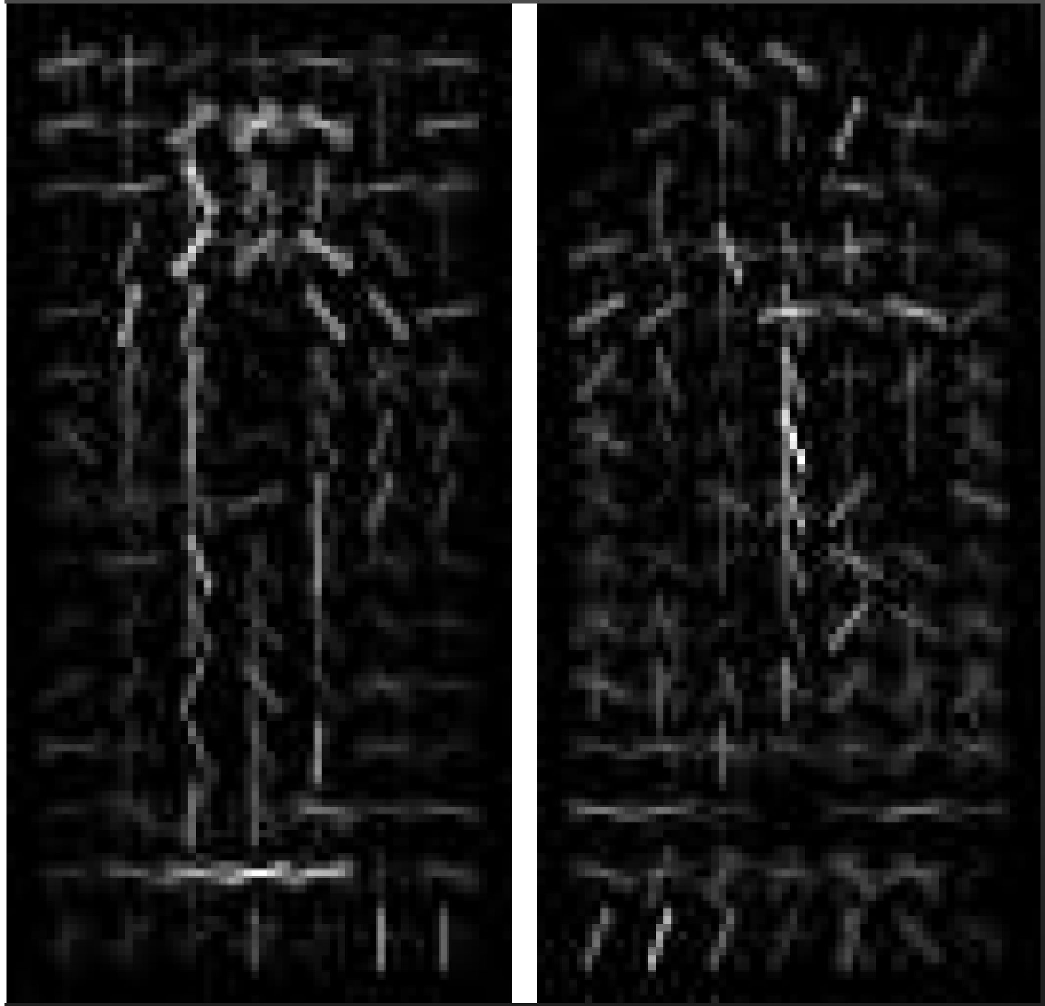


Fergus, Perona & Zisserman (2003)

The “templates and springs” model (Fischler & Elschlager, 1973)

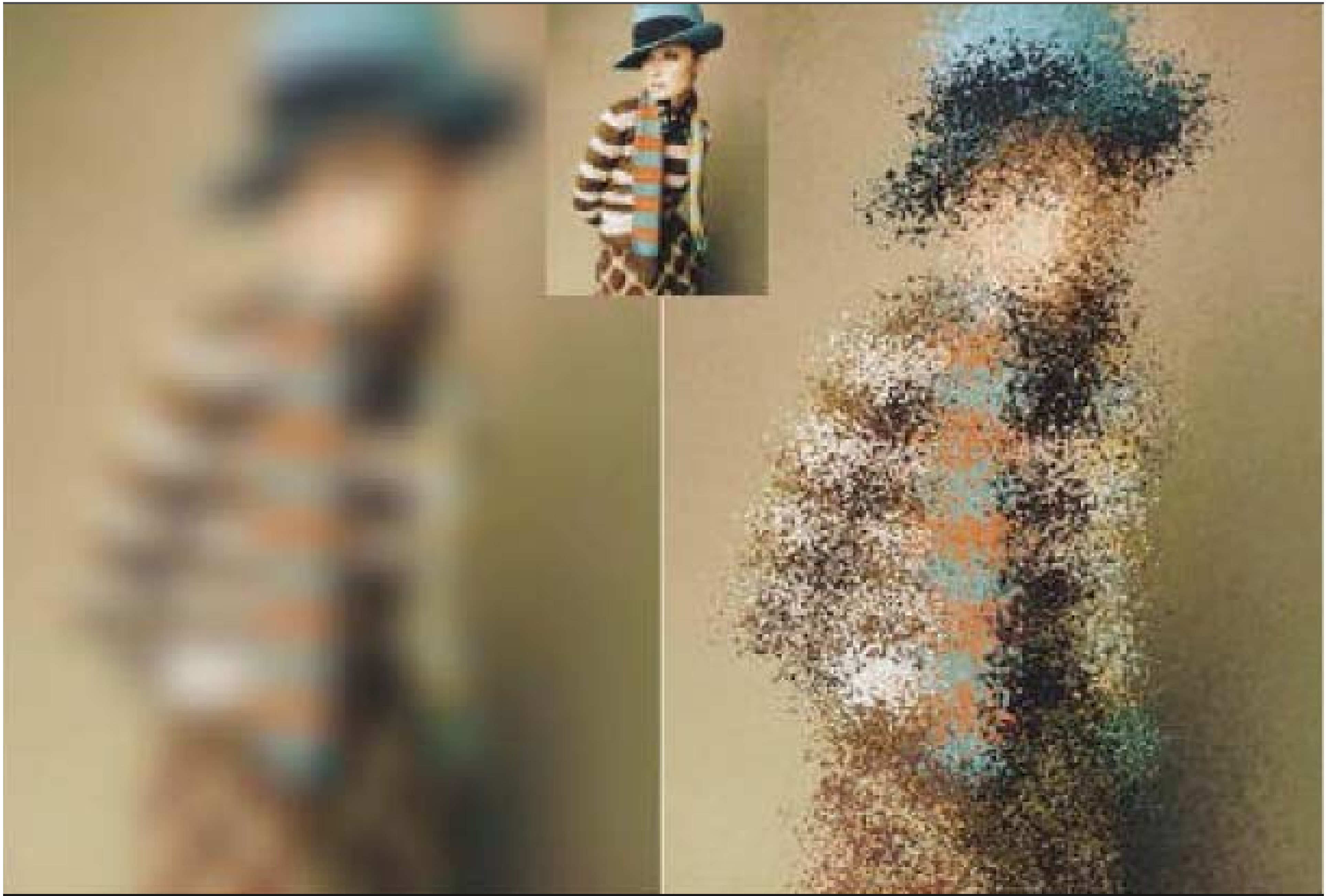


Ballard & Brown (1980, Fig. 11.5). Courtesy Bob Fisher and Ballard & Brown on-line.

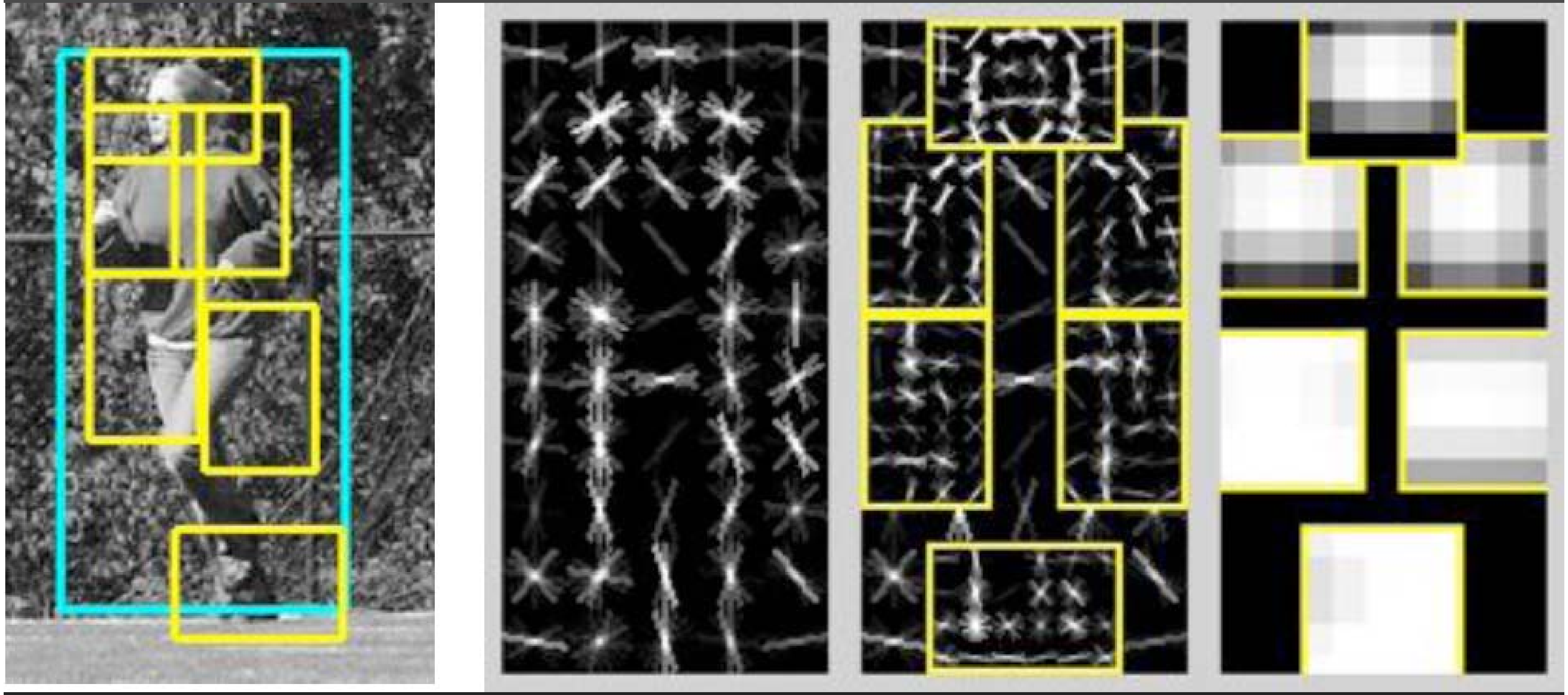


Color histograms (S&B’91)
Local jets (Florack’93)
Spin images (J&H’99)
Sift (Lowe’99)
Shape contexts (B&M’95)

Texton histograms (L&M’97)
Gist (O&T’05)
Spatial pyramids (LSP’06)
Hog (D&T’06)
Phog (B&Z’07)
Convolutional nets (LC’90)



Locally orderless structure of images (K&vD'99)



Felzwenszalb, McAllester, Ramanan (2007)

[Wins on 6 of the Pascal'07 classes, see Chum & Zisserman (2007) for the other big winner.]

Outline

- What computer vision is about
- What this class is about
- A brief history of visual recognition
- Alignment methods

Reconnaissance d'objets et vision artificielle

(Jean Ponce, Cordelia Schmid, Josef Sivic)

La reconnaissance automatique des objets –et de manière plus générale, l'interprétation de la scène– figurant dans une photographie ou une vidéo est le plus grand défi de la vision artificielle. Ce cours présente les modèles d'images, d'objets, et de scènes, ainsi que les méthodes et algorithmes utilisés aujourd'hui pour affronter ce défi.

Plan du cours :

Next time

- Caractéristiques visuelles : points d'intérêt, régions affines, invariants, descripteurs Sift.
- Détection d'objets et de classes spécifiques : alignement 2D et 3D, méthodes de votes, détection de visages et Adaboost.

Today

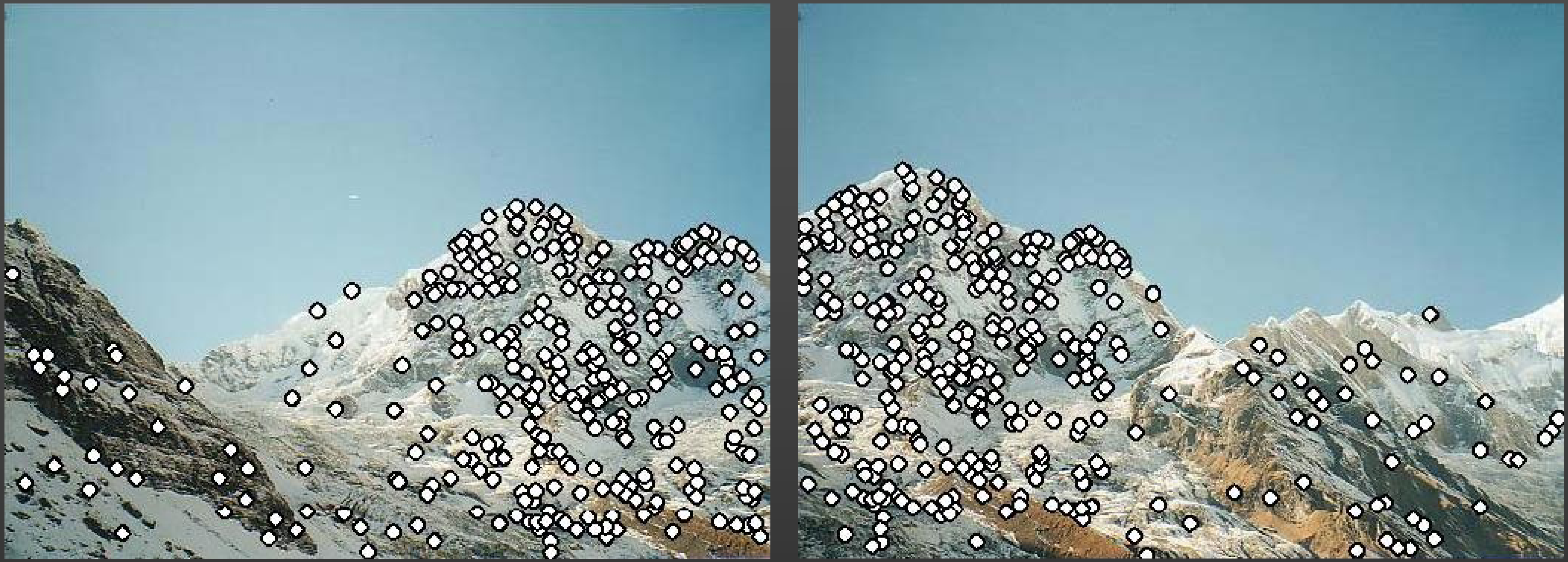
- Classification d'images : sacs de caractéristiques visuelles et machines à vecteurs de support, grilles et pyramides, réseaux convolutionnels.
- Détection de catégories d'objets : constellations de caractéristiques visuelles, assemblages de fragments, méthodes de fenêtre glissantes, apprentissage faiblement supervisé de modèles.
- Aller plus loin : analyse de scène, analyse des activités dans les vidéos.

Bibliographie :

- D.A. Forsyth and J. Ponce, "Computer Vision: A Modern Approach", Prentice-Hall, 2003.
- J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, "Toward Category-Level Object Recognition", Lecture Notes in Computer Science 4170, Springer-Verlag, 2007.

