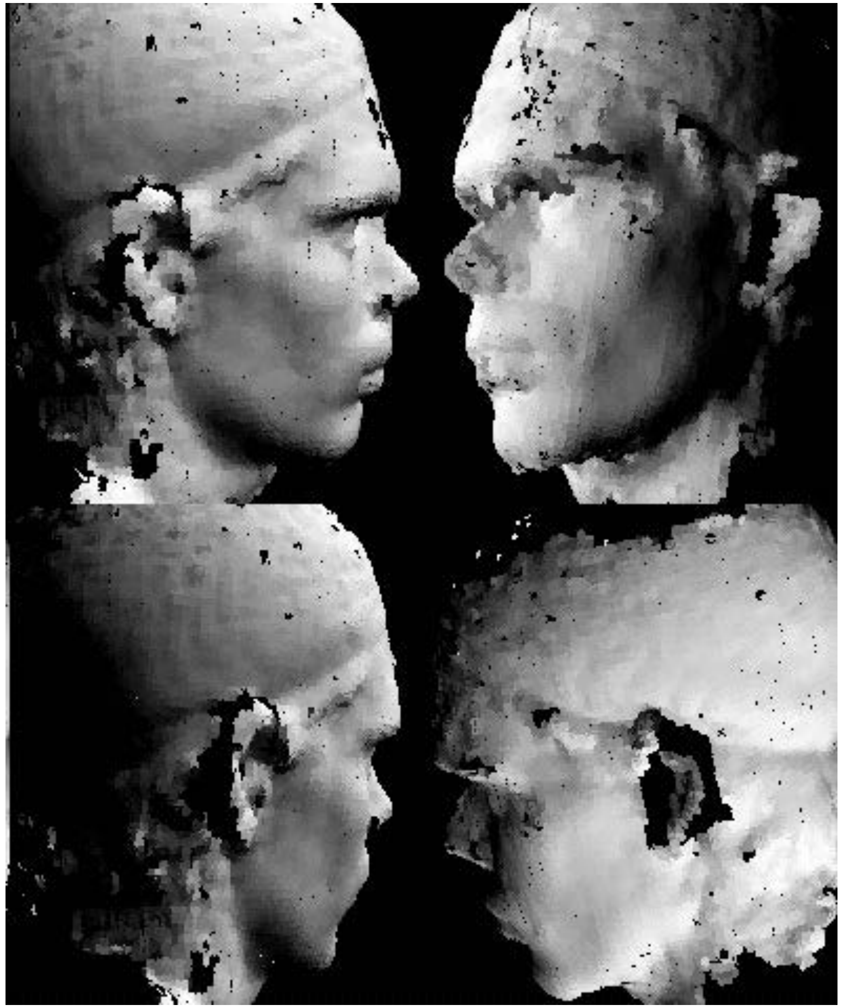
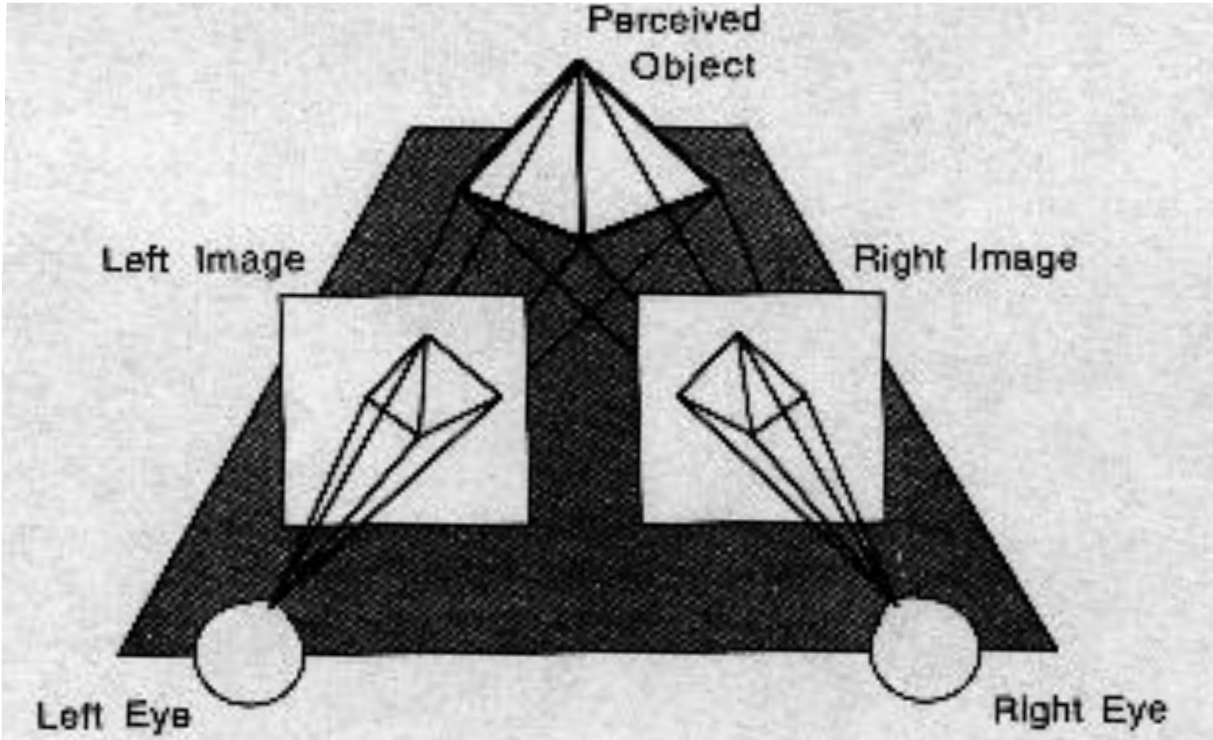
SHAPE FROM X

One image:
- Texture
- Shading

Two images or more:
- **Stereo**
- Contours
- Motion



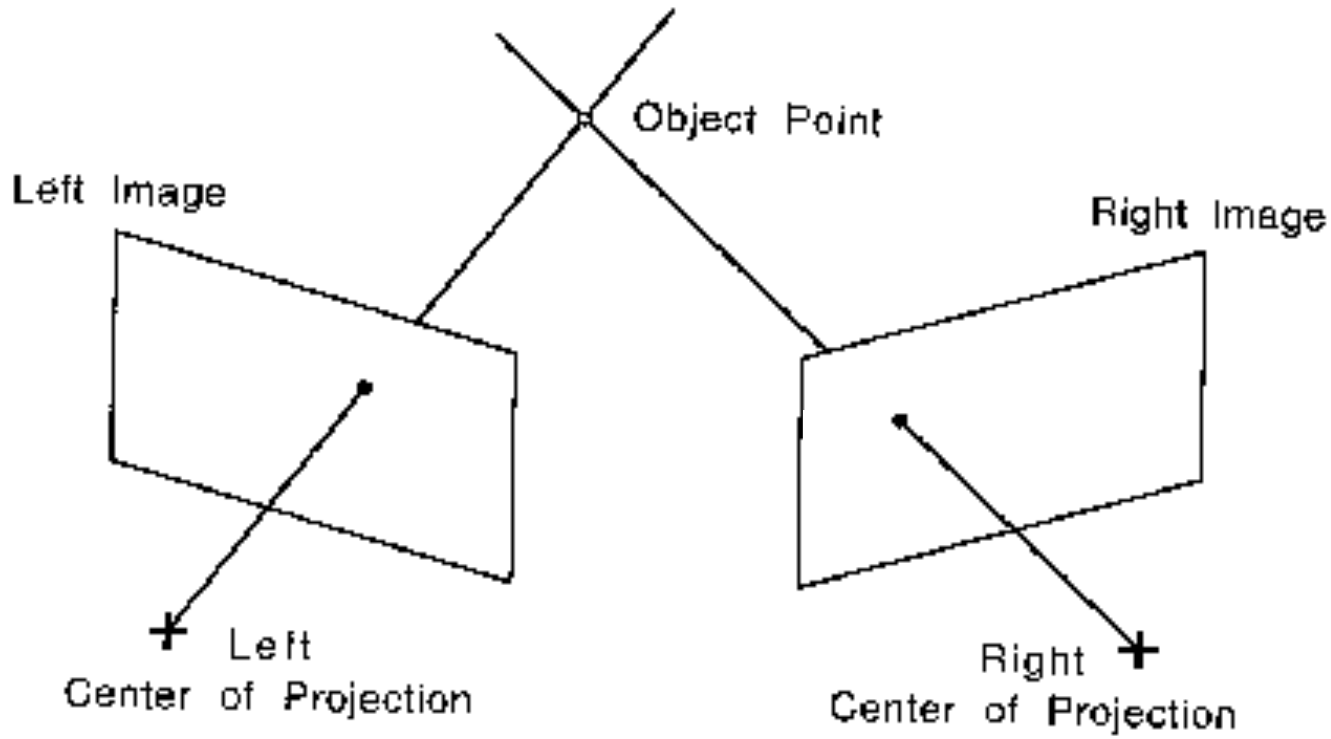
GEOMETRIC STEREO



Depth from two or more images:

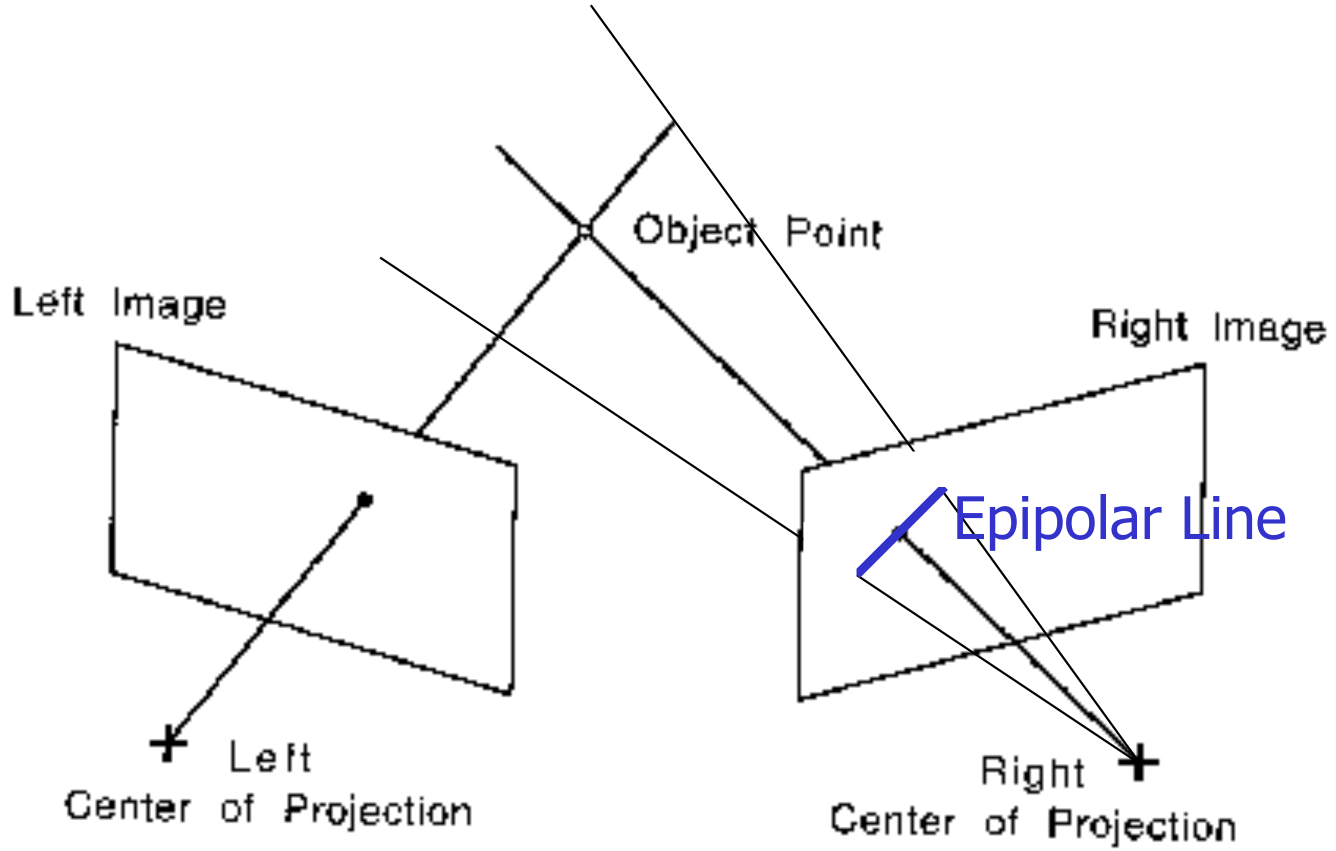
- Geometry of image pairs
- Establishing correspondences

TRIANGULATION



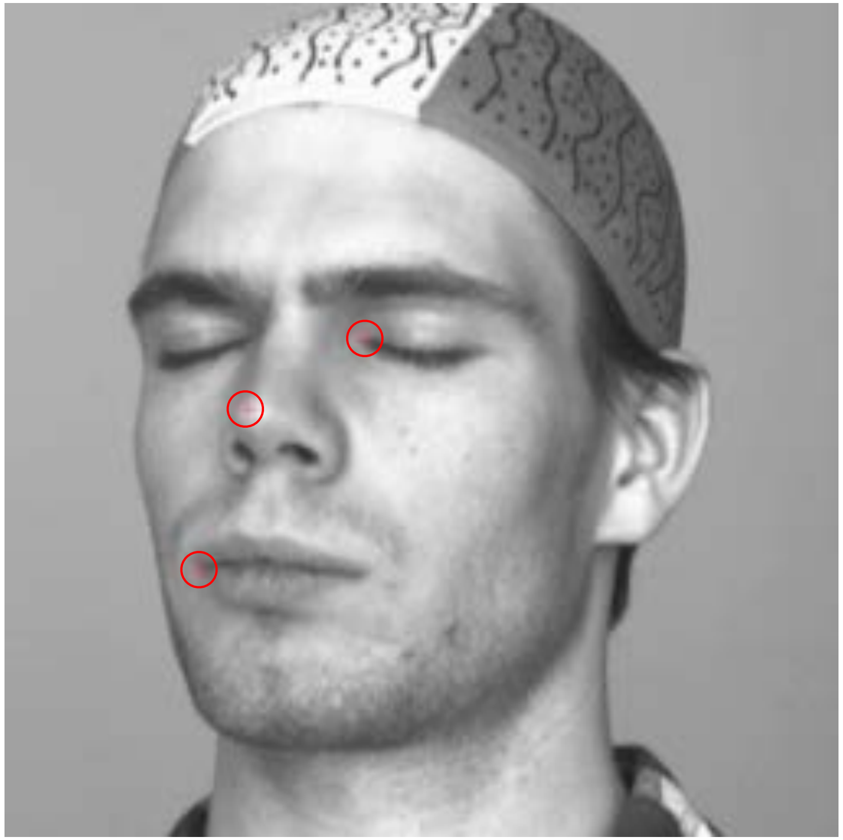
Geometric Stereo: Depth from two images

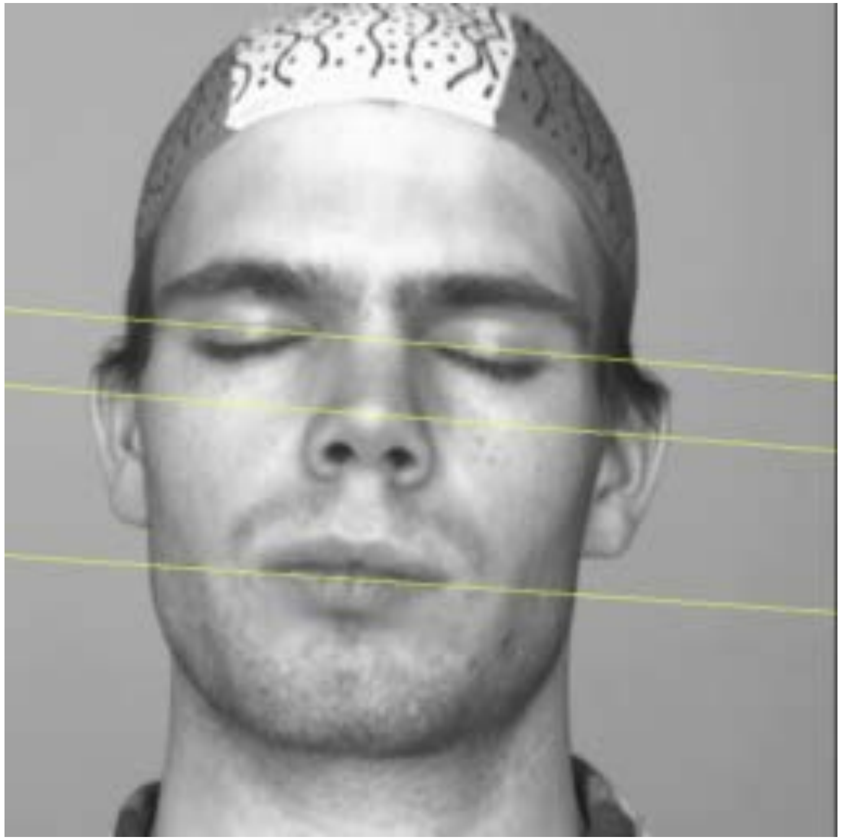
EPIPOLAR LINE



Line on which the corresponding point must lie.

EPIPOLAR LINES

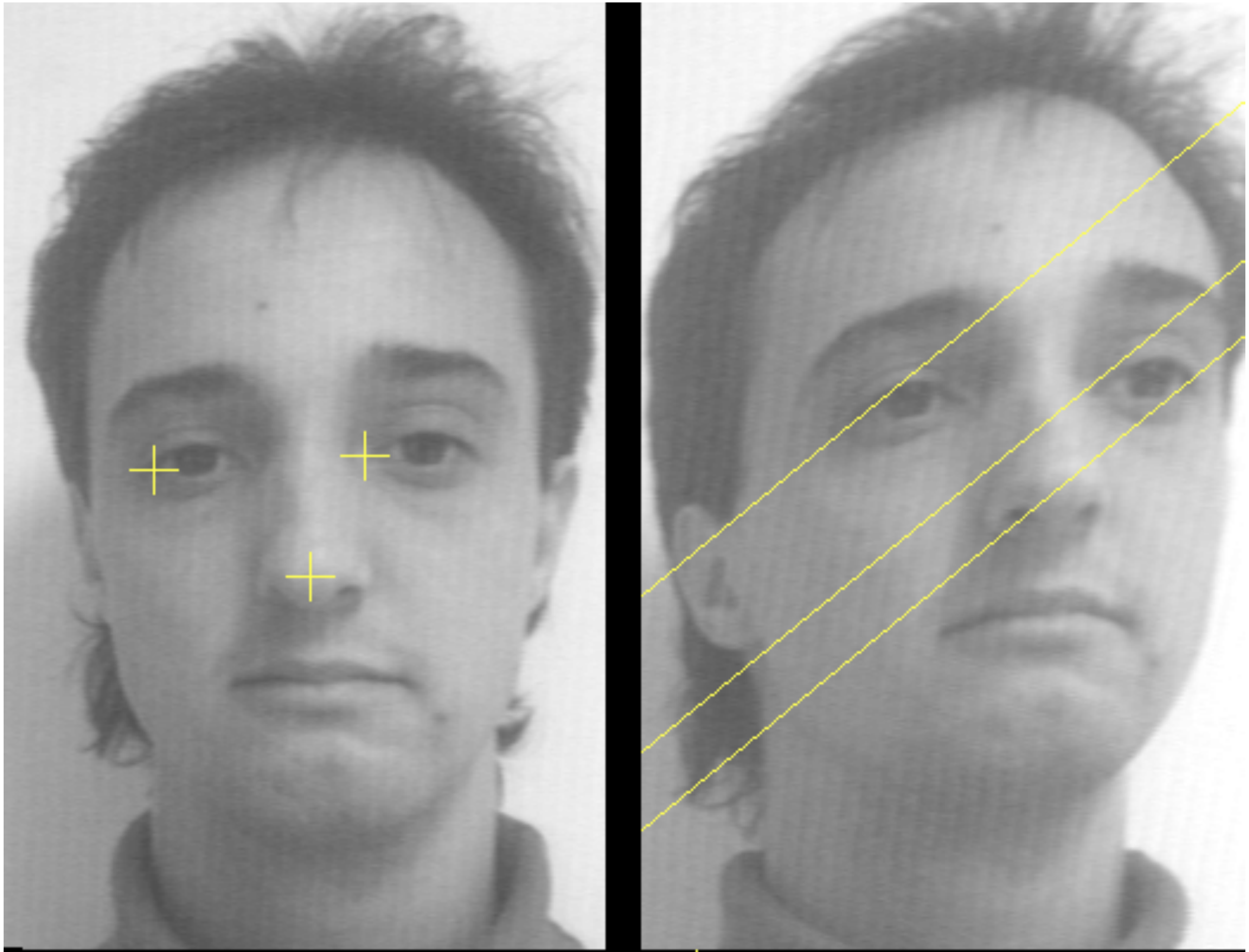




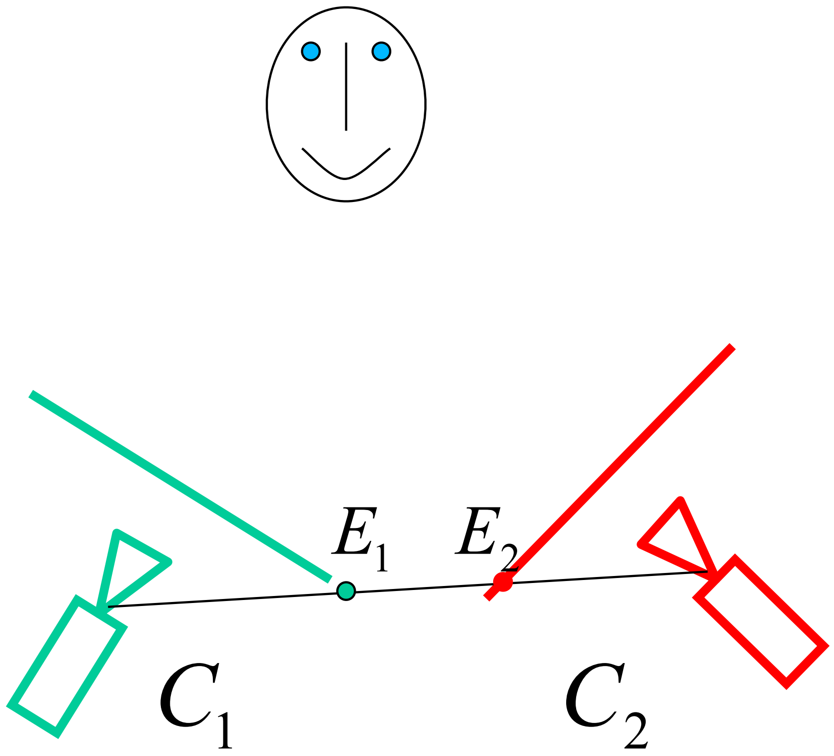
Three points shown as red crosses.

