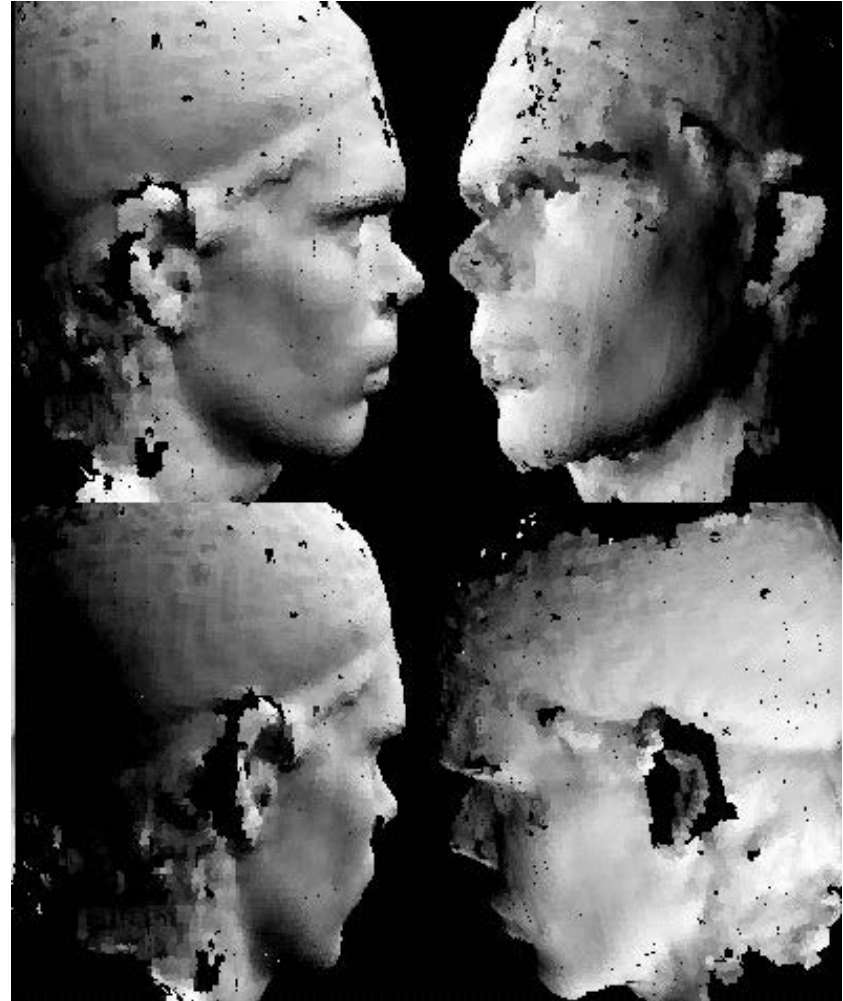
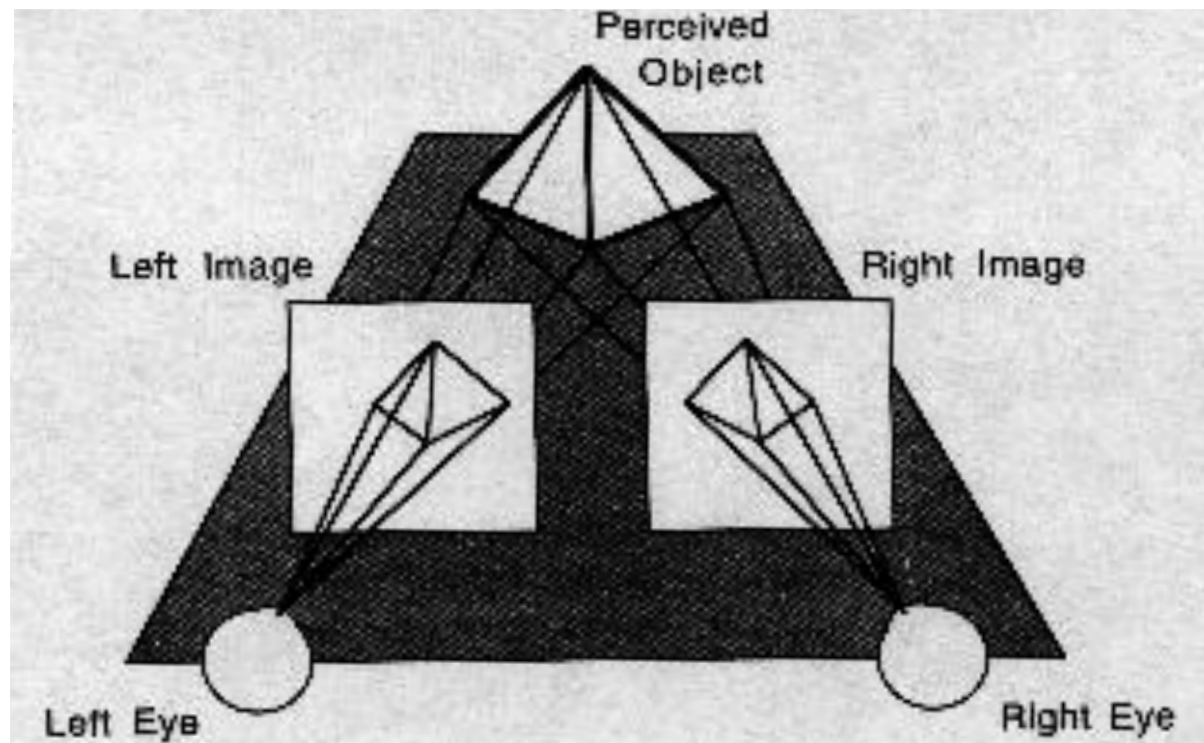


SHAPE FROM X

- One image:
 - Texture
 - Shading
- Two images or more:
 - **Stereo**
 - Contours
 - Motion



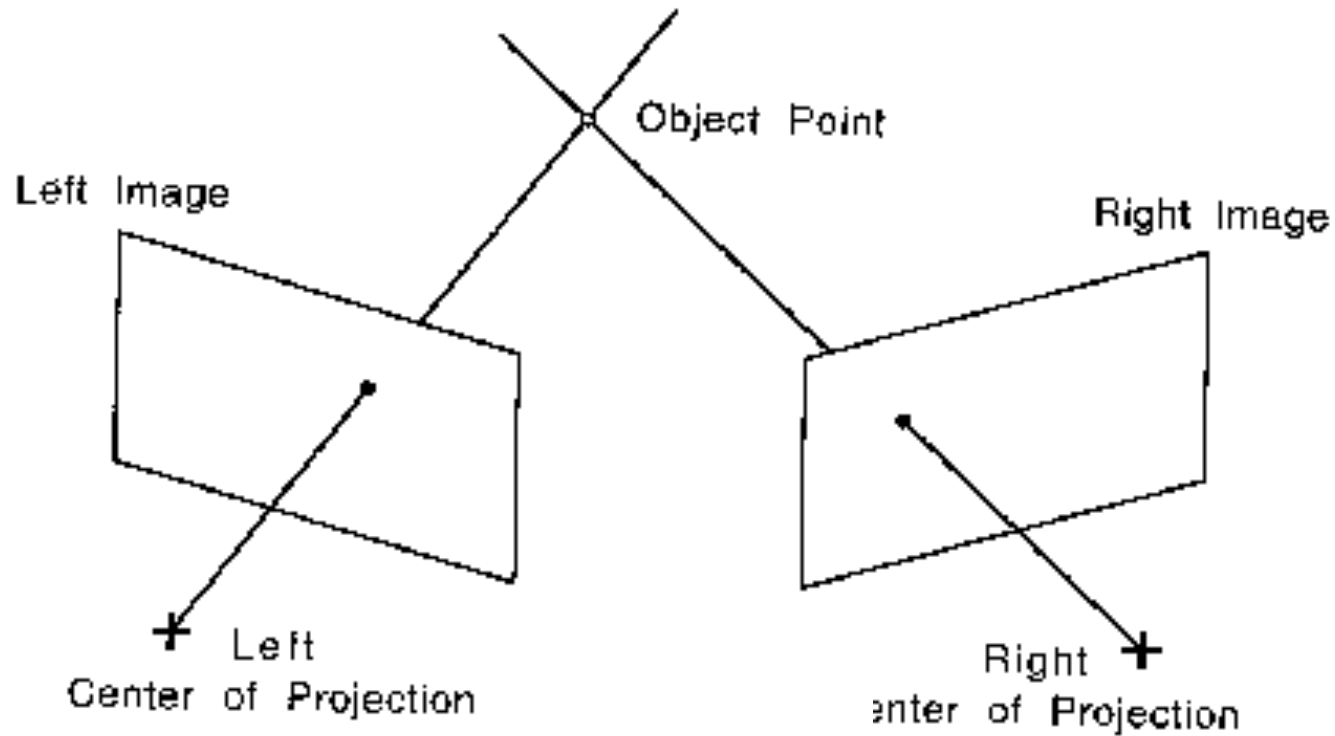
Geometric Stereo



Depth from two or more images:

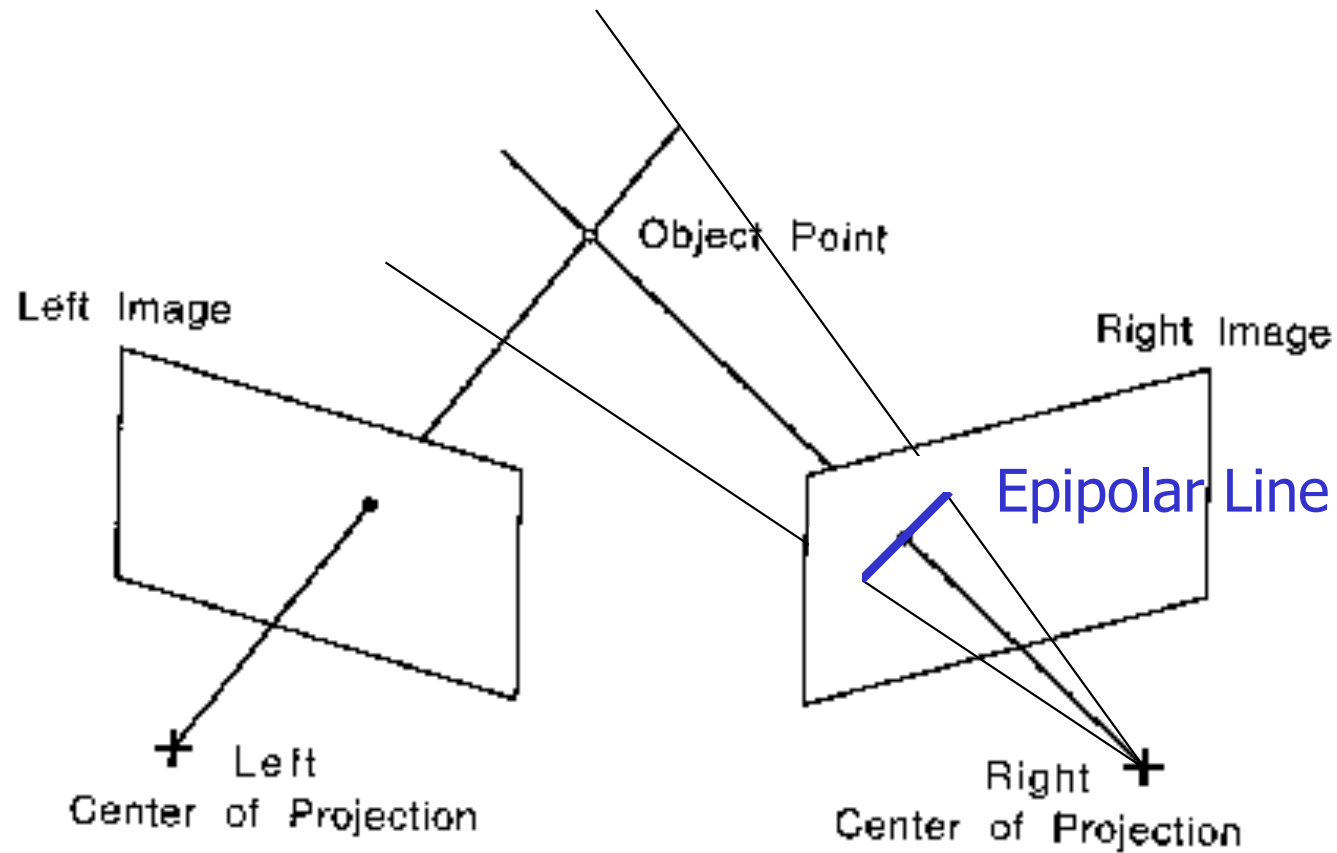
- Geometry of image pairs
- Establishing correspondences

Triangulation



Geometric Stereo: Depth from two images

Epipolar Line



Line on which the corresponding point must lie.

Epipolar Lines

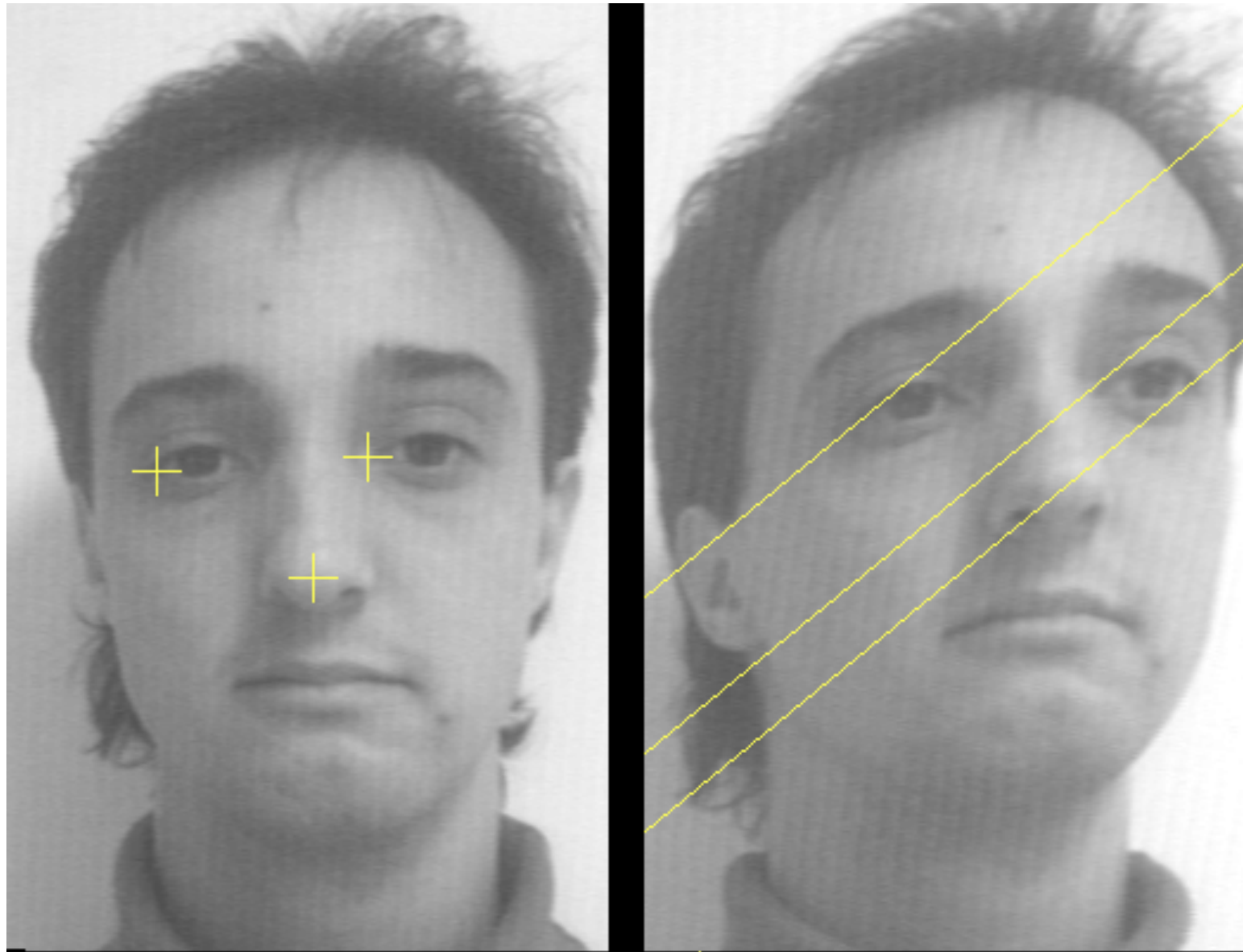


Three points shown
as red crosses.