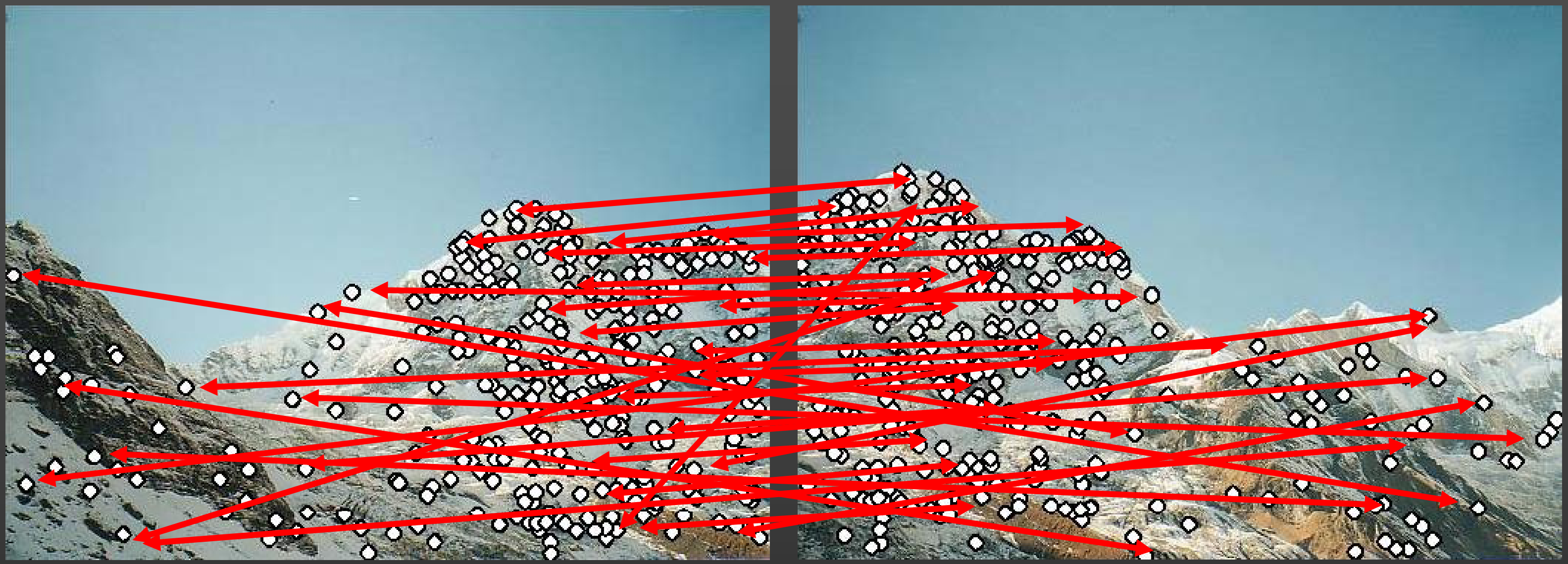
Feature-based alignment outline

Feature-based alignment outline



Extract features

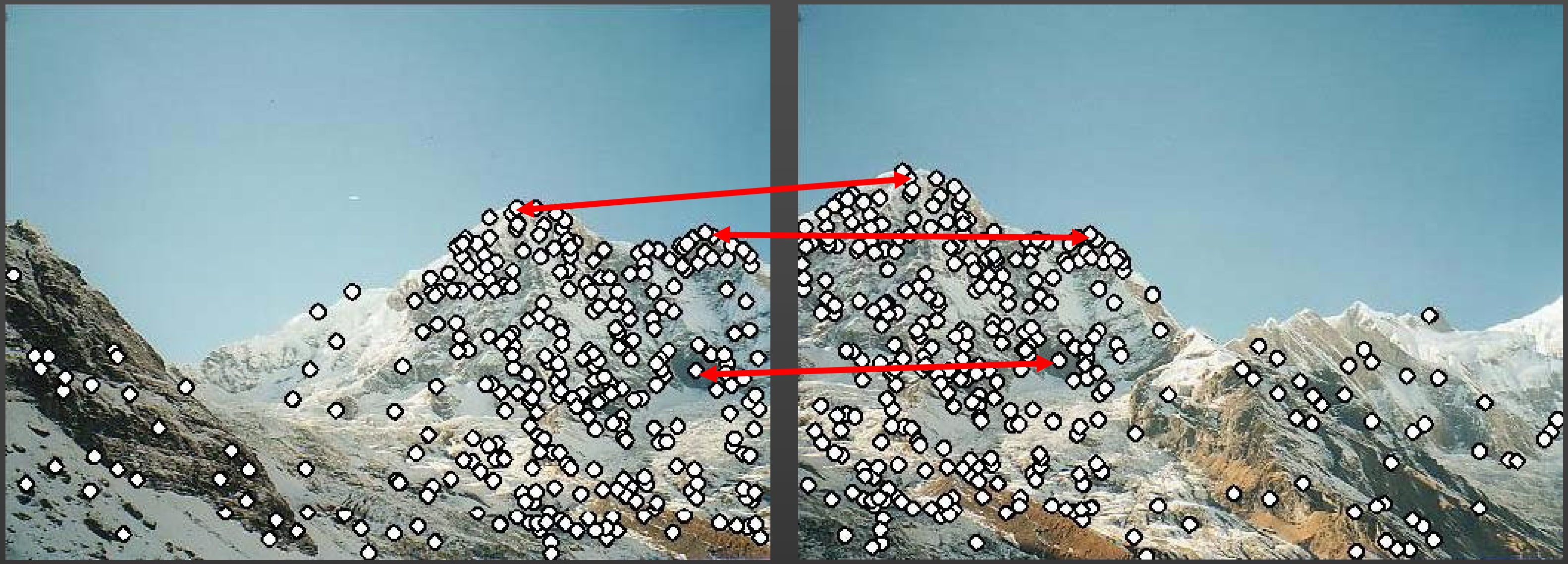
Feature-based alignment outline



Extract features

Compute *putative matches*

Feature-based alignment outline



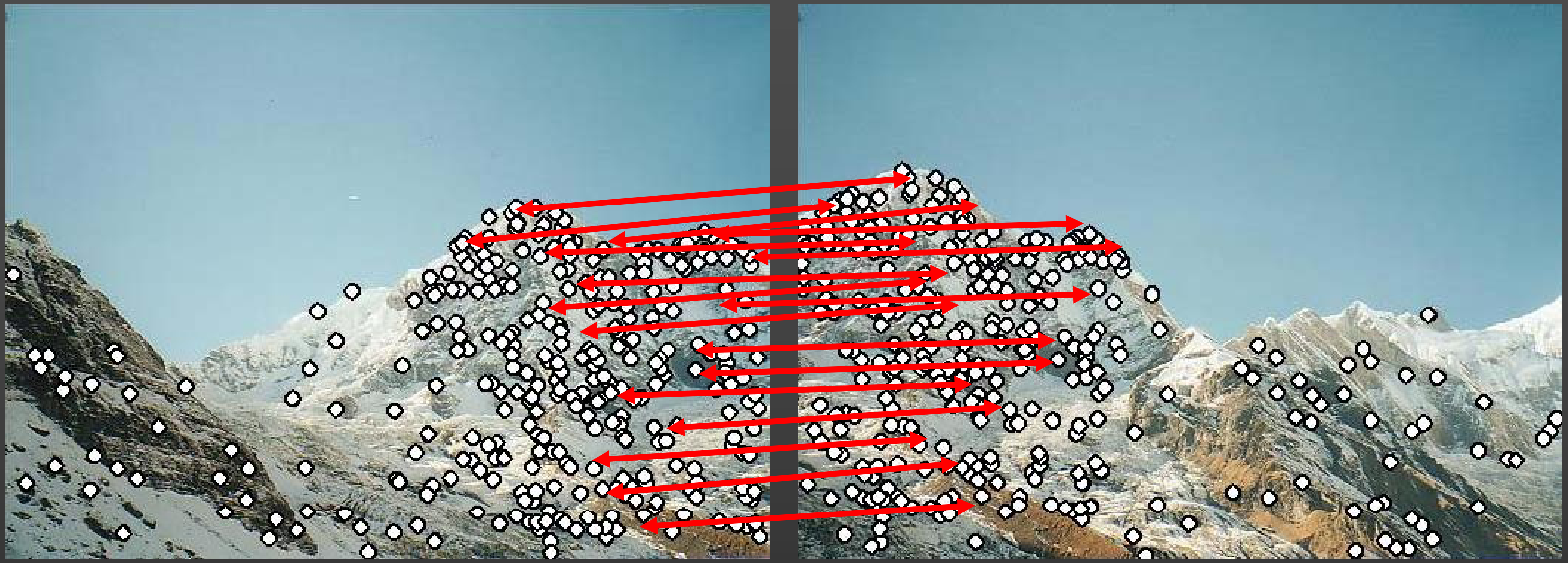
Extract features

Compute *putative matches*

Loop:

- *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)

Feature-based alignment outline



Extract features

Compute *putative matches*

Loop:

- *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
- *Verify* transformation (search for other matches consistent with *T*)

Feature-based alignment outline

Extract features

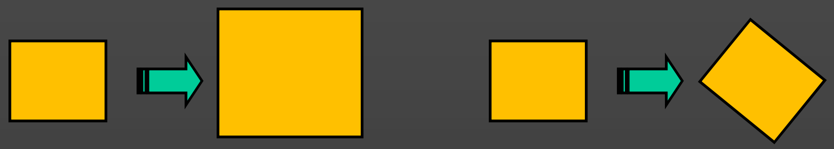
Compute *putative matches*

Loop:

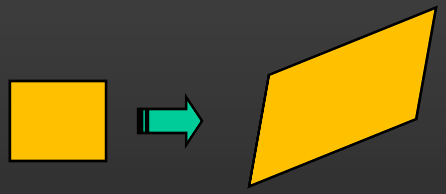
- *Hypothesize* transformation *T* (small group of putative matches that are related by *T*)
- *Verify* transformation (search for other matches consistent with *T*)

2D transformation models

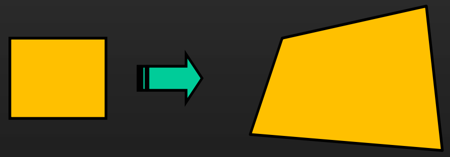
Similarity
(translation,
scale, rotation)



Affine



Projective
(homography)

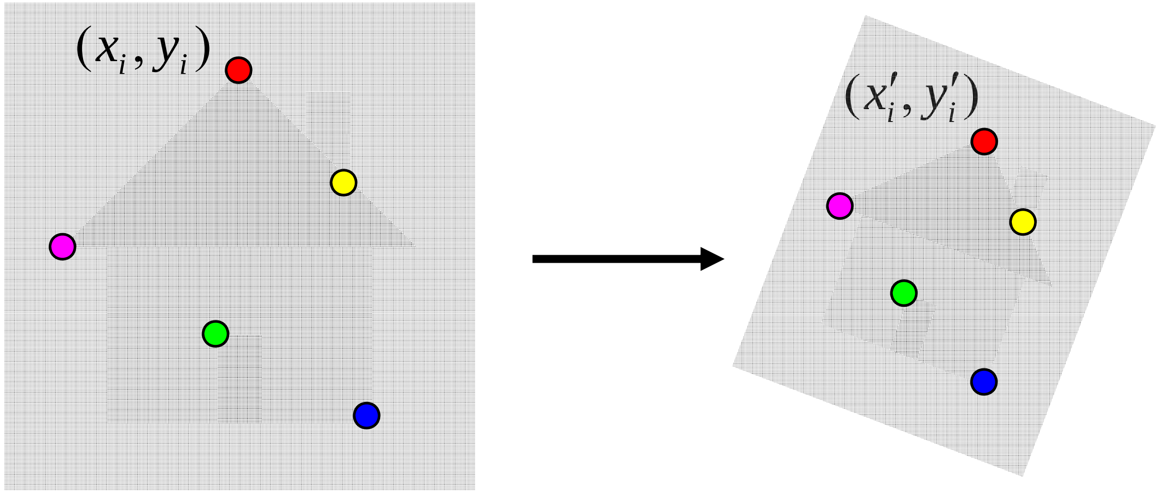


Let us start with affine transformations

- Simple fitting procedure (linear least squares)
- Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras
- Can be used to initialize fitting for more complex models

Fitting an affine transformation

Assume we know the correspondences, how do we get the transformation?



$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

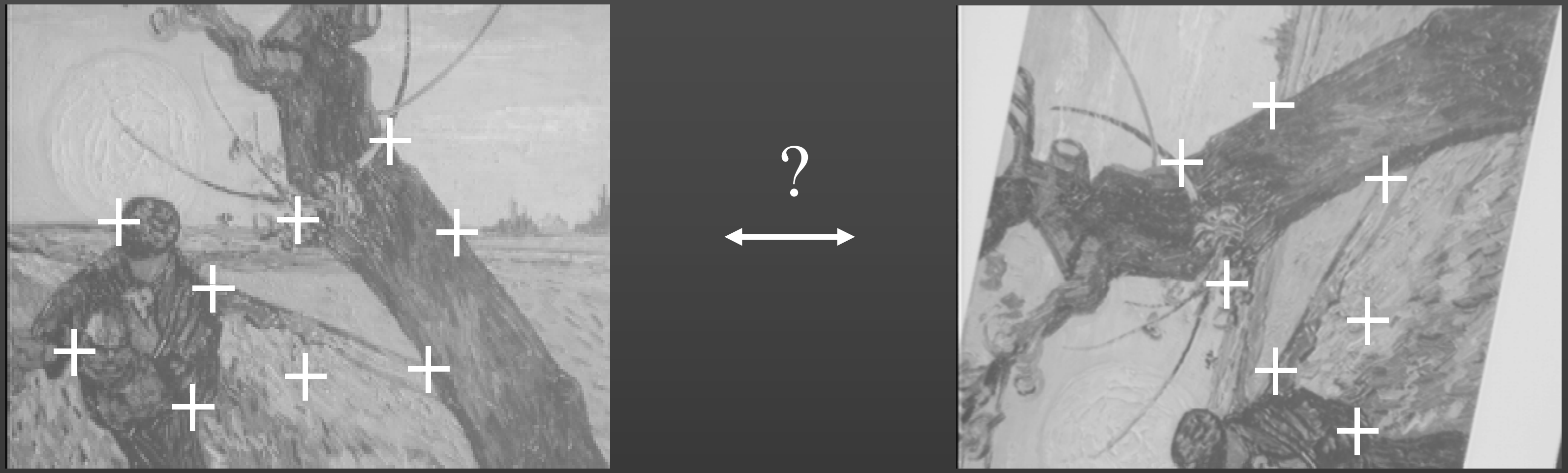
Fitting an affine transformation

$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

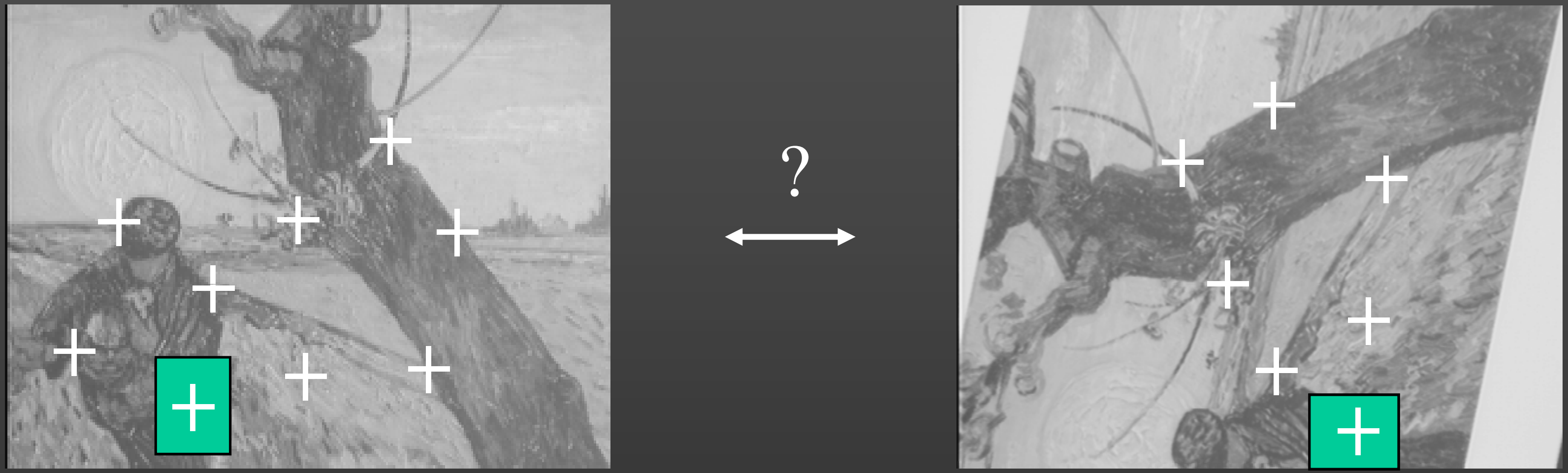
Linear system with six unknowns

Each match gives us two linearly independent equations: need at least three to solve for the transformation parameters

What if we don't know the correspondences?



What if we don't know the correspondences?



- It would help to be able to compare *descriptors* of local patches surrounding interest points (cf next lecture).
- This is not strictly necessary. We will concentrate here on the geometry of the problem.

Dealing with outliers

The set of putative matches still contains a very high percentage of outliers

How do we fit a geometric transformation to a small subset of all possible matches?