Corresponding epipolar lines.

EPIPOLAR LINES



EPIPOLE

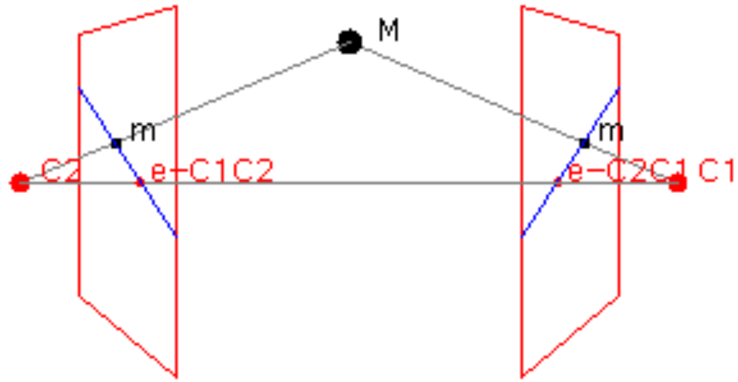


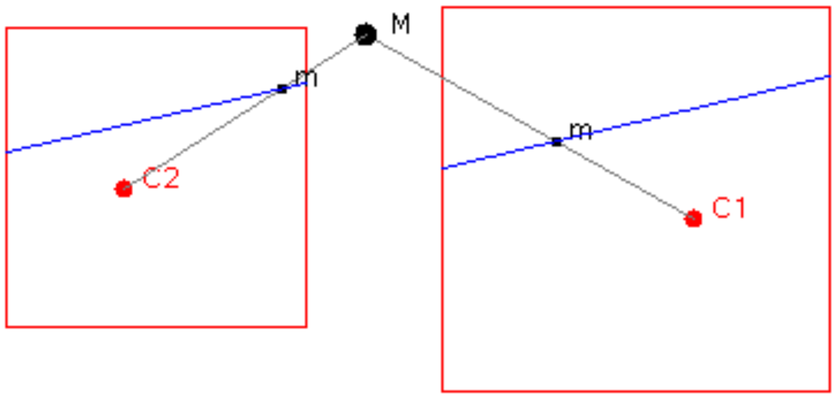
Point at which **all** epipolar lines intersect:

➡ Located at the intersection of line joining optical centers and image plane.

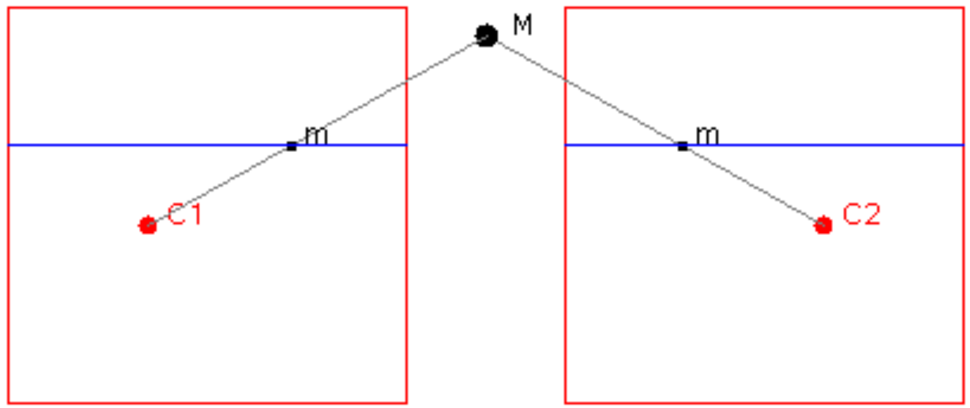
EPIPOLAR GEOMETRY

In general:



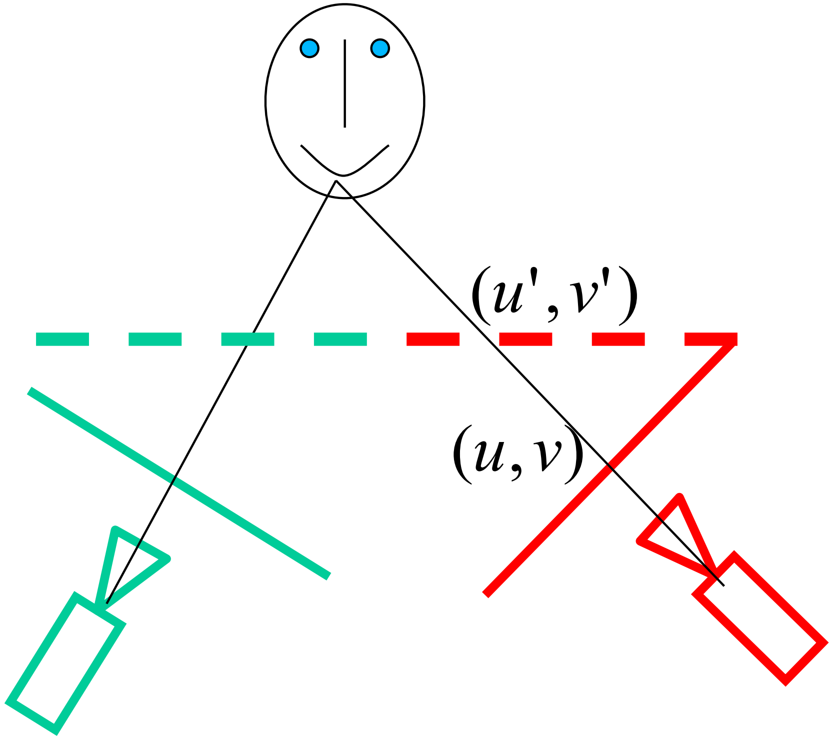


Parallel image planes



Horizontal baseline

RECTIFICATION



$$\begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

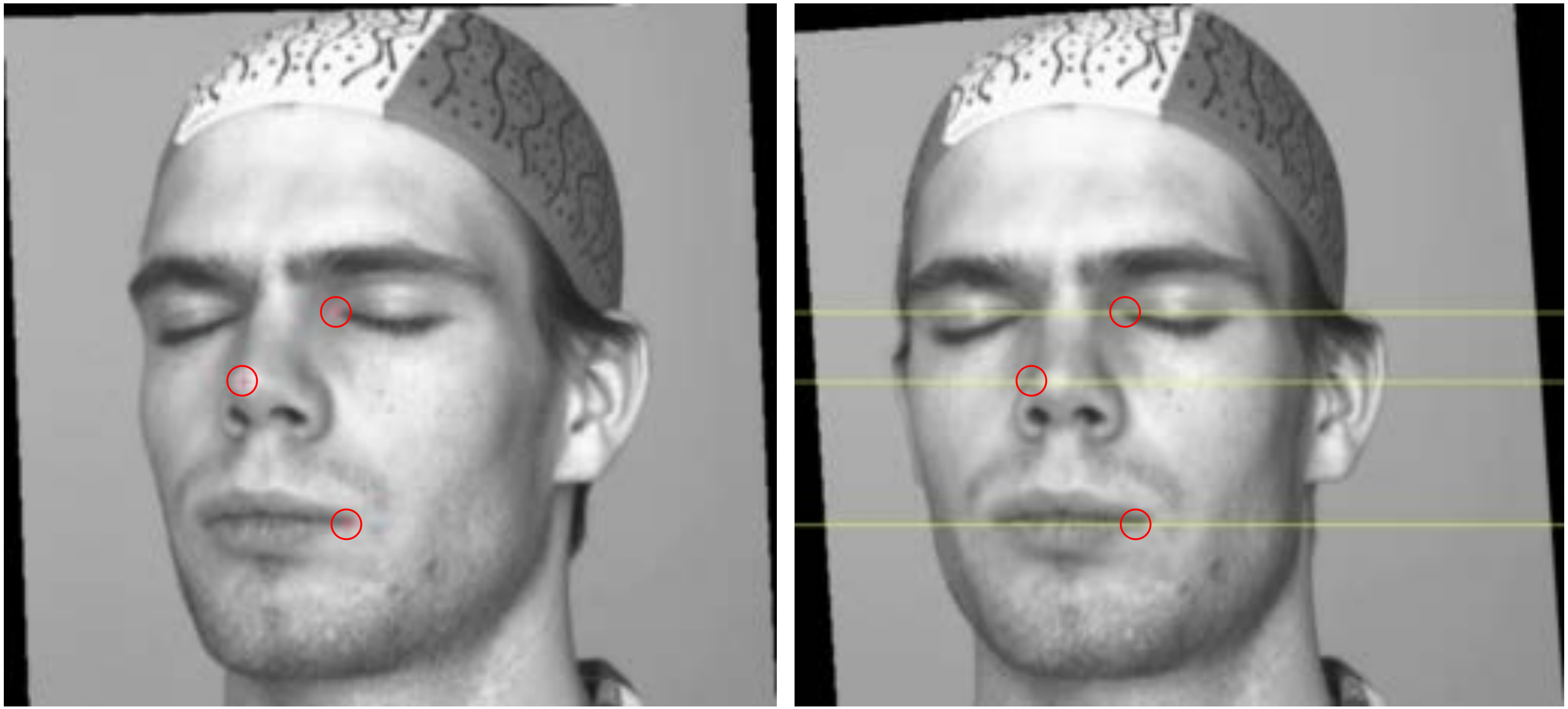
$$u' = U' / W'$$

$$v' = V' / W'$$

Reprojection into parallel virtual image planes:

- Linear operation in projective coordinates
- Real-time implementation possible

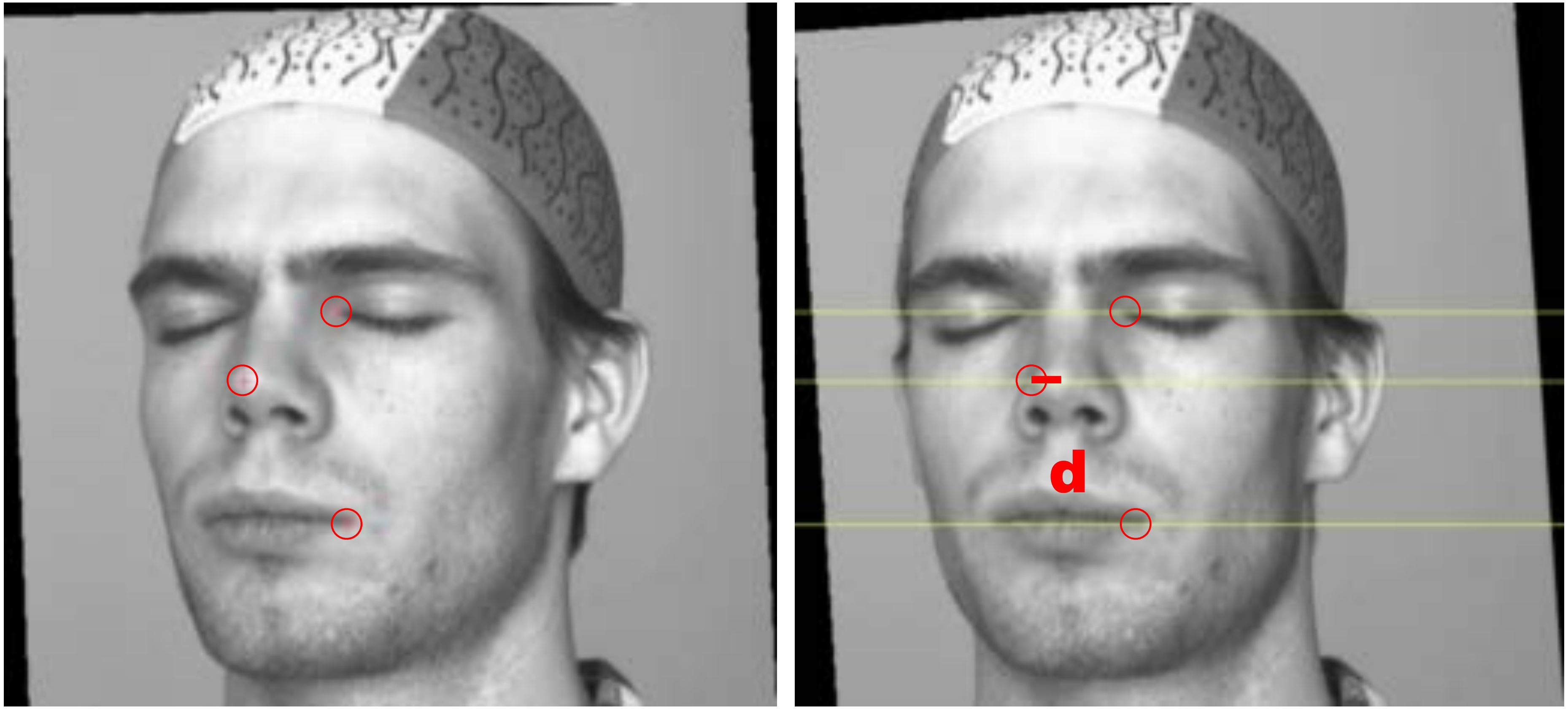
RECTIFICATION



Intersecting epipolar lines

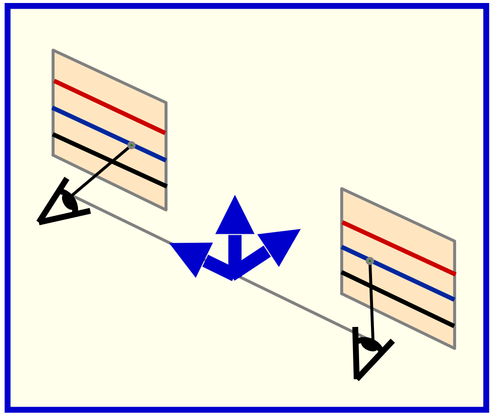
—> Parallel epipolar lines

DISPARITY



Horizontal shift along epipolar line, inversely proportional to distance.

DISPARITY VS DEPTH



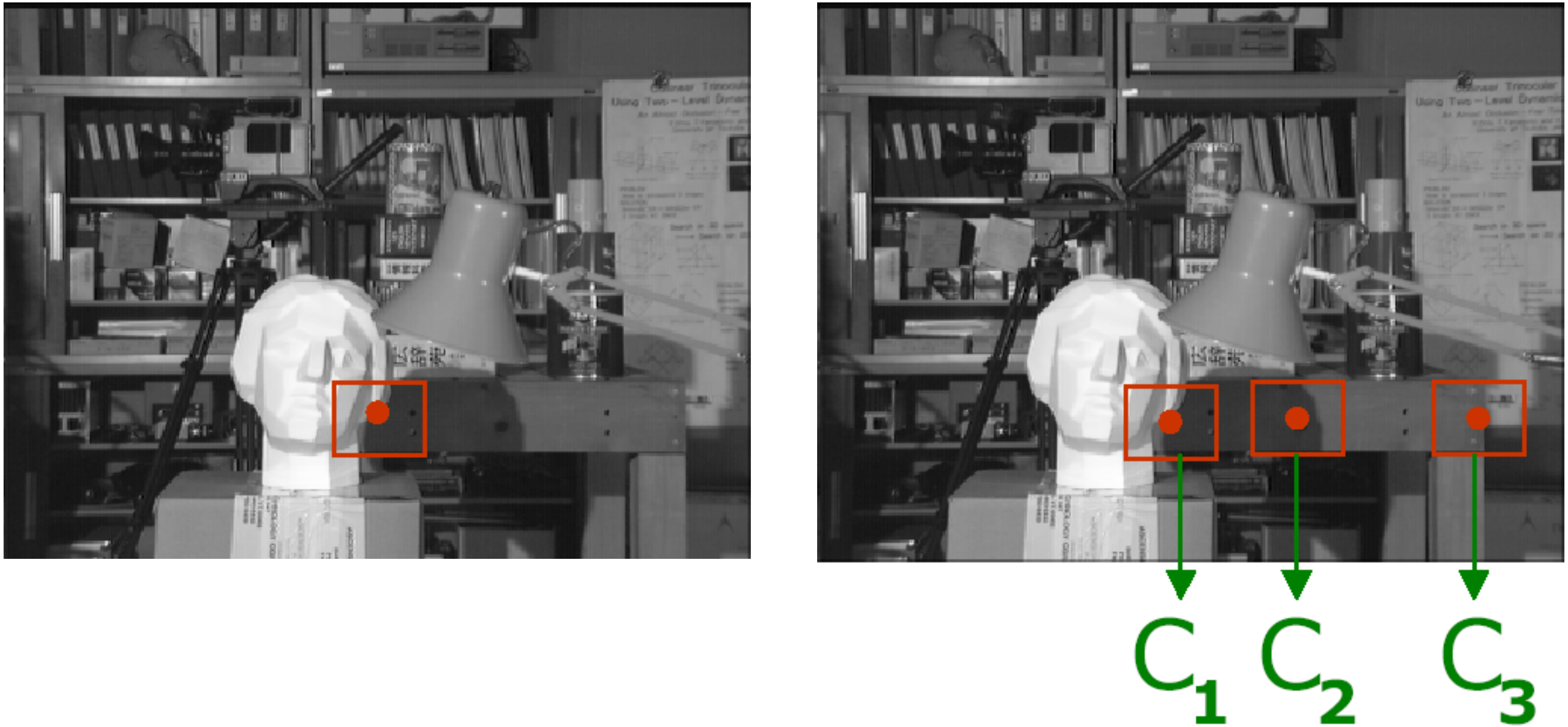
$$u_l = \frac{f(X - b/2)}{Z}, \; v_l = \frac{fY}{Z}$$

$$u_r = \frac{f(X + b/2)}{Z}, \; v_l = \frac{fY}{Z}$$

$$d = f\frac{b}{Z}$$

→ Disparity is inversely proportional to depth.

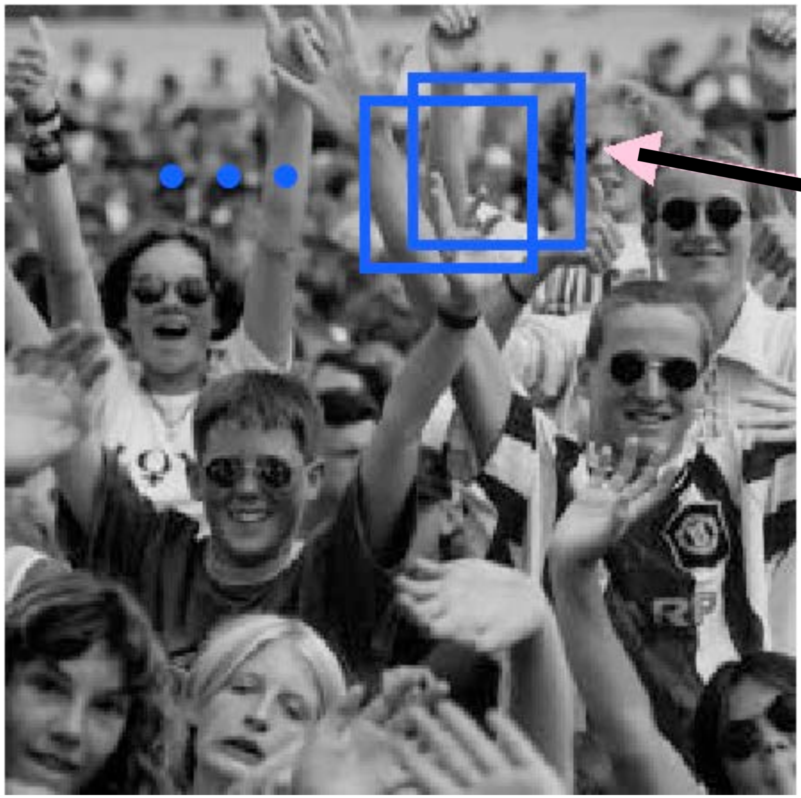
WINDOW BASED APPROACH



- Compute a cost for each C_n location.
- Pick the lowest cost one.

FINDING A PATTERN IN AN IMAGE

Straightforward Approach



Pattern

Move pattern everywhere and compare with image.

But how?