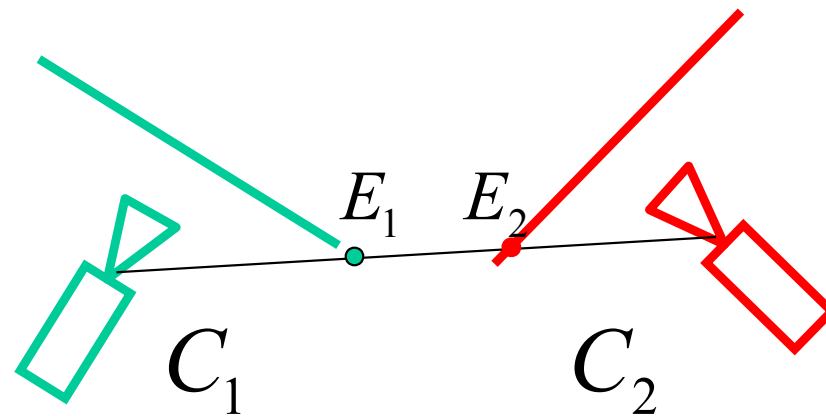
Corresponding epipolar
lines.

Epipolar Lines



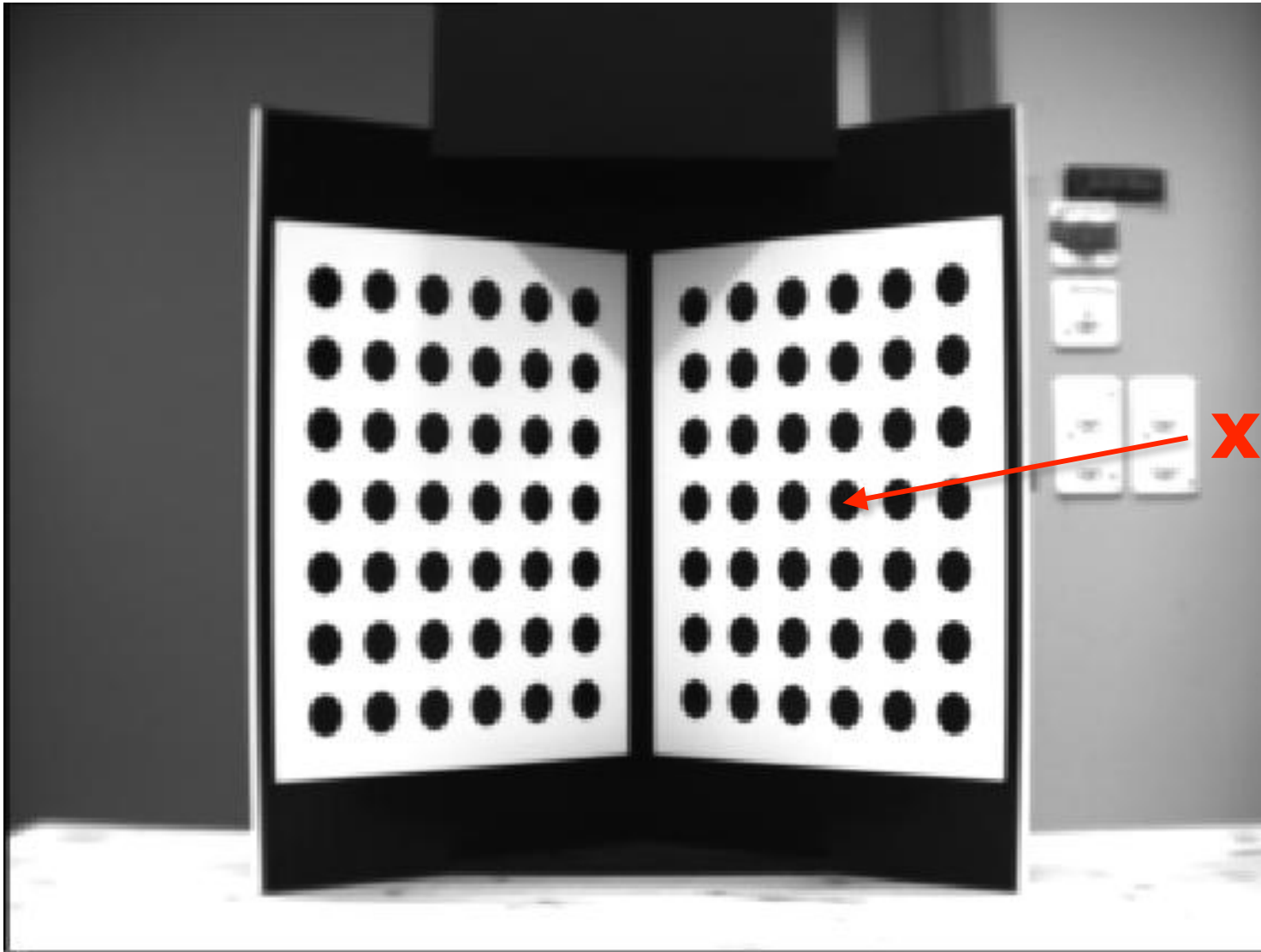
They can have any orientation.

Epipole



- Point at which **all** epipolar lines intersect:
- ➔ Located at the intersection of line joining optical centers and image plane.

Reminder: Calibration Grid



- Take a picture of a calibration grid with each camera.
- Infer the two projection matrices.
- Compute the epipolar lines.

Without a Calibration Grid

There is 3×3 matrix F such that for all corresponding points $\mathbf{x} \leftrightarrow \mathbf{x}'$

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0.$$

Therefore, the epipolar line corresponding to \mathbf{x} is $\mathbf{l} = \mathbf{F} \mathbf{x}$.

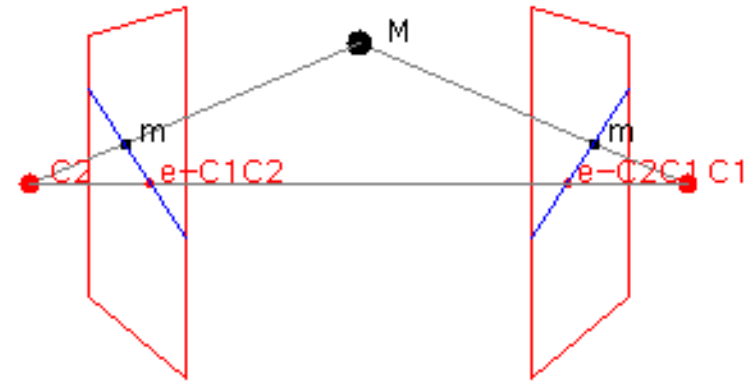
Given a set of n point matches, we write

$$\begin{bmatrix} u'_1 u_1 & u'_1 v_1 & u'_1 & v'_1 u_1 & v'_1 v_1 & v'_1 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u'_n u_n & u'_n v_n & u'_n & v'_n u_n & v'_n v_n & v'_n & u_n & v_n & 1 \end{bmatrix} \mathbf{f} = 0.$$

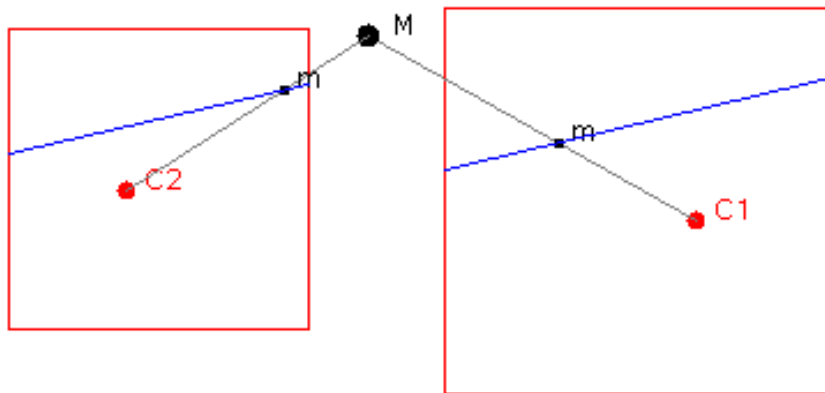
→ DLT or non – linear minimization.

Epipolar Geometry

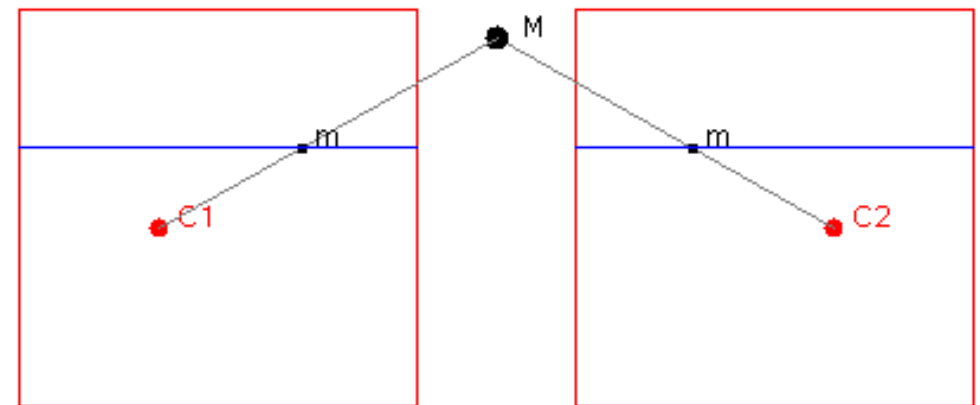
In general:



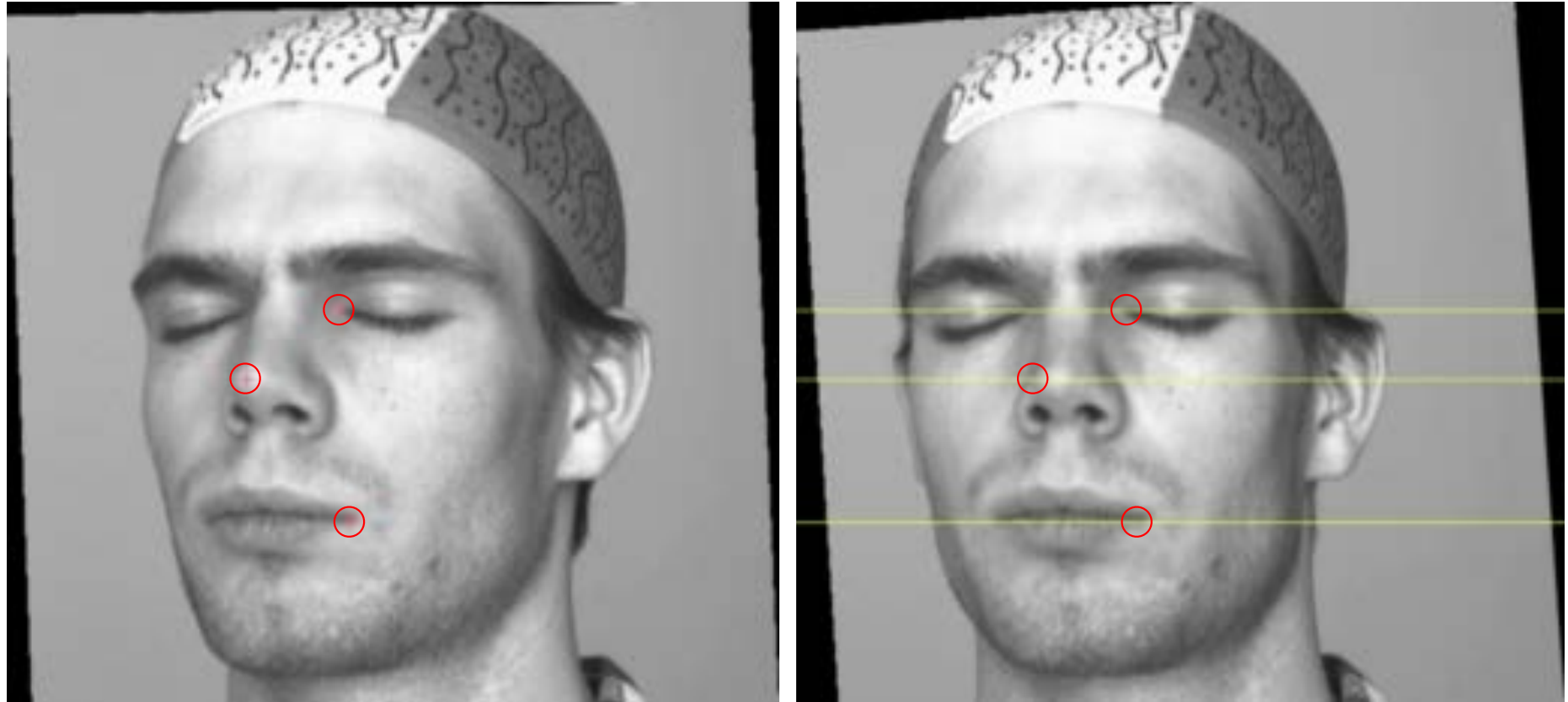
Parallel image planes



Horizontal baseline

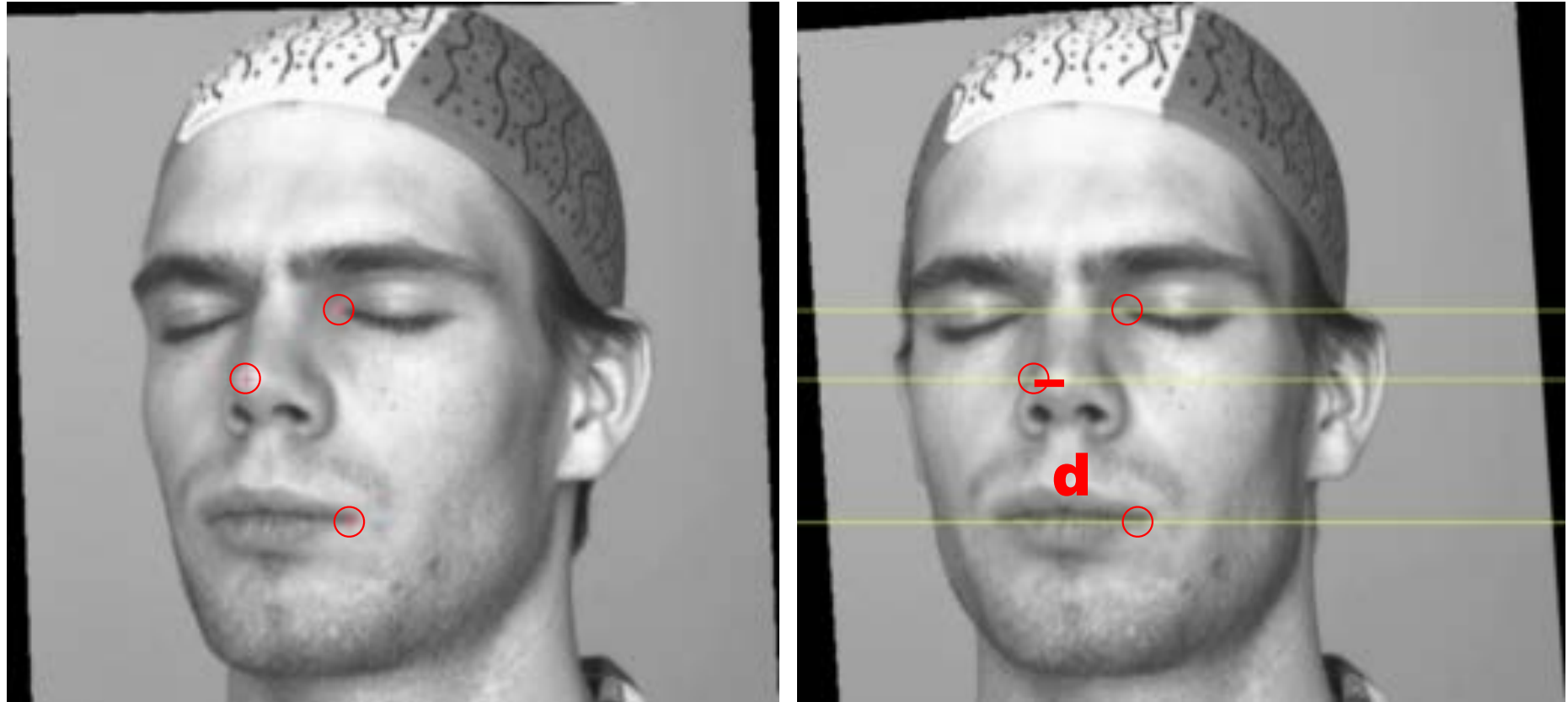


Rectification



From intersecting epipolar lines ...
... to parallel ones.

Disparity



The horizontal shift along an epipolar line, inversely proportional to distance.