Possible strategies:

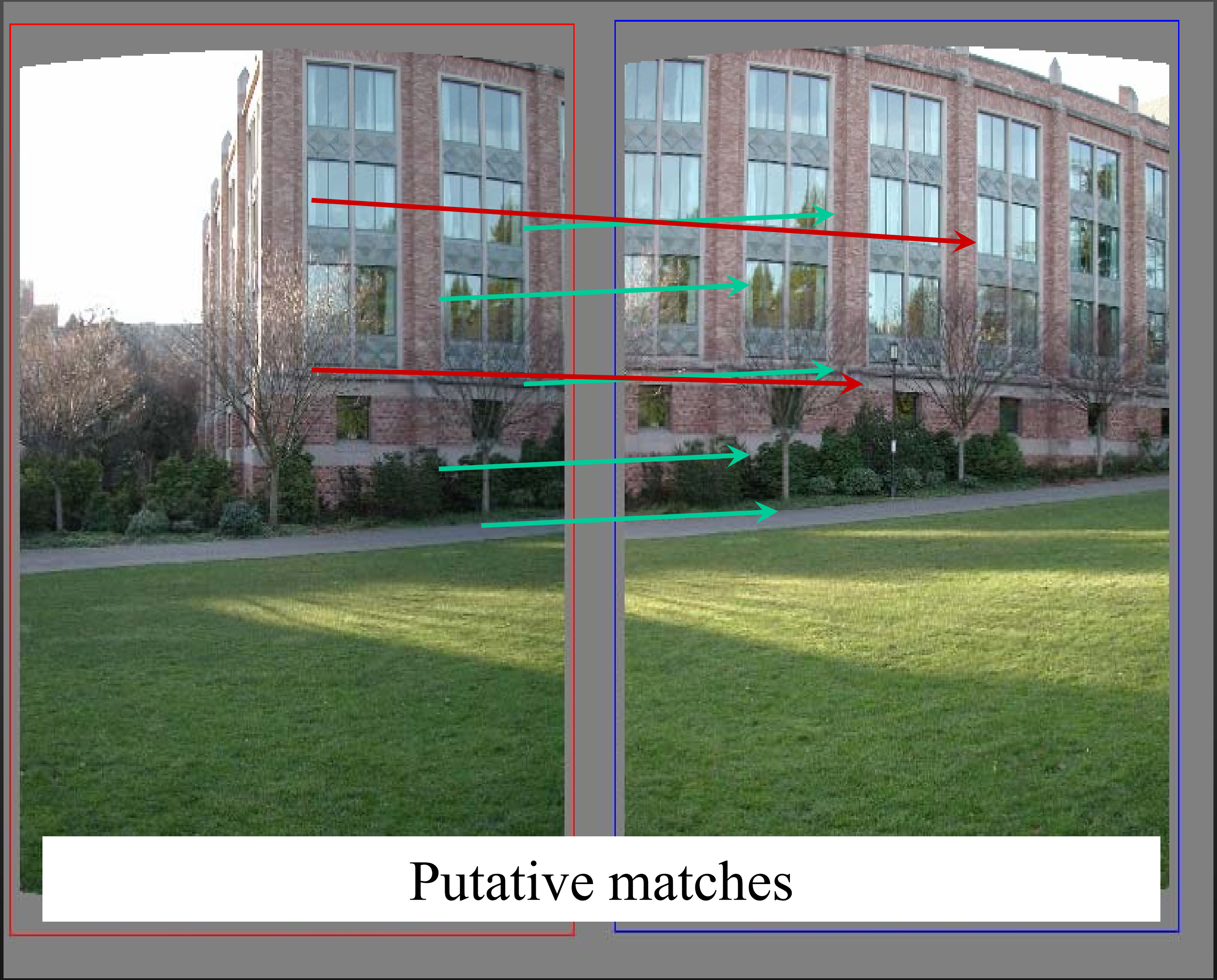
- RANSAC
- Incremental alignment
- Hough transform
- Hashing

Strategy 1: RANSAC

RANSAC loop (Fischler & Bolles, 1981):

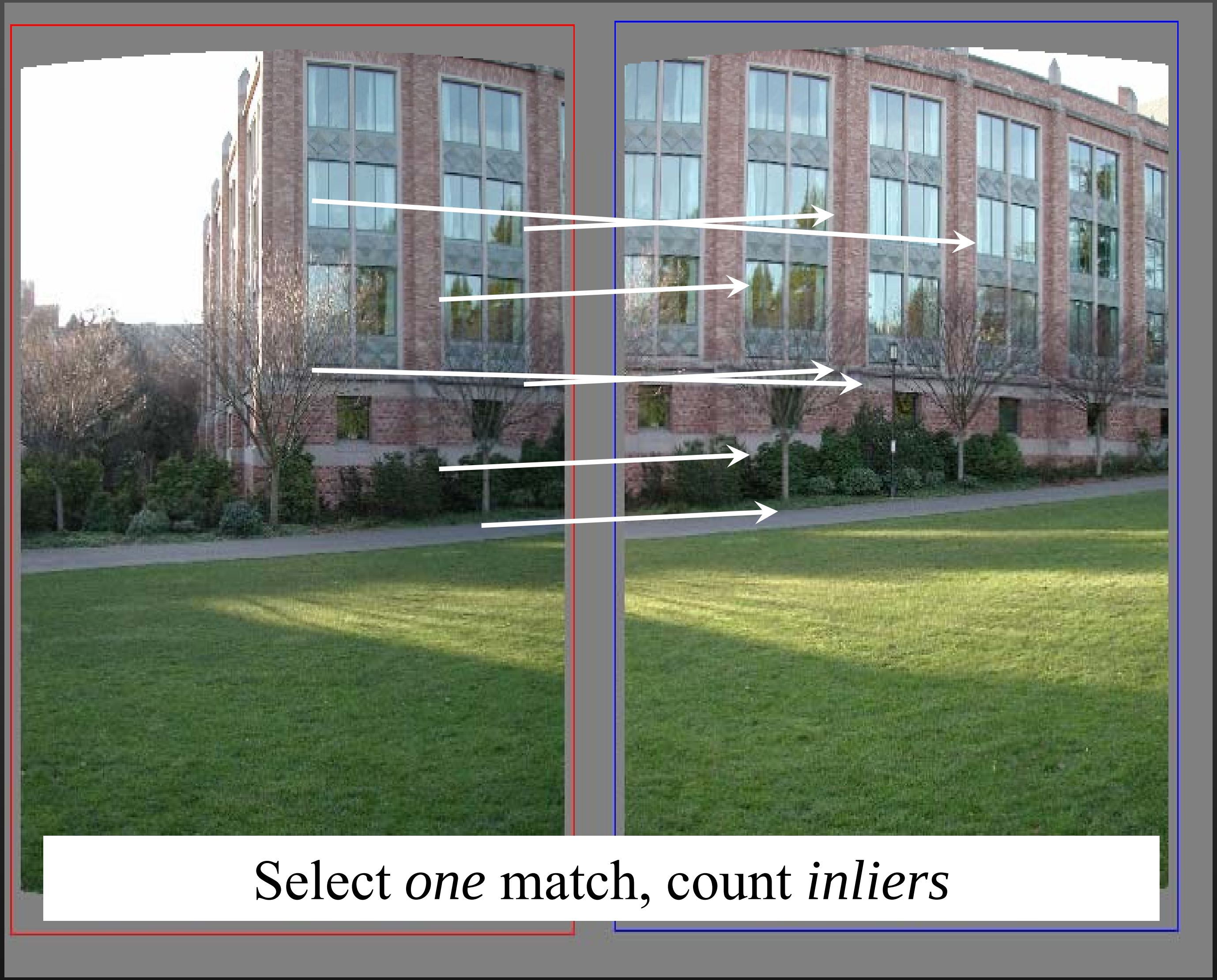
- Randomly select a *seed group* of matches
- Compute transformation from seed group
- Find *inliers* to this transformation
- If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers
- Keep the transformation with the largest number of inliers

RANSAC example: Translation



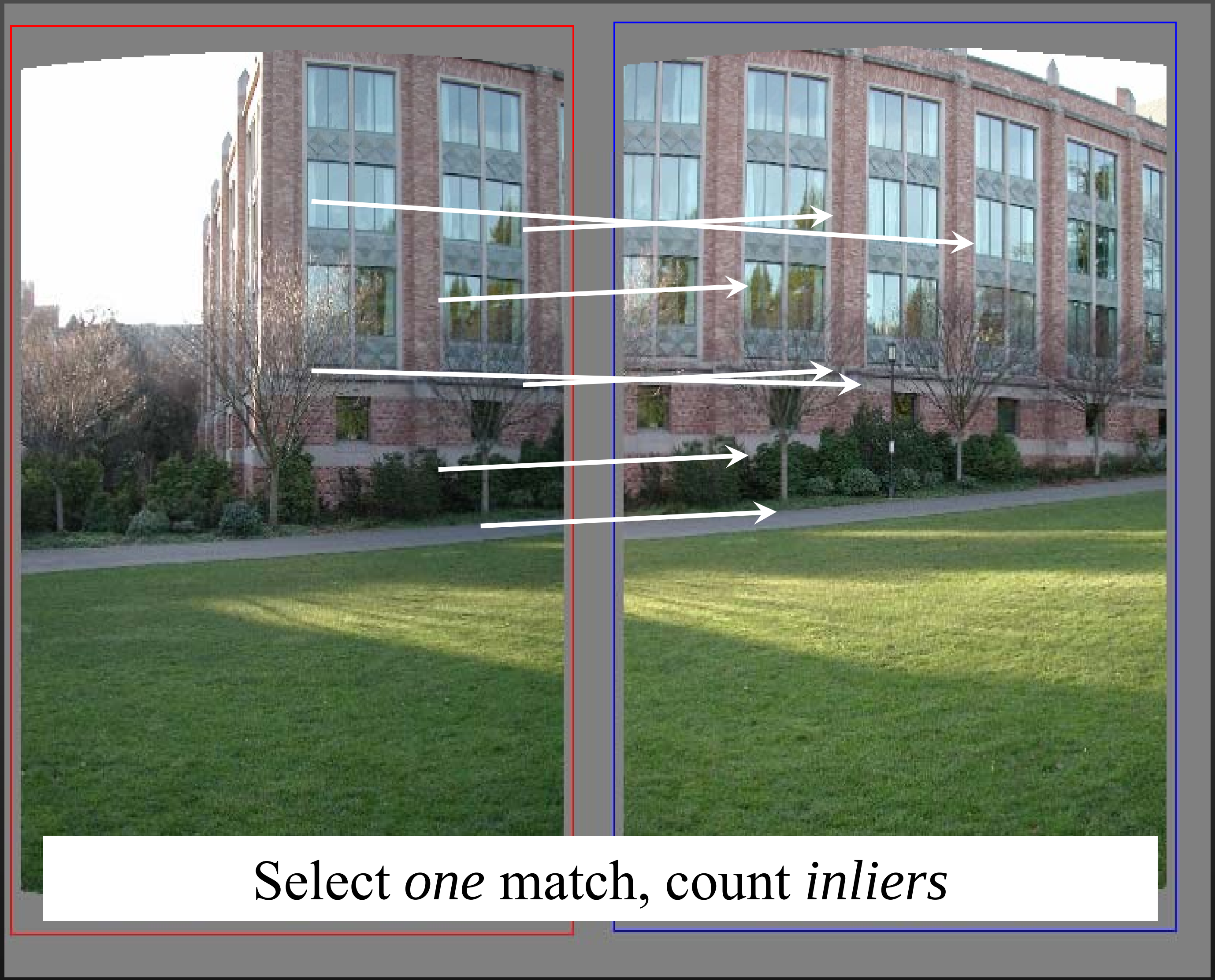
Putative matches

RANSAC example: Translation



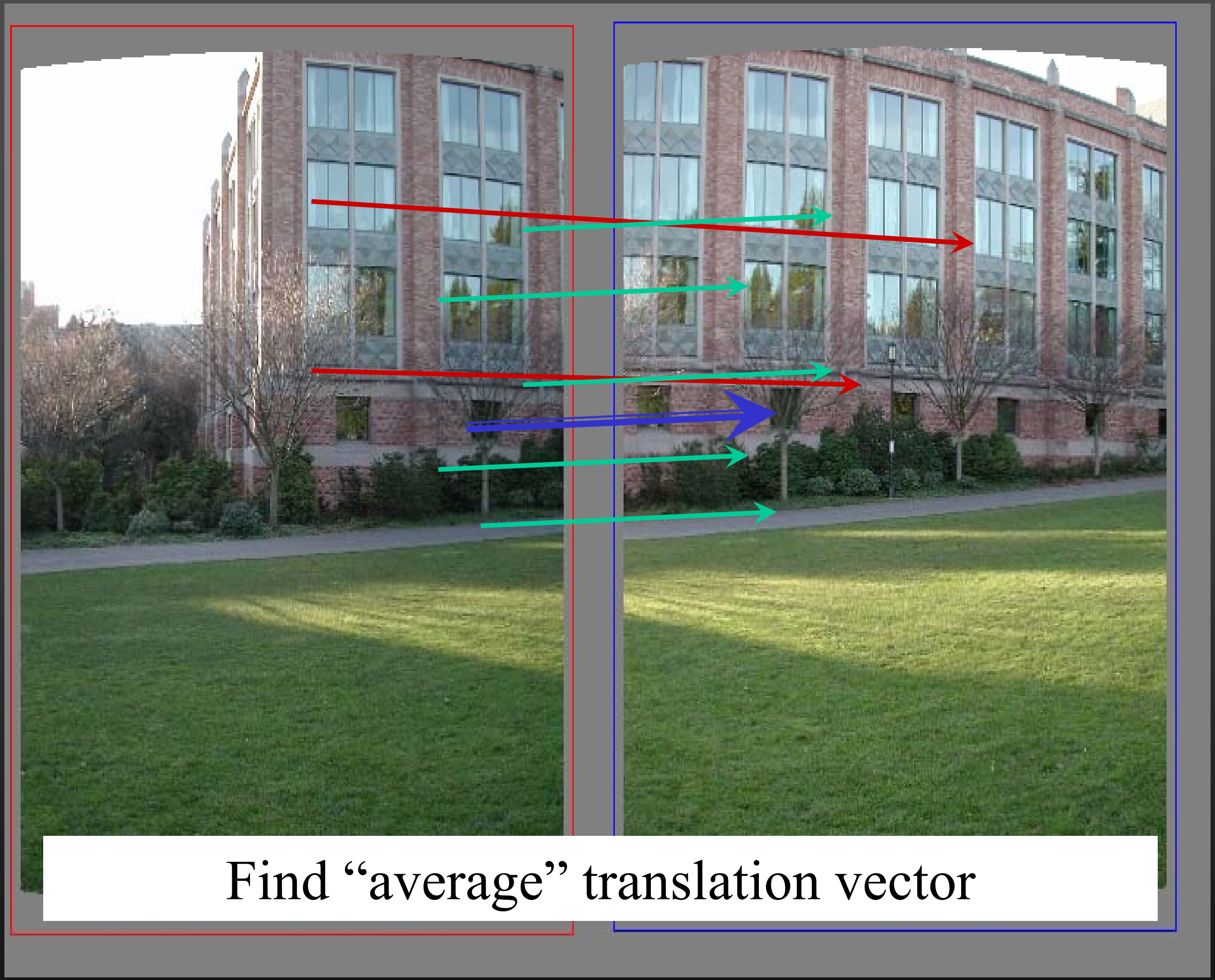
Select *one* match, count *inliers*

RANSAC example: Translation



Select *one* match, count *inliers*

RANSAC example: Translation



Find “average” translation vector

Problem with RANSAC

In many practical situations, the percentage of outliers (incorrect putative matches) is very high (90% or above)

Alternative strategy: restrict search space by using strong locality constraints on seed groups and inliers

➡ Incremental alignment

Strategy 2: Incremental alignment

Take advantage of strong locality constraints: only pick close-by matches to start with, and gradually add more matches in the same neighborhood

Approach introduced in [Ayache & Faugeras, 1982; Hebert & Faugeras, 1983; Gaston & Lozano-Perez, 1984]

Illustrated here with the method from S. Lazebnik, C. Schmid and J. Ponce, “Semi-local affine parts for object recognition”, BMVC 2004

Strategy 2: Incremental alignment

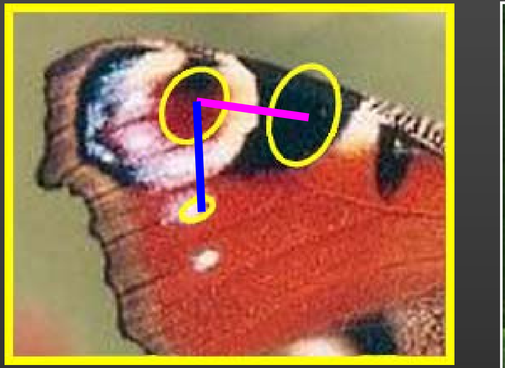
Take advantage of strong locality constraints: only pick close-by matches to start with, and gradually add more matches in the same neighborhood