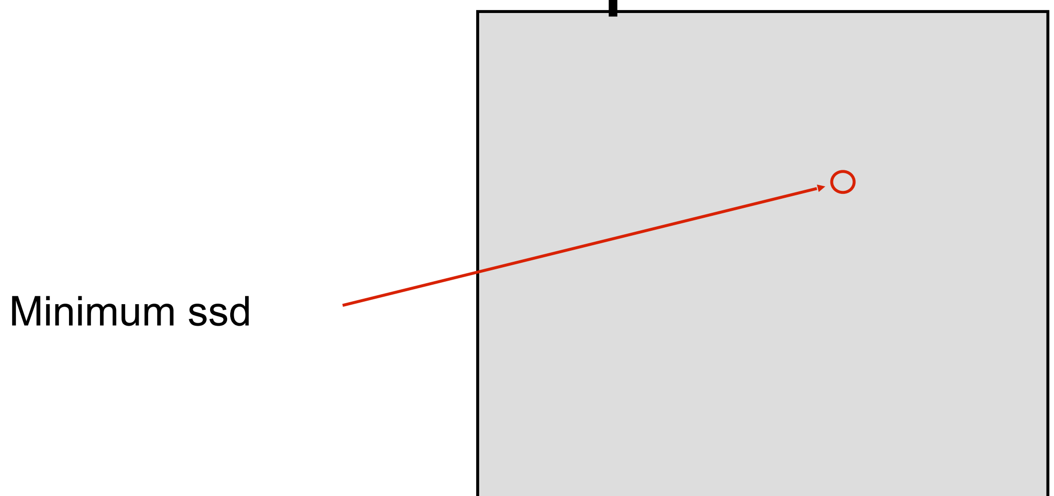
SUM OF SQUARE DIFFERENCES

Subtract pattern and image pixel by pixel and add squares:

$$ssd(u,v) = \sum_{(x,y)\in N} \left[I(u+x, v+y) - P(x,y) \right]^2$$

If identical ssd=0, otherwise ssd >0

→Look for minimum of ssd with respect to u and v.



CORRELATION

$$ssd(u,v) = \sum_{(x,y)\in N}\left[I(u+x,v+y) - P(x,y)\right]^2$$

$$= \cancel{\sum_{(x,y)\in N} I(u+x,v+y)^2} + \cancel{\sum_{(x,y)\in N} P(x,y)^2} - 2\sum_{(x,y)\in N} I(u+x,v+y)P(x,y)$$

Sum of squares of
the window
(slow varying)

Sum of squares of
the pattern
(constant)

Correlation

ssd(u,v) is minimized when correlation is largest

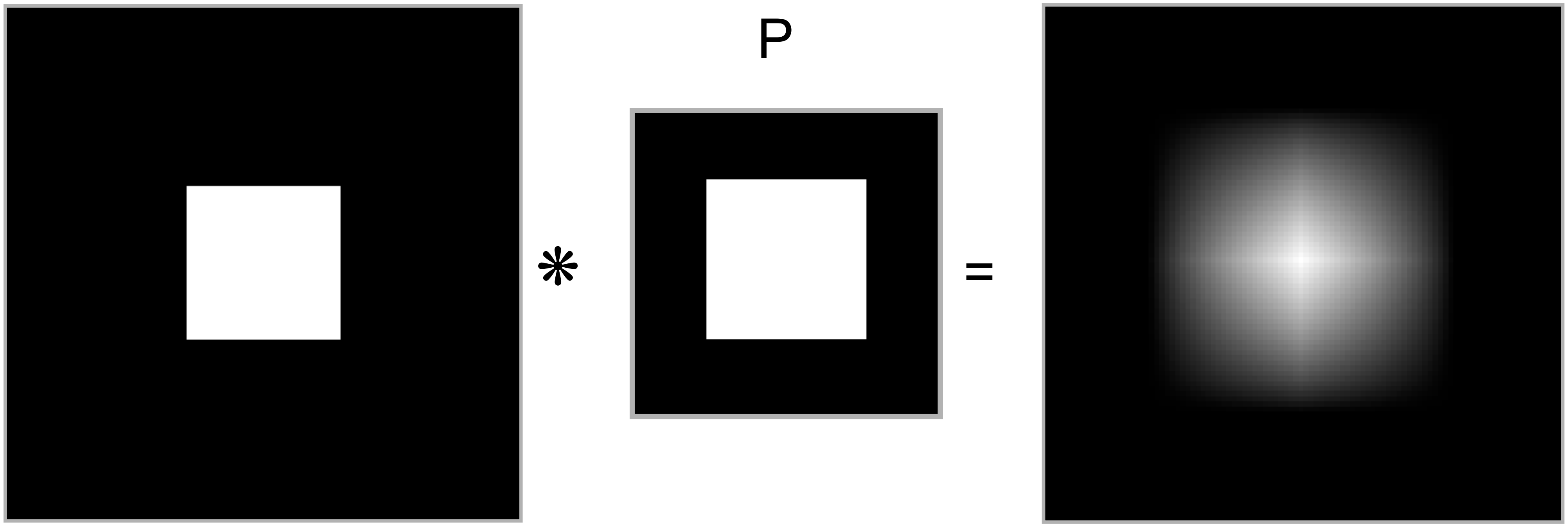
→ Correlation measures similarity

SIMPLE EXAMPLE

I

P

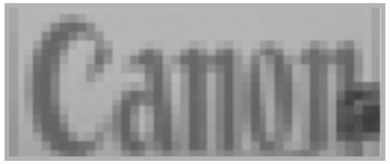
I correlated with P



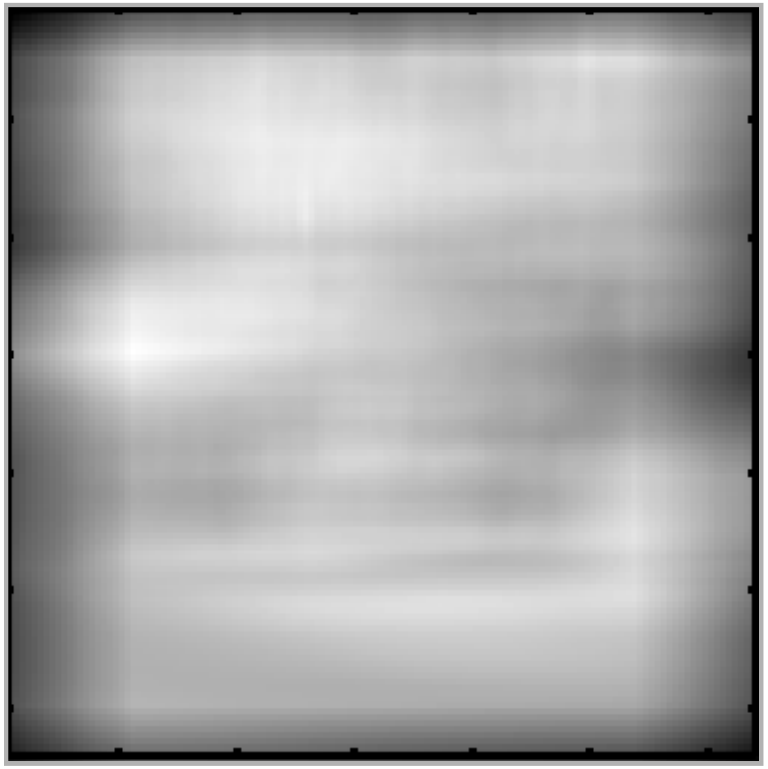
NOT SO SIMPLE EXAMPLE

Image

Pattern



Correlation



- Correlation value depends on the local gray levels of the pattern and image window.
- Need to normalize.

NORMALIZED CROSS CORRELATION

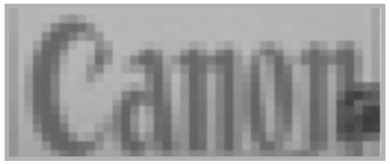
$$ncc(u,v) = \frac{\sum_{(x,y)\in N}\left[I(u+x,v+y)-\bar{I}\right]\left[P(x,y)-\bar{P}\right]}{\sqrt{\sum_{(x,y)\in N}\left[I(u+x,v+y)-\bar{I}\right]^2 \sum_{(x,y)\in N}\left[P(x,y)-\bar{P}\right]^2}}$$

- Between -1 and 1
- Invariant to linear transforms
- Independent of the average gray levels of the pattern and the image window

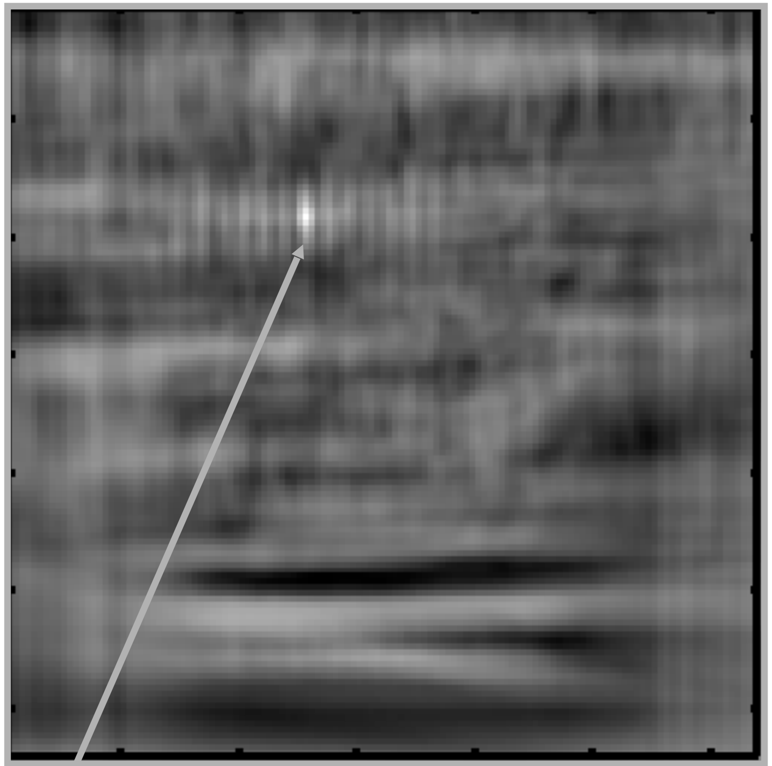
NORMALIZED EXAMPLE

Image

Pattern

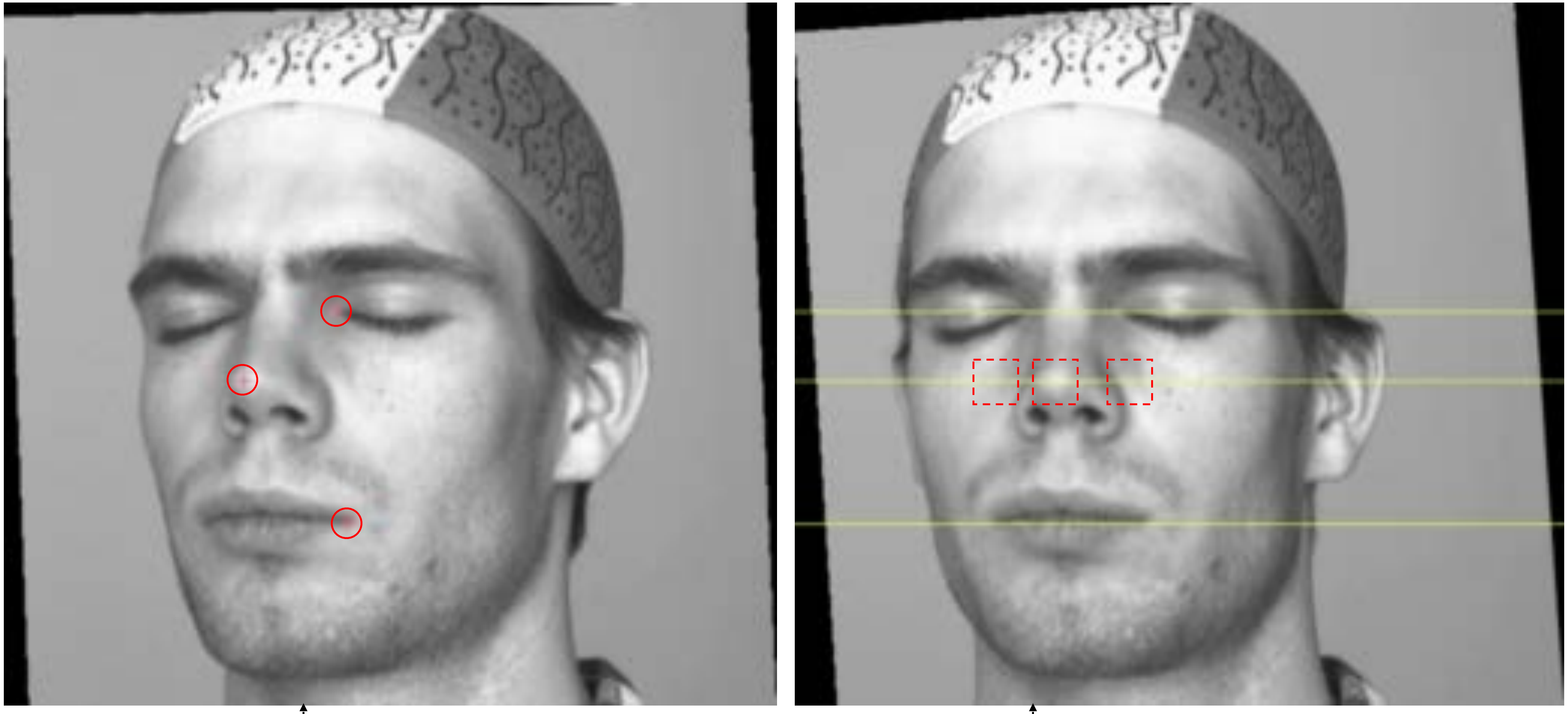


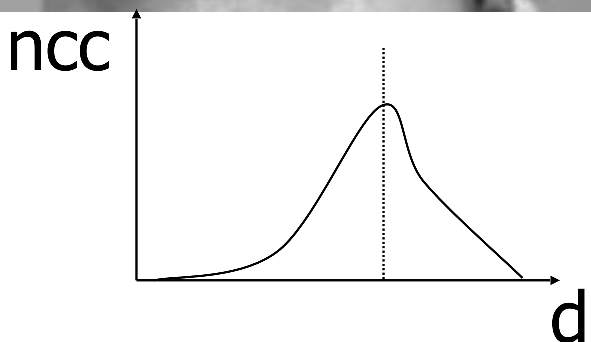
Normalized Correlation



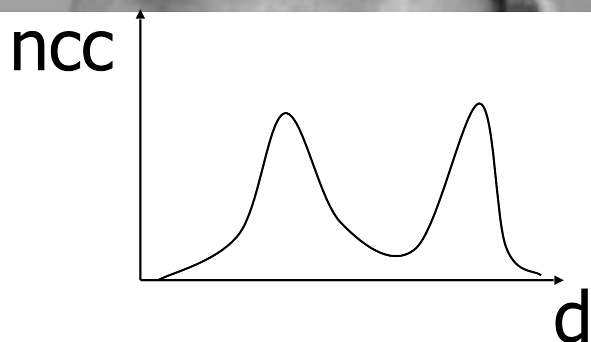
Point of maximum correlation

SEARCHING ALONG EPIPOLAR LINES

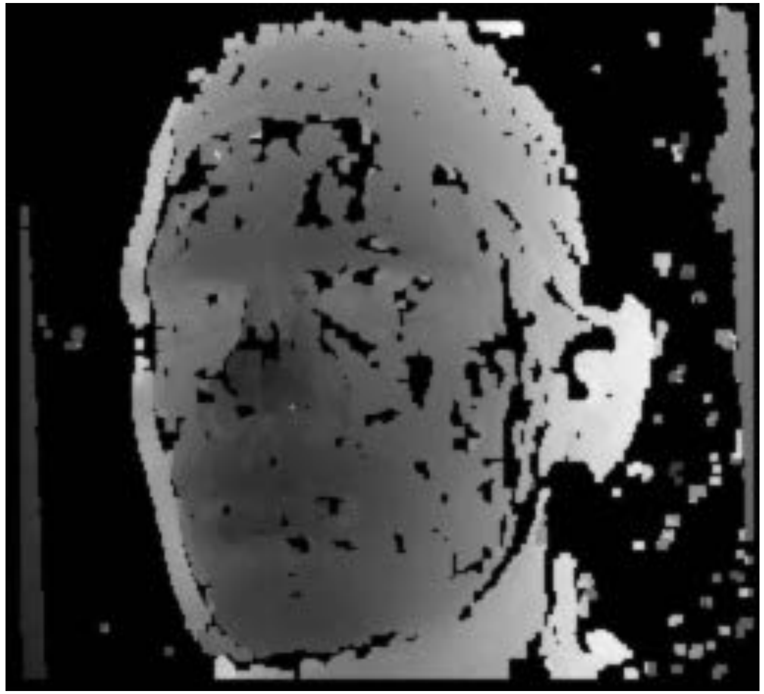




or

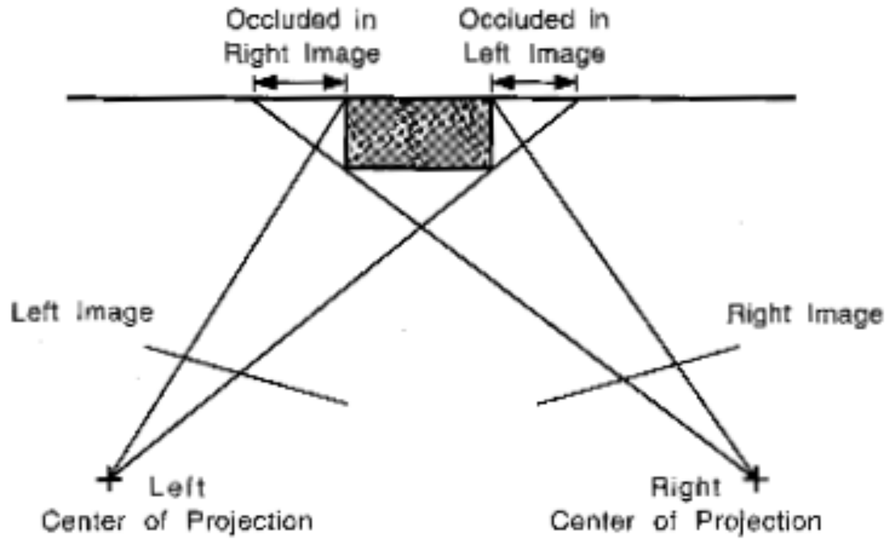


DISPARITY MAP

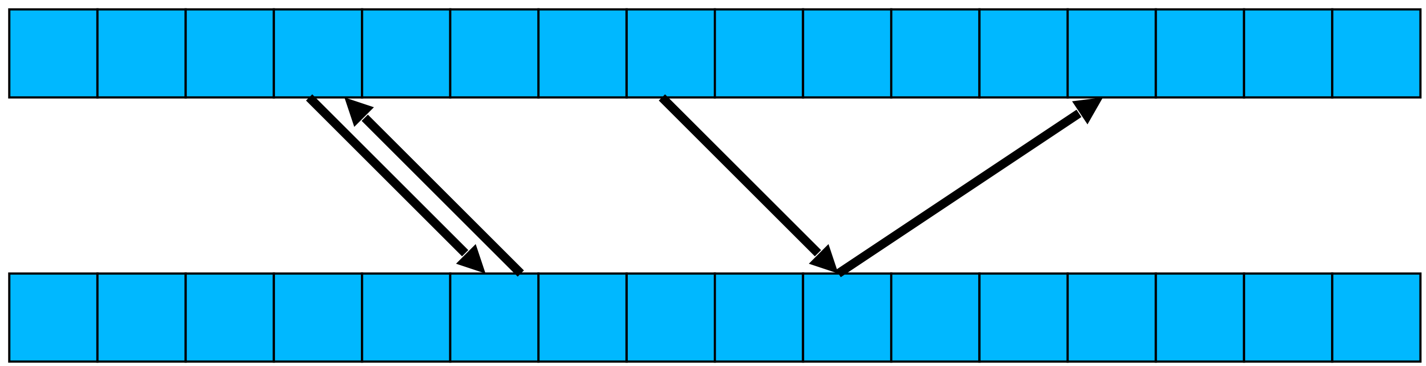


Black pixels: No disparity.