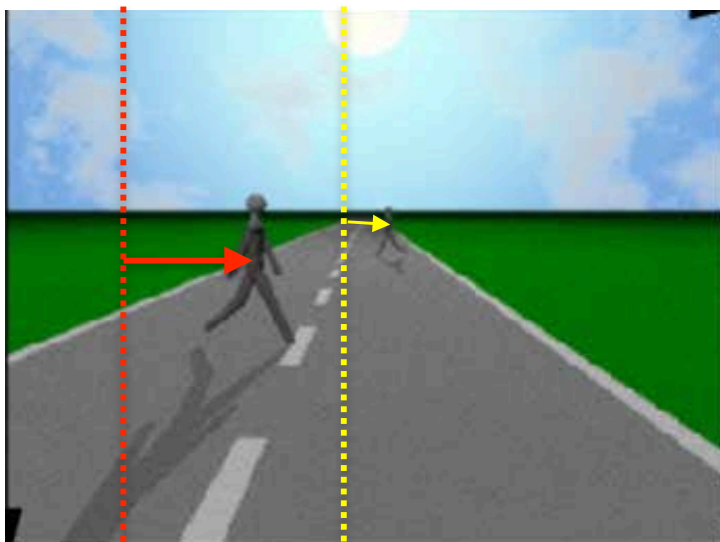
Superposed Images



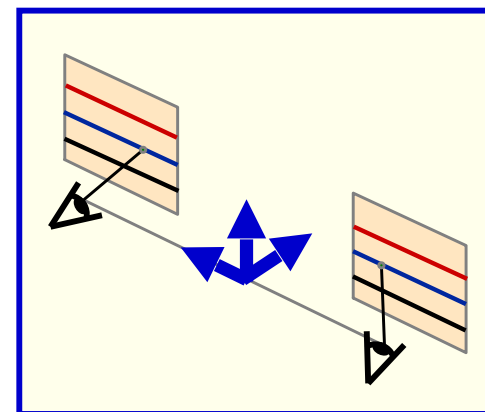
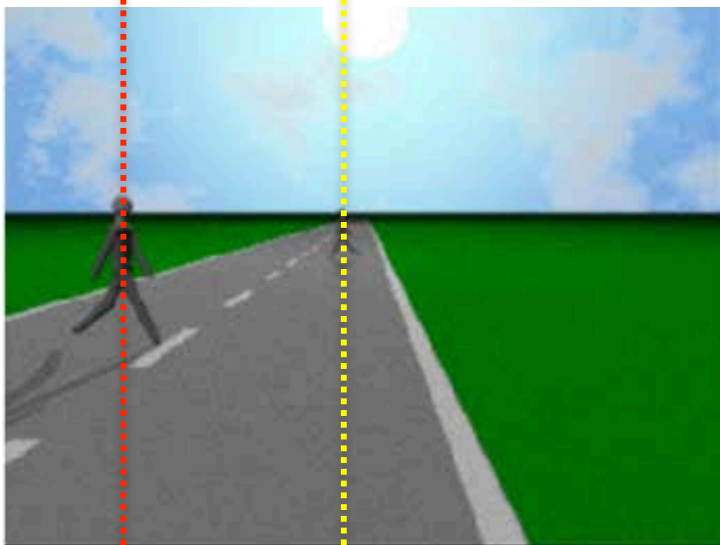
Red: Left Image
Cyan: Right Image

Disparity vs Depth

Left



Right



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

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

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

→ Disparity is inversely proportional to depth.

Window Based Approach to Establishing Correspondences



C_1 C_2 C_3

- 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.

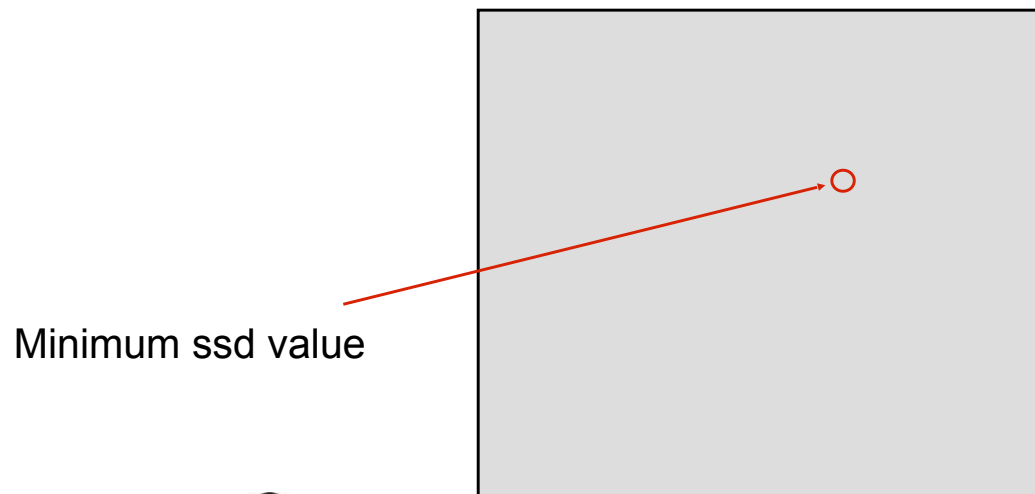
But how?

Sum of Square Differences

- Subtract pattern and image pixel by pixel and add squares:

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

- If identical $ssd=0$, otherwise $ssd > 0$
→ Look for minimum of ssd with respect to u and v .



Correlation

$$\begin{aligned} \text{ssd}(u,v) &= \sum_{(x,y) \in N} [I(u+x, v+y) - P(x,y)]^2 \\ &= \sum_{(x,y) \in N} I(u+x, v+y)^2 + \sum_{(x,y) \in N} P(x,y)^2 - 2 \sum_{(x,y) \in N} I(u+x, v+y)P(x,y) \end{aligned}$$

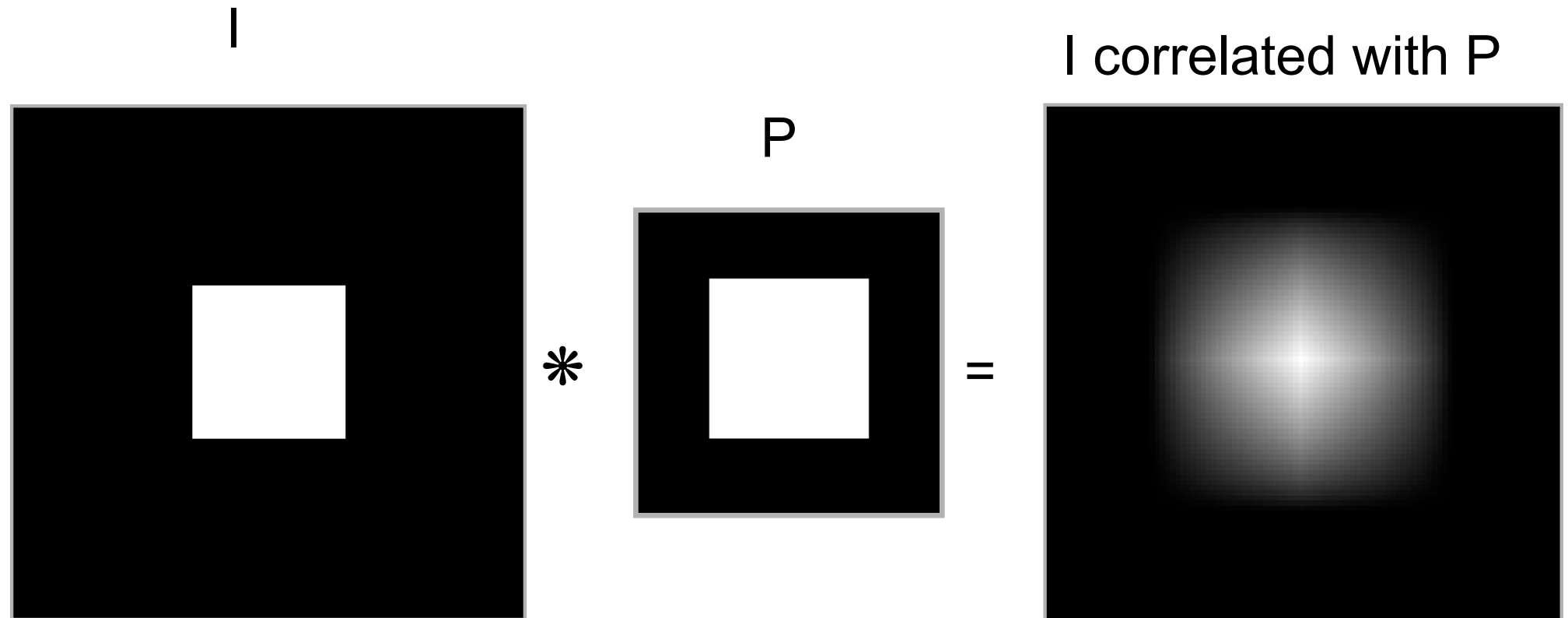
Sum of squares of
the window
(slow varying)

Sum of squares of
the pattern
(constant)

Correlation

$\text{ssd}(u,v)$ is smallest when correlation is largest
→ Correlation measures similarity

Synthetic Example

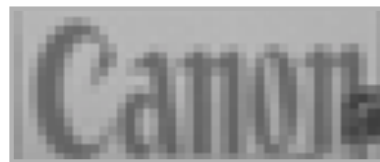


Real World Example

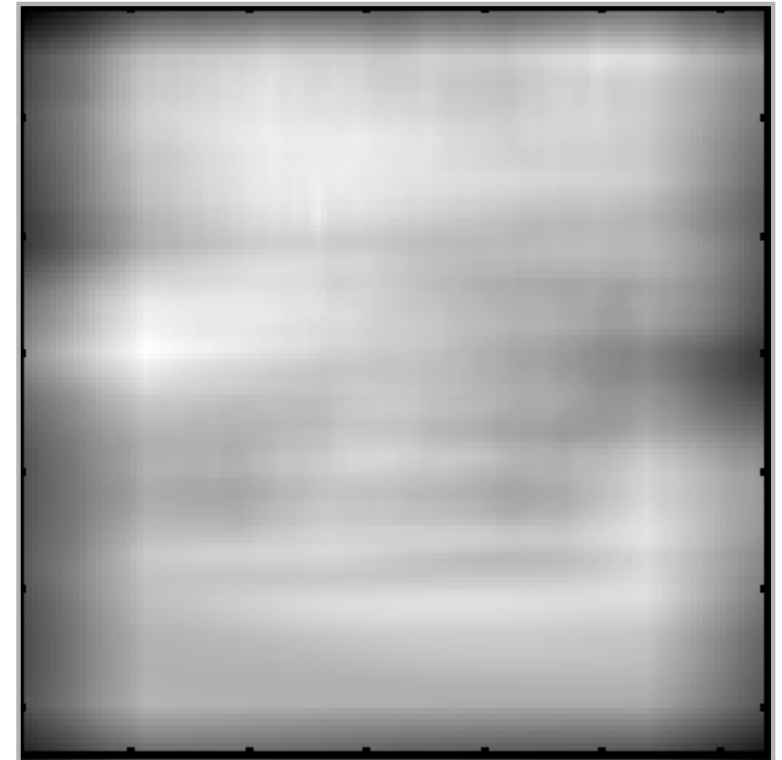
Image



Pattern



Correlation



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

Normalized Cross Correlation

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

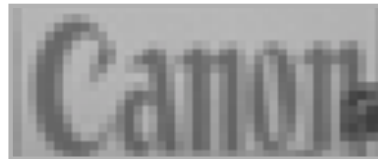
- Between -1 and 1
- Invariant to linear transforms
- Independent of the average gray levels of the pattern and the image window

Normalized Example

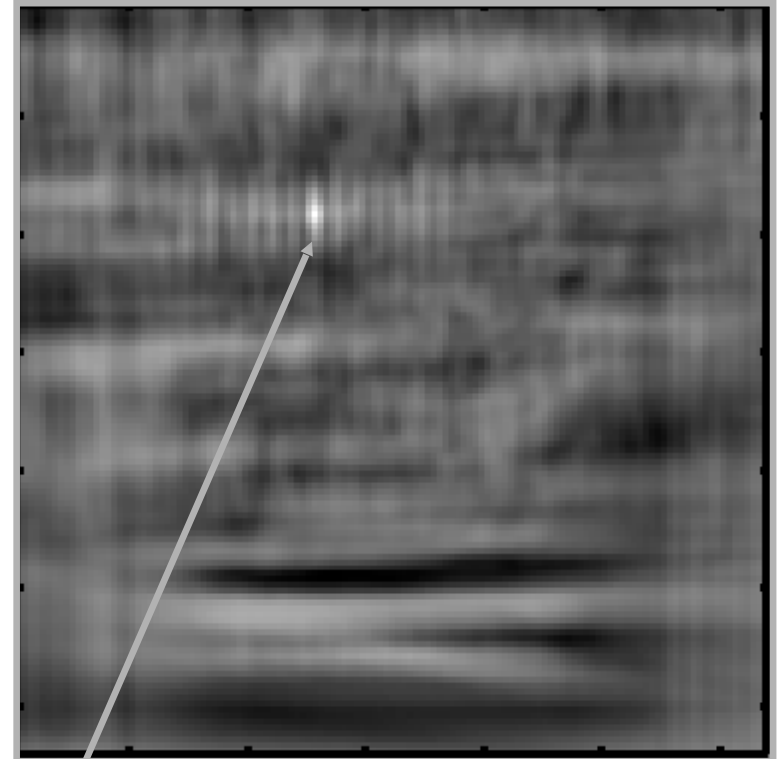
Image



Pattern

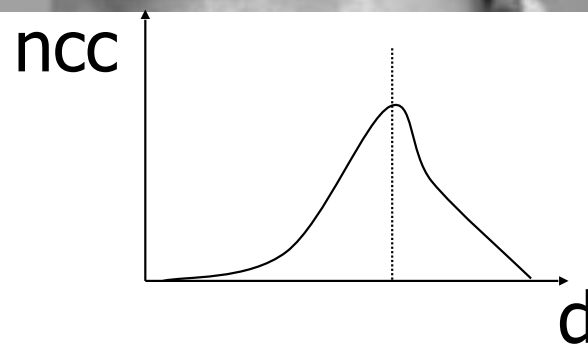


Normalized Correlation

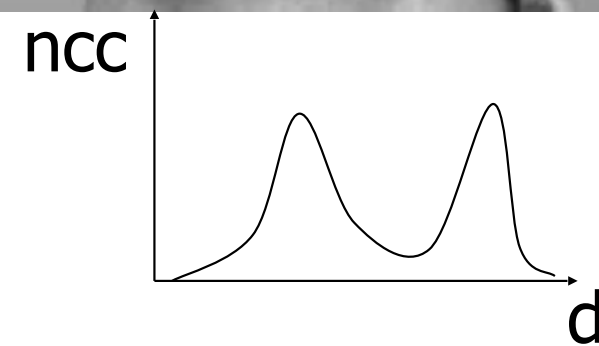


Point of maximum correlation

Searching along Epipolar Lines

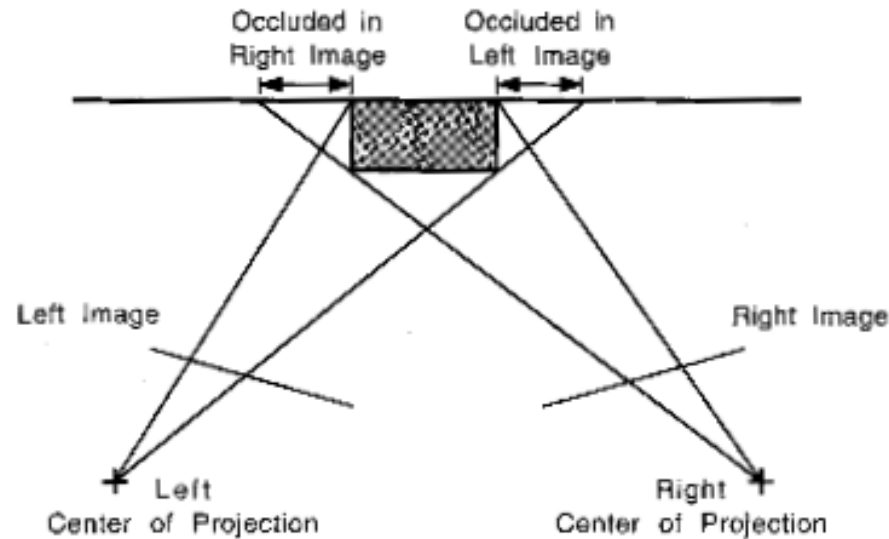


or

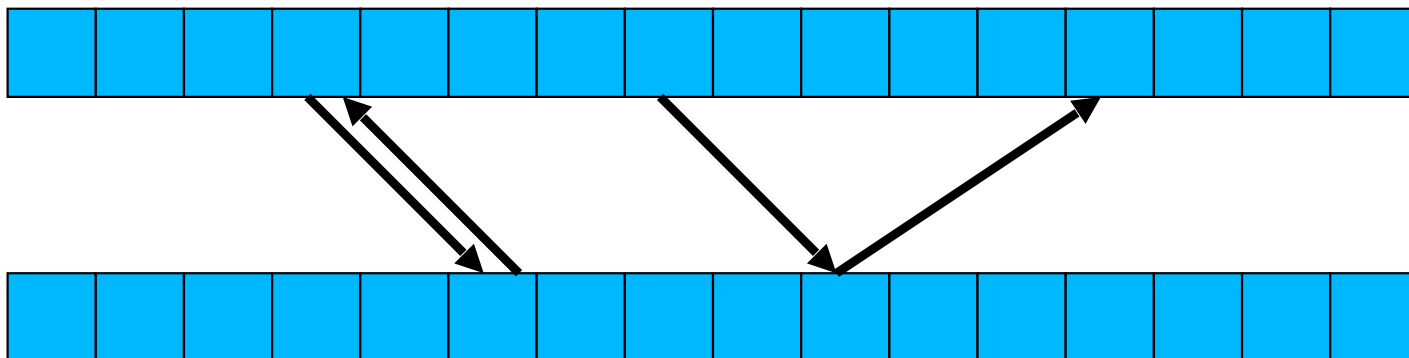


Ignoring Occluded Pixels

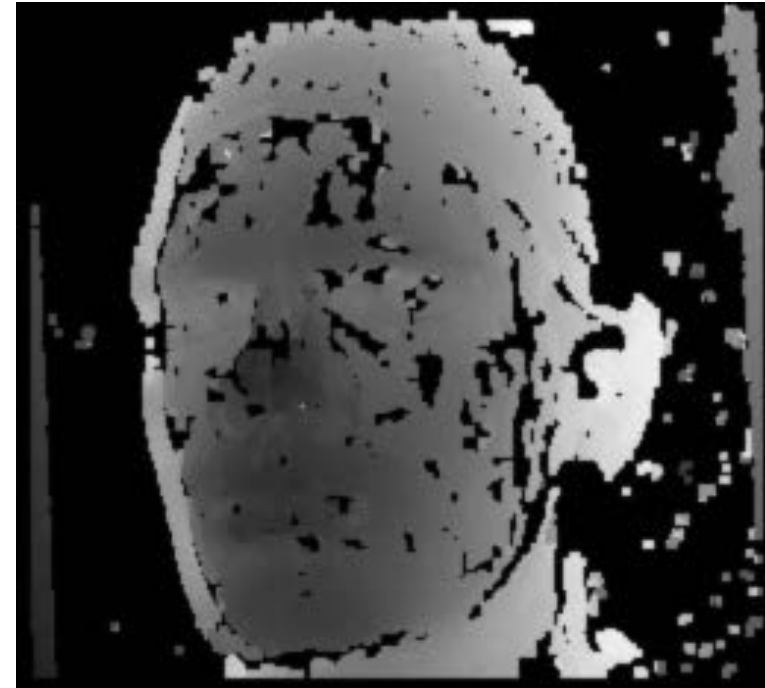
Some pixels have no corresponding pixel in the other image:



Left right consistency test:

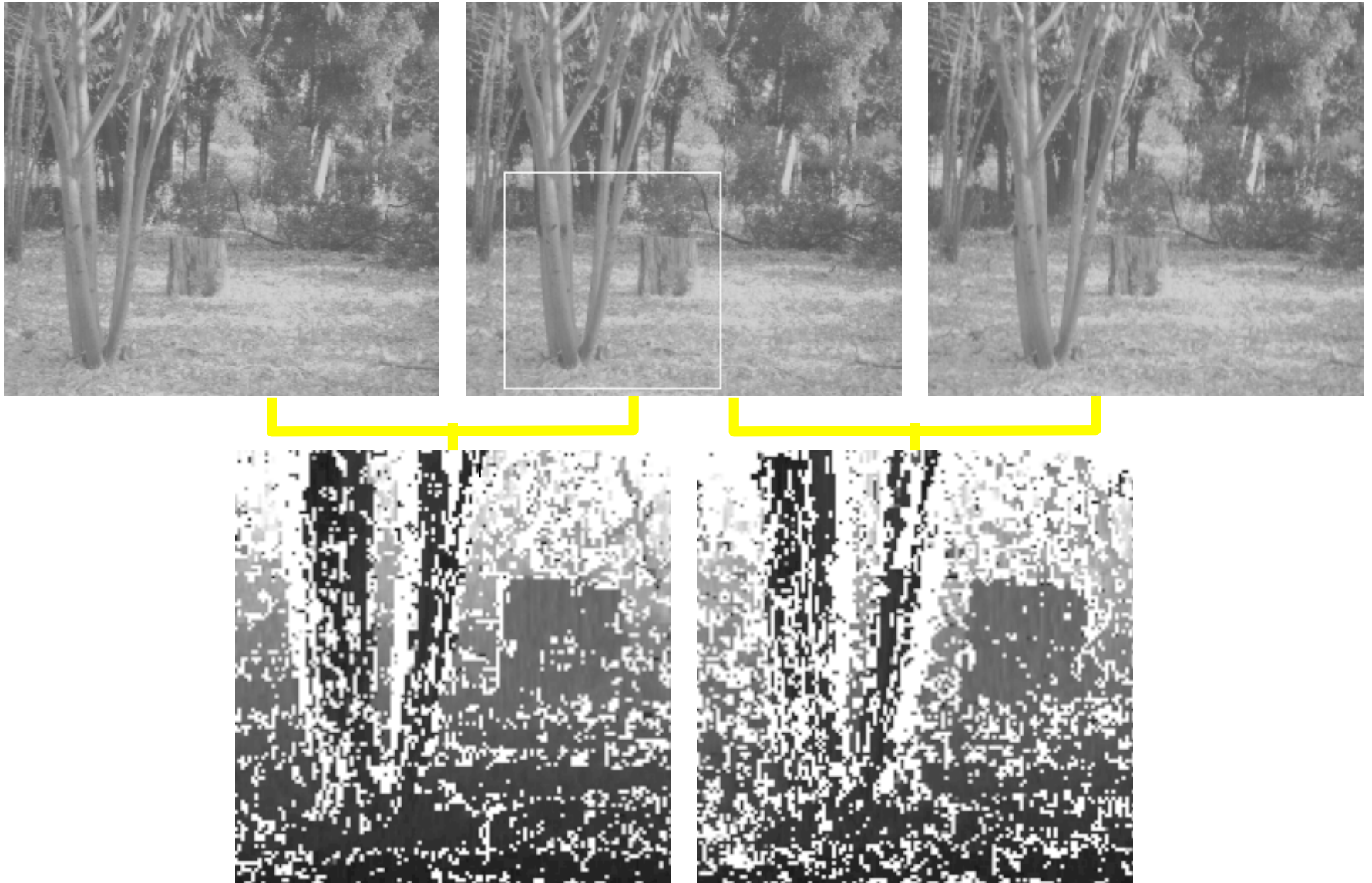


Disparity Map

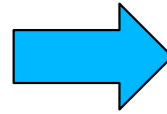


Black pixels: No disparity.

Ground Level Stereo



Combining Disparity Maps



- Merging several disparity maps.
- Smoothing the resulting map.

Variational Approach



$$\mathcal{C} = \int s(w - w_0)^2 + \lambda_x \left(\frac{\partial w}{\partial x}\right)^2 + \lambda_y \left(\frac{\partial w}{\partial y}\right)^2$$

s = Correlation score if w_0 has been measured, 0 otherwise.

$$\lambda_x = c_x f\left(\frac{\partial I}{\partial x}\right)$$

$$\lambda_y = c_y f\left(\frac{\partial I}{\partial y}\right)$$

$$f(x) = \begin{cases} 1 & \text{if } x < x_0 \\ \frac{x_1 - x}{x_1 - x_0} & \text{if } x_0 < x < x_1 \\ 0 & \text{if } x_1 < x \end{cases}$$

Solving the Variational Problem

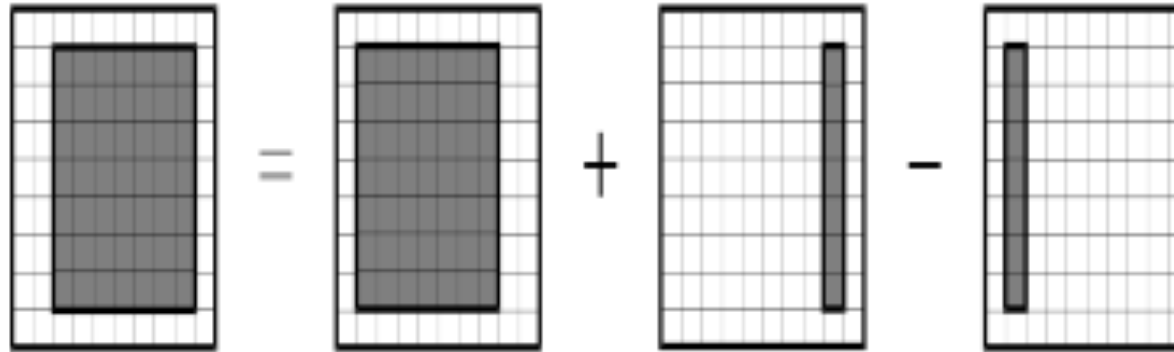
Discretize the integral and solve a linear problem:

$$\begin{aligned} \mathcal{C} &= \sum_{ij} s_{ij} (w_{ij} - w_{0ij})^2 + \lambda_x \sum_{ij} (w_{i+1,j} - w_{i,j})^2 + \lambda_y \sum_{ij} (w_{i,j+1} - w_{i,j})^2 \\ &= (W - W_0)^t S (W - W_0) + W^t K W \end{aligned}$$

$$\Rightarrow \frac{\partial \mathcal{C}}{\partial W} = 0$$

$$\Rightarrow (K + S)W = SW_0$$

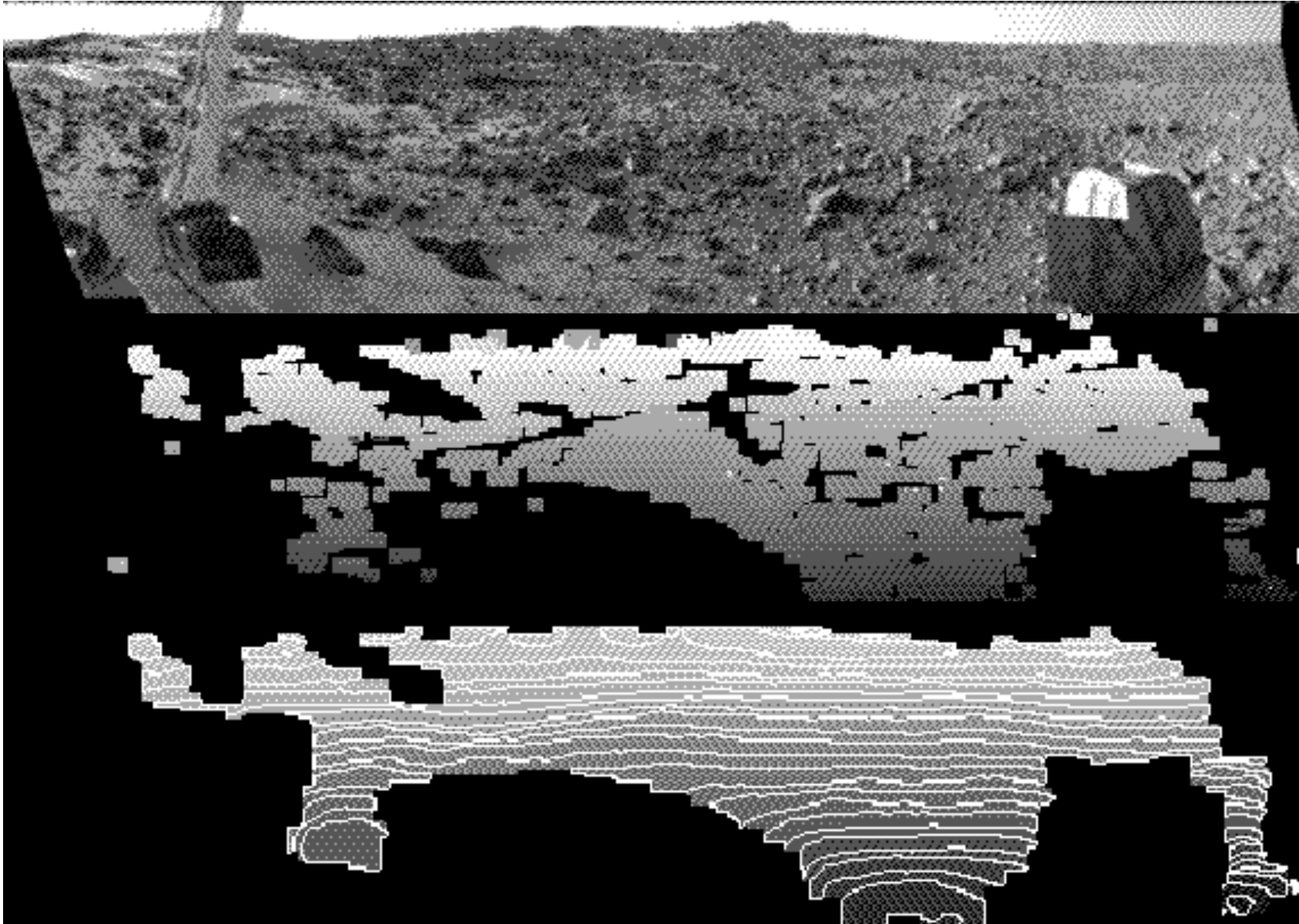
Real-Time Implementation



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

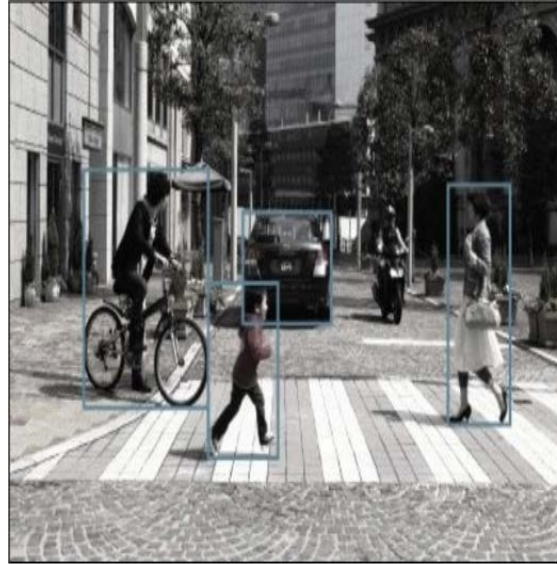
- Many duplicated computations.
- Can be implemented so that it is fast.
- Speed is independent from window size.

Then



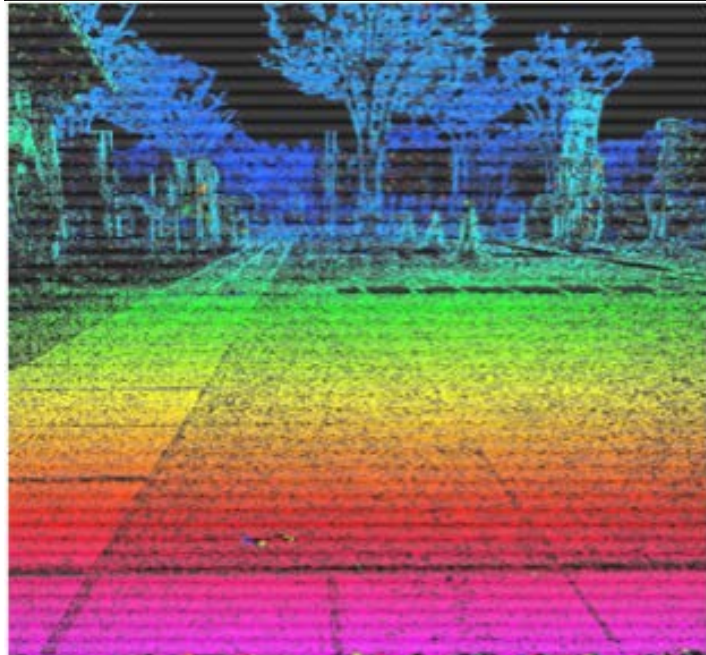
1993:
256x256,
60 disps,
7 fps.