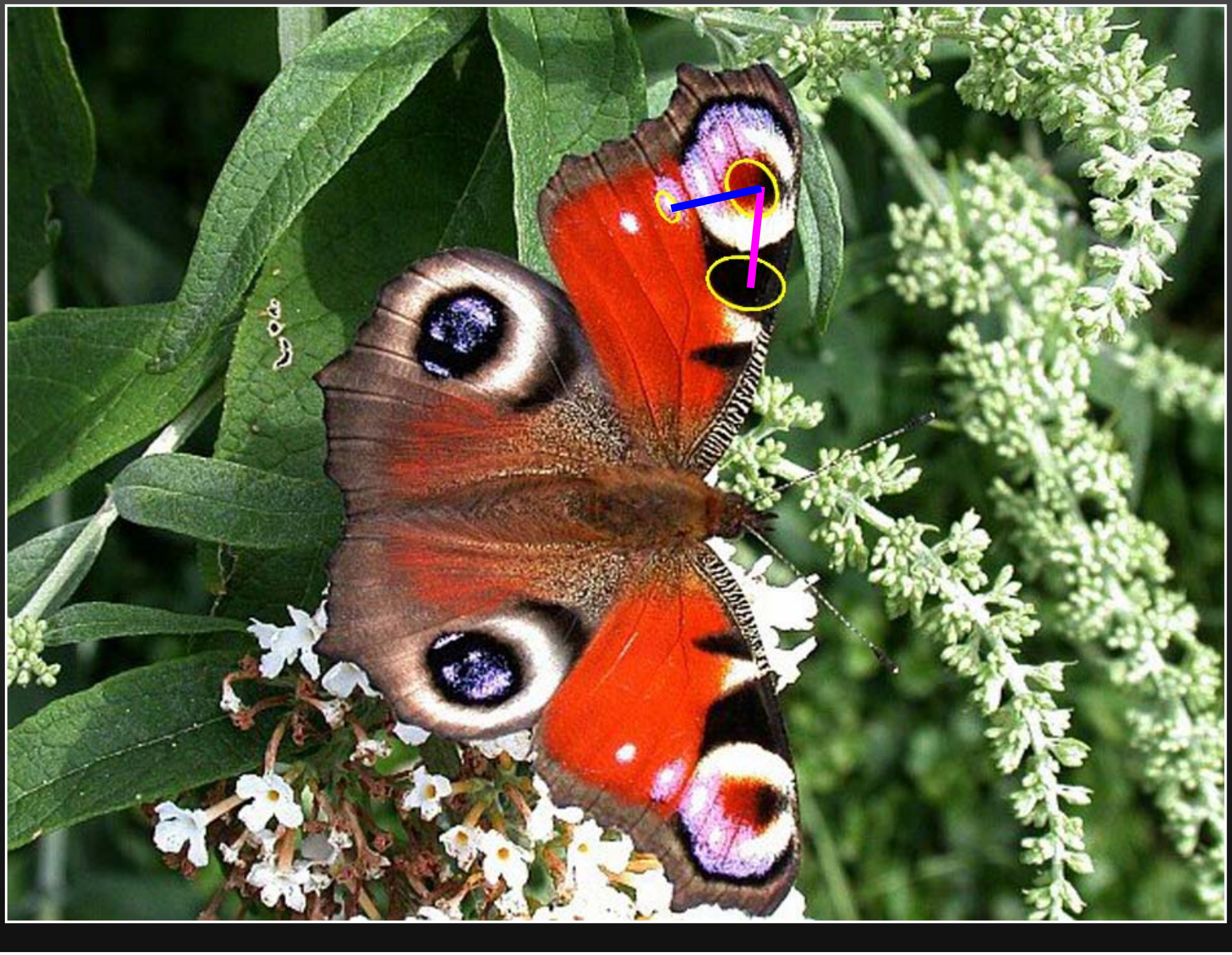
Strategy 2: Incremental alignment

Take advantage of strong locality constraints: only pick close-by matches to start with, and gradually add more matches in the same neighborhood

Strategy 2: Incremental alignment

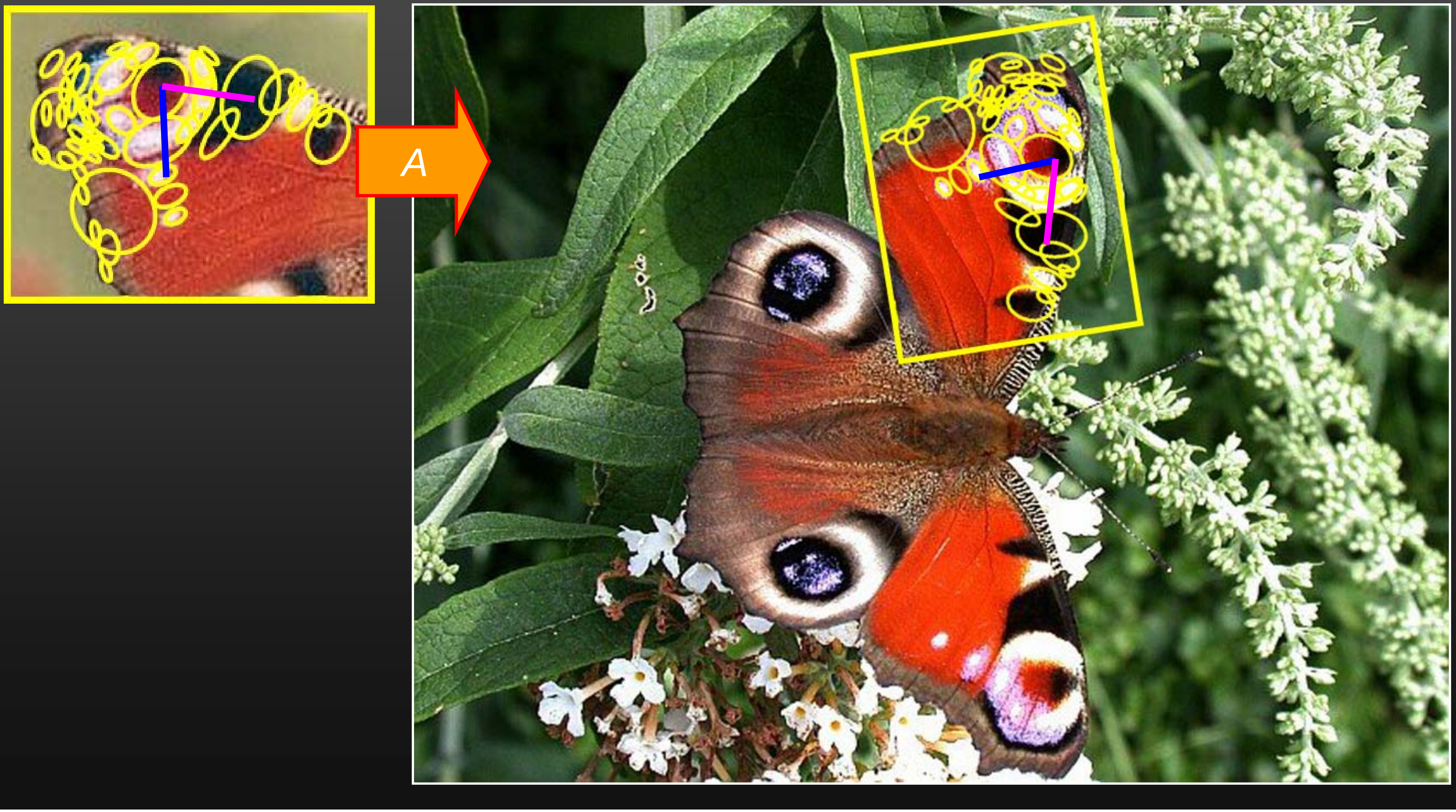
Take advantage of strong locality constraints: only pick close-by matches to start with, and gradually add more matches in the same neighborhood



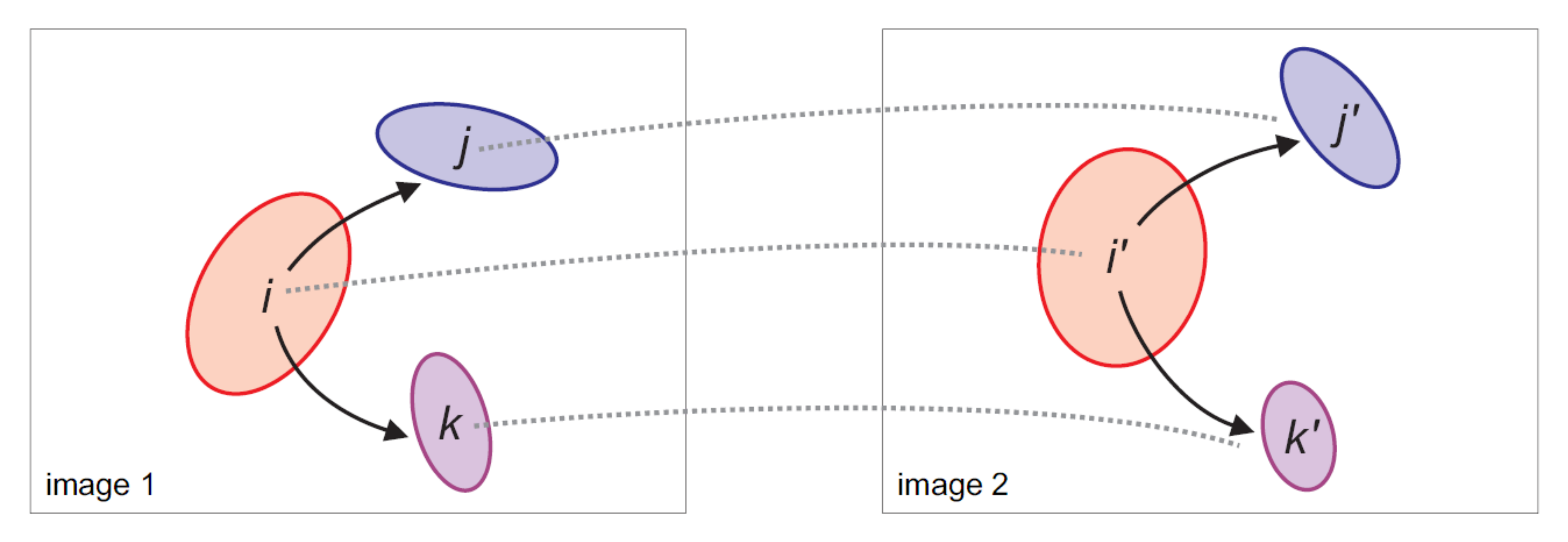


Strategy 2: Incremental alignment

Take advantage of strong locality constraints: only pick close-by matches to start with, and gradually add more matches in the same neighborhood



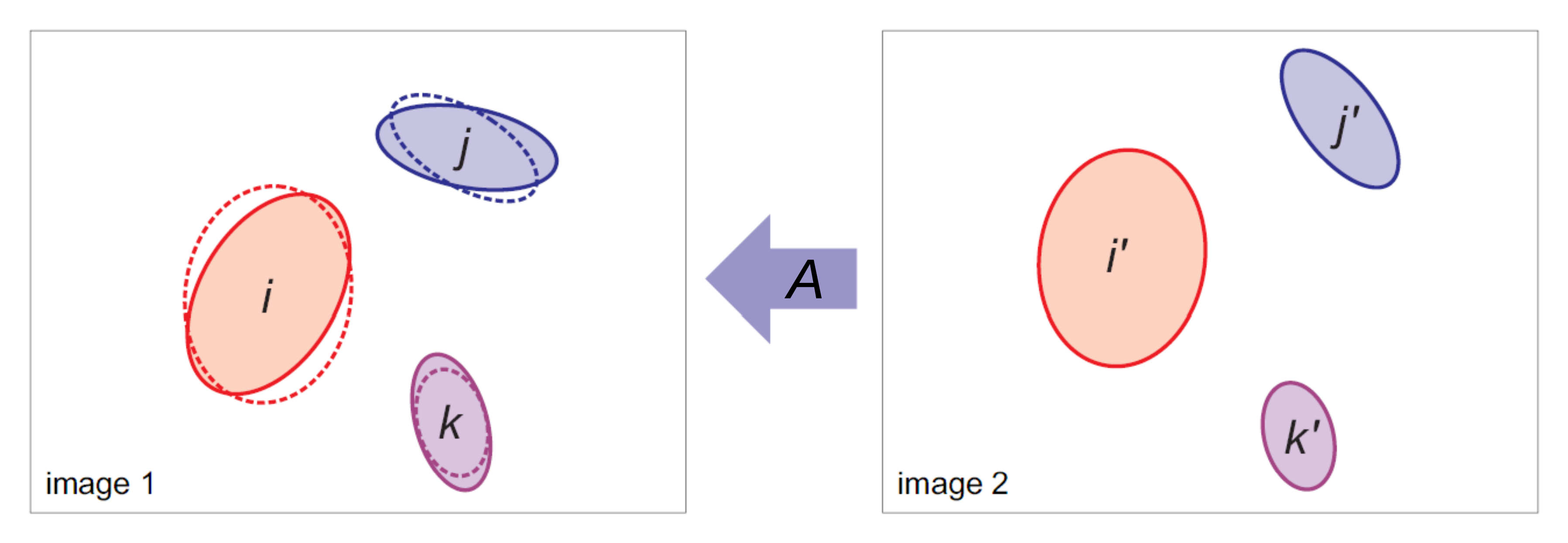
Incremental alignment: Details



Generating seed groups:

- Identify triples of neighboring features (*i*, *j*, *k*) in first image
- Find all triples (*i*′, *j*′, *k*′) in the second image such that *i*′ (resp. *j*′, *k*′) is a putative match of *i* (resp. *j*, *k*), and *j*′, *k*′ are neighbors of *i*′

Incremental alignment: Details

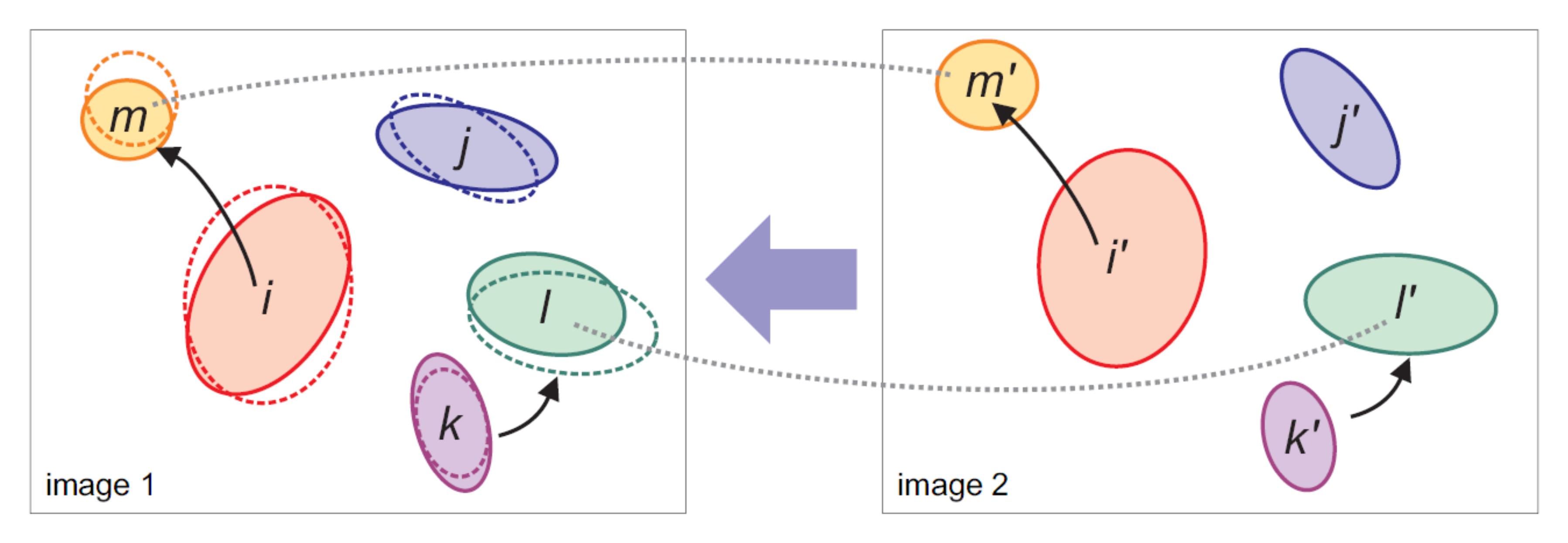


Beginning with each seed triple, repeat:

- Estimate the aligning transformation between corresponding features in current group of matches
- Grow the group by adding other consistent matches in the neighborhood

Until the transformation is no longer consistent or no more matches can be found

Incremental alignment: Details

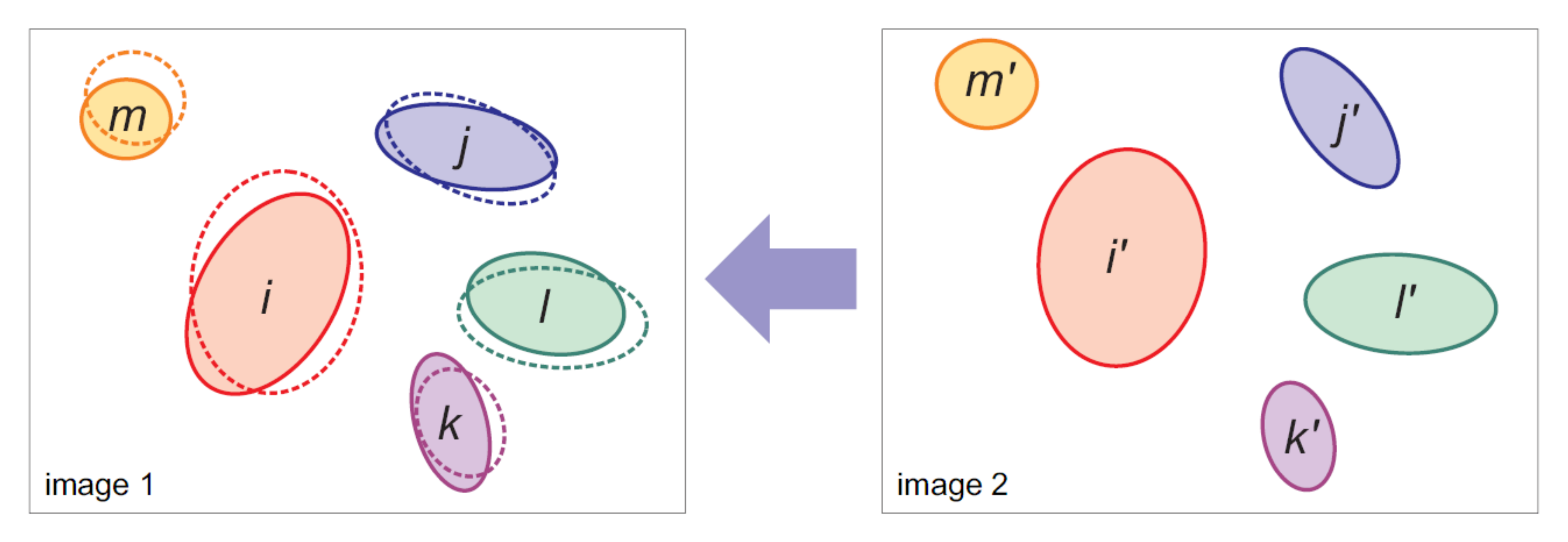


Beginning with each seed triple, repeat:

- Estimate the aligning transformation between corresponding features in current group of matches
- Grow the group by adding other consistent matches in the neighborhood

Until the transformation is no longer consistent or no more matches can be found

Incremental alignment: Details

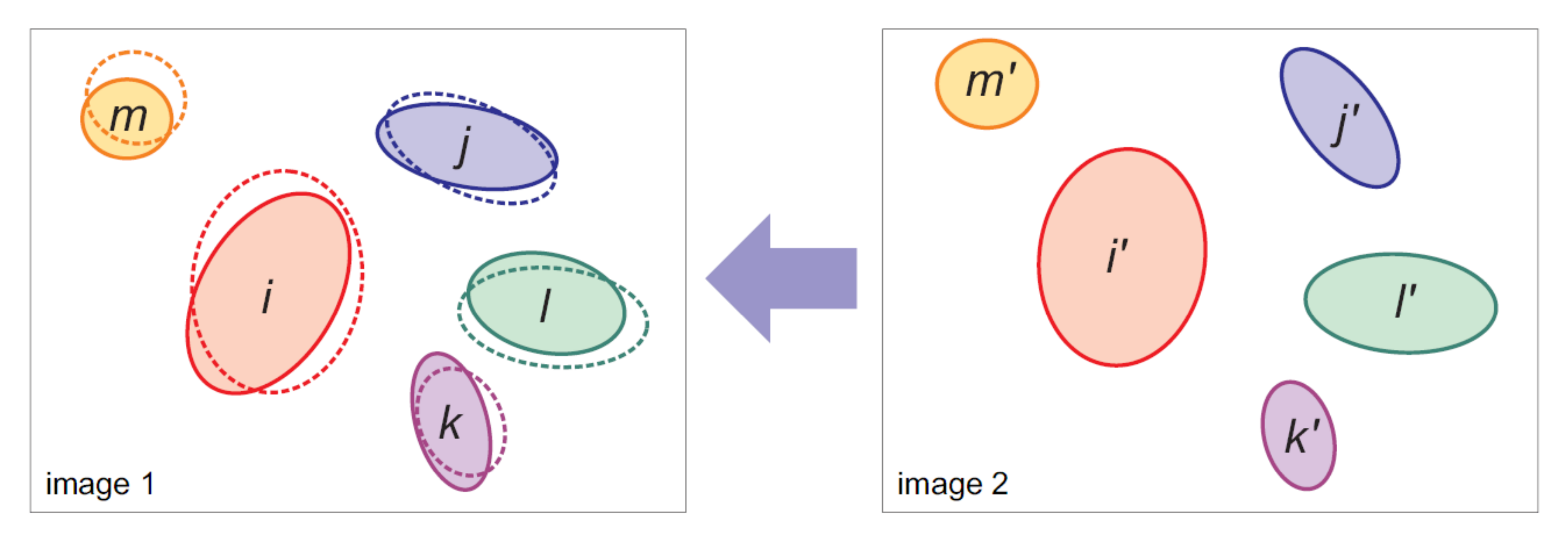


Beginning with each seed triple, repeat:

- Estimate the aligning transformation between corresponding features in current group of matches
- Grow the group by adding other consistent matches in the neighborhood

Until the transformation is no longer consistent or no more matches can be found

Incremental alignment: Details



Beginning with each seed triple, repeat:

- Estimate the aligning transformation between corresponding features in current group of matches
- Grow the group by adding other consistent matches in the neighborhood

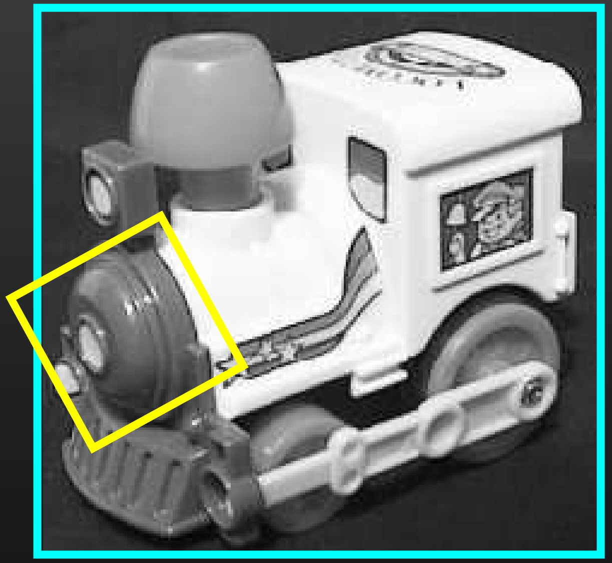
Until the transformation is no longer consistent or no more matches can be found

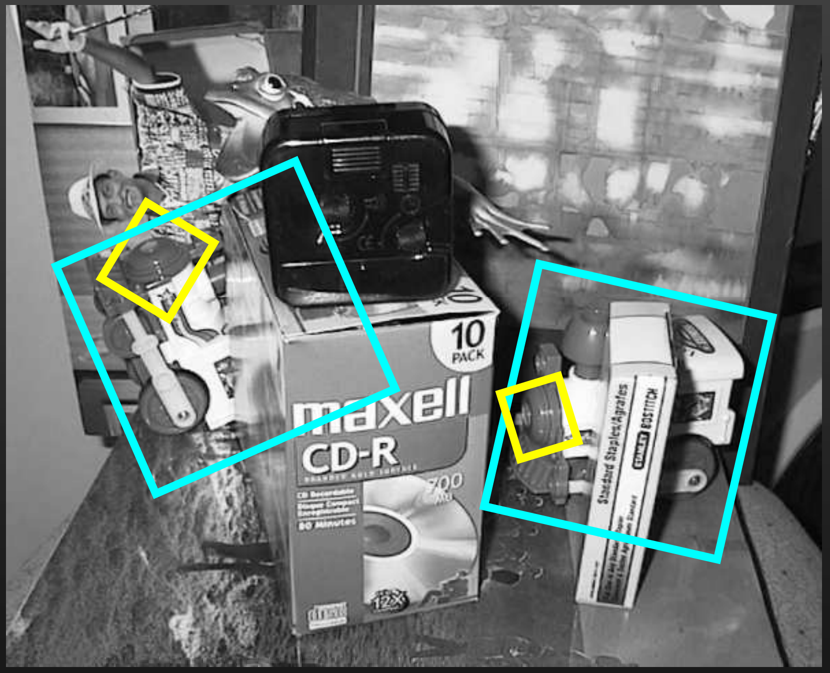
Strategy 3: Hough transform

Suppose our features are scale- and rotation-covariant

- Then a single feature match provides an alignment hypothesis (translation, scale, orientation)

model



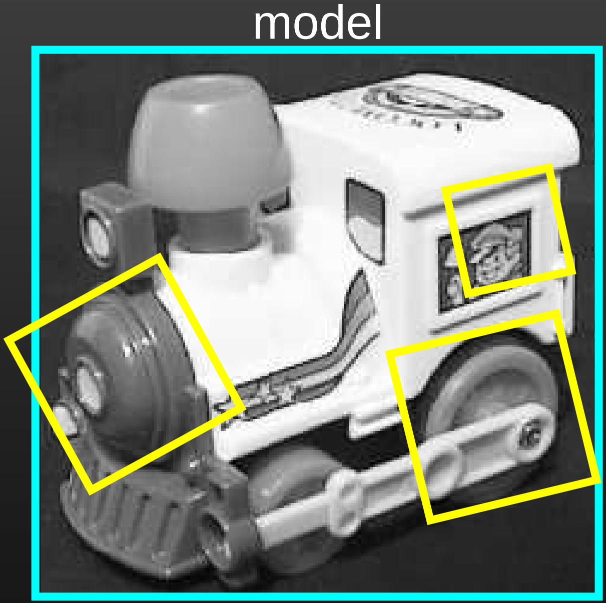


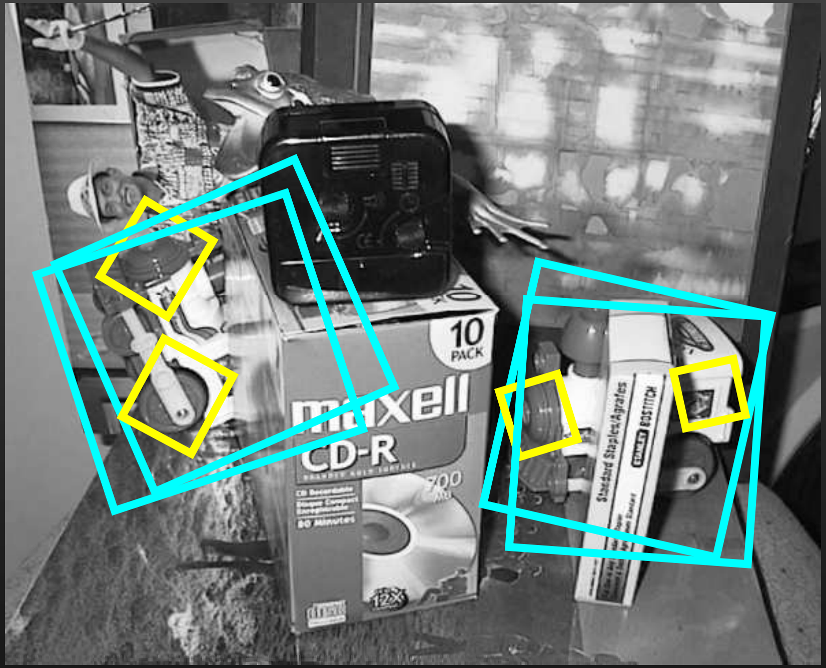
David G. Lowe. **“Distinctive image features from scale-invariant keypoints”,** *IJCV* 60 (2), pp. 91-110, 2004.

Strategy 3: Hough transform

Suppose our features are scale- and rotation-covariant

- Then a single feature match provides an alignment hypothesis (translation, scale, orientation)
- Of course, a hypothesis obtained from a single match is unreliable
- Solution: let each match vote for its hypothesis in a Hough space with very coarse bins





David G. Lowe. **“Distinctive image features from scale-invariant keypoints”,** *IJCV* 60 (2), pp. 91-110, 2004.

Hough transform details (D. Lowe's system)

Training phase: For each model feature, record 2D location, scale, and orientation of model (relative to normalized feature frame)

Test phase: Let each match between a test and a model feature vote in a 4D Hough space

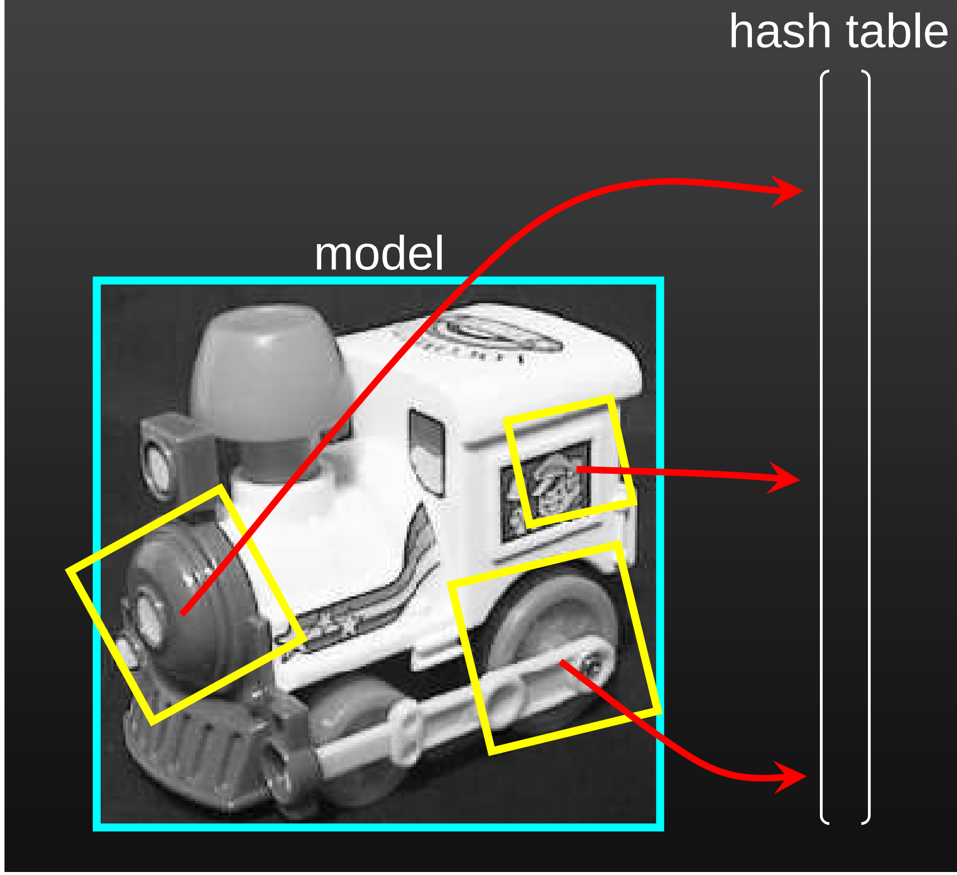
- Use broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times image size for location
- Vote for two closest bins in each dimension

Find all bins with at least three votes and perform geometric verification

- Estimate least squares *affine* transformation
- Use stricter thresholds on transformation residual
- Search for additional features that agree with the alignment