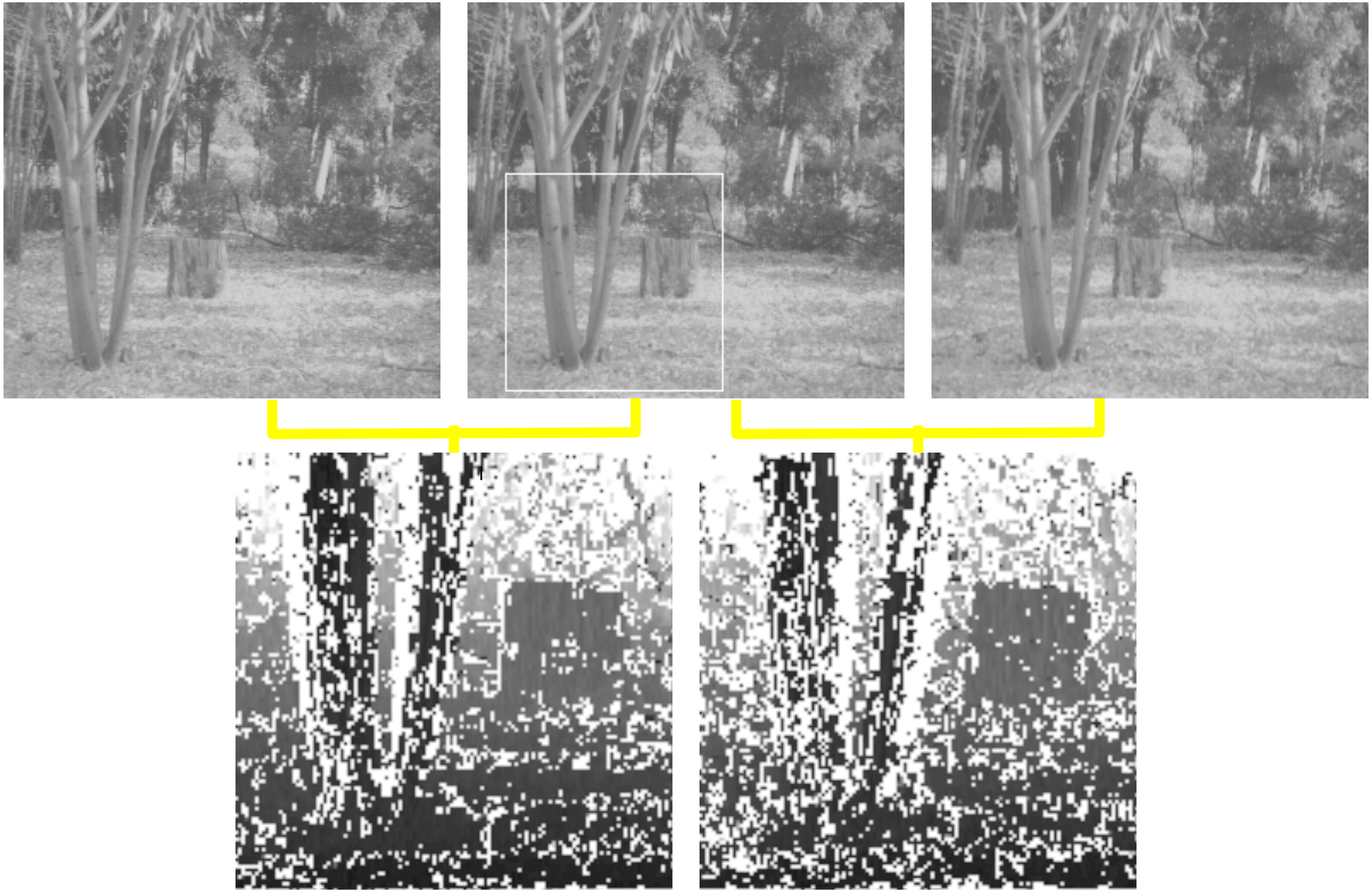
OCCLUSIONS



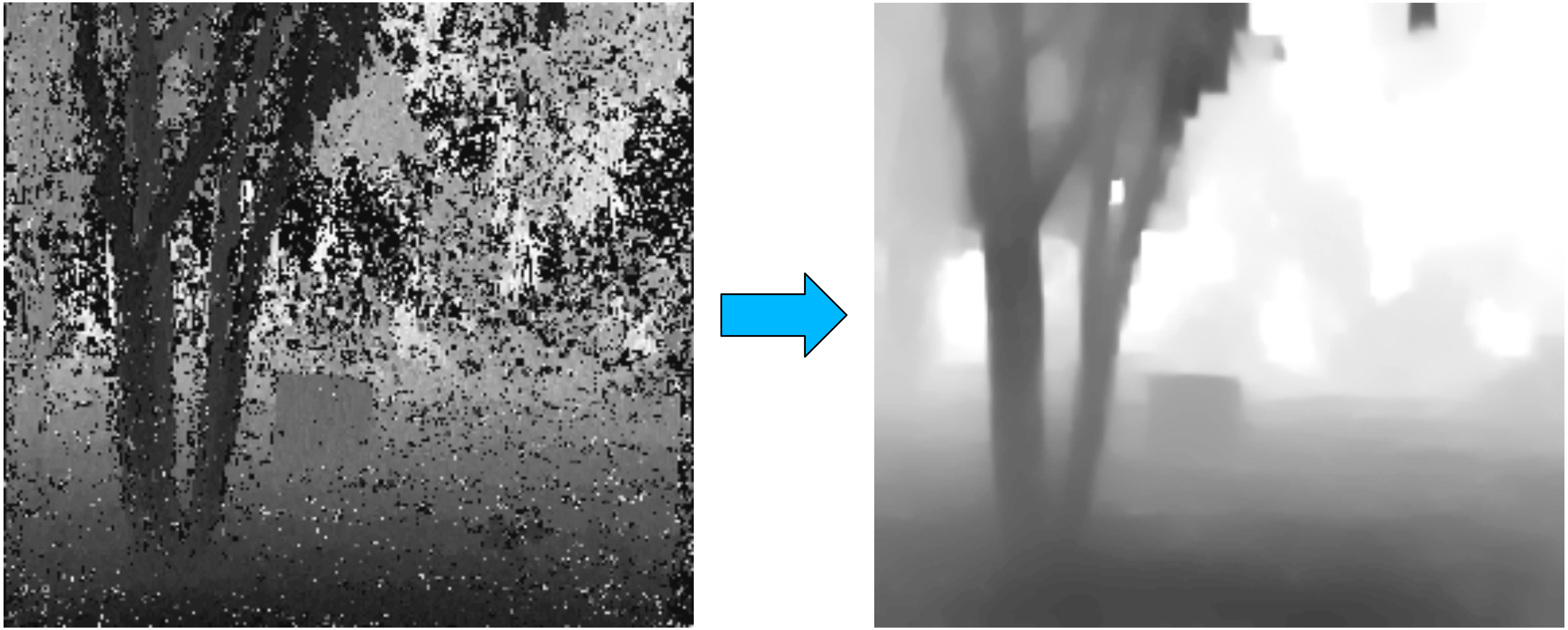
→ Consistency test



GROUND LEVEL STEREO



COMBINING DISPARITY MAPS

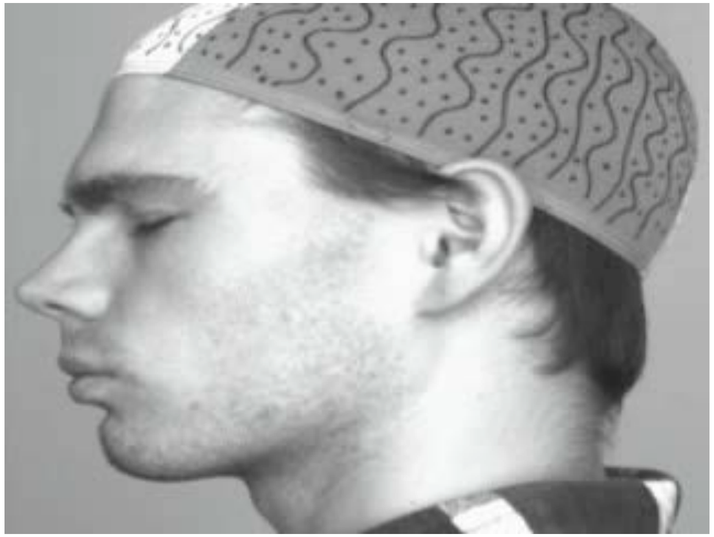


• Merging several disparity maps.

• Smoothing the resulting map.

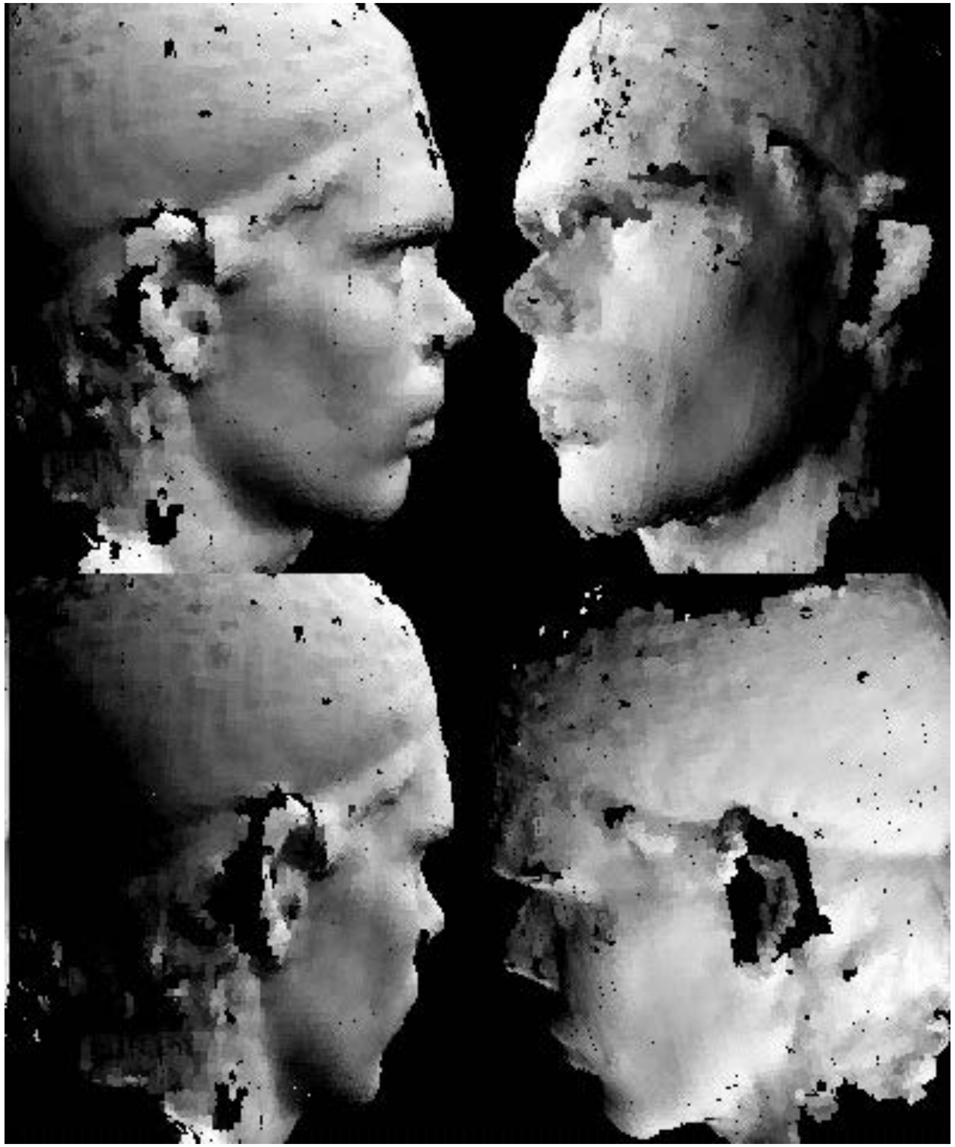
Fua, MVA'91

SHAPE FROM VIDEO



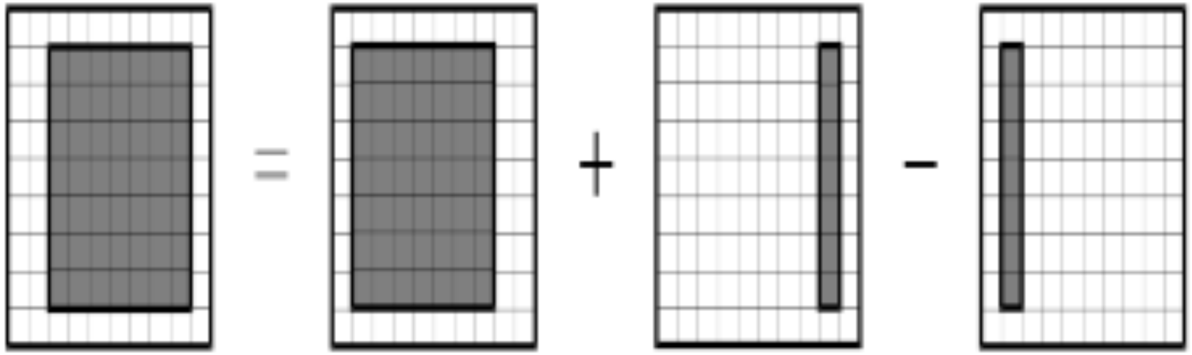
Treat consecutive images as stereo pairs.

1. Compute disparity maps.
2. Merge 3-D point clouds.
3. Represent as particles.



Fua. IJCV'97

REAL-TIME IMPLEMENTATION

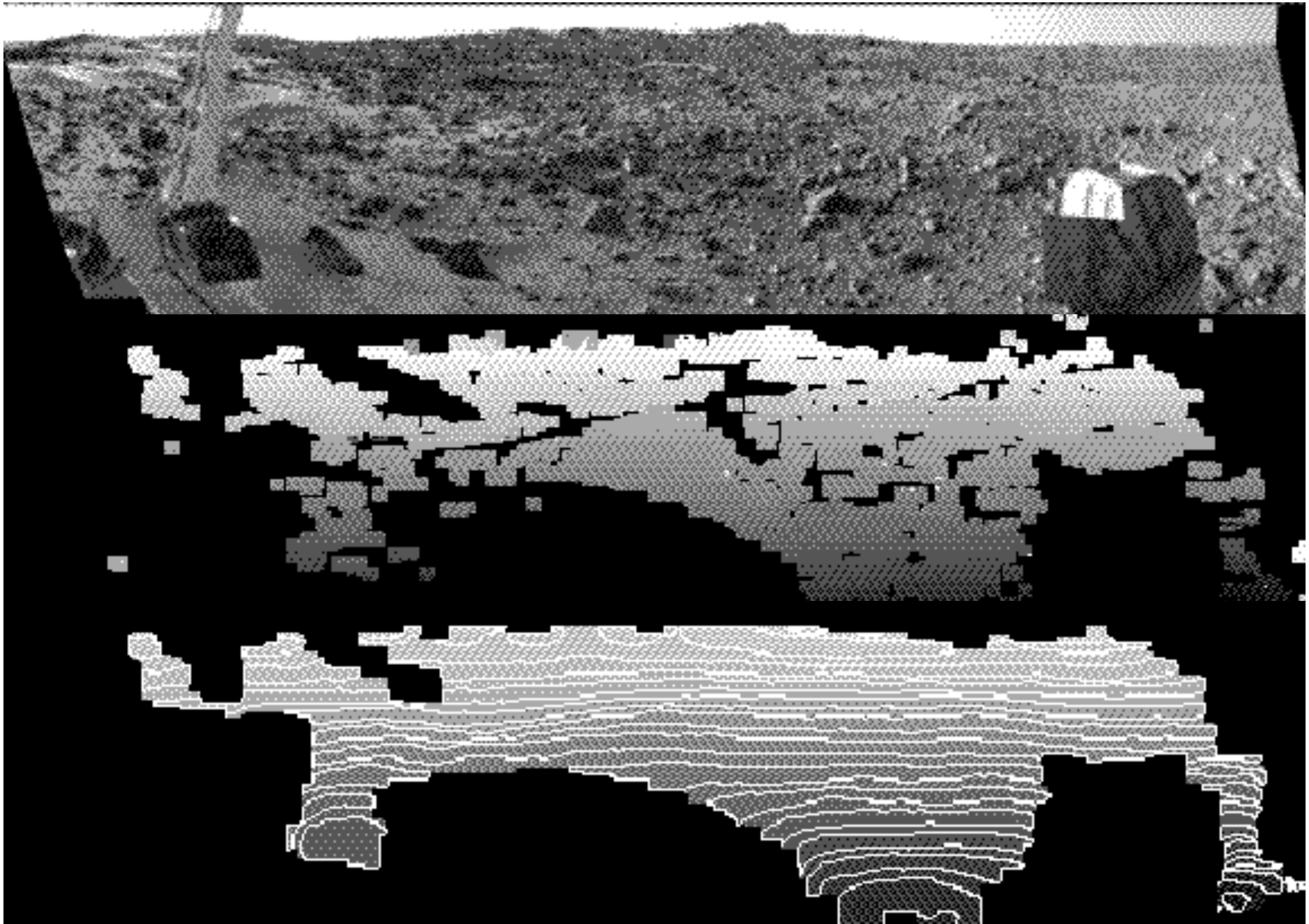


$$C(x,y,d) \quad \propto \quad \frac{\sum_{i,j} I_1(x+i,y+j) \times I_2(x+d+i,y+j)}{\sqrt{\sum_{i,j} I_2(x+d+i,y+j)^2}}$$

$$C(x+1,y,d) \quad \propto \quad \frac{\sum_{i,j} I_1(x+1+i,y+j) \times I_2(x+1+d+i,y+j)}{\sqrt{\sum_{i,j} I_2(x+1+d+i,y+j)^2}}$$

$$\propto \quad \frac{\sum_{i',j} I_1(x+i',y+j) \times I_2(x+d+i',y+j)}{\sqrt{\sum_{i,j} I_2(x+d+i',y+j)^2}}$$

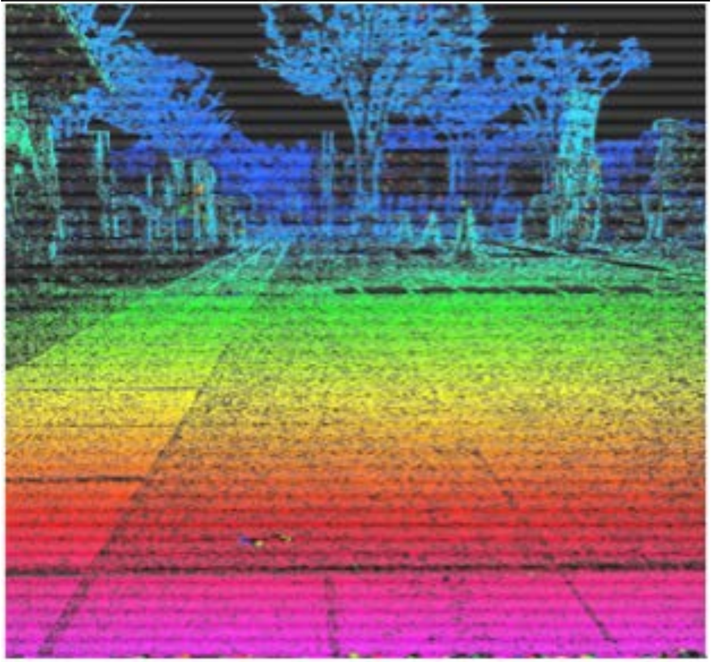
THEN ....



1993:
256x256,
60 disps,
7 fps.

Faugeras et al.,
INRIA'93

… AND MORE RECENTLY



Subaru's EyeSight System

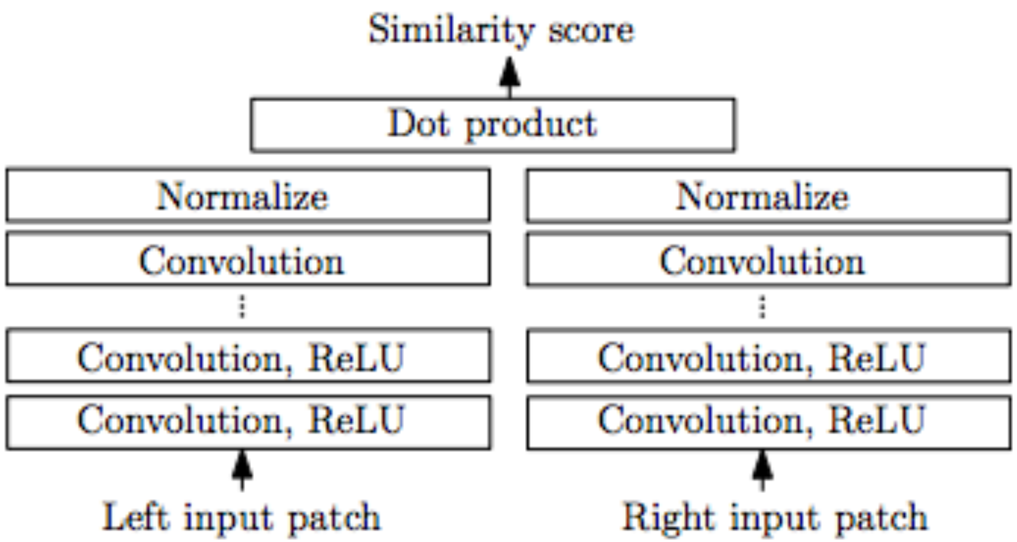
http://www.gizmag.com/subaru-new-eyesight-stereoscopic-vision-system/14879/

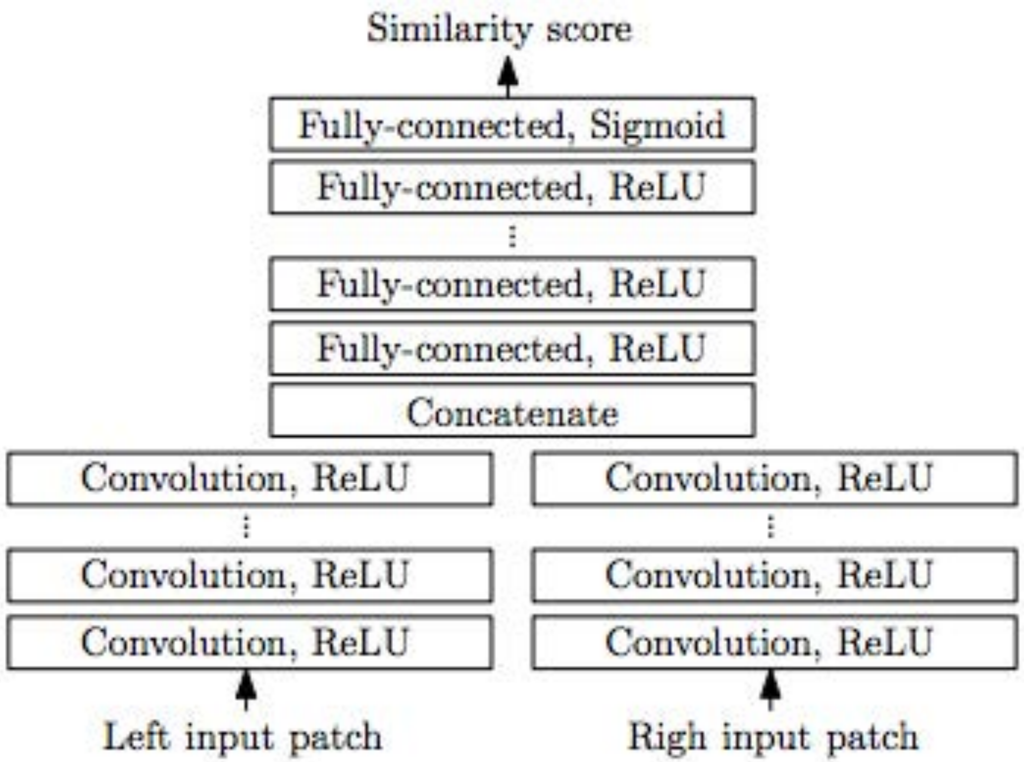
2011:
1312x688,
176 disps,
160 fps.