... and more Recently



Subaru's EyeSight System

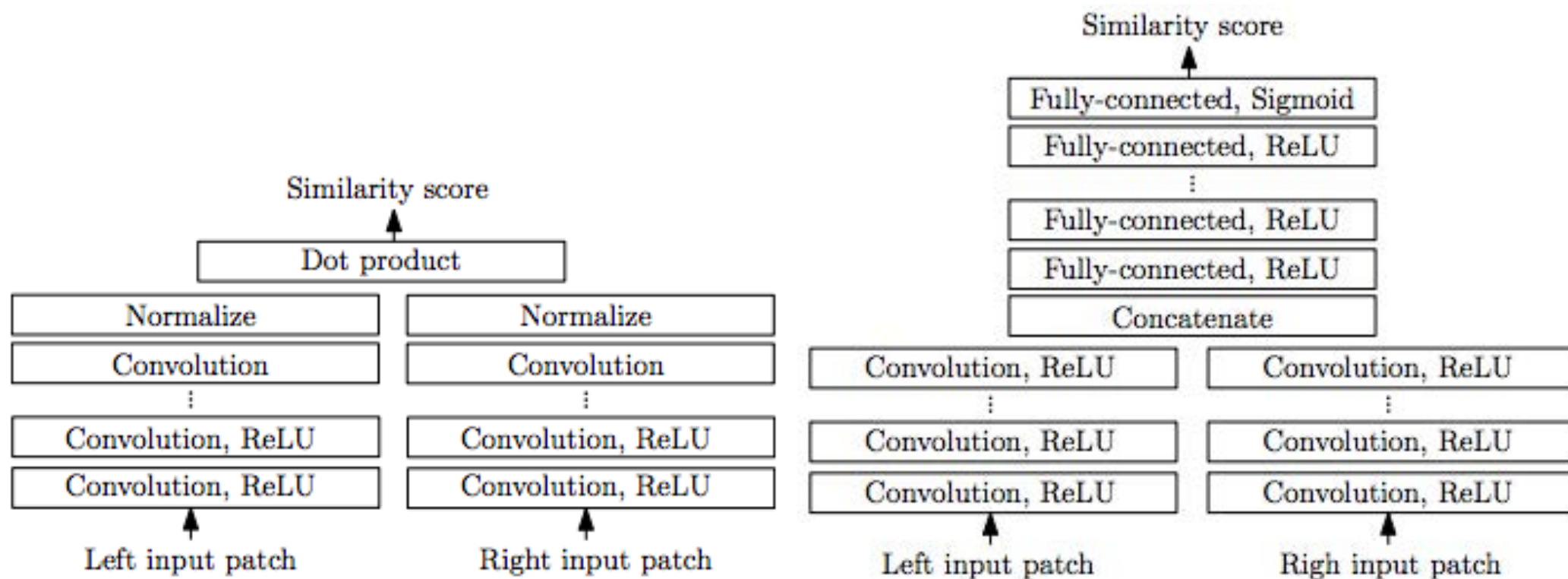
<http://www.gizmag.com/subaru-new-eyesight-stereoscopic-vision-system/14879/>



2011:
1312x688,
176 disps,
160 fps.

... and even More Recently

Replace Normalized Cross Correlation by Siamese nets designed to return a similarity score for potentially matching patches.



Tesla's non LiDAR Approach

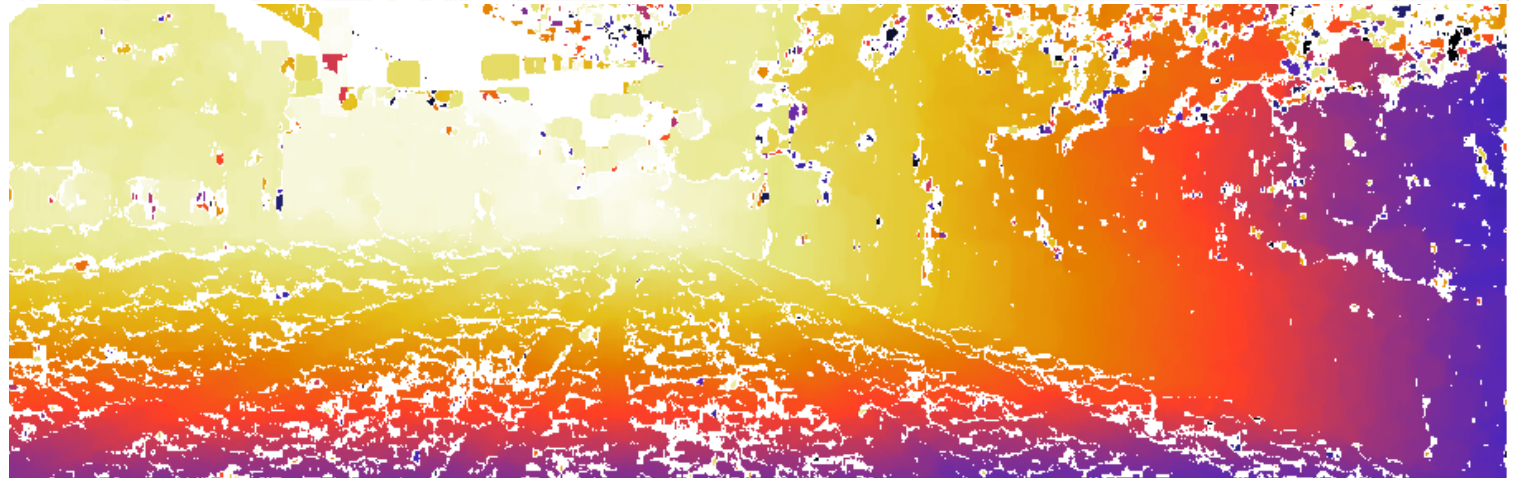


<https://www.therobotreport.com/researchers-back-teslas-non-lidar-approach-to-self-driving-cars/>

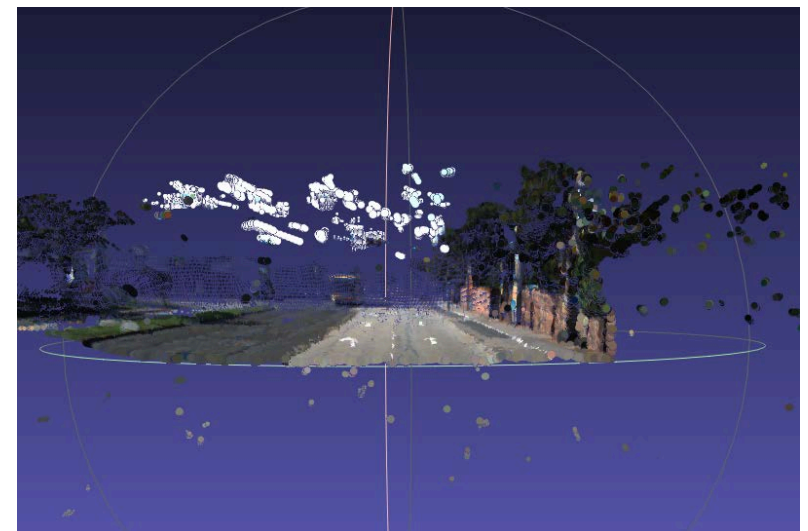
3D Point Cloud



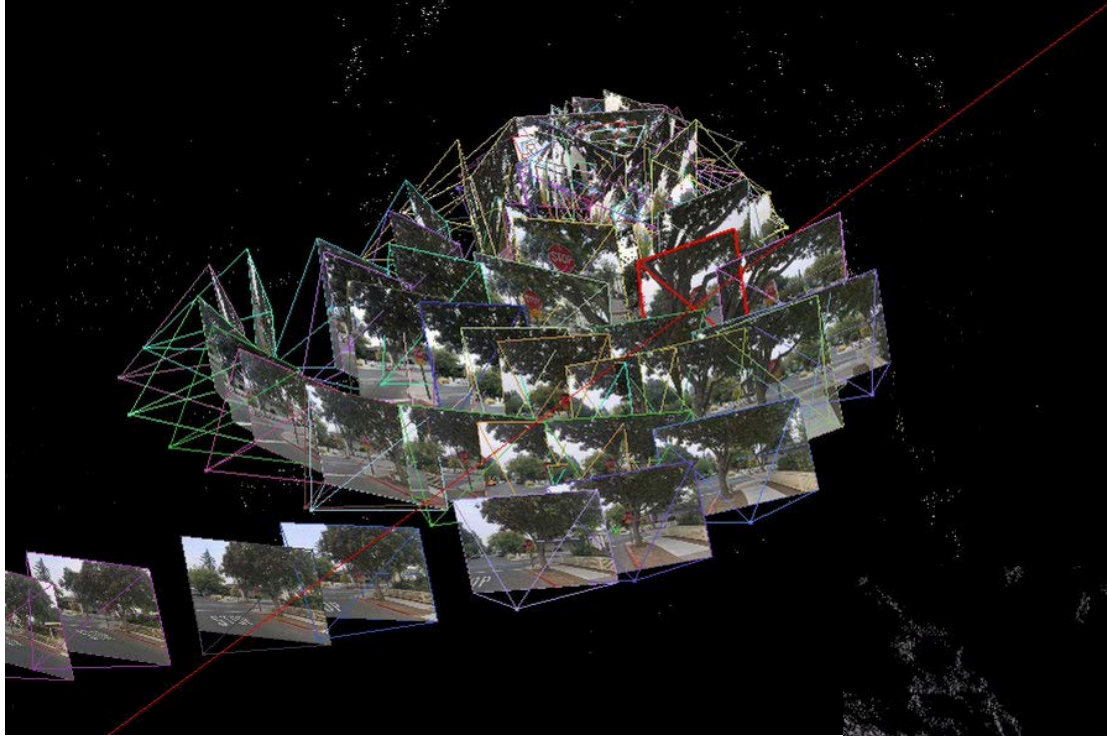
Disparity map



We can transform each triplet (u,v,d) into a 3D point (x,y,z) .
—> A 3D point cloud.



Merging Point Clouds

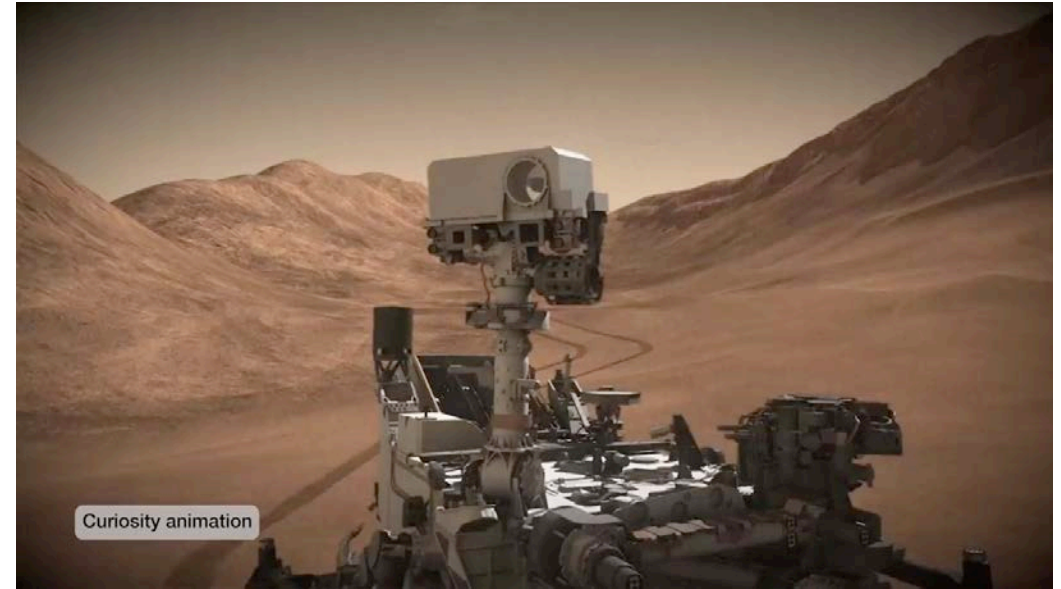


We can treat image pairs in a video sequence as stereo pairs and compute a 3D cloud for each.

We can then merge the 3D point clouds.
—> Potential dense models when we have enough images.



Perseverance 2022



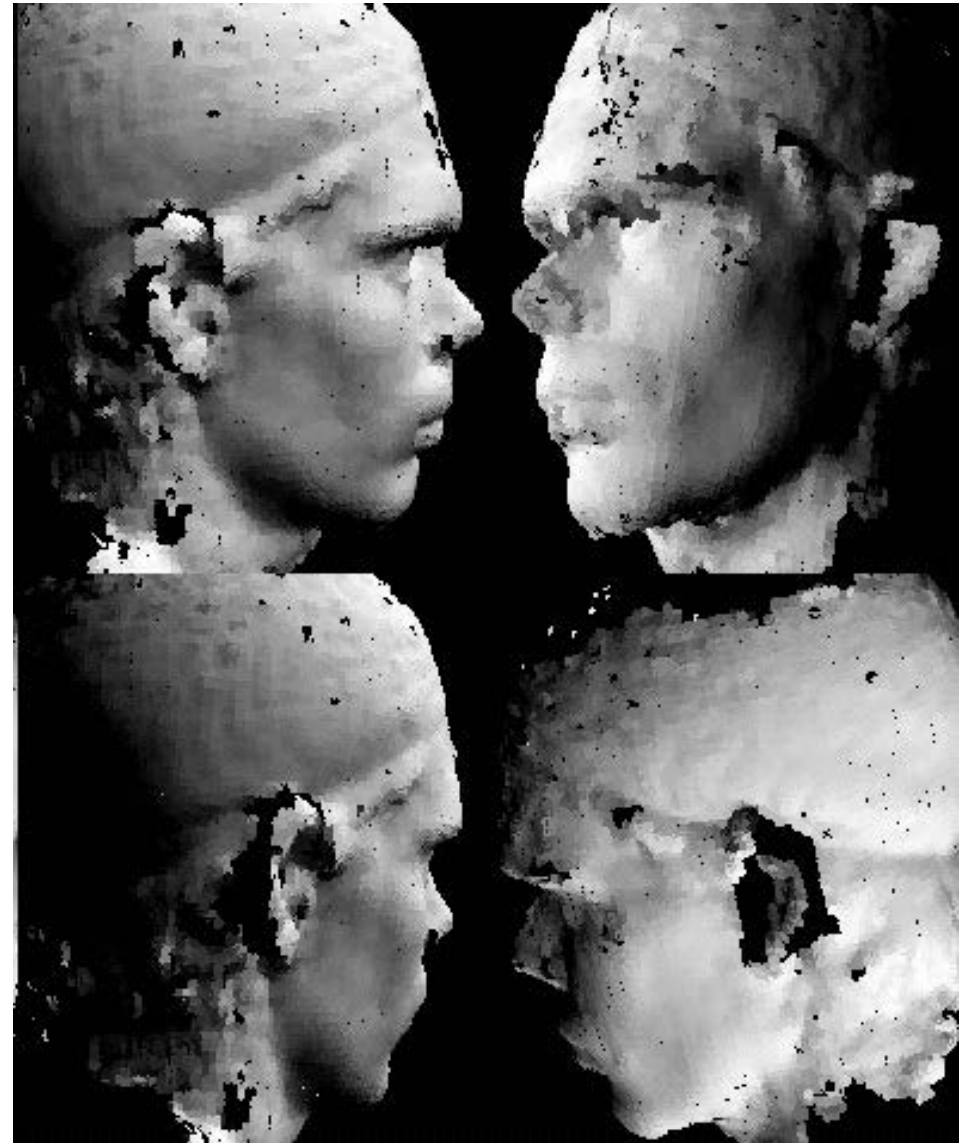
Max distance in a single day:

219m (Opportunity, 2008)

319m (Perseverance, 2022)

Impressive when 260 million kilometers away!

Head from Video



Treat consecutive images as stereo pairs.

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

Window Size

Small windows:

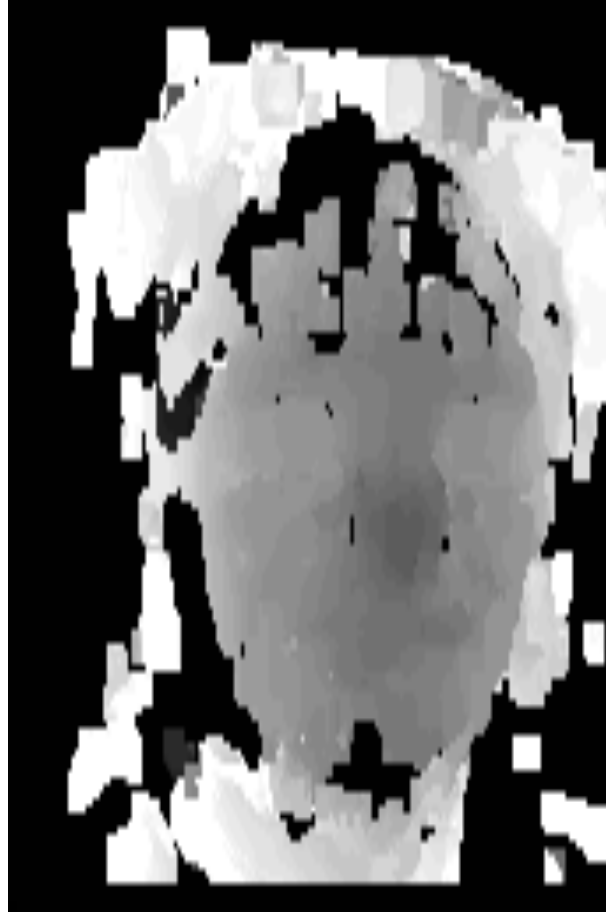
- Good precision
- Sensitive to noise

Large windows:

- Diminished precision
- Increased robustness to noise

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

Window Size



15x15



7x7

Scale-Space Revisited



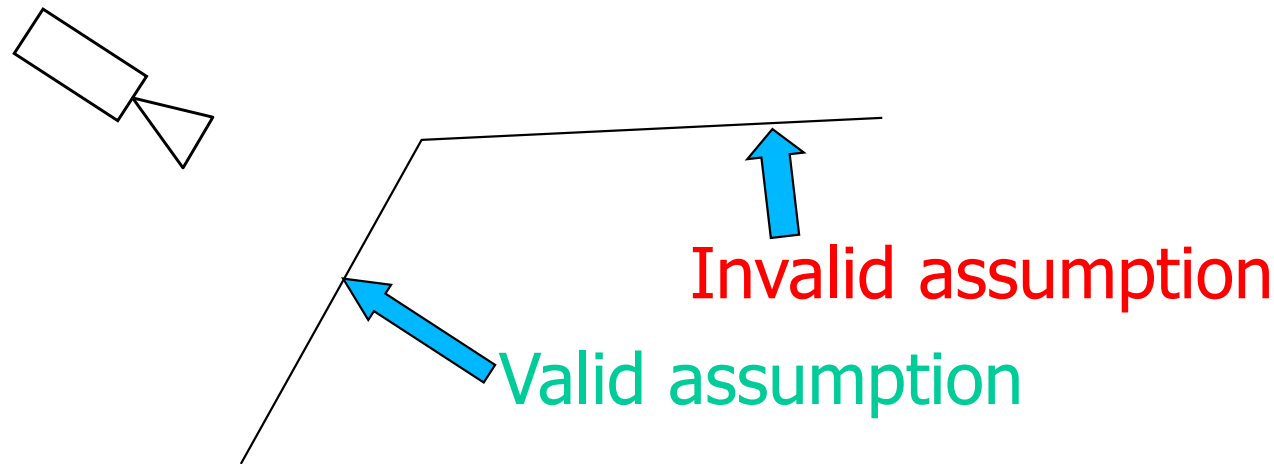
Gaussian
pyramid

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 over the entire correlation window, which is equivalent to assuming constant depth.