Strategy 4: Hashing

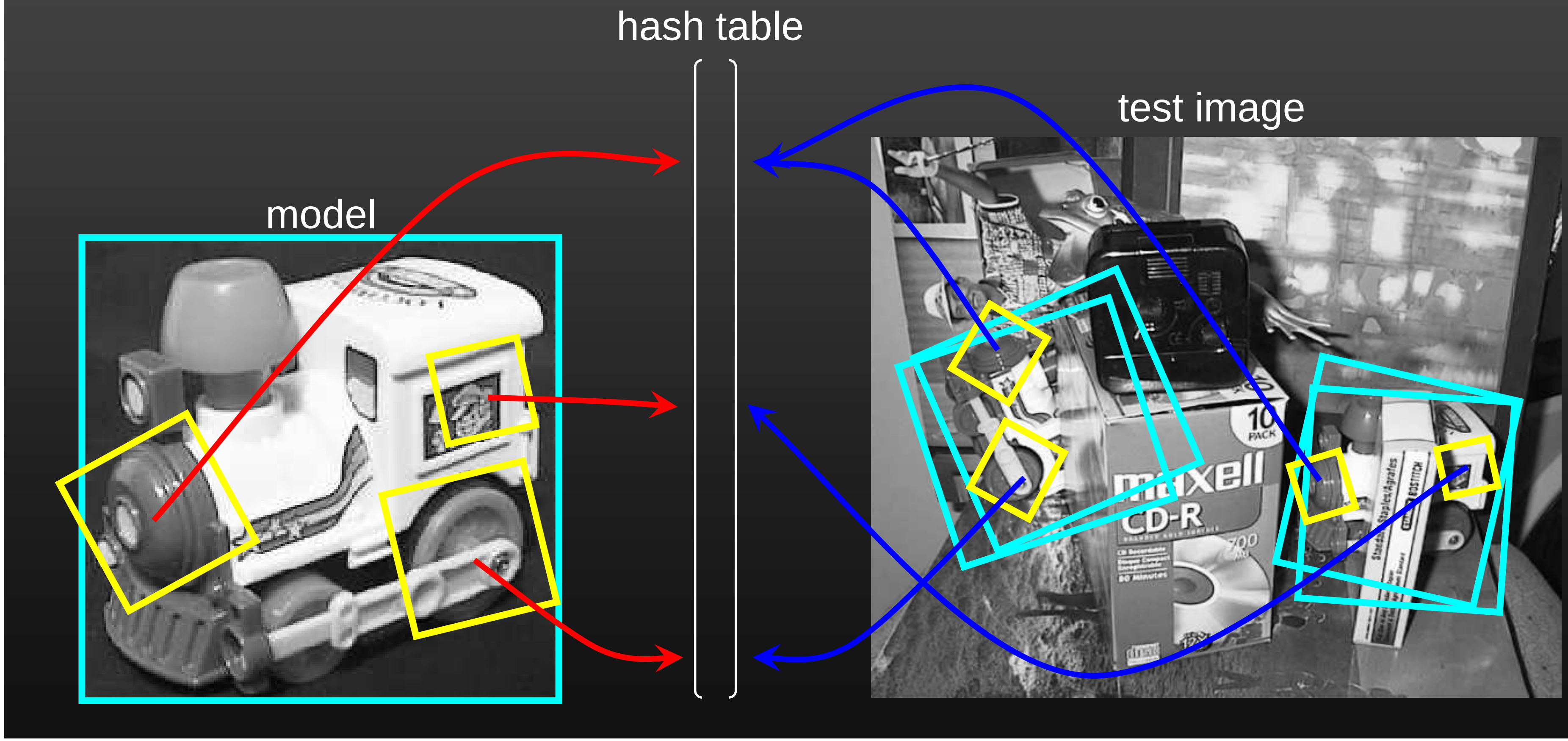
Make each invariant image feature into a low-dimensional "key" that indexes into a table of hypotheses



Strategy 4: Hashing

Make each invariant image feature into a low-dimensional “key” that indexes into a table of hypotheses

Given a new test image, compute the hash keys for all features found in that image, access the table, and look for consistent hypotheses

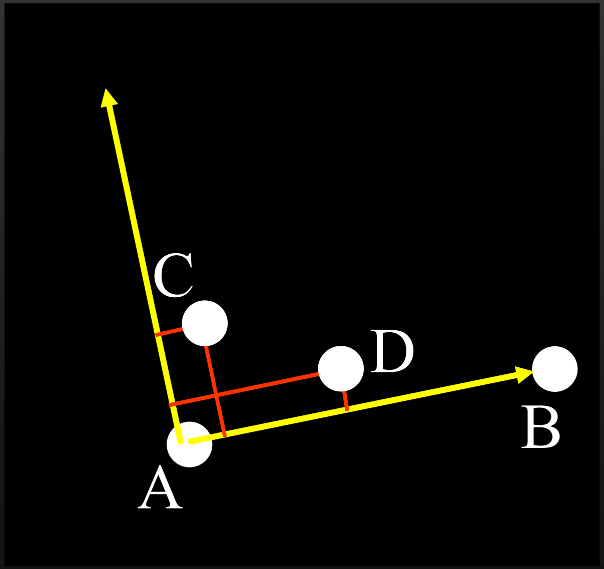


Strategy 4: Hashing

Make each invariant image feature into a low-dimensional “key” that indexes into a table of hypotheses

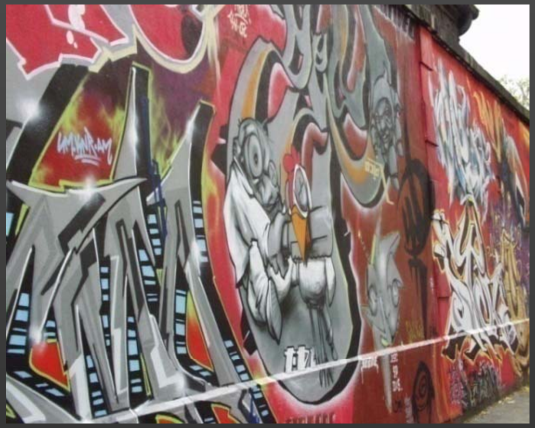
Given a new test image, compute the hash keys for all features found in that image, access the table, and look for consistent hypotheses

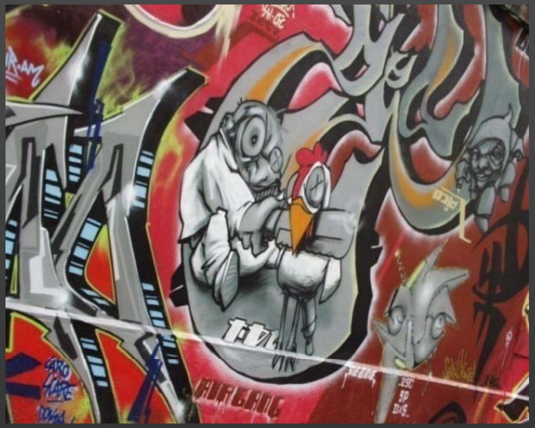
This can even work when we don’t have any feature descriptors: we can take n-tuples of neighboring features and compute invariant hash codes from their geometric configurations



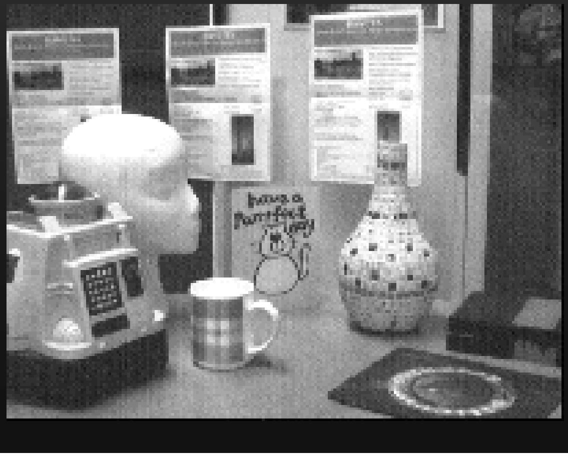
Beyond affine transformations

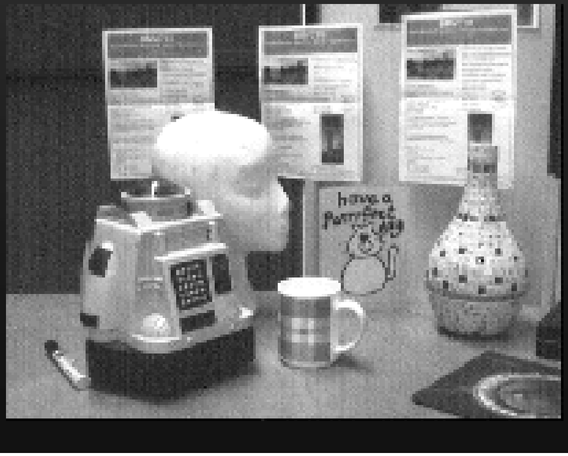
What is the transformation between two views of a planar surface?





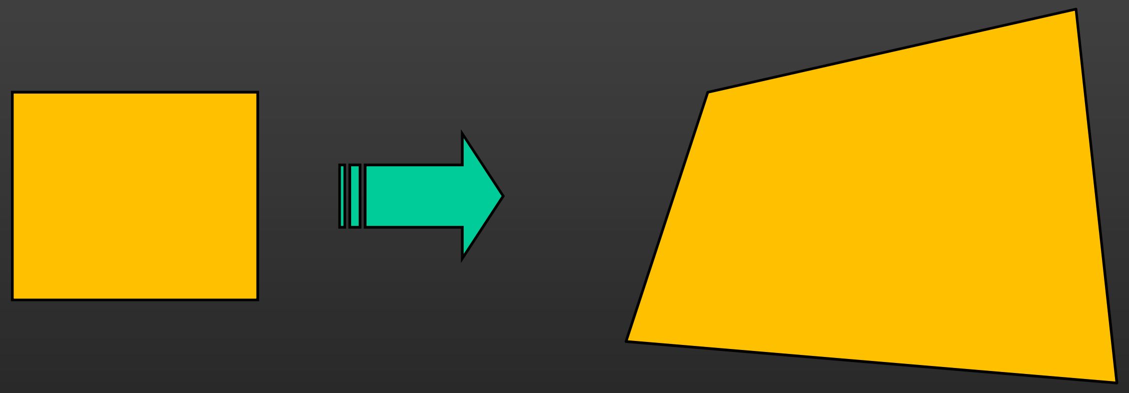
What is the transformation between images from two cameras that share the same center?





Beyond affine transformations

Homography: plane projective transformation (transformation taking a quad to another arbitrary quad)



Fitting a homography

Recall: homogenenous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Converting *to* homogenenous image coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

Converting *from* homogenenous image coordinates

Fitting a homography

Recall: homogenenous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Converting *to* homogenenous image coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

Converting *from* homogenenous image coordinates

Equation for homography:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Fitting a homography

Equation for homography:

$$\lambda \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$\lambda \mathbf{x}_i' = \mathbf{H}\,\mathbf{x}_i = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{bmatrix} \mathbf{x}_i$$

9 entries, 8 degrees of freedom
(scale is arbitrary)

$$\mathbf{x}_i' \times \mathbf{H}\,\mathbf{x}_i = 0$$

$$\mathbf{x}_i' \times \mathbf{H}\,\mathbf{x}_i = \begin{bmatrix} y_i'\,\mathbf{h}_3^T\mathbf{x}_i - \mathbf{h}_2^T\mathbf{x}_i \\ \mathbf{h}_1^T\mathbf{x}_i - x_i'\,\mathbf{h}_3^T\mathbf{x}_i \\ x_i'\,\mathbf{h}_2^T\mathbf{x}_i - y_i'\,\mathbf{h}_1^T\mathbf{x}_i \end{bmatrix}$$

$$\begin{bmatrix} 0^T & -\mathbf{x}_i^T & y_i'\,\mathbf{x}_i^T \\ \mathbf{x}_i^T & 0^T & -x_i'\,\mathbf{x}_i^T \\ -y_i'\,\mathbf{x}_i^T & x_i'\,\mathbf{x}_i^T & 0^T \end{bmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = 0$$

3 equations, only 2 linearly independent

Direct linear transform

$$\begin{bmatrix} 0^T & \mathbf{x}_1^T & -y_1'\,\mathbf{x}_1^T \\ \mathbf{x}_1^T & 0^T & -x_1'\,\mathbf{x}_1^T \\ \cdots & \cdots & \cdots \\ 0^T & \mathbf{x}_n^T & -y_n'\,\mathbf{x}_n^T \\ \mathbf{x}_n^T & 0^T & -x_n'\,\mathbf{x}_n^T \end{bmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = 0$$

$$\mathbf{A}\,\mathbf{h} = 0$$

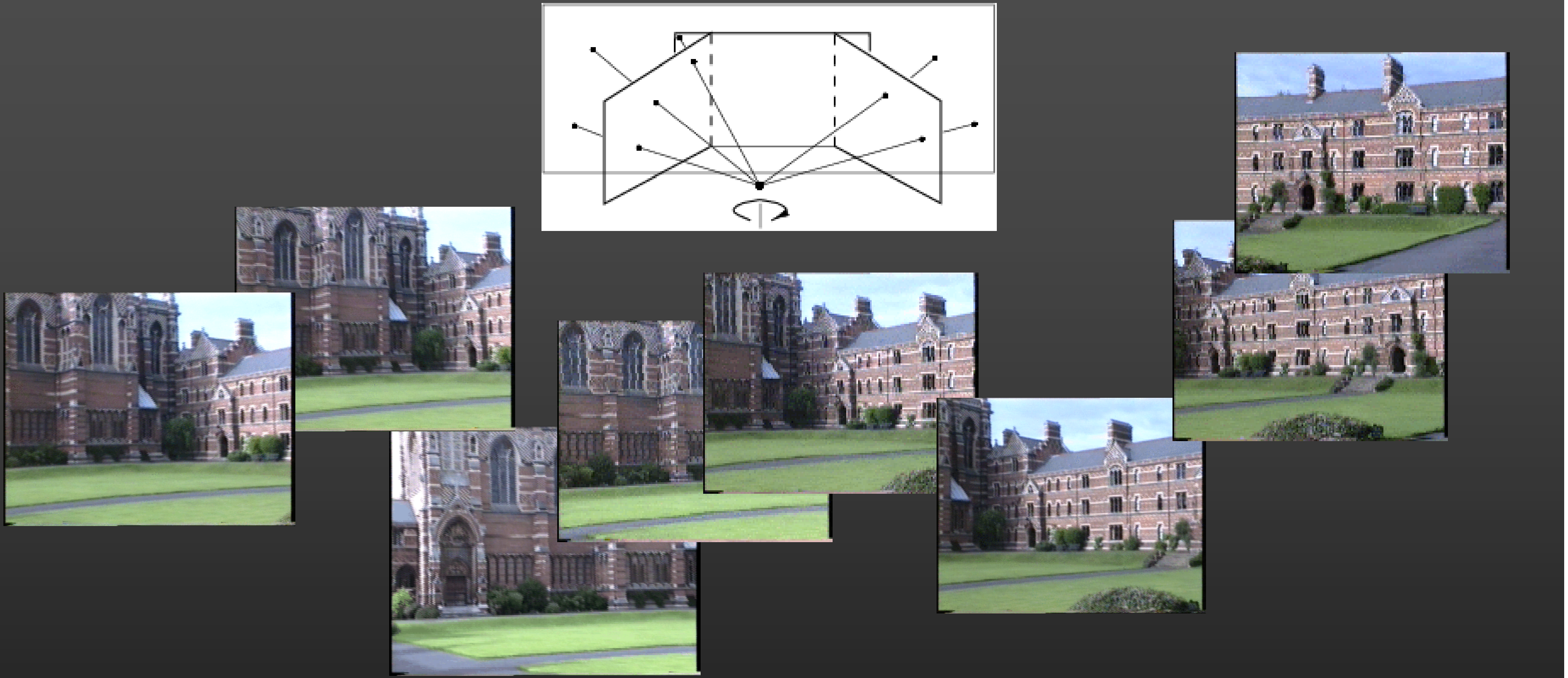
H has 8 degrees of freedom (9 parameters, but scale is arbitrary)

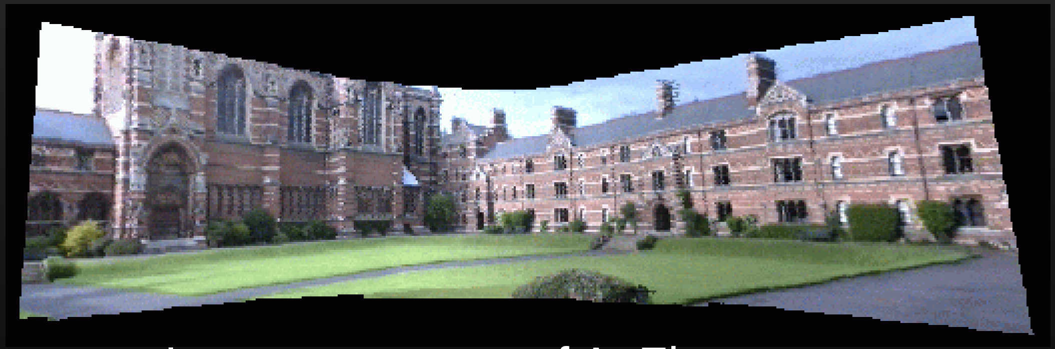
One match gives us two linearly independent equations