Saneyoshi, CMVA'11

... AND EVEN MORE RECENTLY

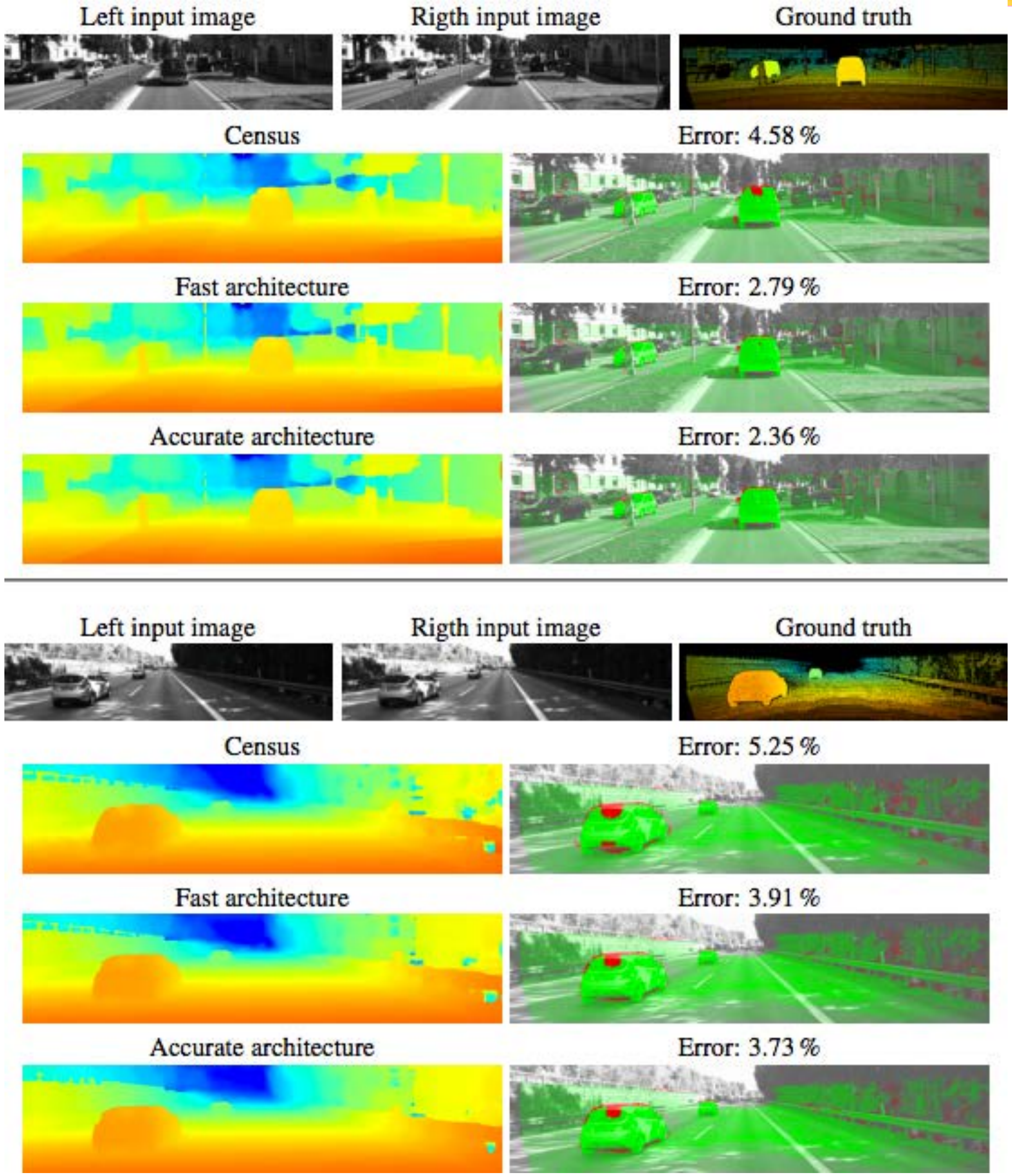
Train Siamese nets to return a similarity score.





Zbontar and Lecun, 2015

COMPARATIVE RESULTS



Improved performance on test data but

- How well will it generalize to unseen images?

- Is it worth the much heavier computational load?

Time will tell.

WINDOW SIZE

Small windows:

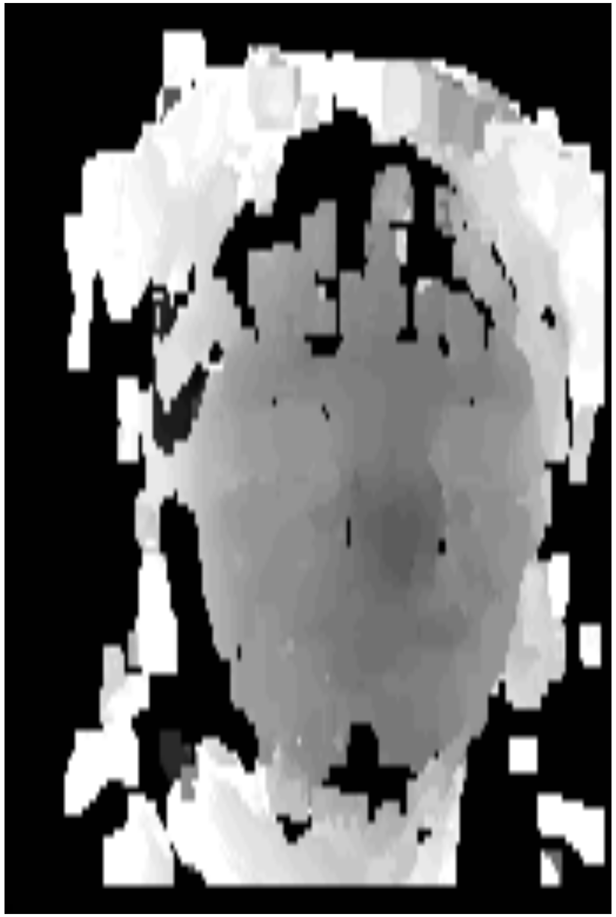
- Good precision
- Sensitive to noise

Large windows:

- Diminished precision
- Increased robustness to noise

→ Same kind of trade-off as for edge-detection.

WINDOW SIZE

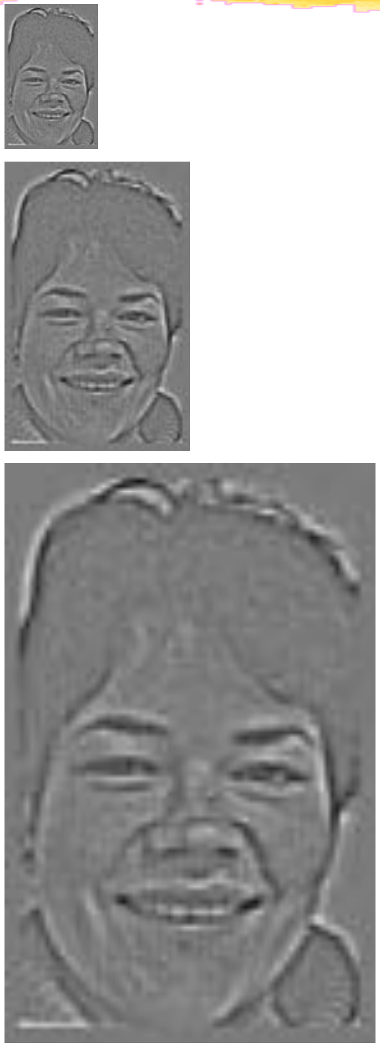


15x15

7x7

SCALE-SPACE REVISITED

Gaussian pyramid



Difference of Gaussians

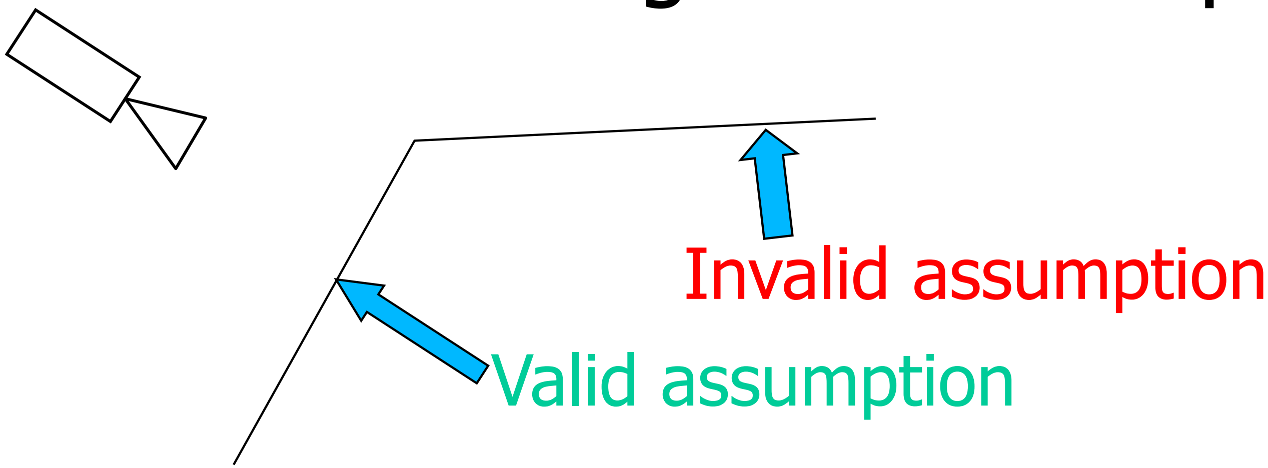
- Using a small window on a reduced image is equivalent to using a large one on the original image.

- Using difference of Gaussian images is an effective way of achieving normalization.

→It becomes natural to use results obtained using low resolution images to guide the search at higher resolution.

FRONTO-PARALLEL ASSUMPTION

The disparity is assumed to be the same in the whole correlation window, which is equivalent to assuming constant depth.



→ Ok when the surface faces the camera but breaks down otherwise.

MULTI-VIEW STEREO

Multi-view reconstruction setup

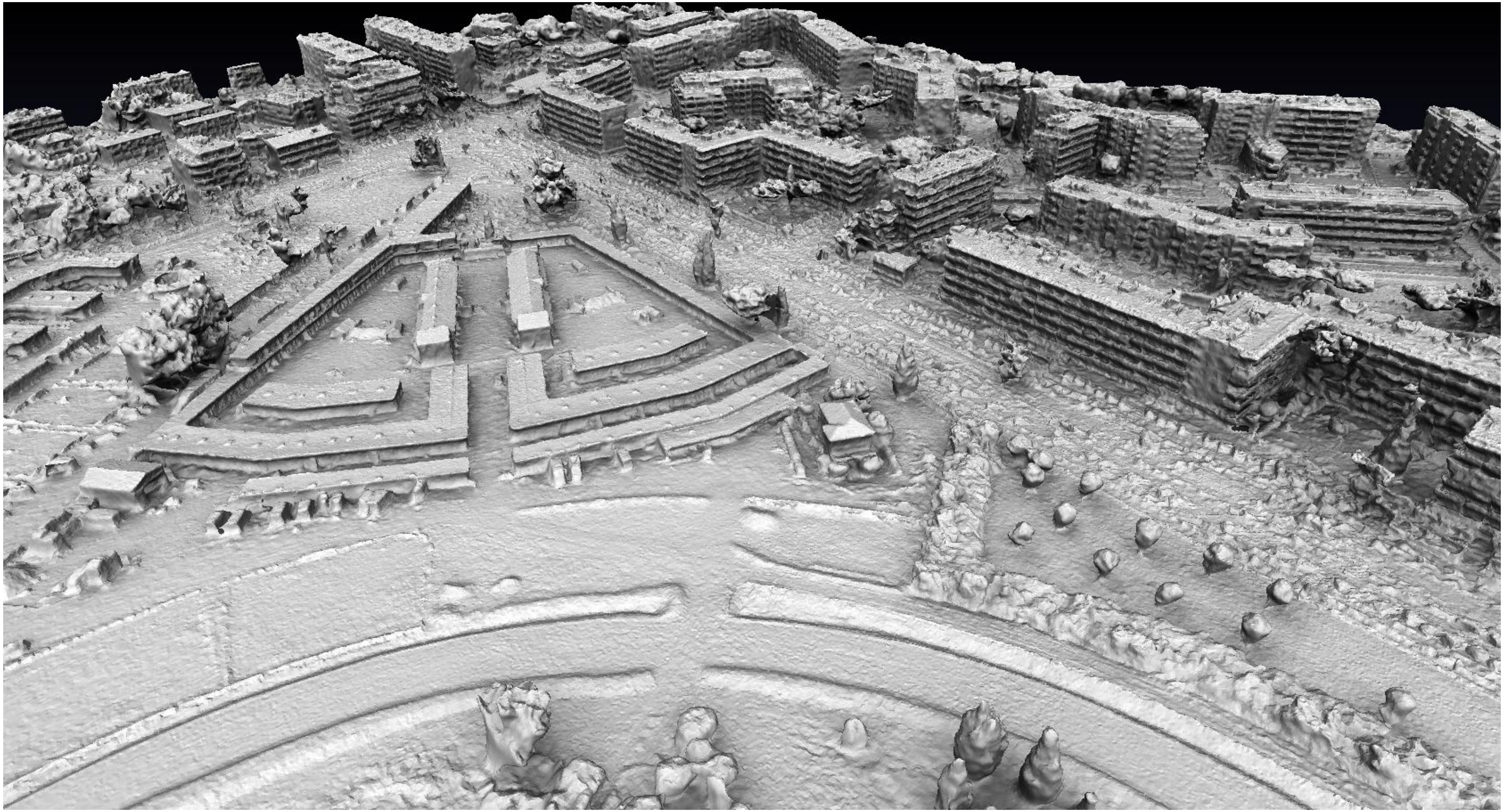
➢ Adjust correlation window shapes to handle orientation.

Texture-Mapped 3D Model

Shaded 3D Model

Furukawa&Ponce ECCV'06

MULTI-VIEW STEREO



Aganj et al. ICCV'09

SMALL DRONES

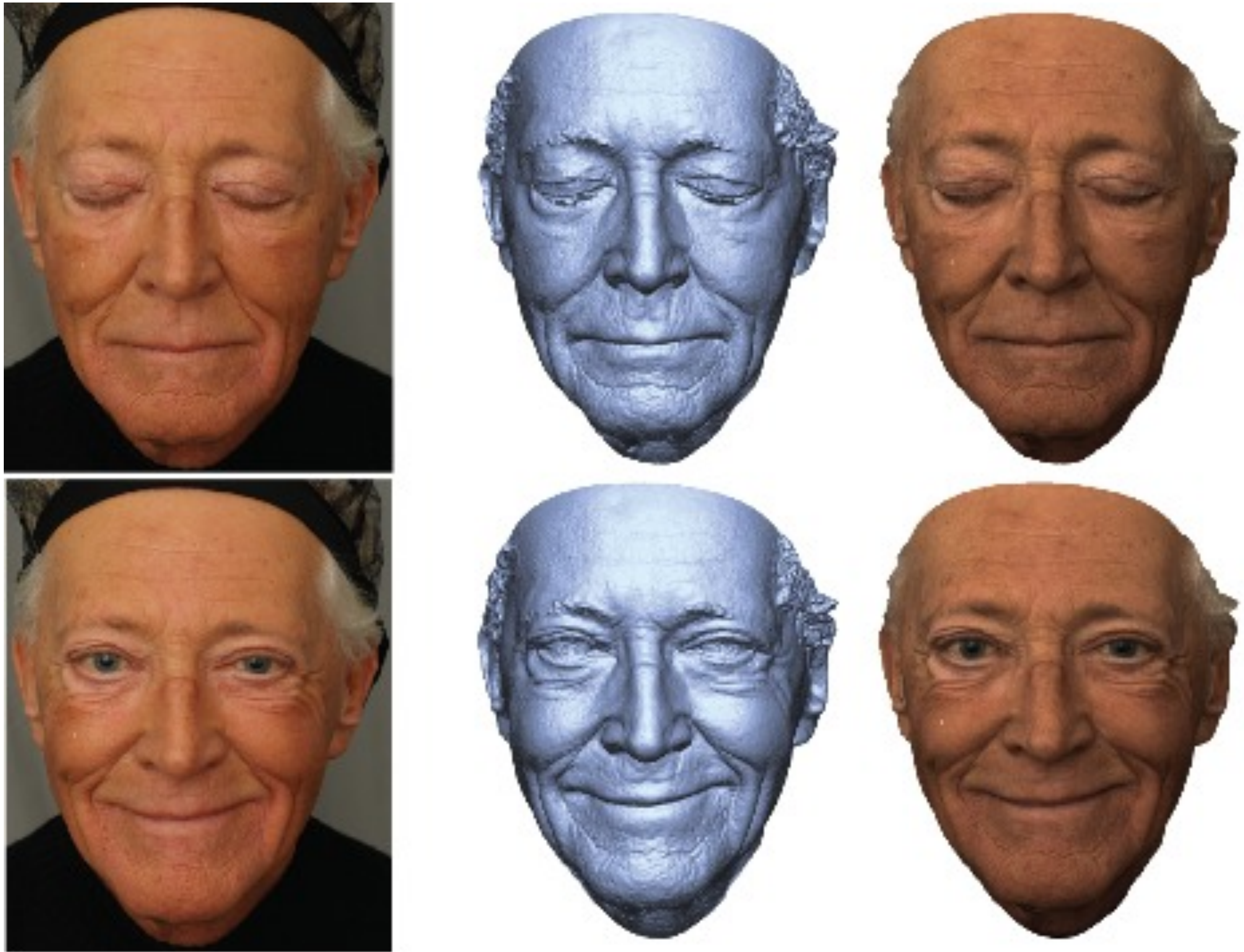
SenseFly:
www.sensefly.com

Gatewing:
www.gatewing.com

MATTERHORN

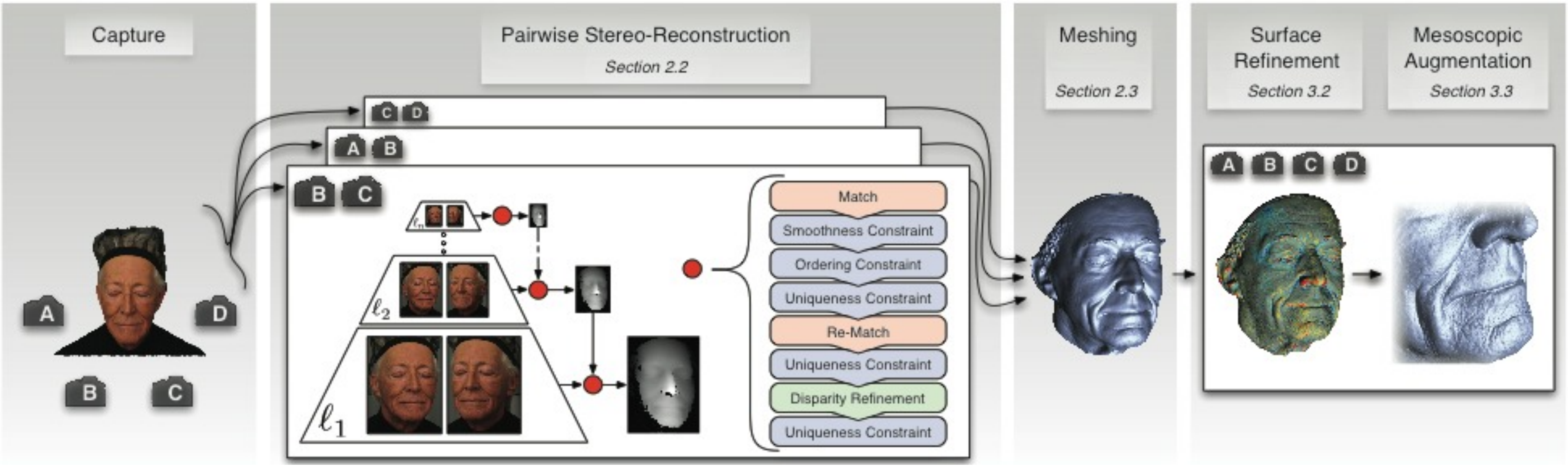
Drone: www.sensefly.com Mapping: www.pix4d.com