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

Multi-View Stereo



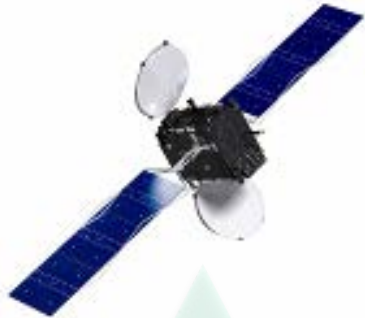
Multi-view reconstruction setup

—> Adjust correlation window shapes to handle orientation.

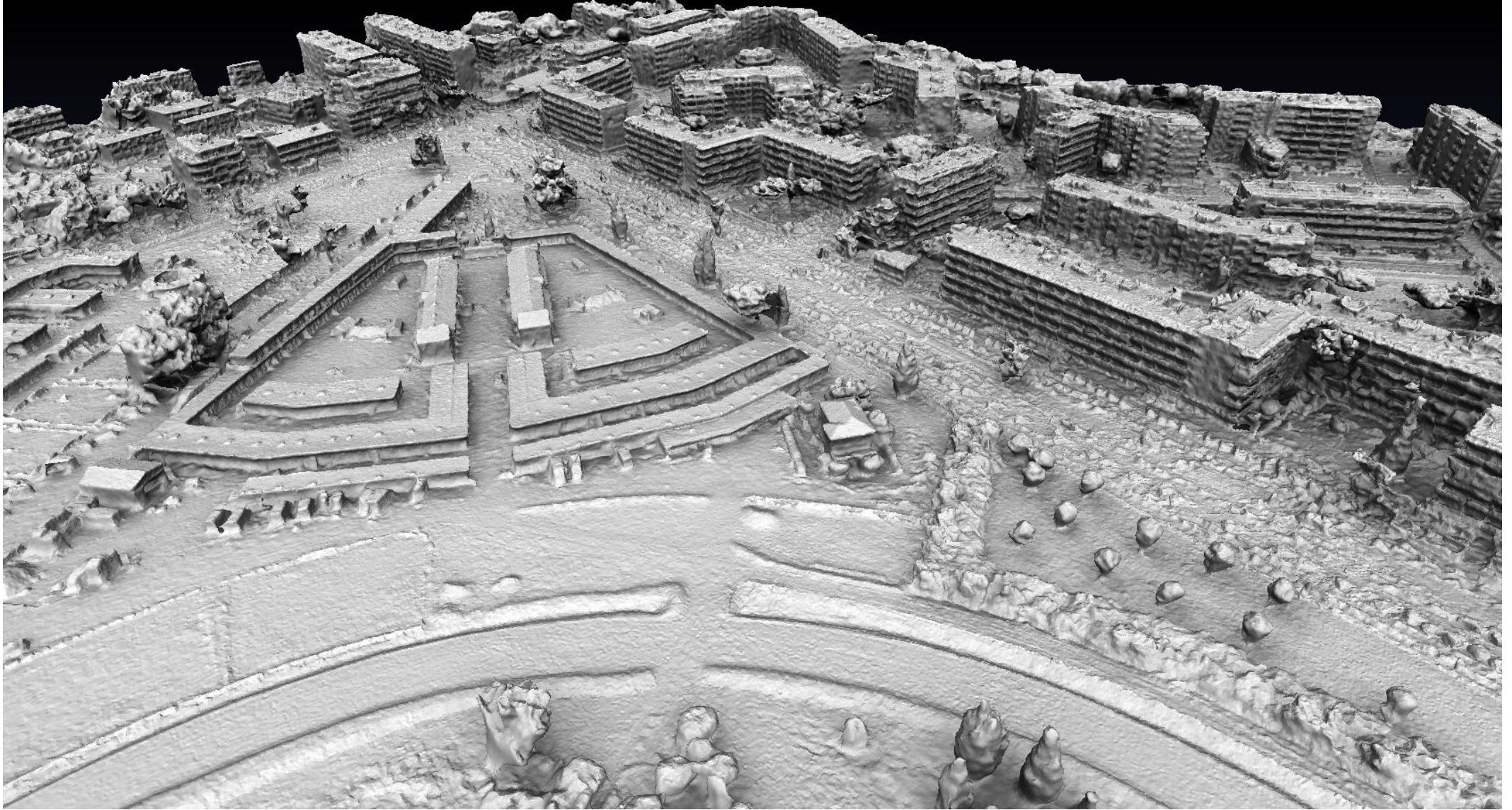


Textured and Mapped 3D Model

Flying Cameras



Multi-View Stereo



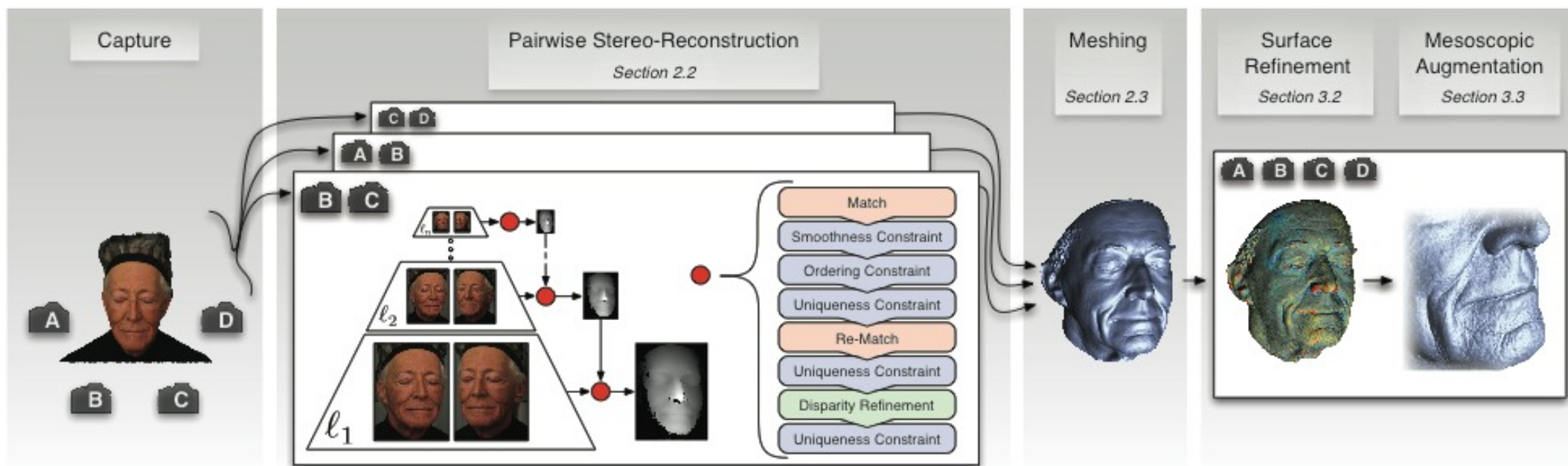
Matterhorn



Face Reconstruction



Face Reconstruction



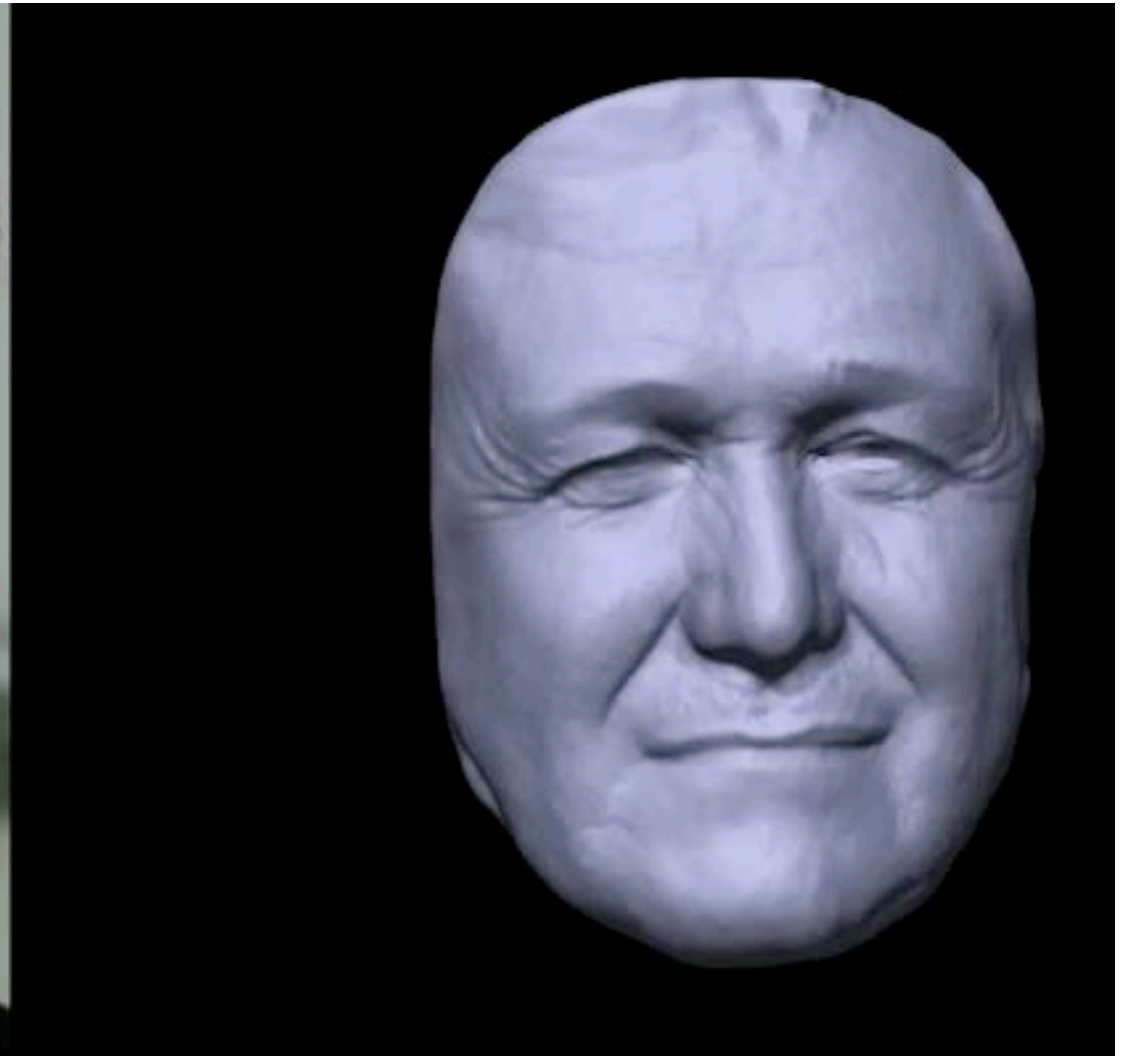
Face Reconstruction



Dynamic Shape

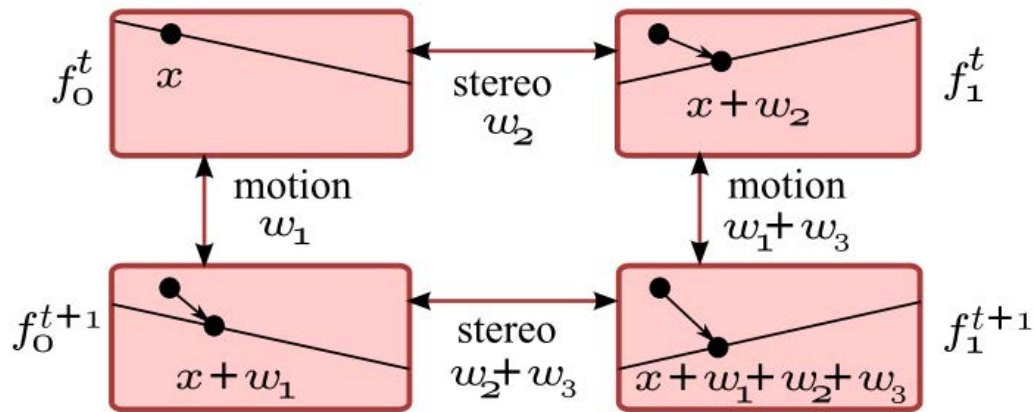


One of two Input Views

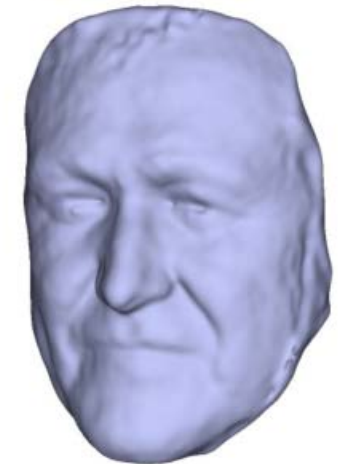
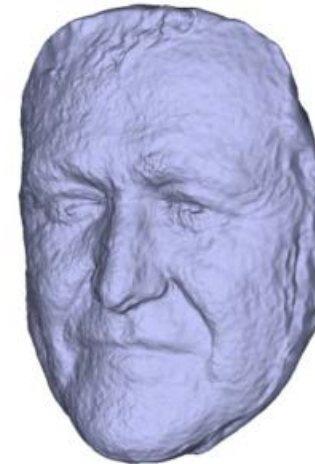