Four matches needed for a minimal solution (null space of 8x9 matrix)

More than four: homogeneous least squares

Application: Panorama stitching

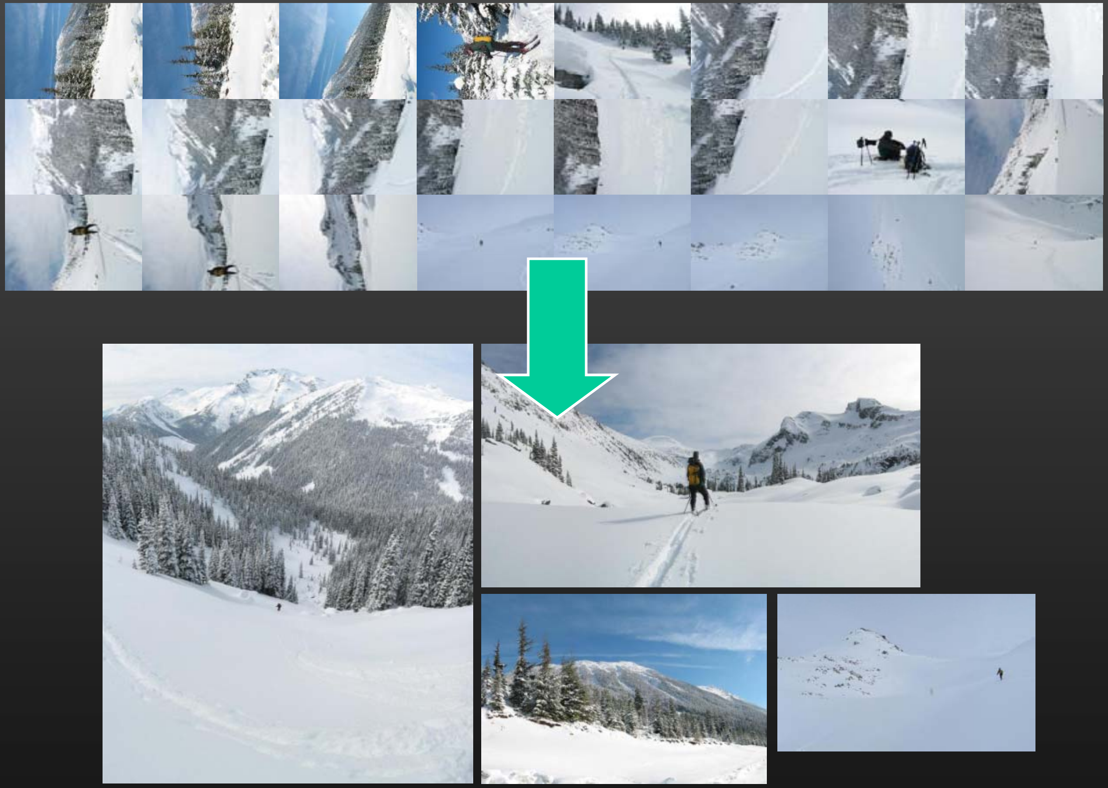




Images courtesy of A. Zisserman.

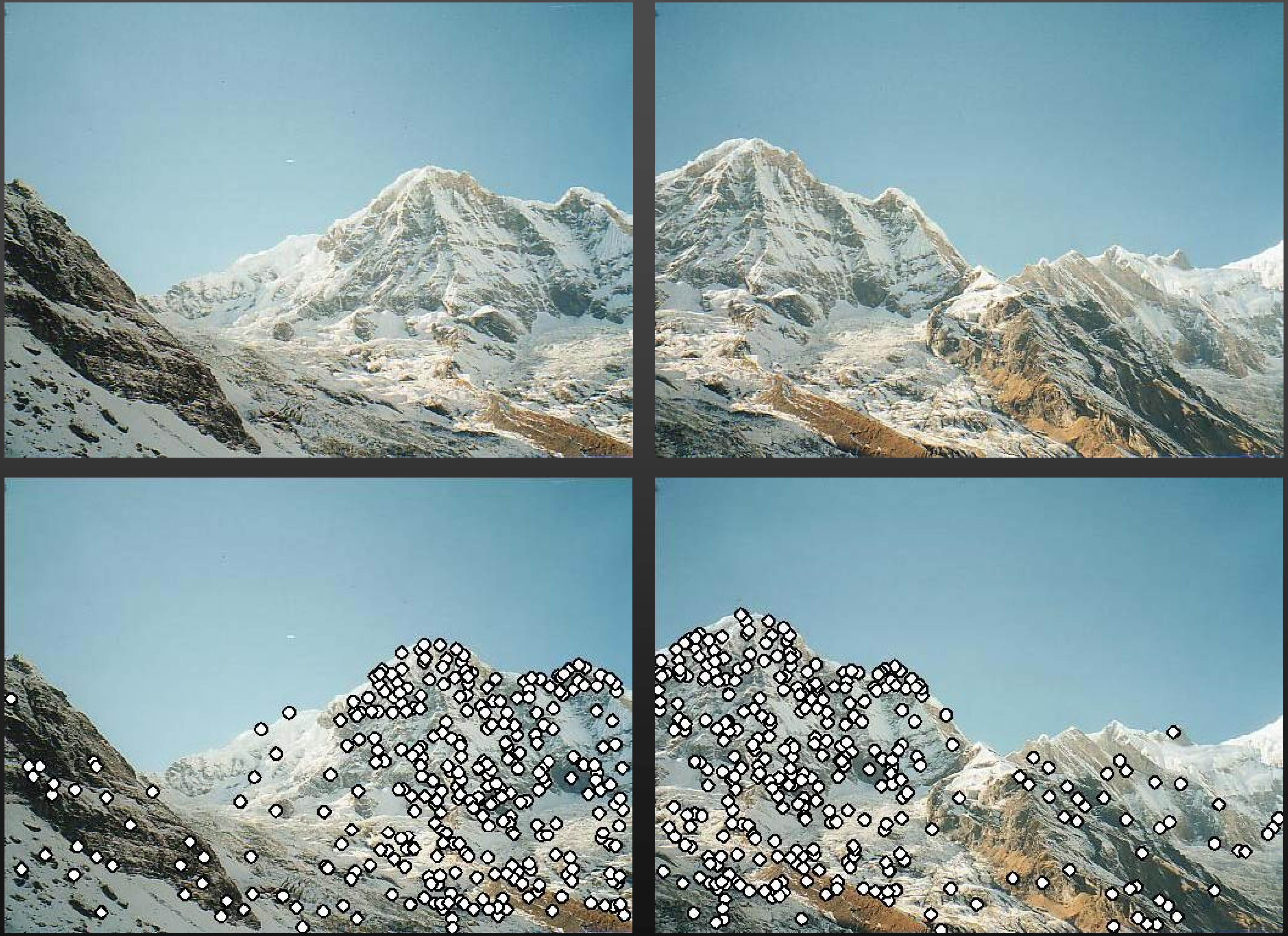
Recognizing panoramas

Given contents of a camera memory card, automatically figure out which pictures go together and stitch them together into panoramas

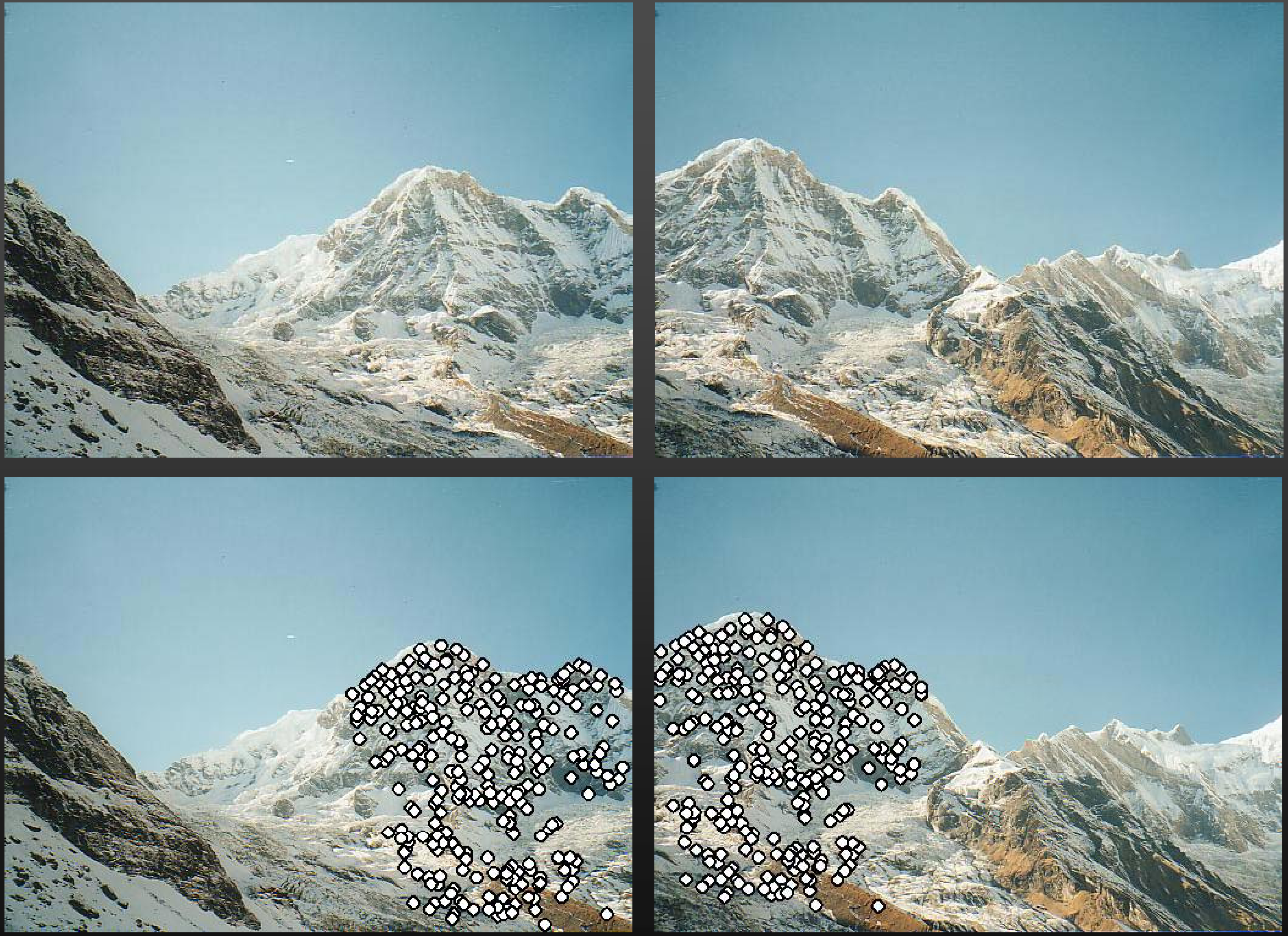


M. Brown and D. Lowe, “Recognizing panoramas”, ICCV 2003.

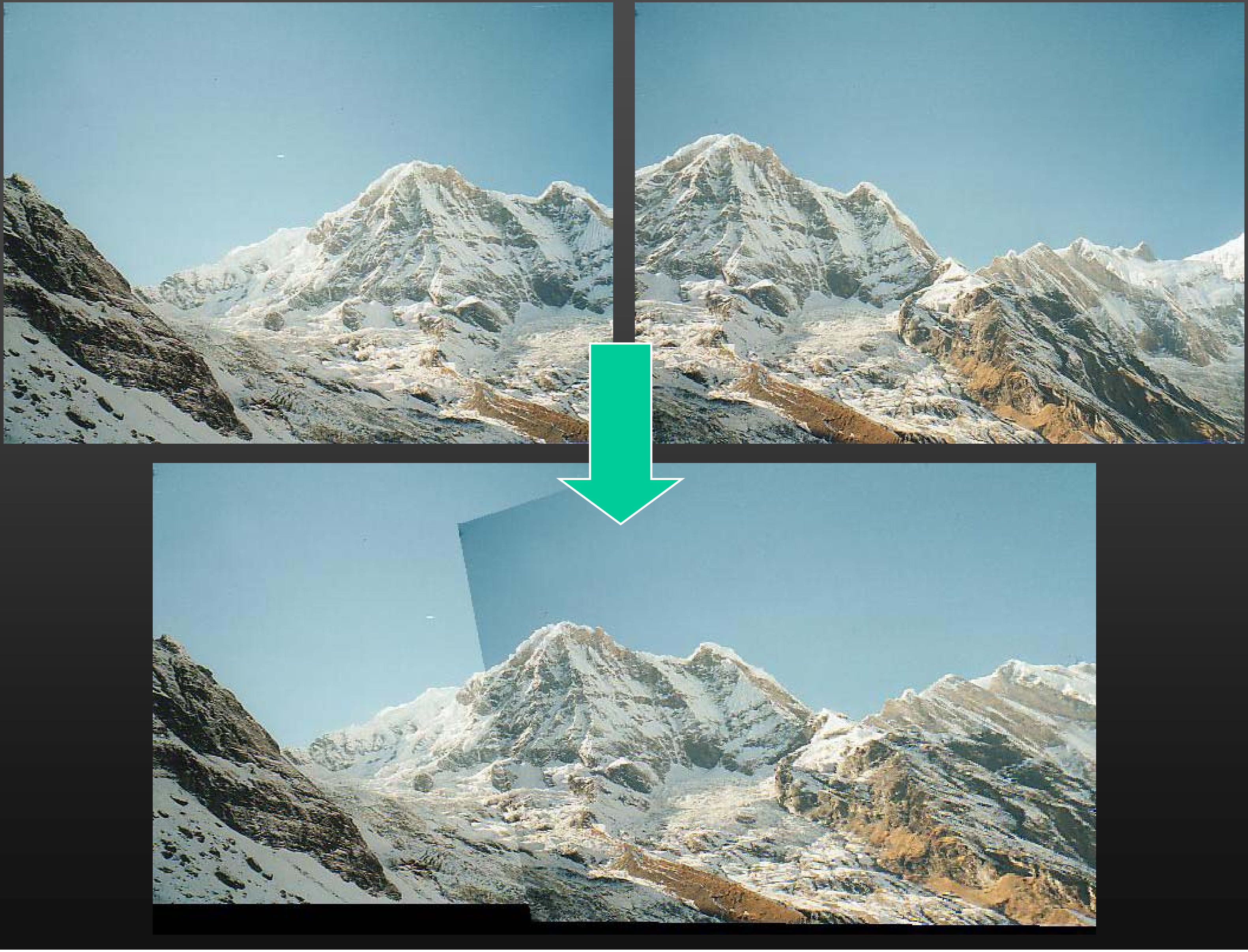
1. Estimate homography (RANSAC)



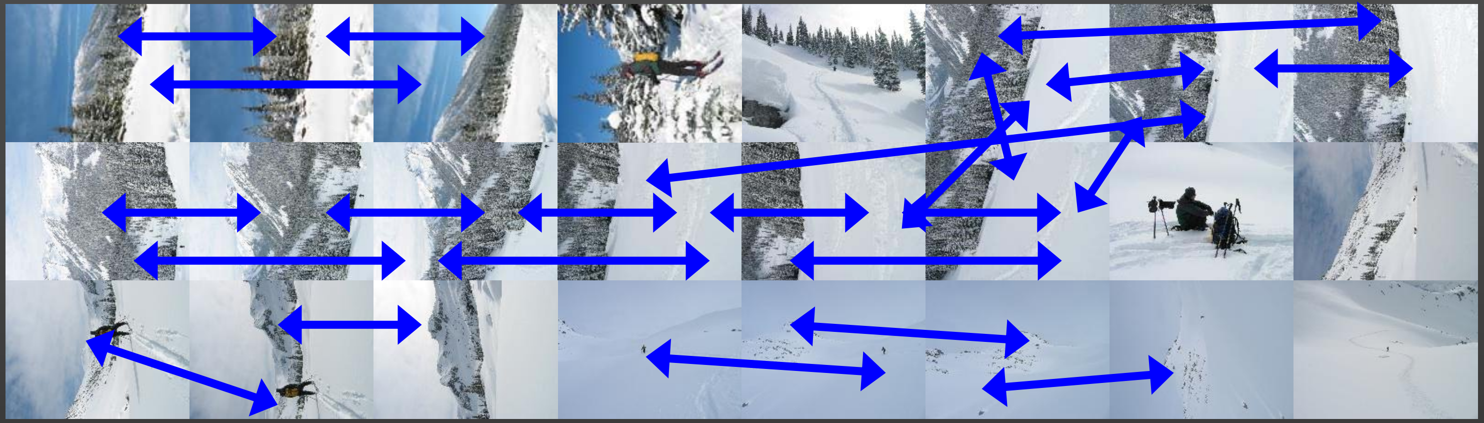
1. Estimate homography (RANSAC)