FACE RECONSTRUCTION



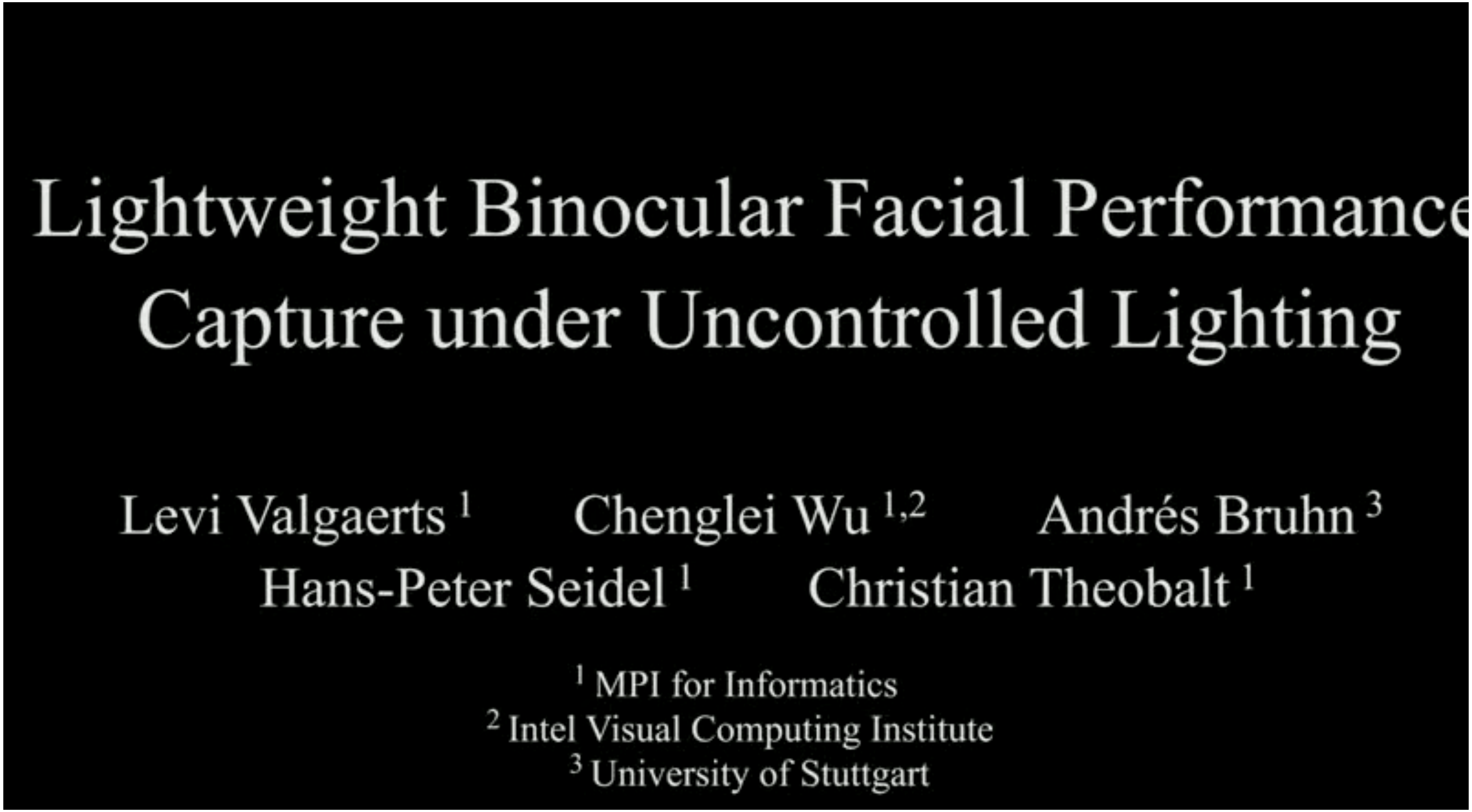
Beeler et al. SIGGRAPH'10

FACE RECONSTRUCTION



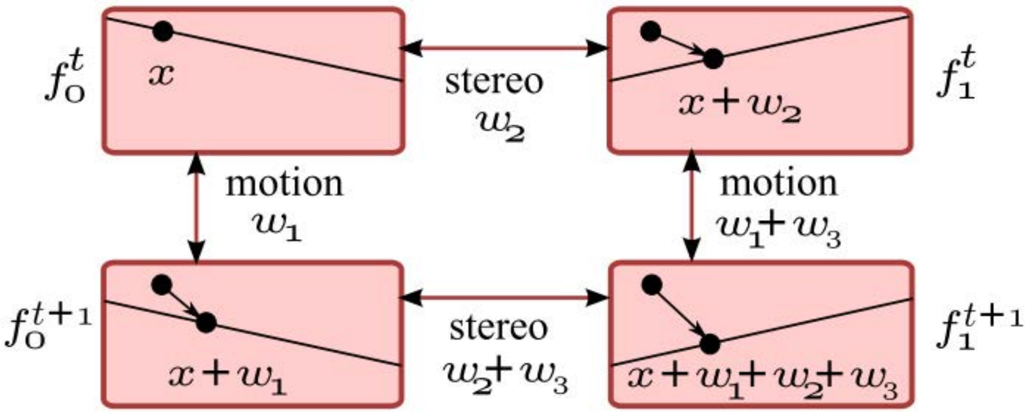
Beeler et al. SIGGRAPH'10

DYNAMIC SHAPE

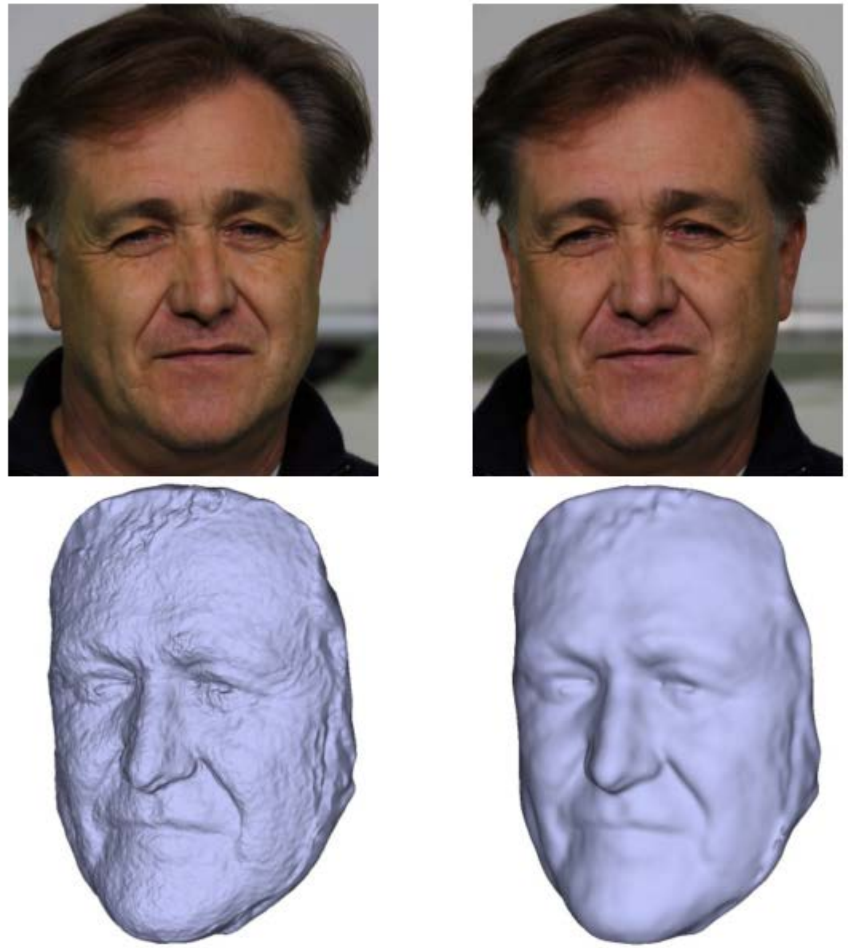


Valgaerts et al. SIGGRAPH Asia'12

SCENE FLOW



Correspondences across cameras and across time

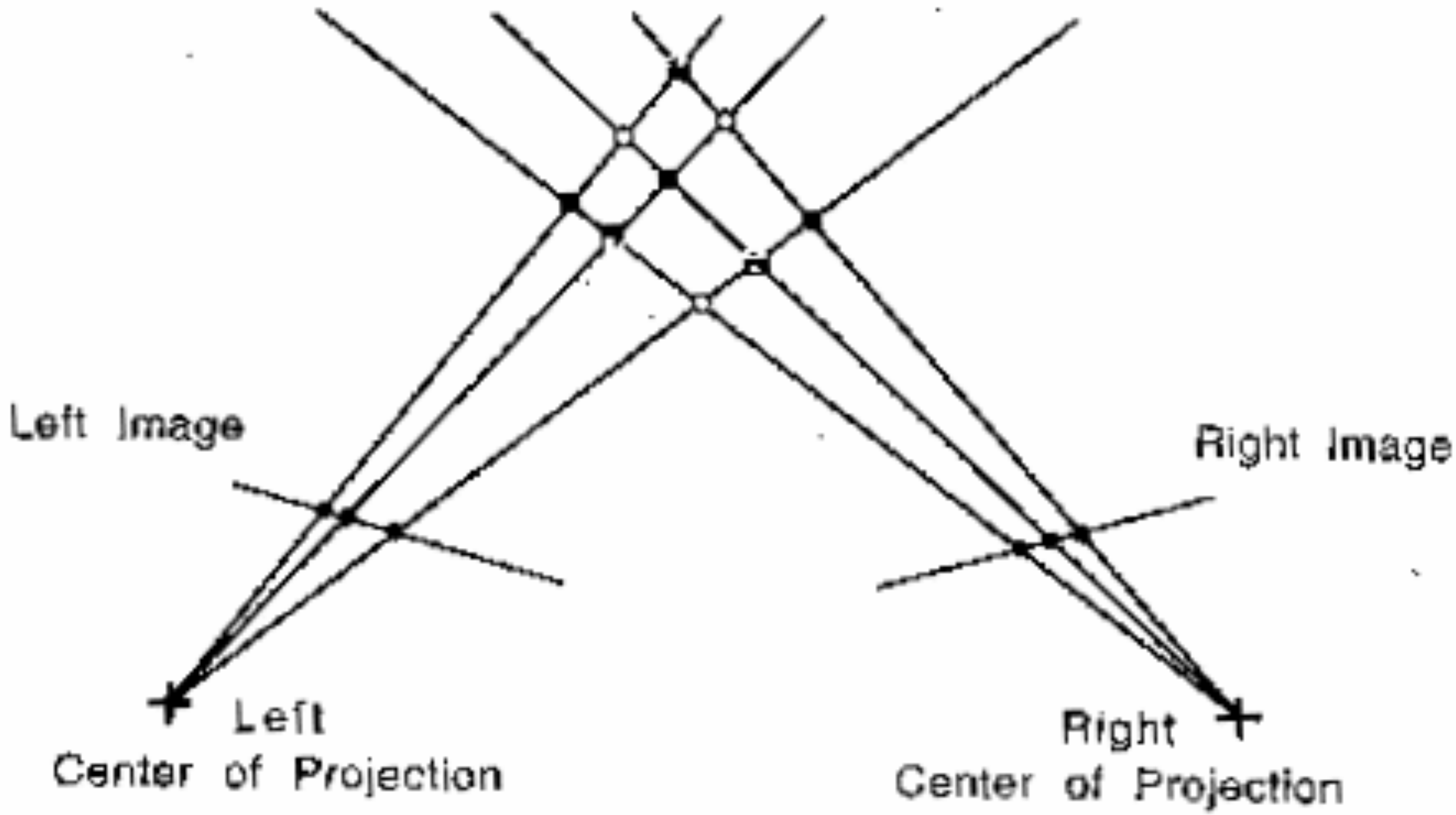


Stereo Only   Stereo + Flow

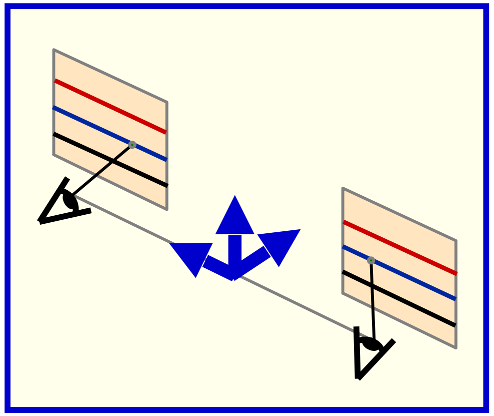
SHAPE FROM SHADING

Shape-from-shading is used to refine the shape and provide high-frequency details.

UNCERTAINTY



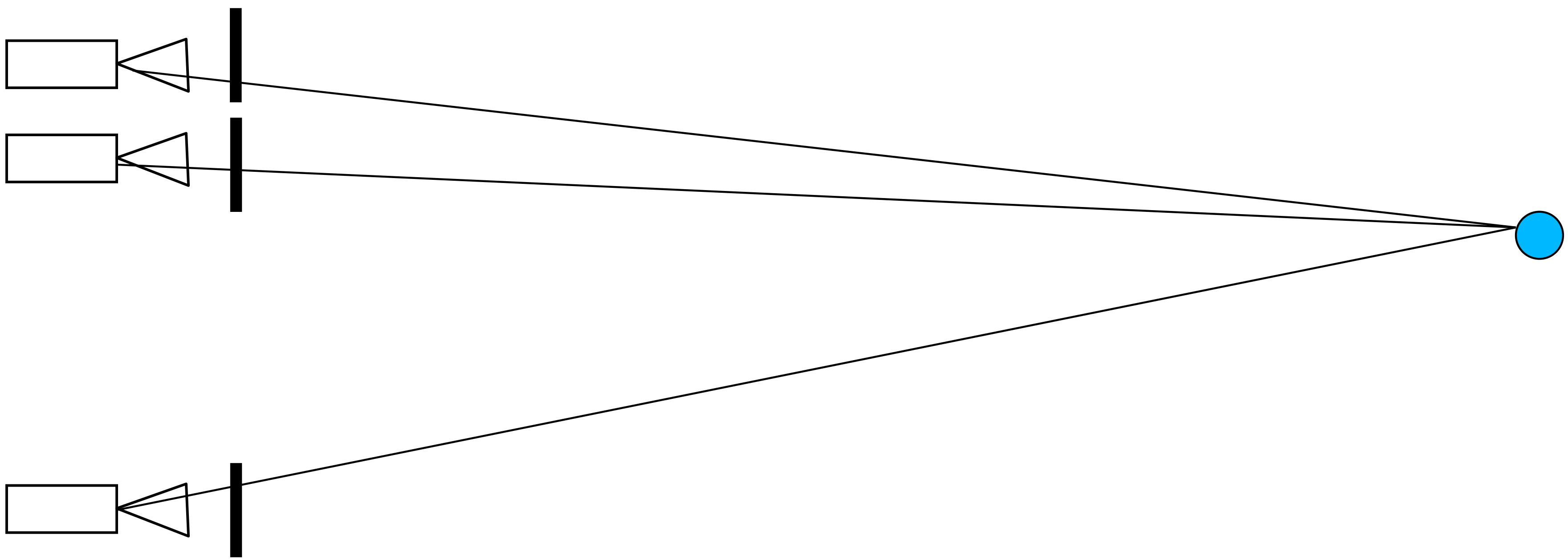
PRECISION vs BASELINE



$$
\begin{aligned}
d &= f\frac{b}{Z} \\
\Rightarrow Z &= f\frac{b}{d} \\
\Rightarrow \frac{\delta Z}{\delta d} &= -f\frac{b}{d^2} = -\frac{Z^2}{fb}
\end{aligned}
$$

- Beyond a certain depth stereo stops being useful.
- Precision is inversely proportional to baseline length.

SHORT vs LONG BASELINE



Long baseline:
- Harder to match
- More occlusions
- Better precision

Short baseline:
- Good matches
- Few occlusions
- Poor precision

MARS ROVER

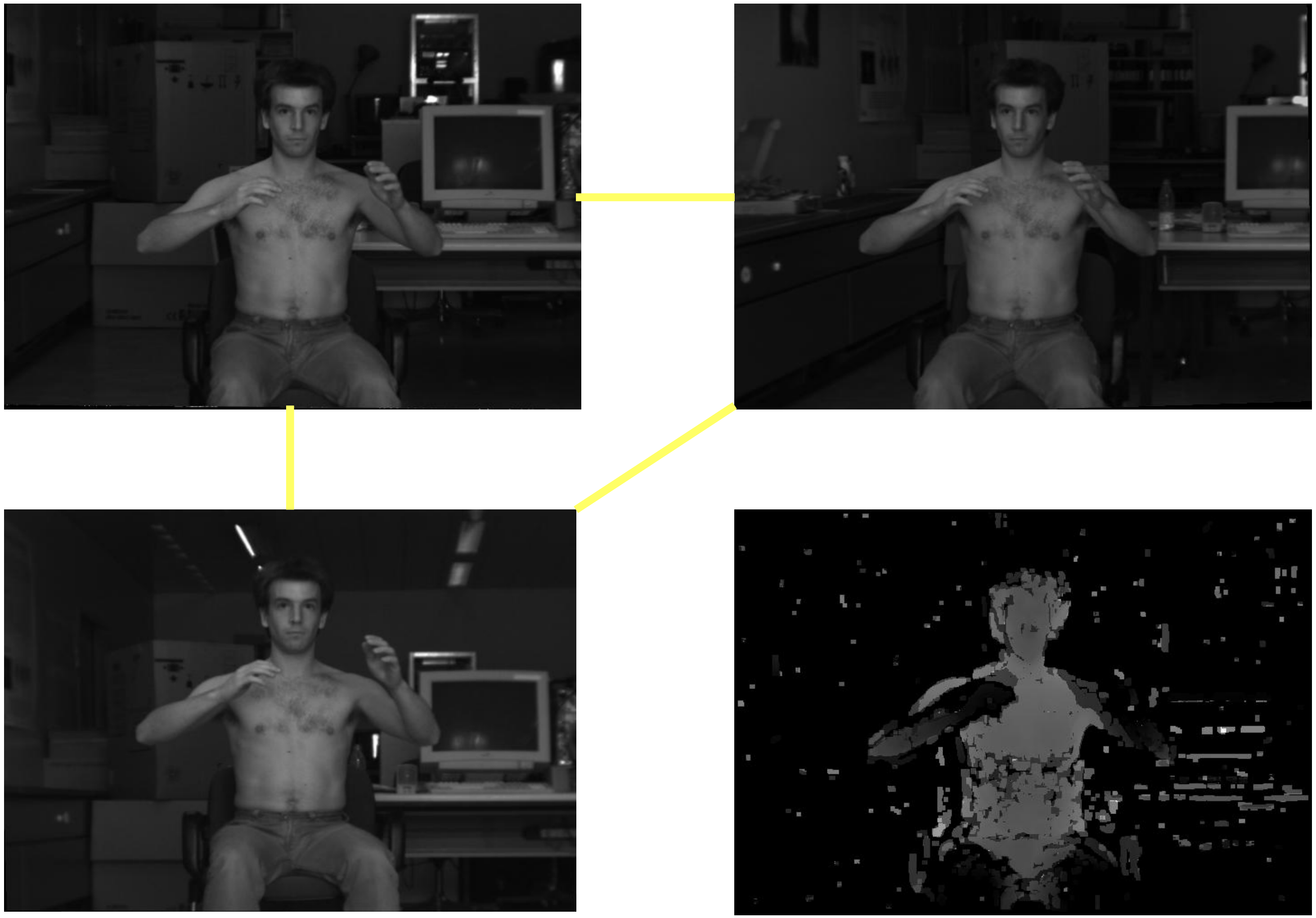
There are four cameras!

VIDEO-BASED MOTION CAPTURE

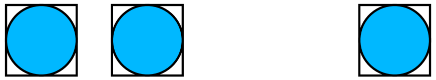
Fitting an articulated body model to stereo data.

Plankers & Fua, PAMI'03

TRINOCULAR STEREO

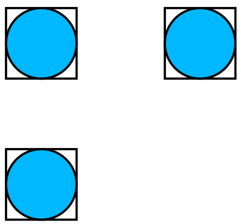


MULTI-CAMERA CONFIGURATIONS



3 cameras give both robustness and precision

4 cameras give additional redundancy

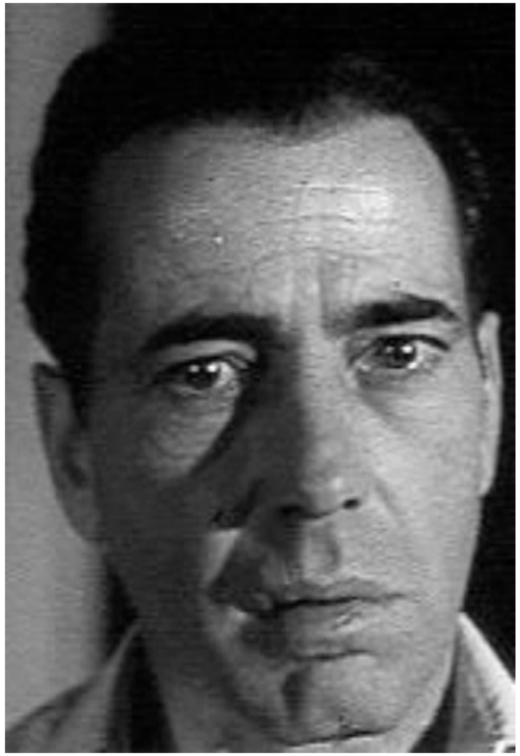


3 cameras in a T arrangement allow the system to see vertical lines.

KINECT: STRUCTURED LIGHT

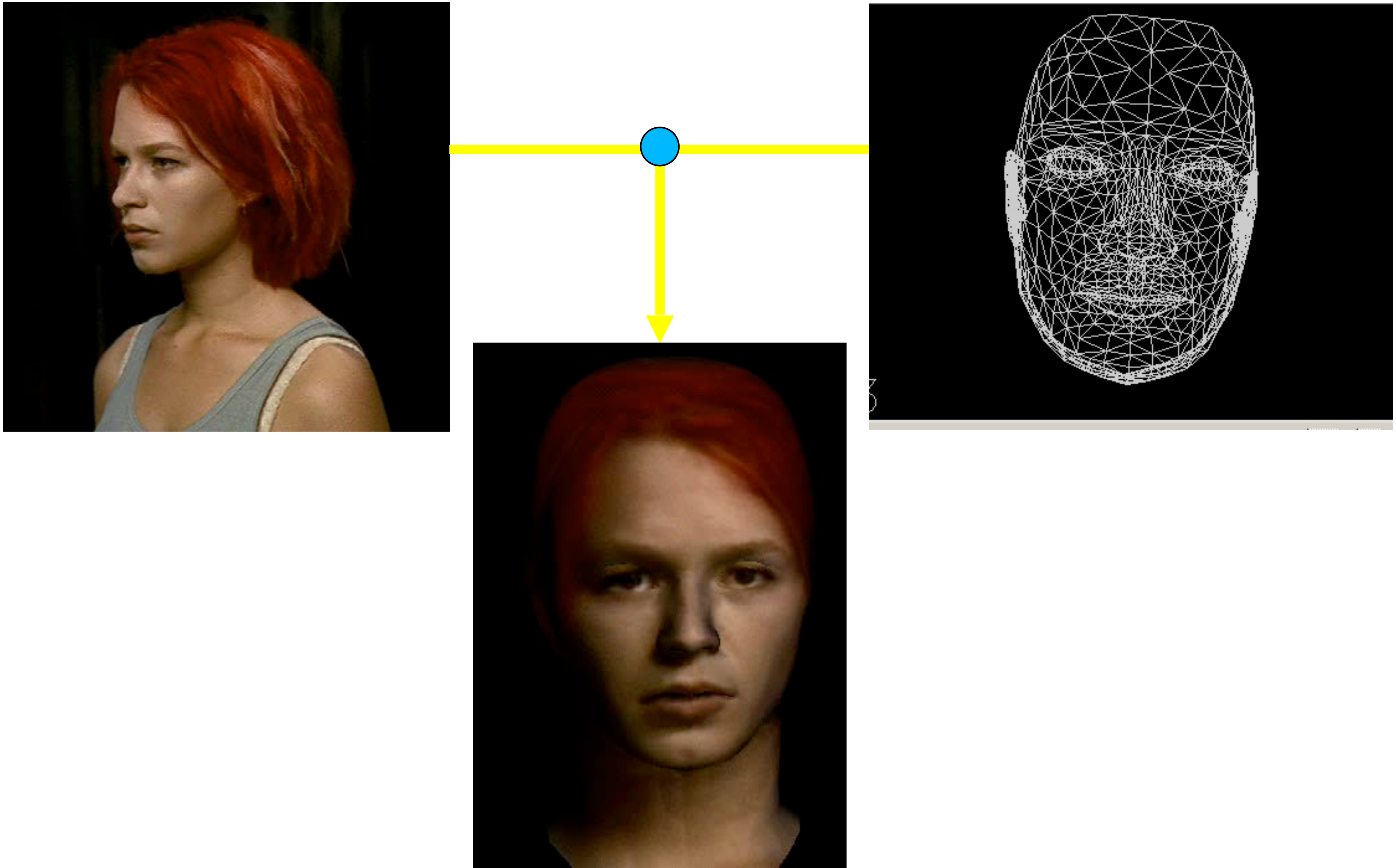
- The Kinect camera projects a IR pattern and measures depth from its distortion.
- Same principle but the second camera is replaced by the projector.