Reconstruction

Scene Flow



Correspondences across cameras and across time



Stereo Only

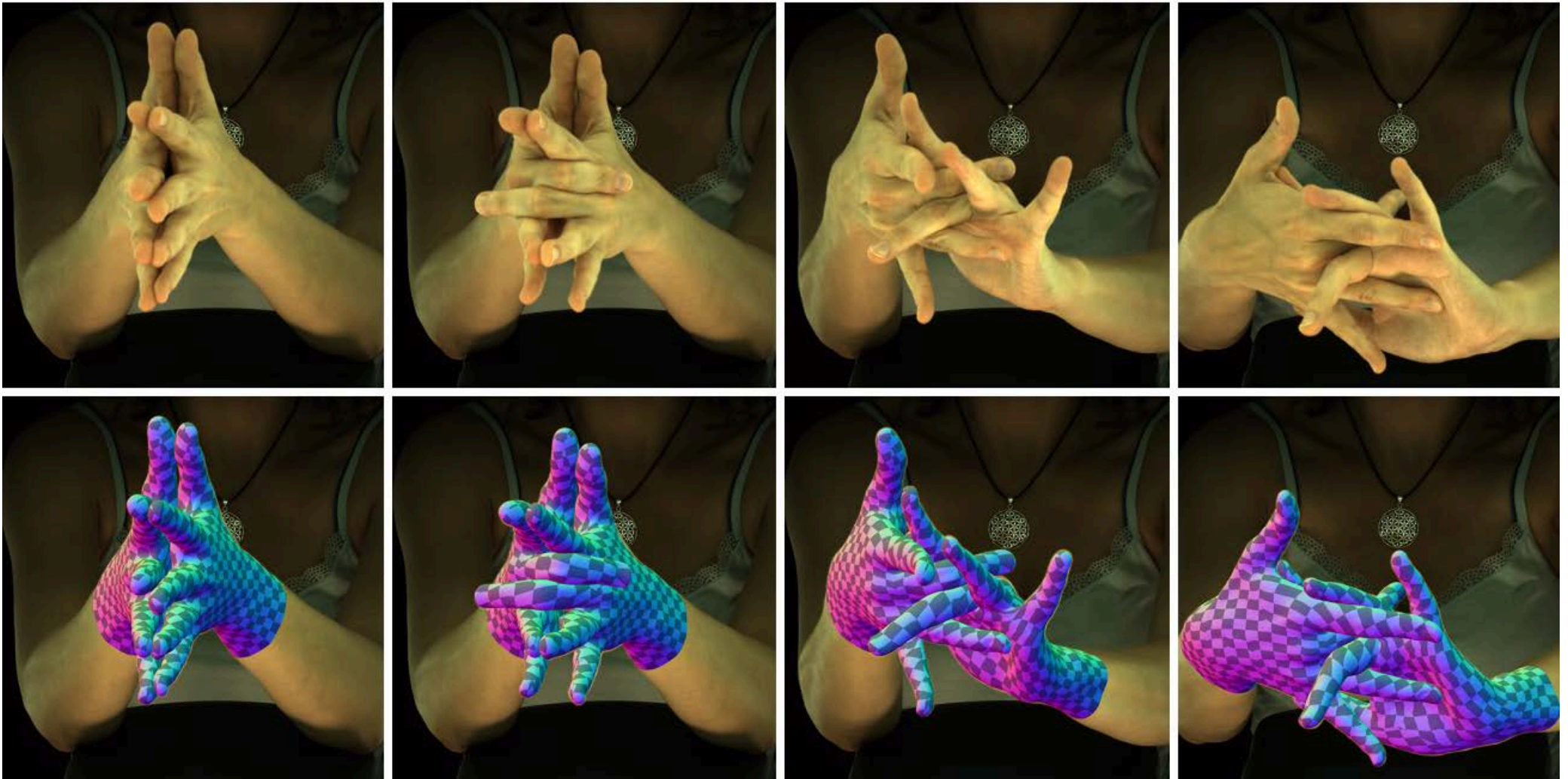
Stereo + Flow

Refining using Shape From Shading



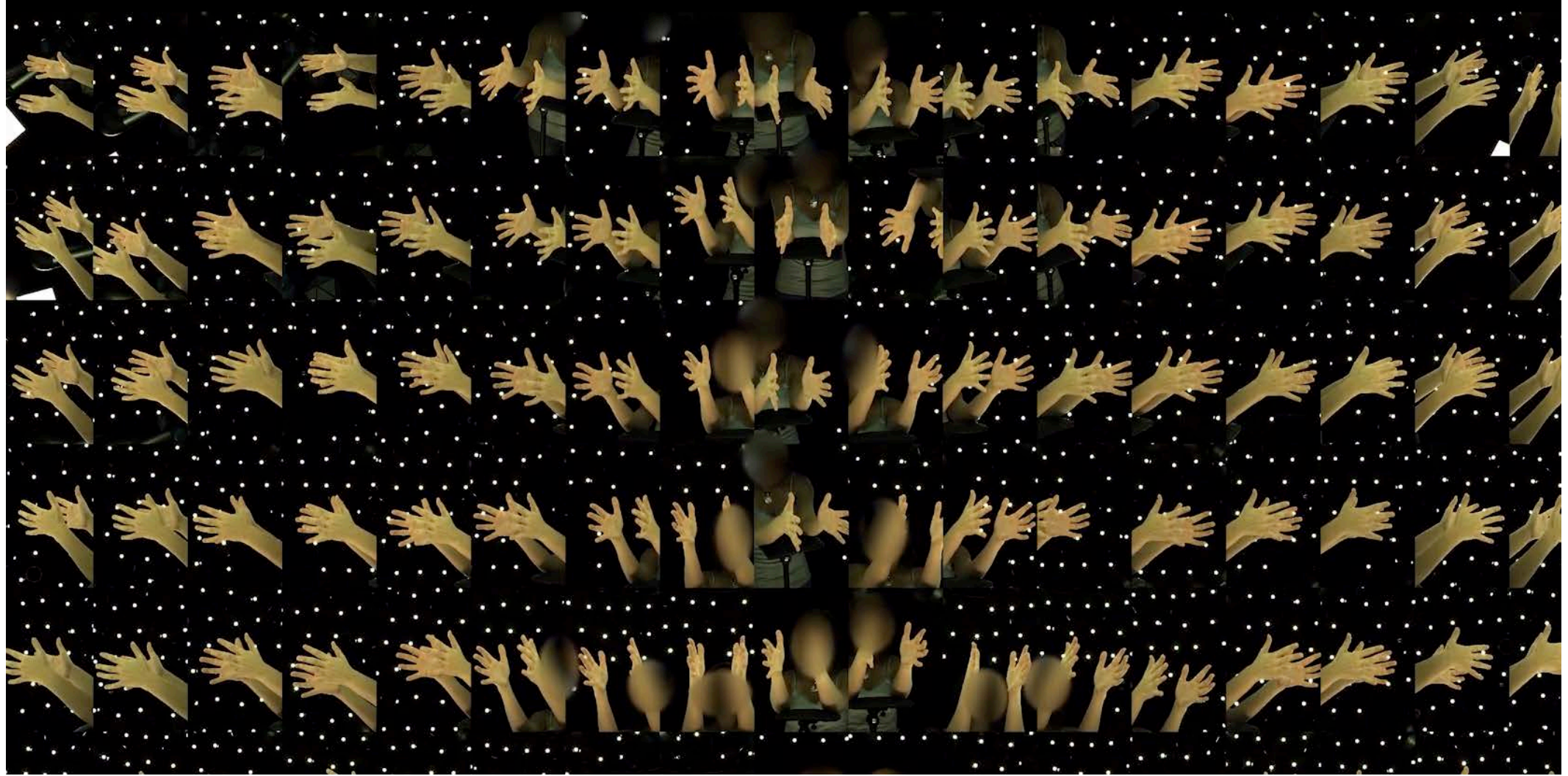
Shape-from-shading can be used to refine the shape and provide high-frequency details.

Using Many Cameras

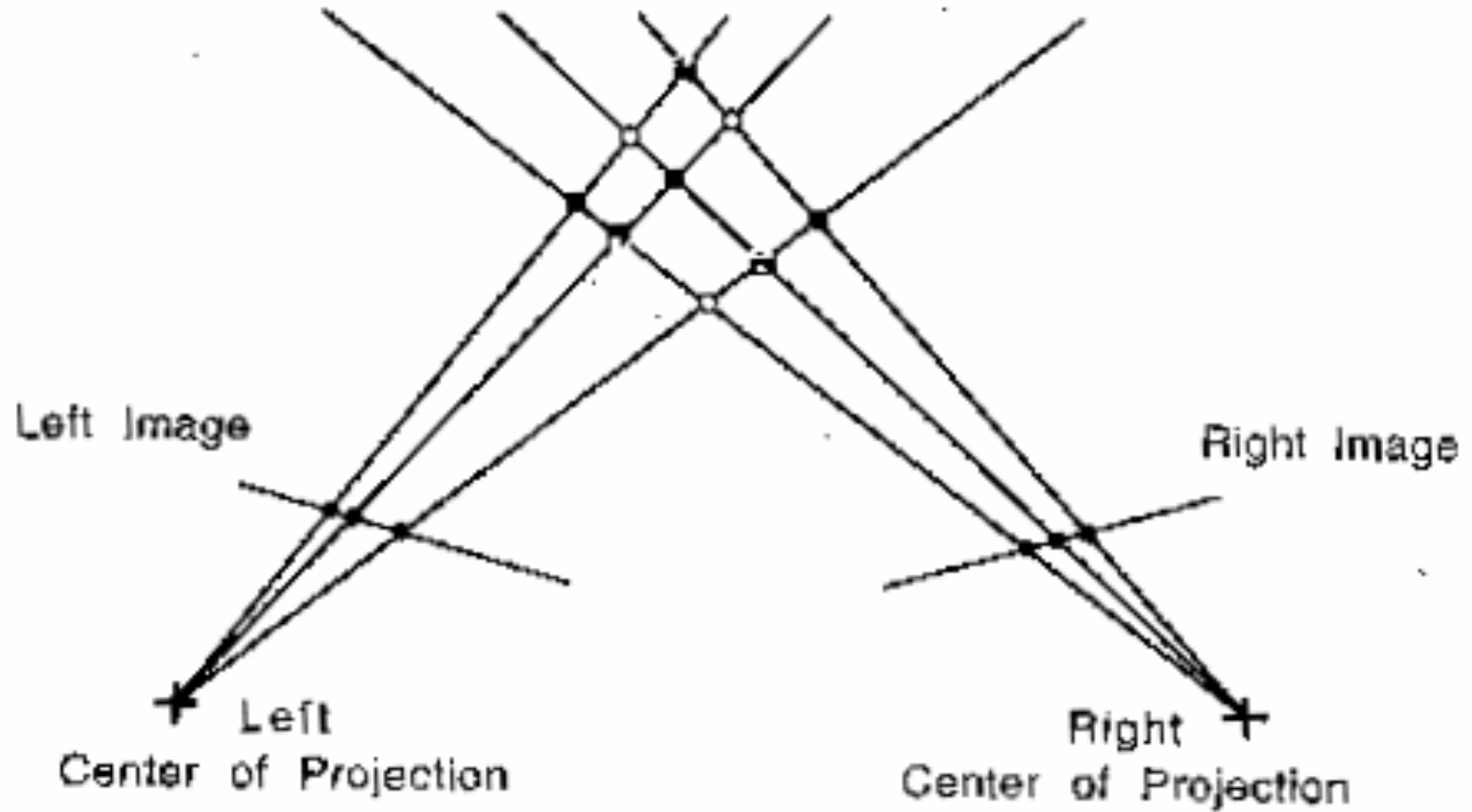


Using 124 calibrated cameras with hardware synchronization

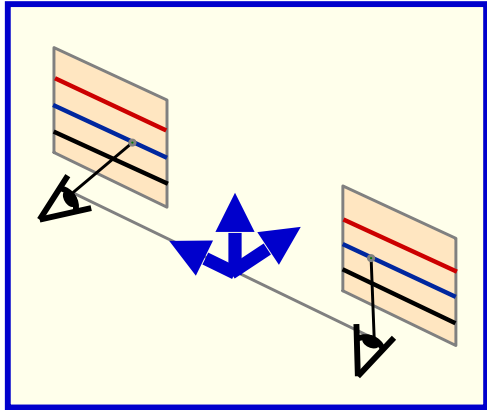
Captured Images



Uncertainty



Precision vs Baseline



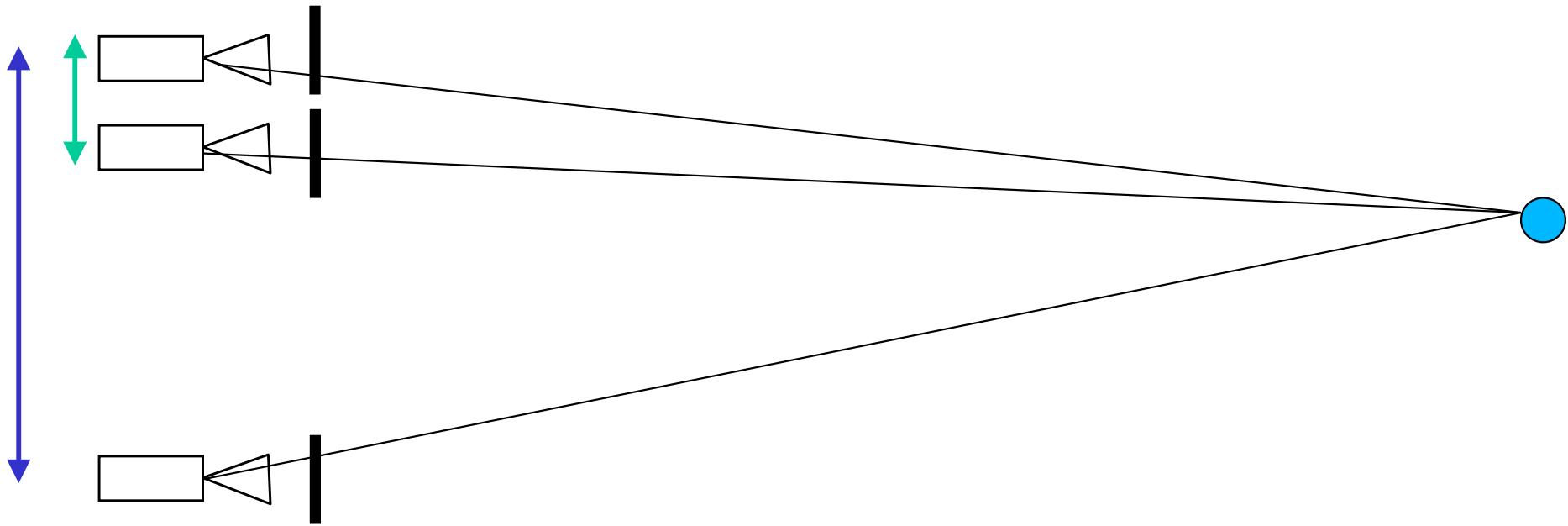
$$d = f \frac{b}{Z}$$

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

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

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

Short vs Long Baseline



Short baseline:

- Good matches
- Few occlusions
- Poor precision

Long baseline:

- Harder to match
- More occlusions
- Better precision

Mars Rover



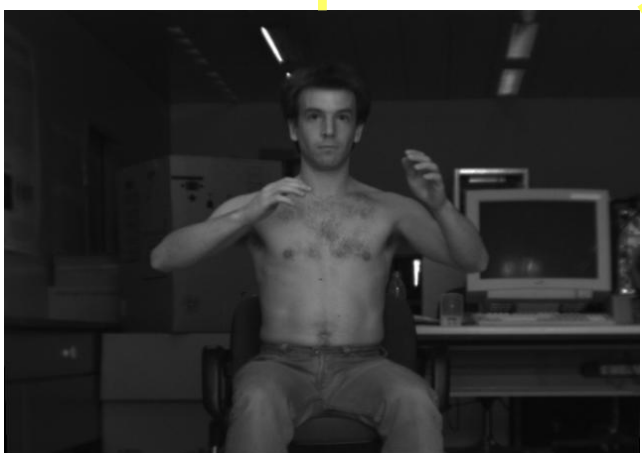
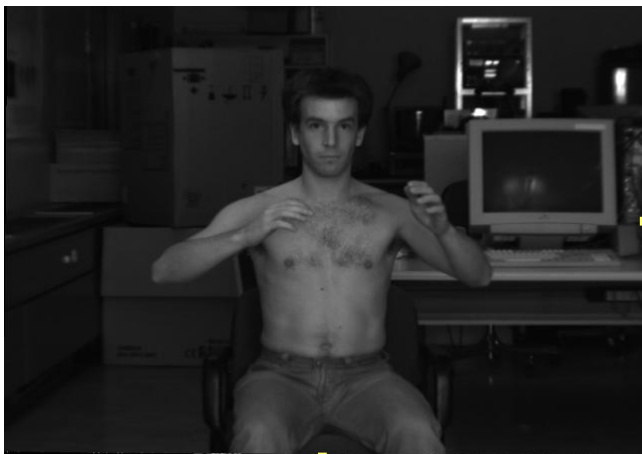
There are four cameras!

Video-Based Motion Capture



Fitting an articulated body model to stereo data.

Trinocular Stereo



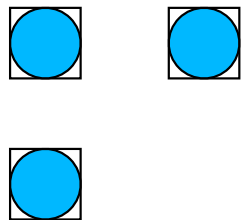
Multi-Camera Configurations



3 cameras give both robustness and precision.



4 cameras give additional redundancy.