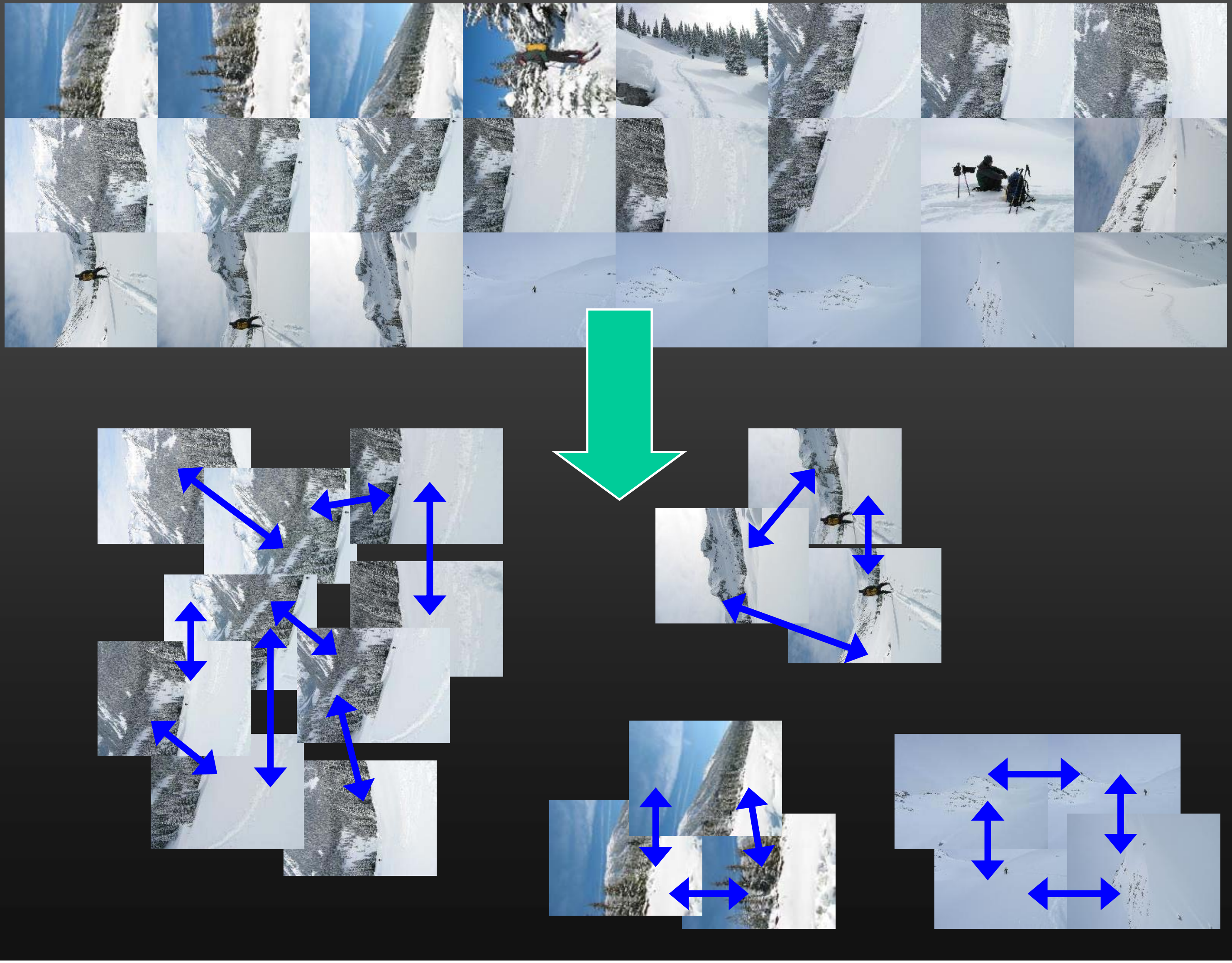
1. Estimate homography (RANSAC)



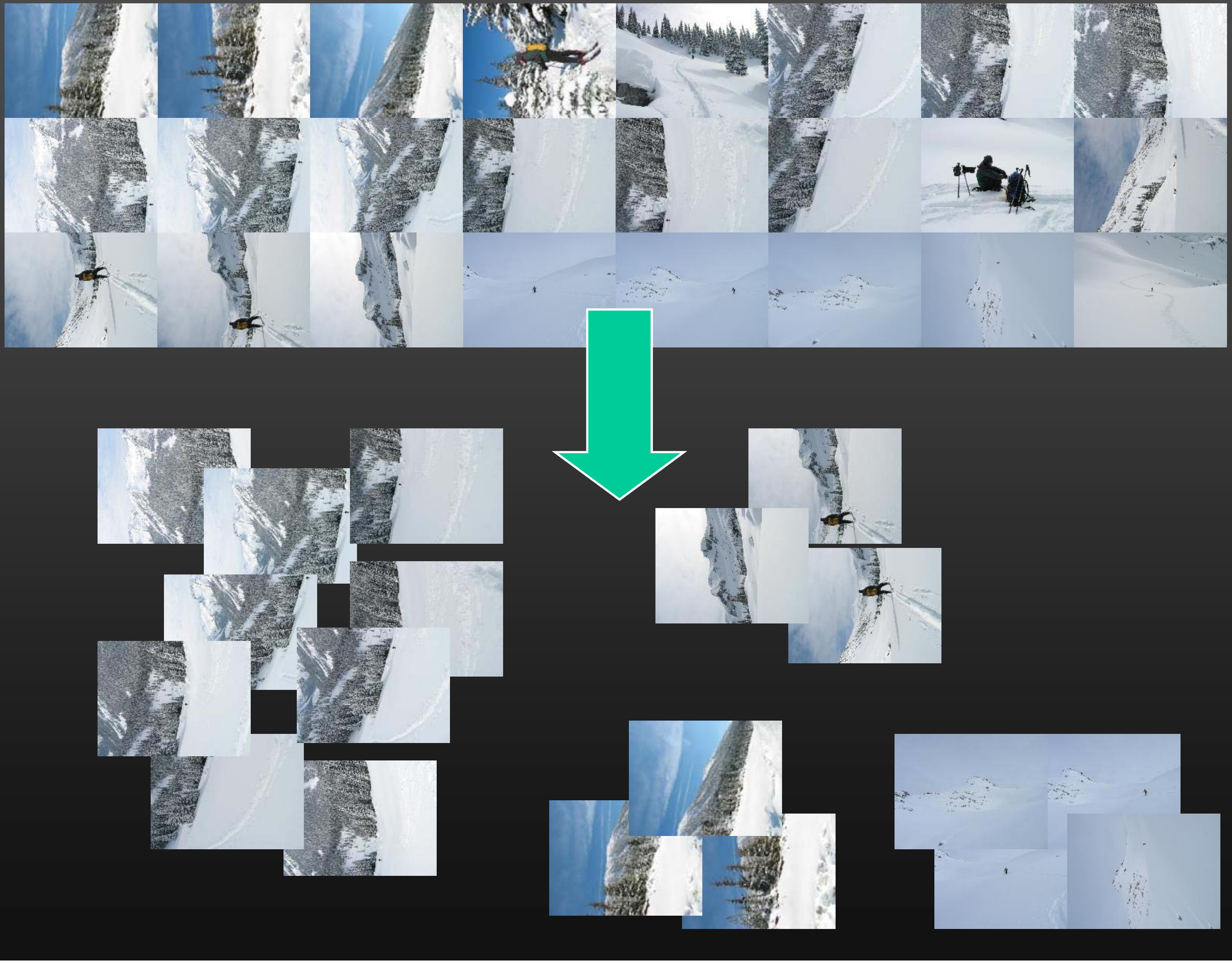
2. Find connected sets of images



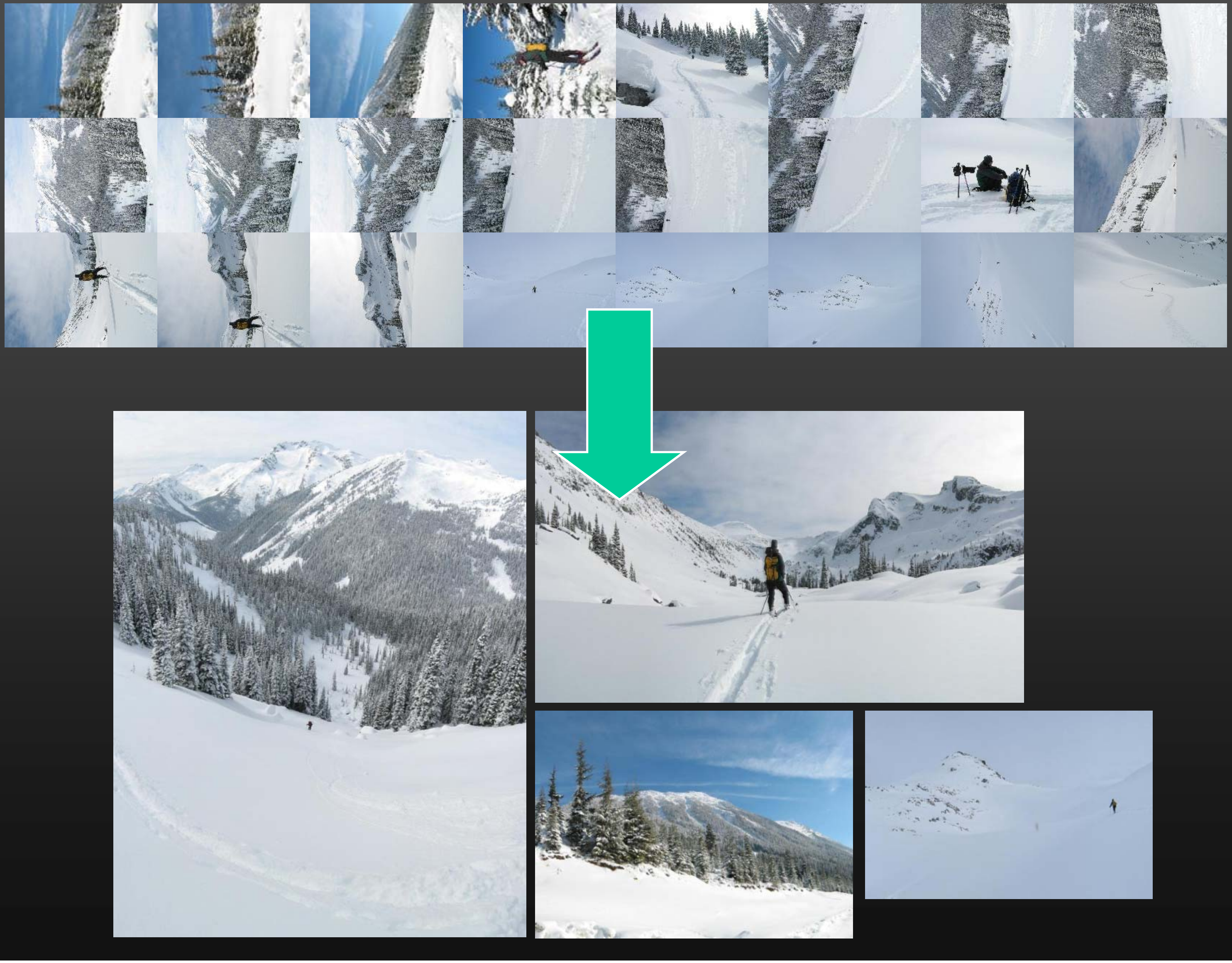
2. Find connected sets of images



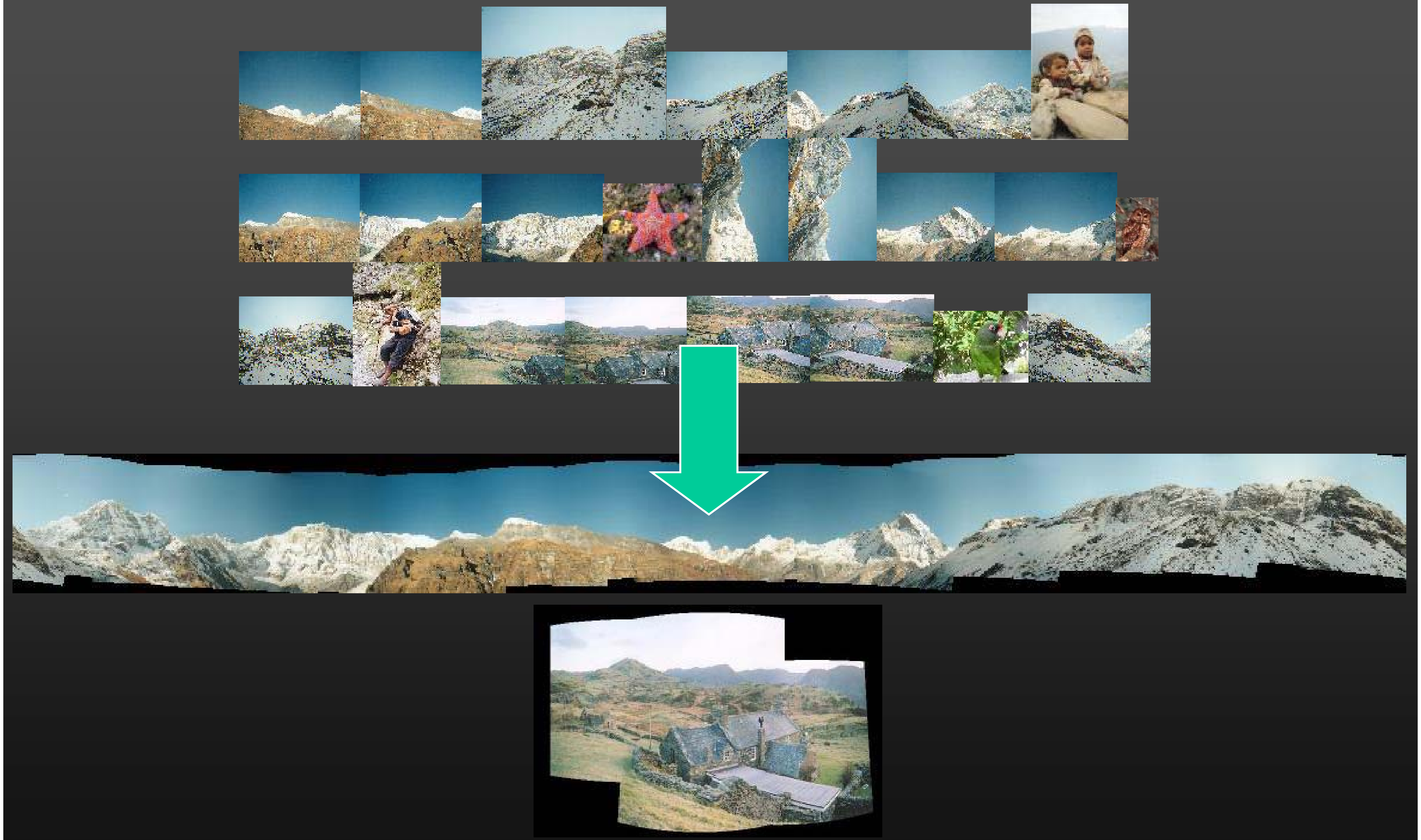
2. Find connected sets of images



3. Stitch and blend the panoramas



Results



Issues in alignment-based applications

Choosing the geometric alignment model

- Tradeoff between “correctness” and robustness (also, efficiency)

Choosing the descriptor

- “Rich” imagery (natural images): high-dimensional patch-based descriptors (e.g., SIFT)
- “Impoverished” imagery (e.g., star fields): need to create invariant geometric descriptors from k-tuples of point-based features

Strategy for finding putative matches

- Small number of images, one-time computation (e.g., panorama stitching): brute force search
- Large database of model images, frequent queries: indexing or hashing
- Heuristics for feature-space pruning of putative matches

Issues in alignment-based applications

Choosing the geometric alignment model

Choosing the descriptor

Strategy for finding putative matches

Hypothesis generation strategy

- Relatively large inlier ratio: RANSAC
- Small inlier ratio: locality constraints, Hough transform

Hypothesis verification strategy

- Size of consensus set, residual tolerance depend on inlier ratio and expected accuracy of the model
- Possible refinement of geometric model
- Dense verification