FACES FROM LOW-RESOLUTION VIDEOS



- No calibration data
- Relatively little texture
- Difficult lighting

SIMPLE FACE MODEL



PCA FACE MODEL

$$S = \overline{S} + \sum_{i=1}^{99} a_i S_i$$

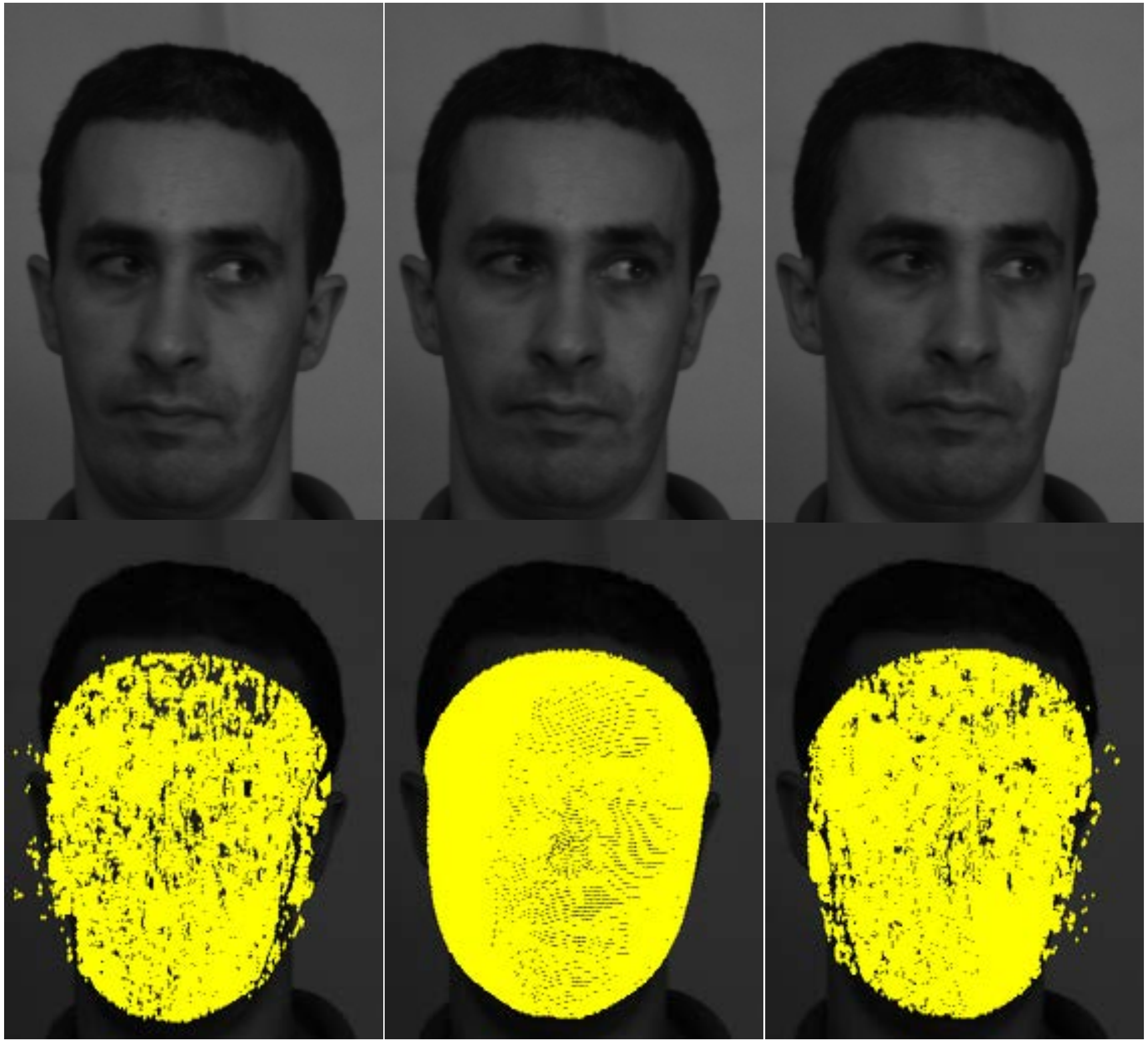
$\overline{S}$: Average shape

S_i : Shape vector

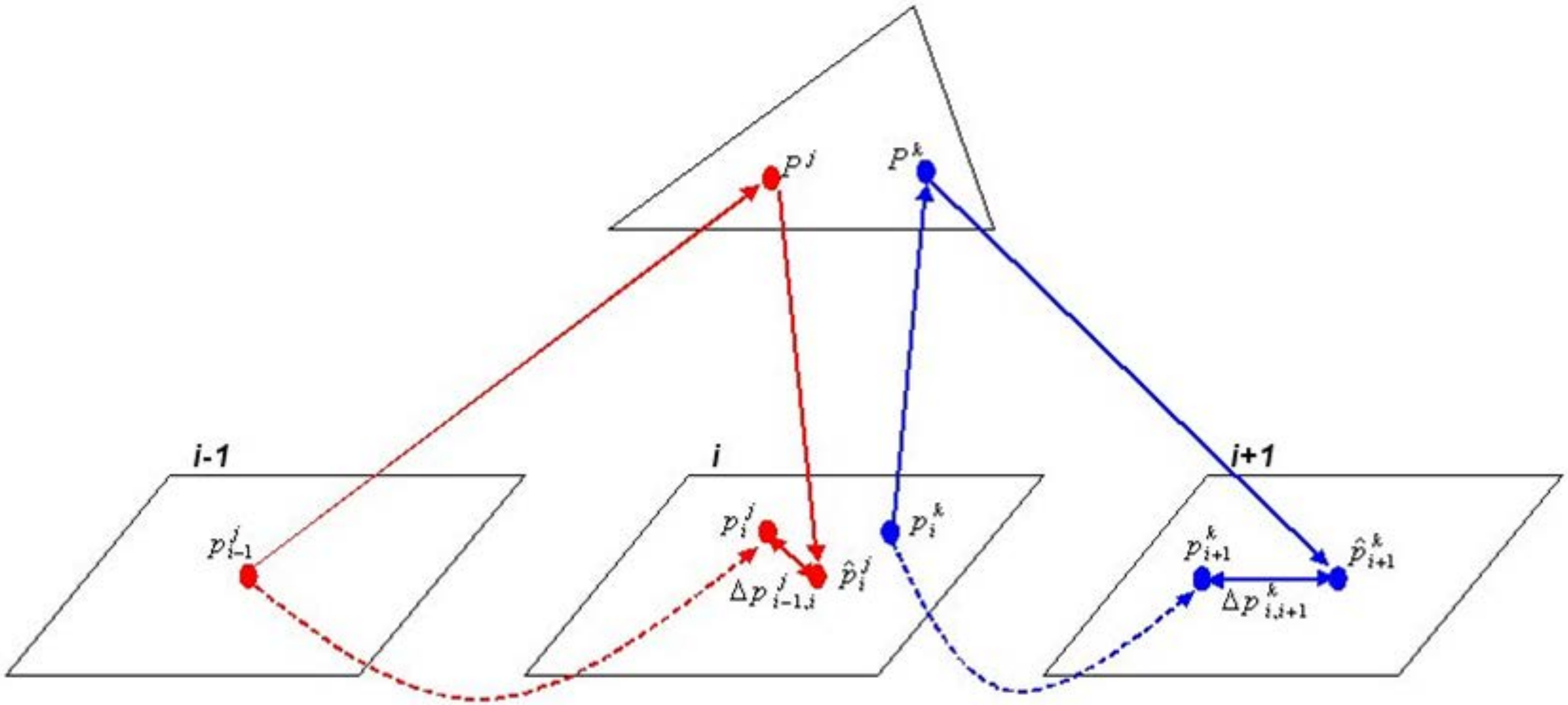
a_i : Shape coefficients

Blanz&Vetter, SIGGRAPH'99

CORRESPONDENCES

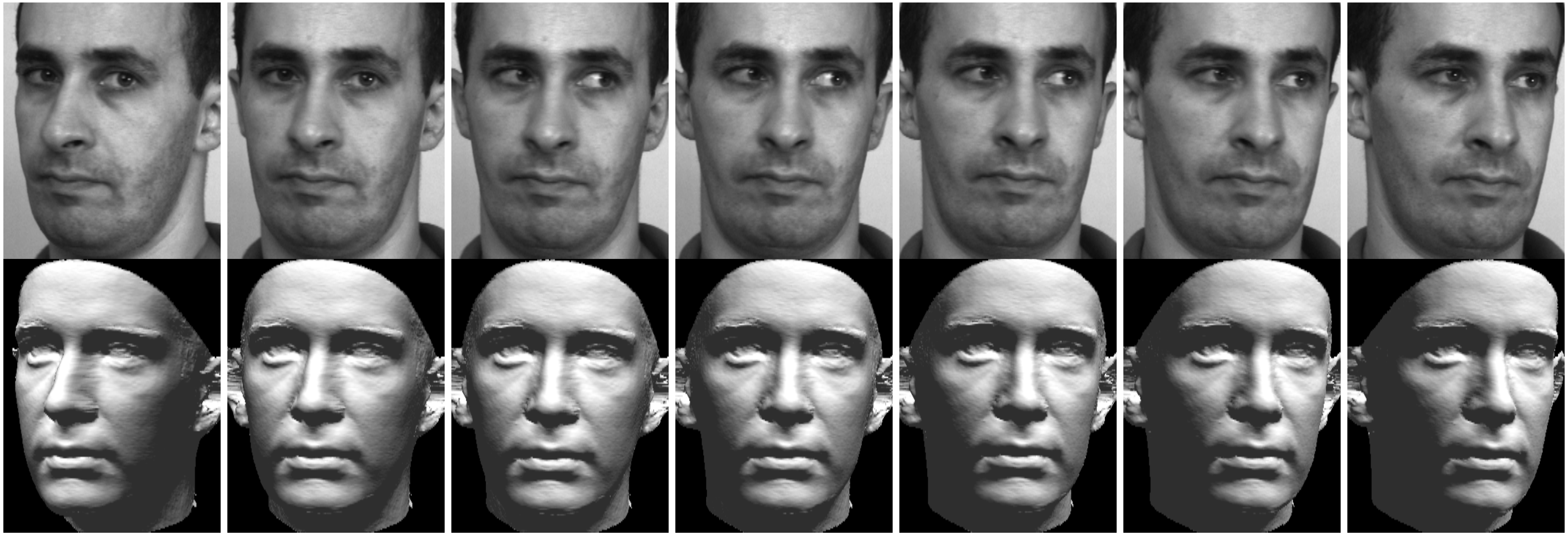


TRANSFER FUNCTION



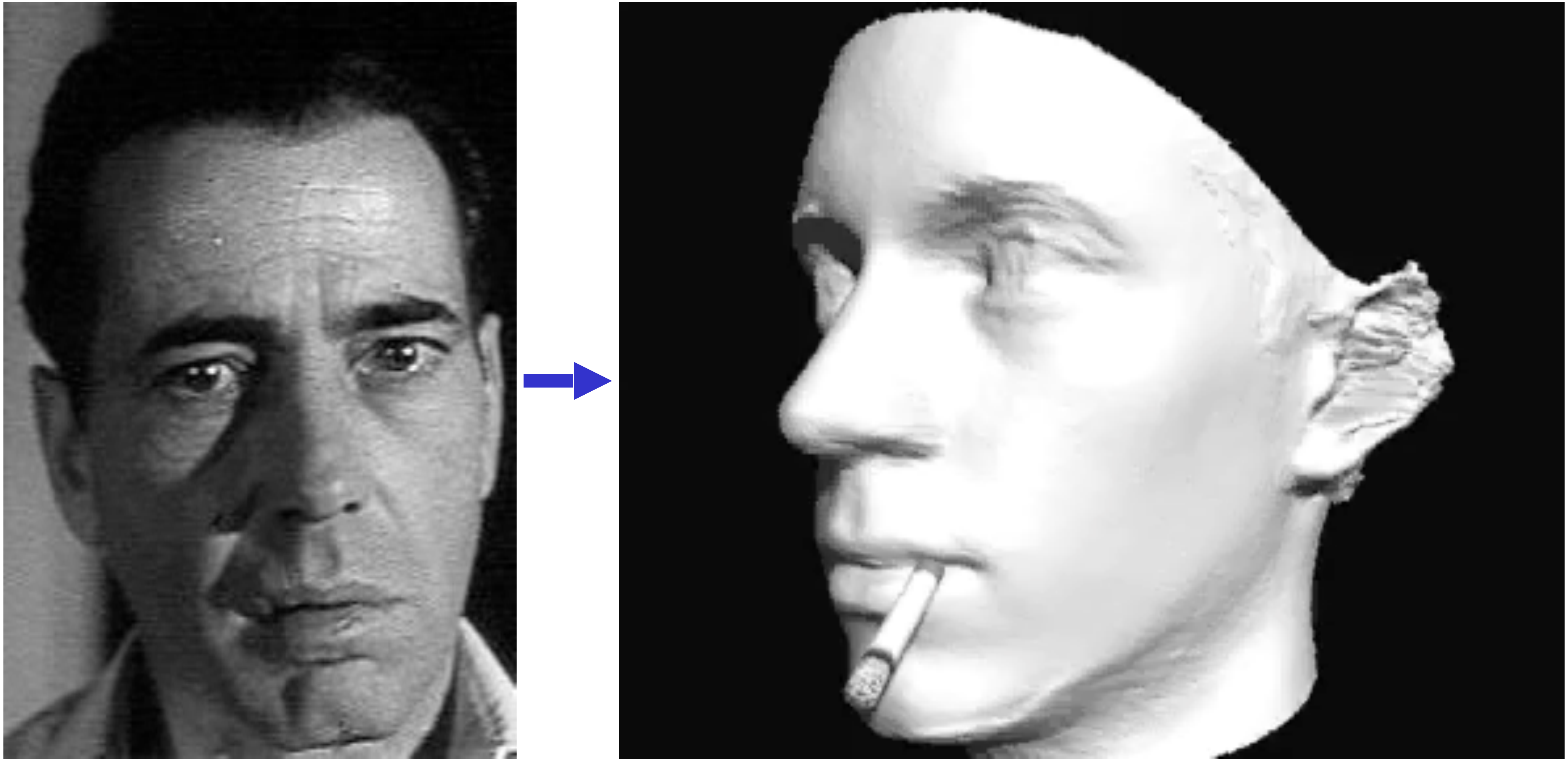
$$F_3(A, C_{i-1}, C_i, C_{i+1}) = \sum_{j \in Q_{i-1}} \left\| \Delta p^j_{i-1,i} \right\|^2 + \sum_{k \in Q_i} \left\| \Delta p^k_{i,i+1} \right\|^2$$

MODEL BASED BUNDLE ADJUSTMENT

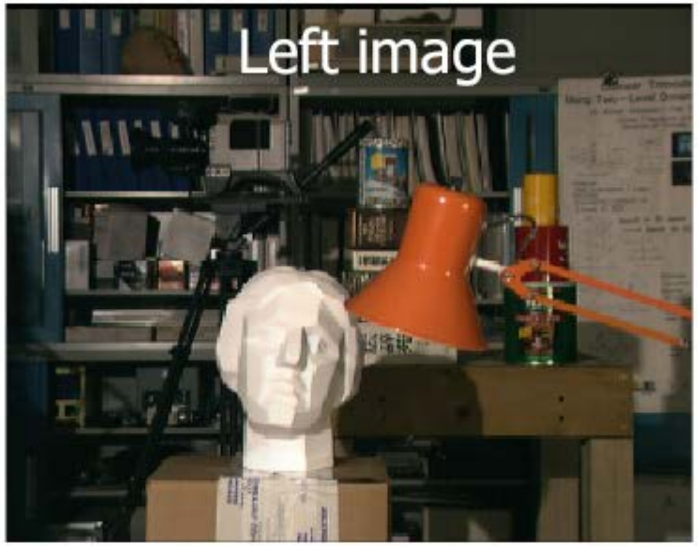


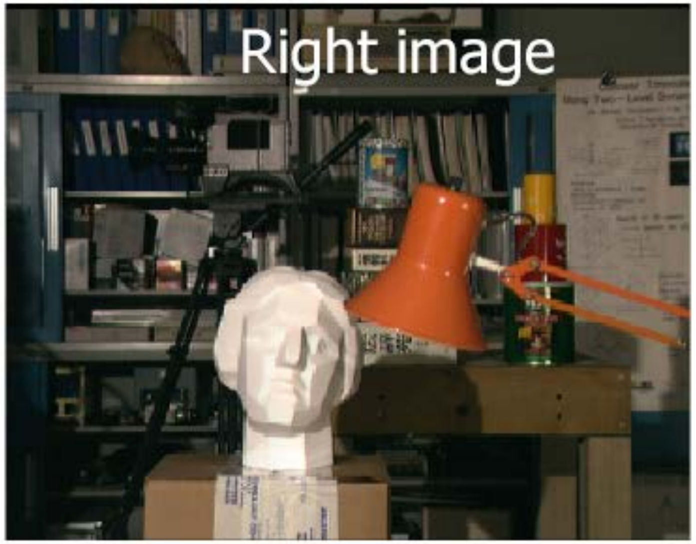
→ Median accuracy greater than 0.5mm

MODEL FROM OLD MOVIE

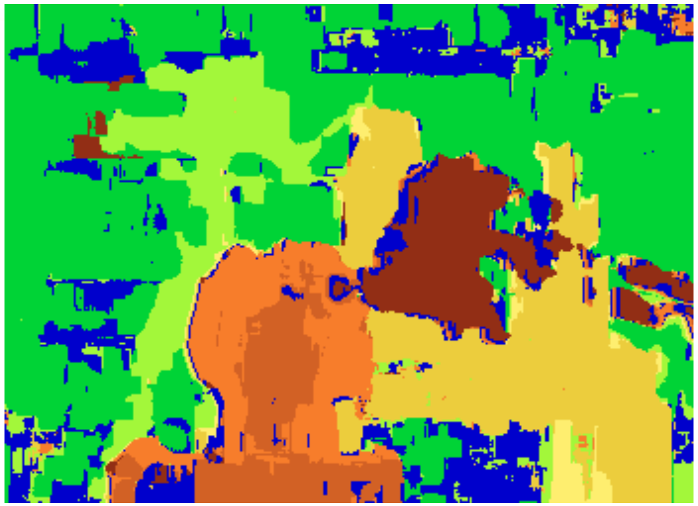


LIMITATIONS OF WINDOW BASED METHODS





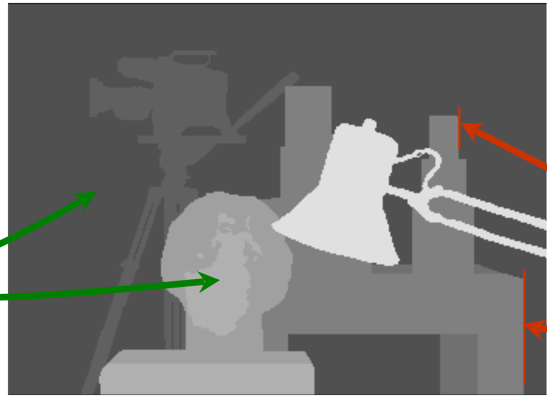
Ground truth



Correlation result

ENERGY MINIMIZATION

Disparity continuous in most places,



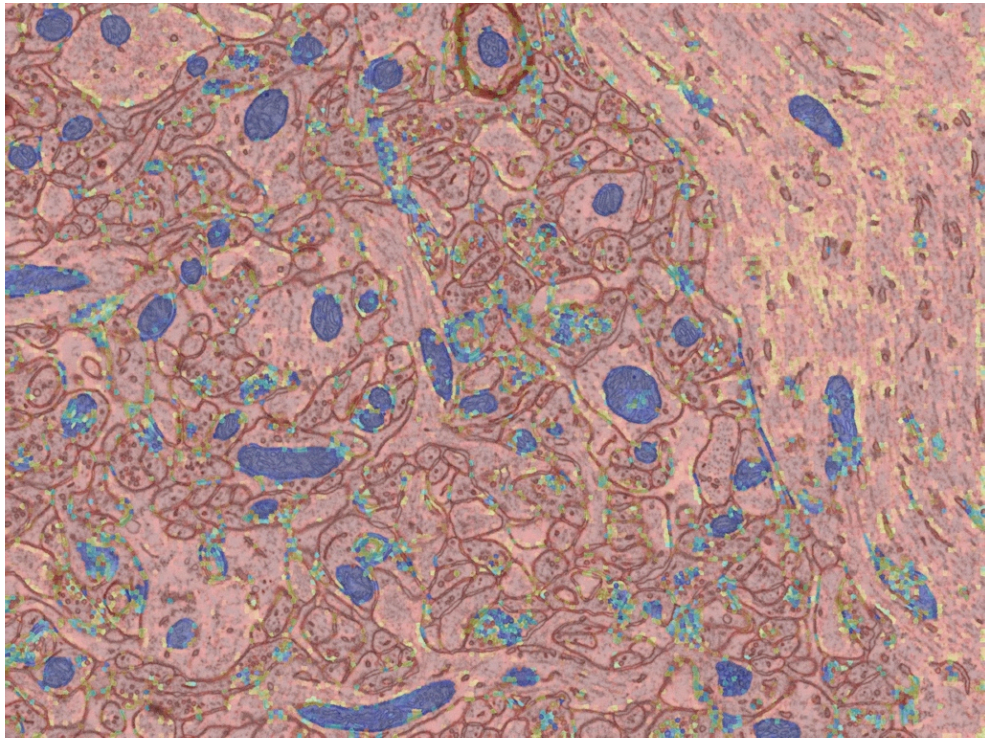
except at depth discontinuities

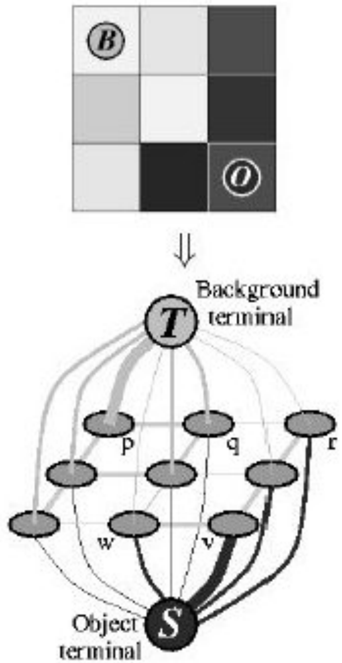
1. Matching pixels should have similar intensities.
2. Most nearby pixels should have similar disparities

→ Minimize

$$\sum [I_2(x+D(x,y),y)-I_1(x,y)]^2+\lambda \sum [D(x+1,y)-D(x,y)]^2+\mu \sum [D(x,y+1)-D(x,y)]^2$$

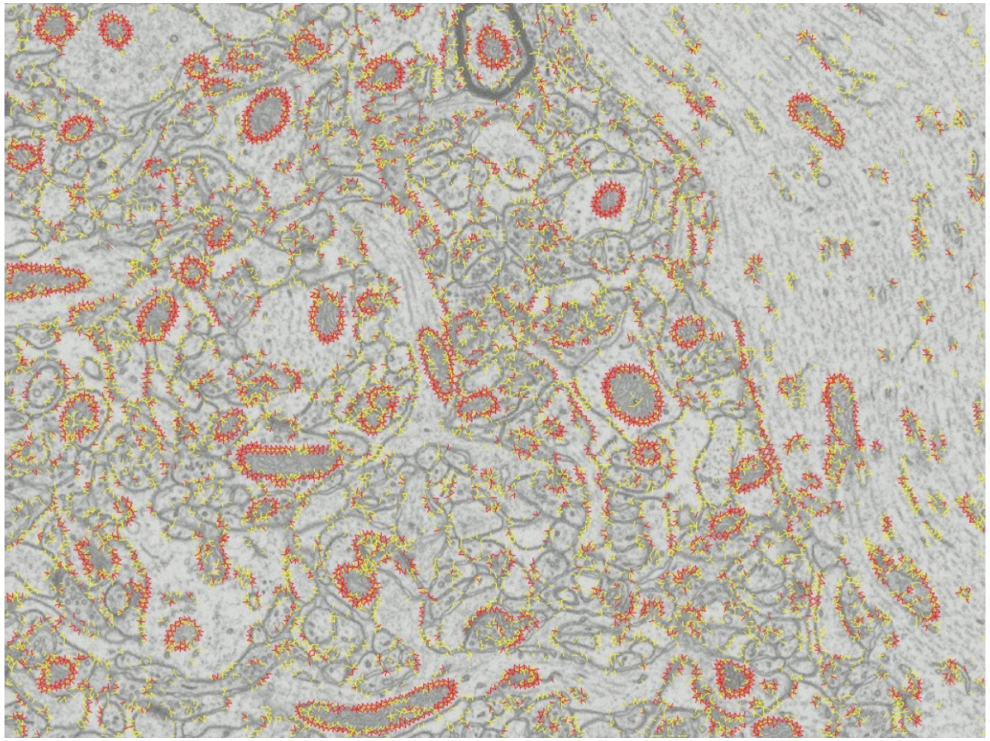
MITOCHONDRIA REMINDER

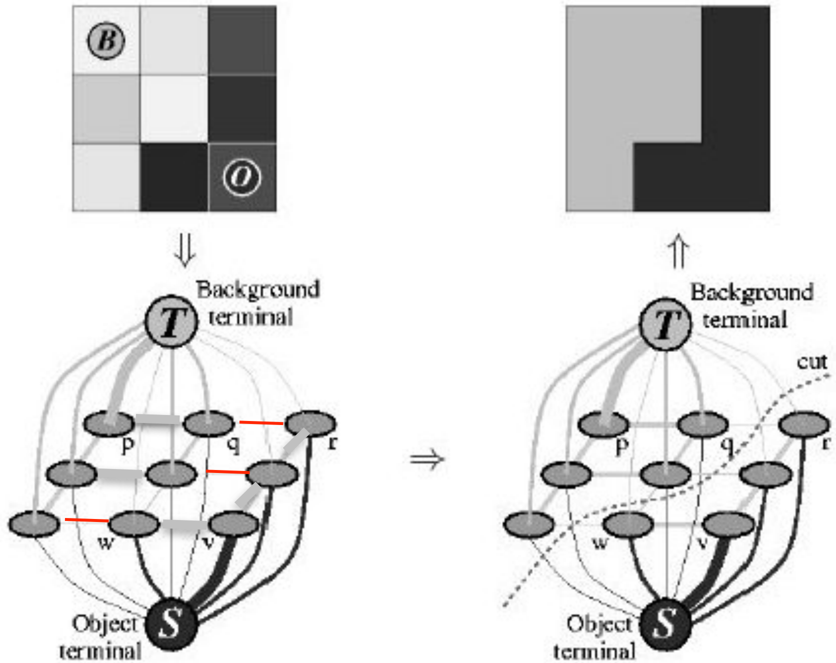




- A high probability of being a mitochondria can be represented by a strong edge connecting a supervoxel to the source and a weak one to the sink.
- And conversely for a low probability.

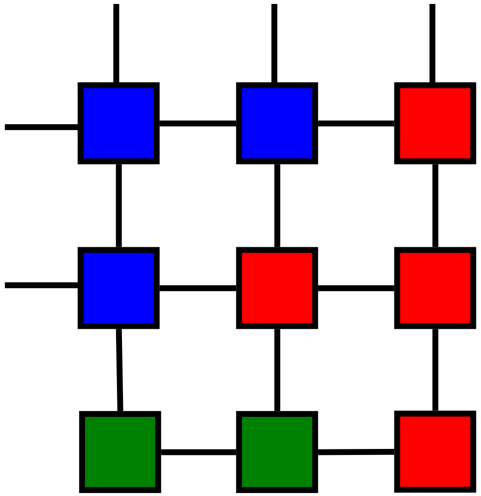
MITOCHONDRIA REMINDER





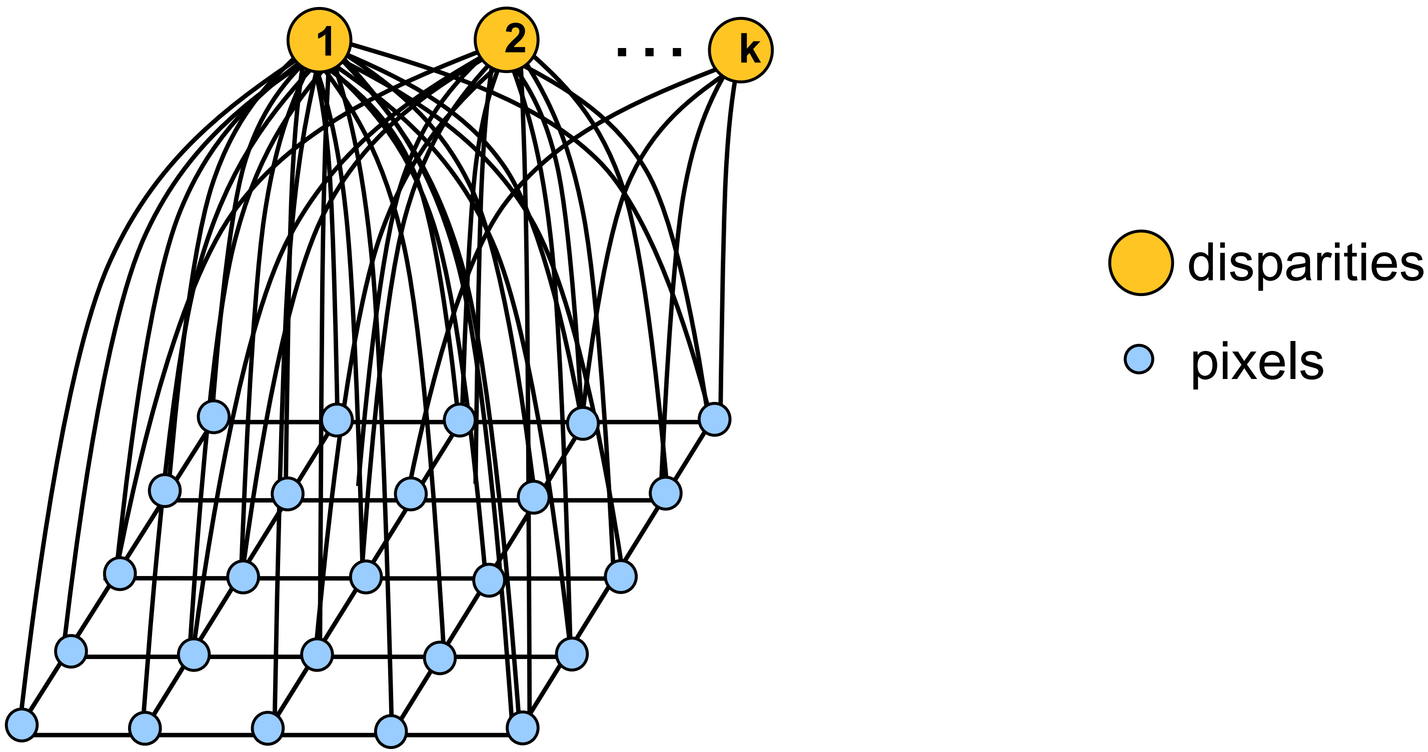
- Another classifier can be trained to assign a high-weight to edges connecting supervoxels belonging to the same class and a low one to others.

GRAPH CUTS



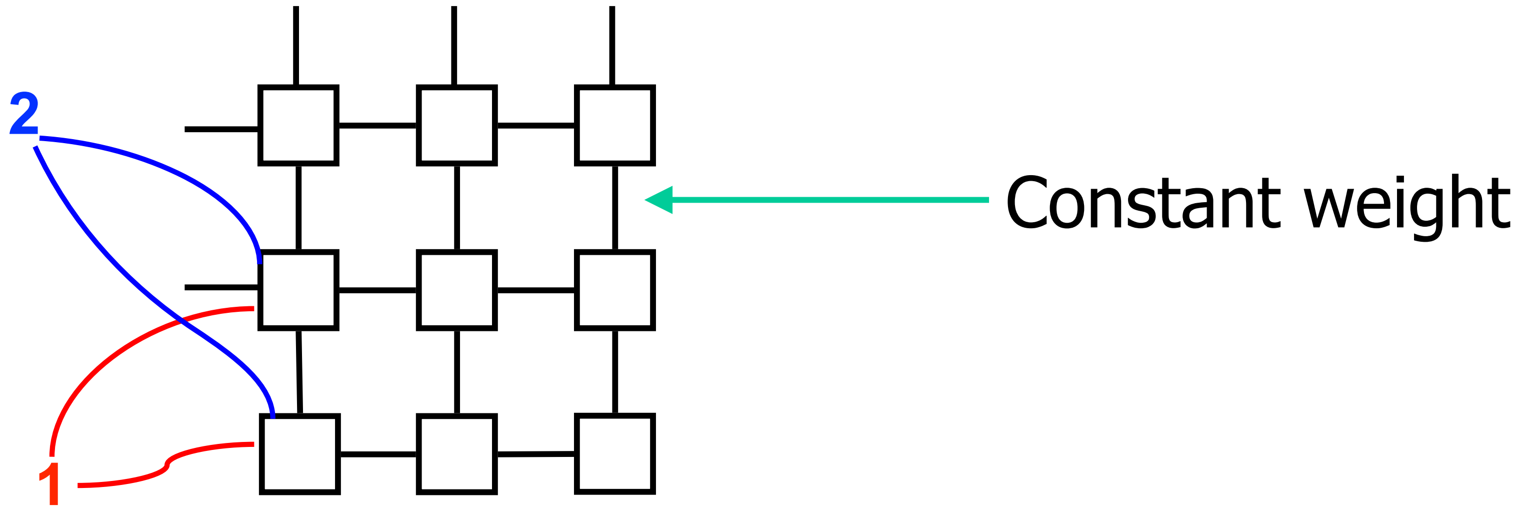
1. Stereo is a labeling problem. —> Use graph cut.
2. Connect each pixel to each possible disparity value.

Building The Graph



Connect each pixel to each possible disparity value.

Assigning Edge Weights



Assign a weight that is inversely proportional to |I2(x+1,y)-I1(x,y)|

Assign a weight that is inversely proportional to |I2(x+2,y)-I1(x,y)|

……

Minimizing the Objective Function

Minimize:

$$\sum[I_2(x+D(x,y),y)-I_1(x,y)]^2+\lambda\sum[D(x+1,y)-D(x,y)]^2+\mu\sum[D(x,y+1)-D(x,y)]^2$$

Graph cut algorithm:

- Guarantees an absolute minimum only when there are only two possible disparities.
- Effective heuristics (α-expansion, α–β swap) otherwise.

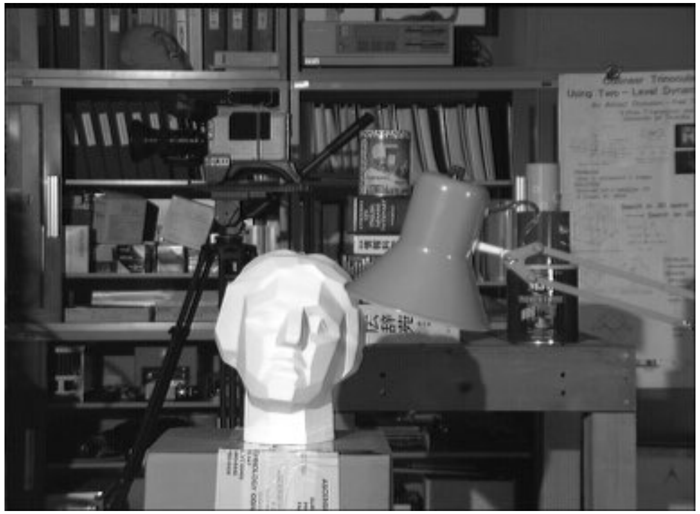
NCC vs GRAPH CUTS

Normalized correlation

Graph Cuts

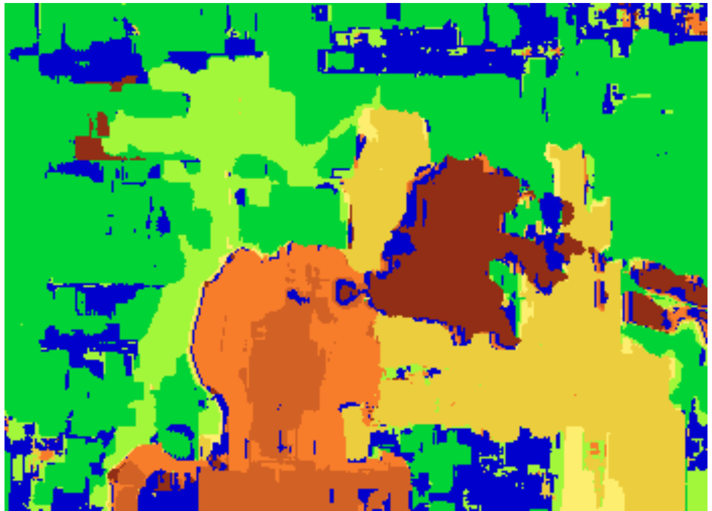
NCC vs GRAPH CUTS

left image

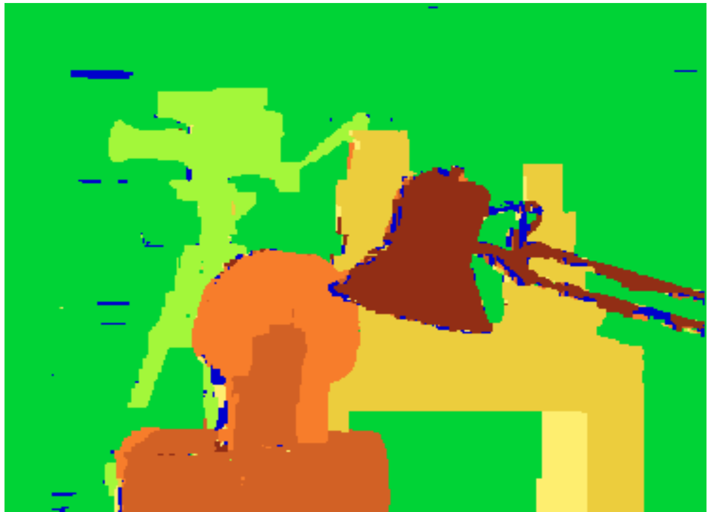


true disparities

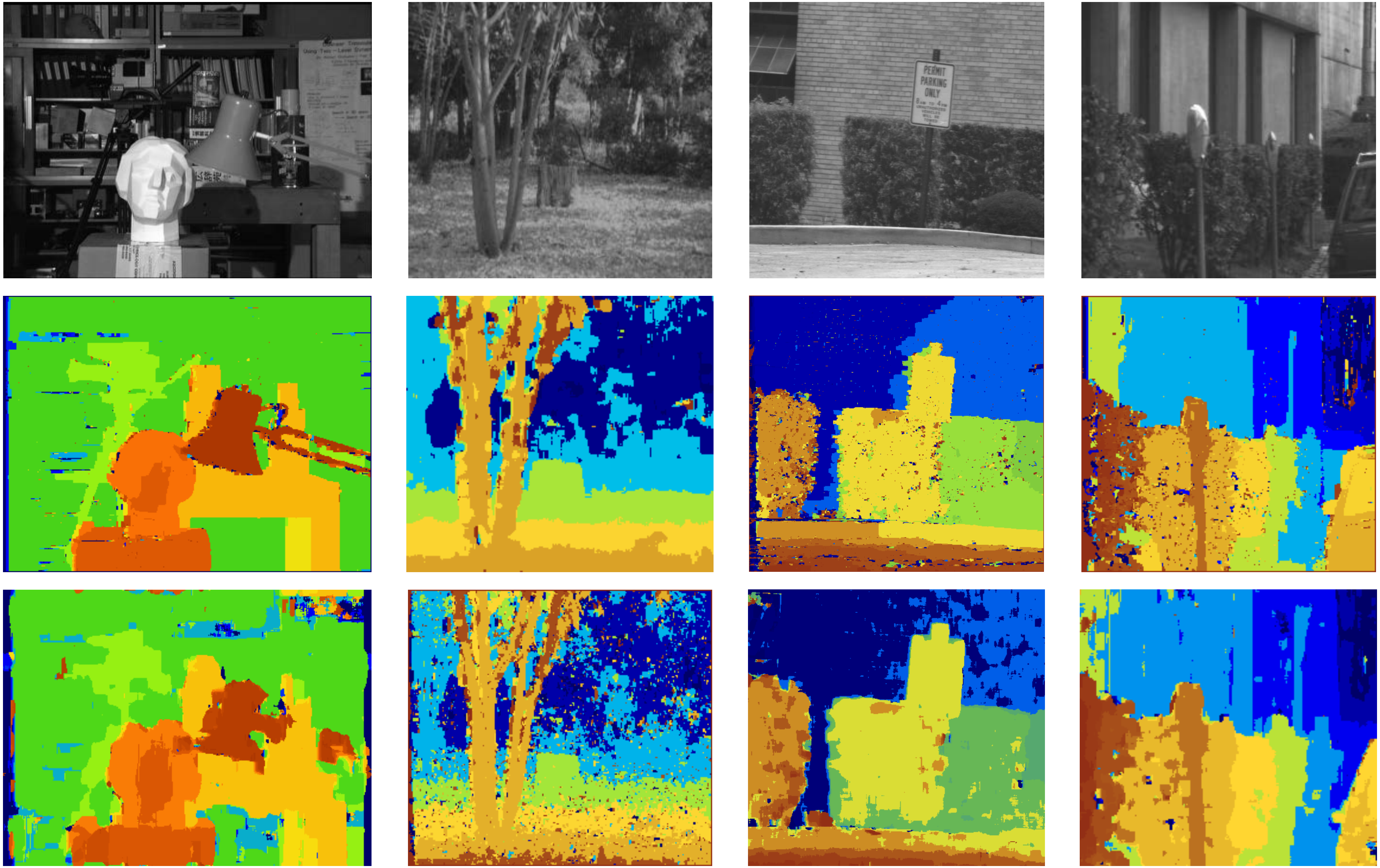
Normalized correlation



Graph Cuts



GRAPH CUT RESULTS



STRENGTHS AND LIMITATIONS

Strengths:

- Practical method for depth recovery.
- Runs in real-time on ordinary hardware.

Limitations:

- Requires multiple views.
- Only applicable to reasonably textured objects.