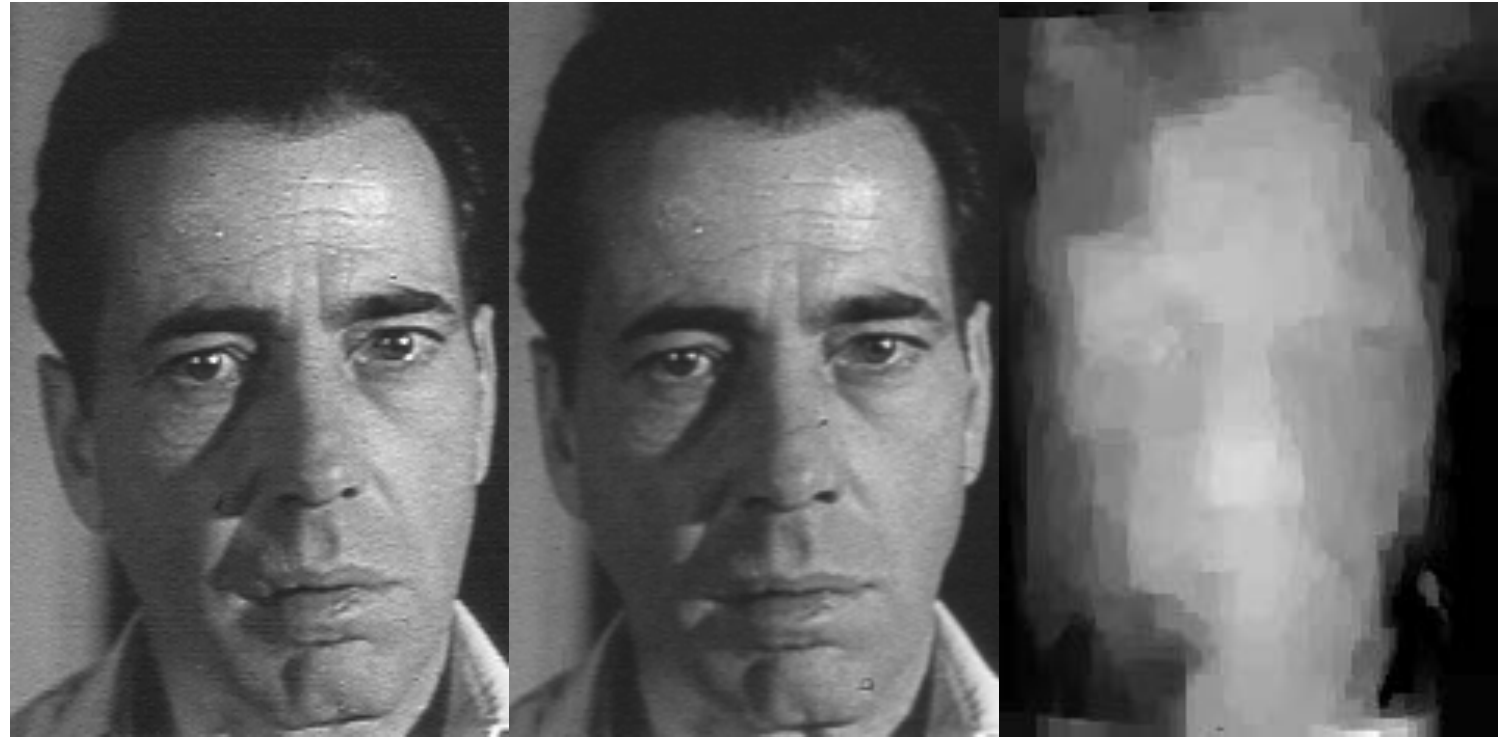
3 cameras in a T arrangement allow the system to see vertical lines.

Kinect: Structured Light



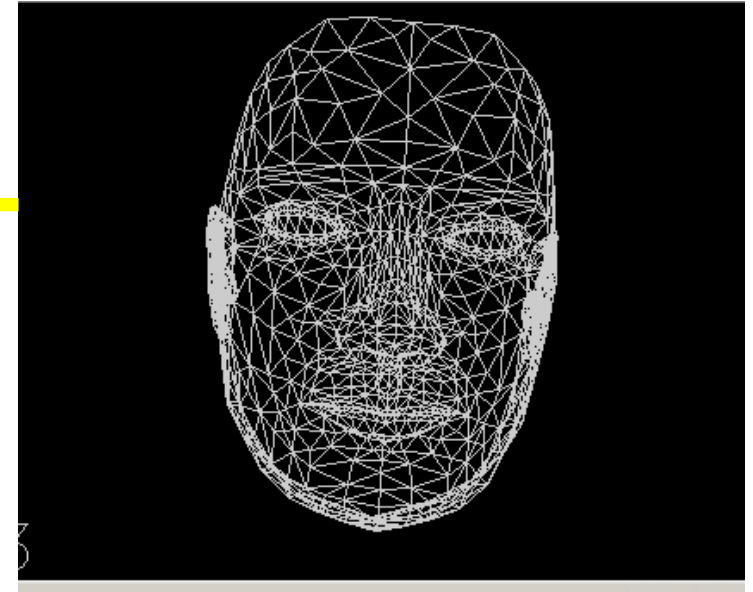
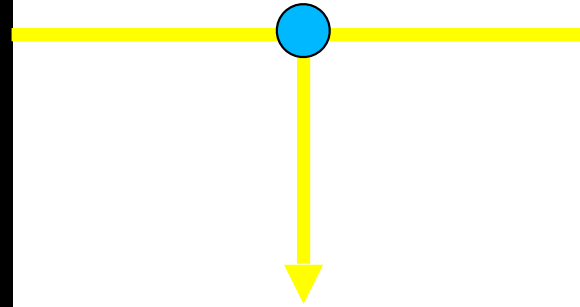
- The Kinect camera projects a IR pattern and measures depth from its distortion.
- Same principle but the second camera is replaced by the projector.

Faces from Low-Resolution Videos



- No calibration data
- Relatively little texture
- Difficult lighting

Simple Face Model

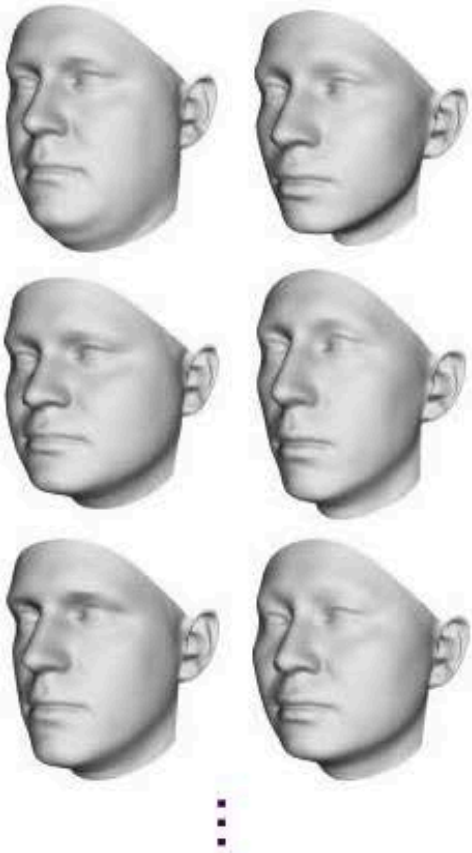


PCA Face Model



-2σ

$+2\sigma$



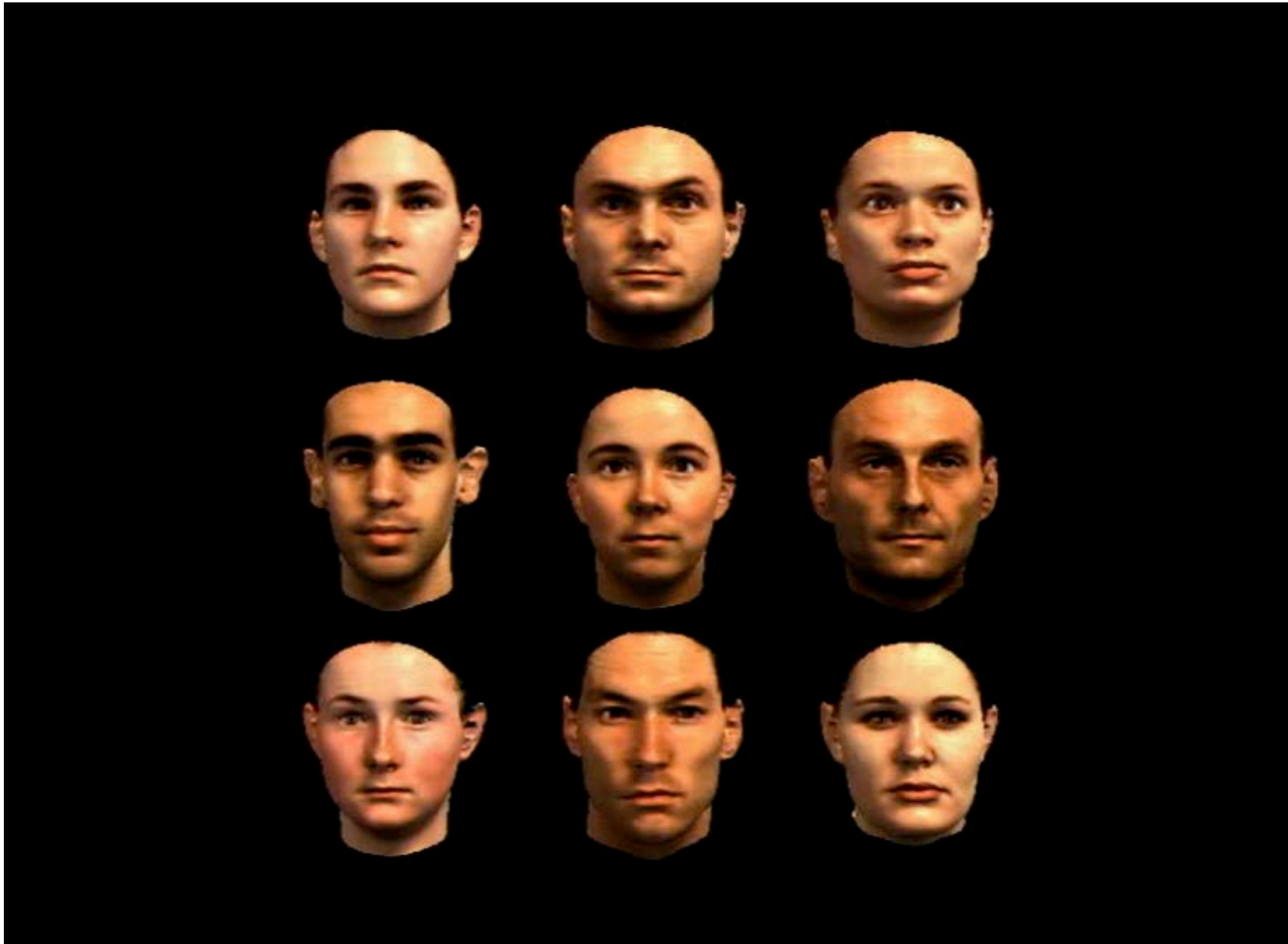
$$S = \bar{S} + \sum_{i=1}^{99} \alpha_i S_i$$

\bar{S} : Average shape

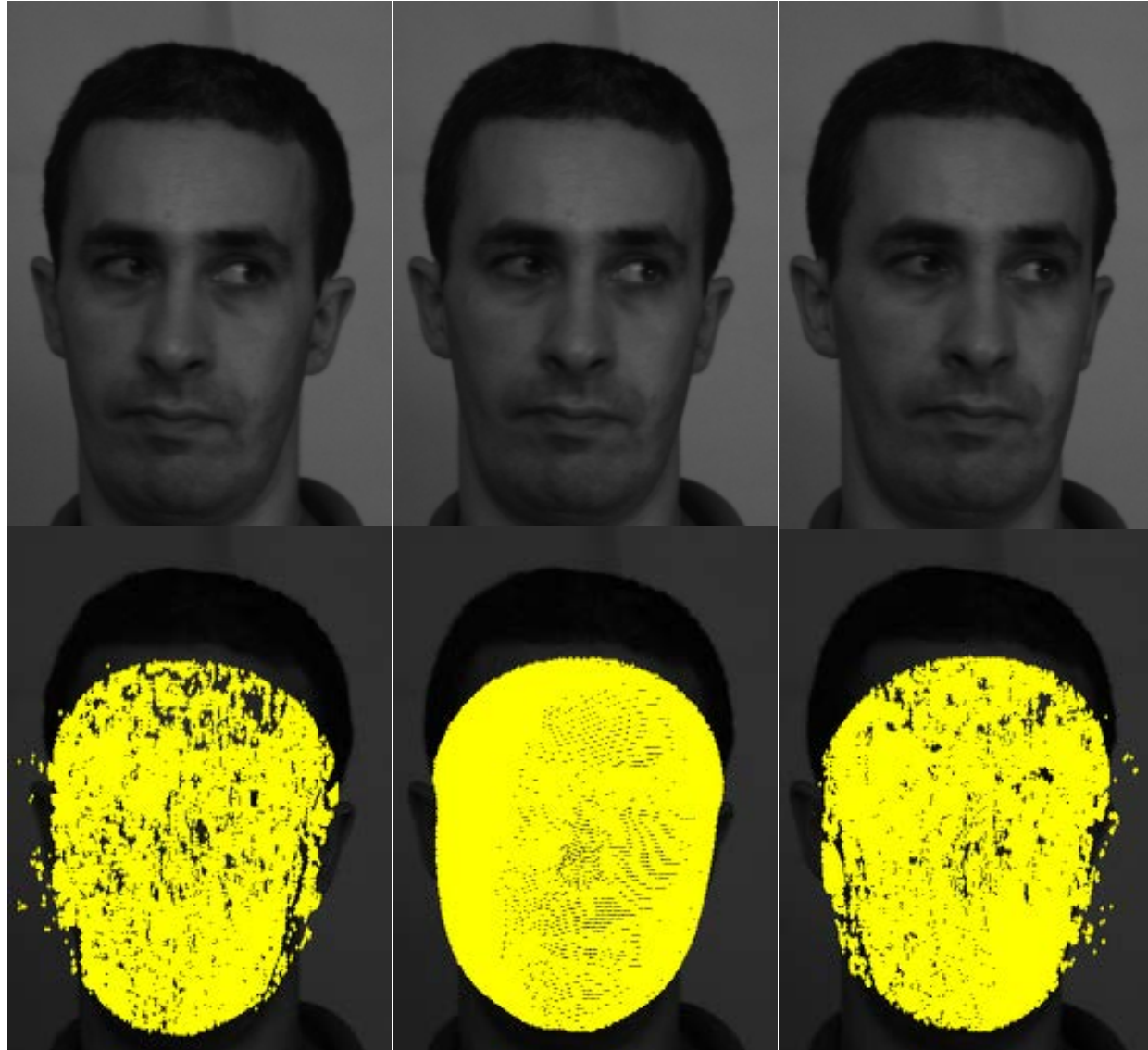
S_i : Shape vector

α_i : Shape coefficients

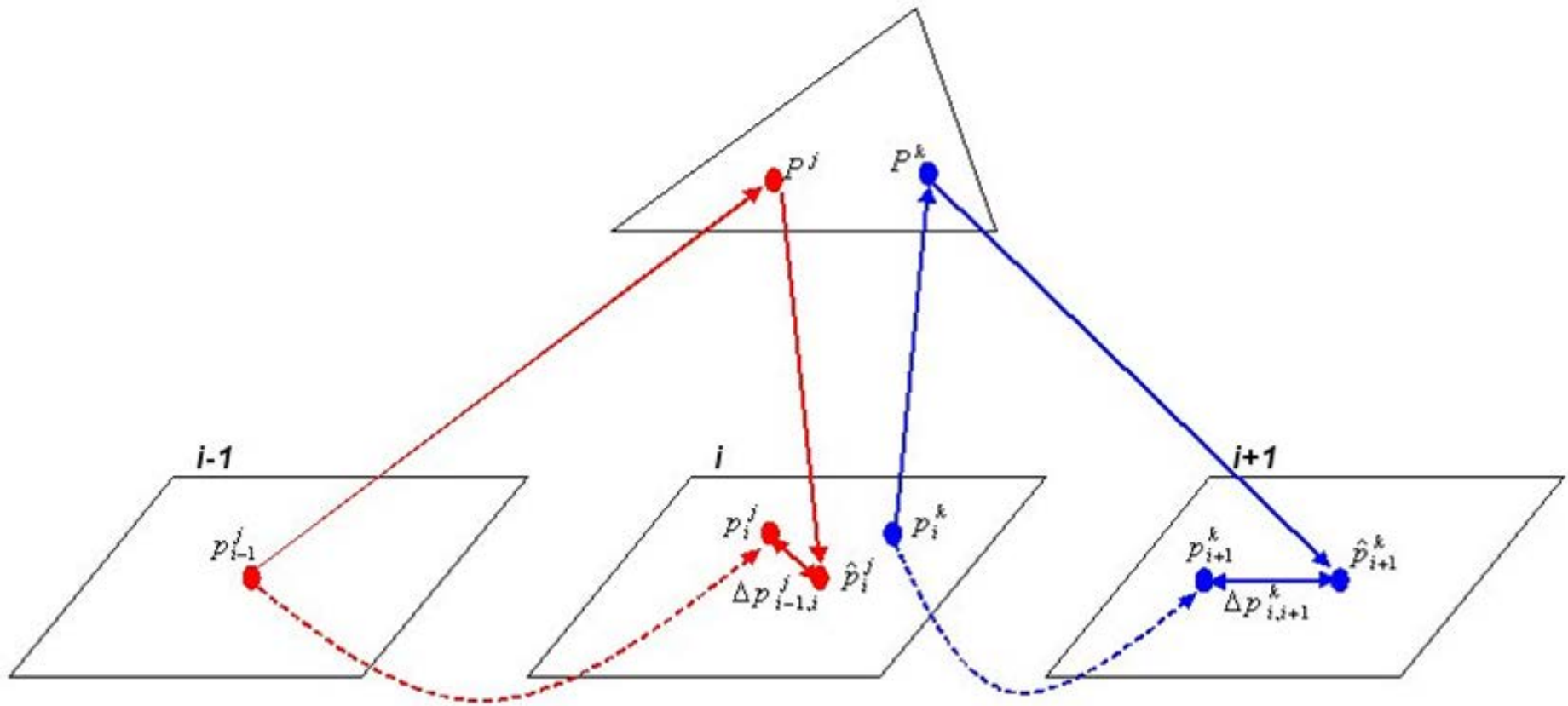
3D Face Modeling



Correspondences

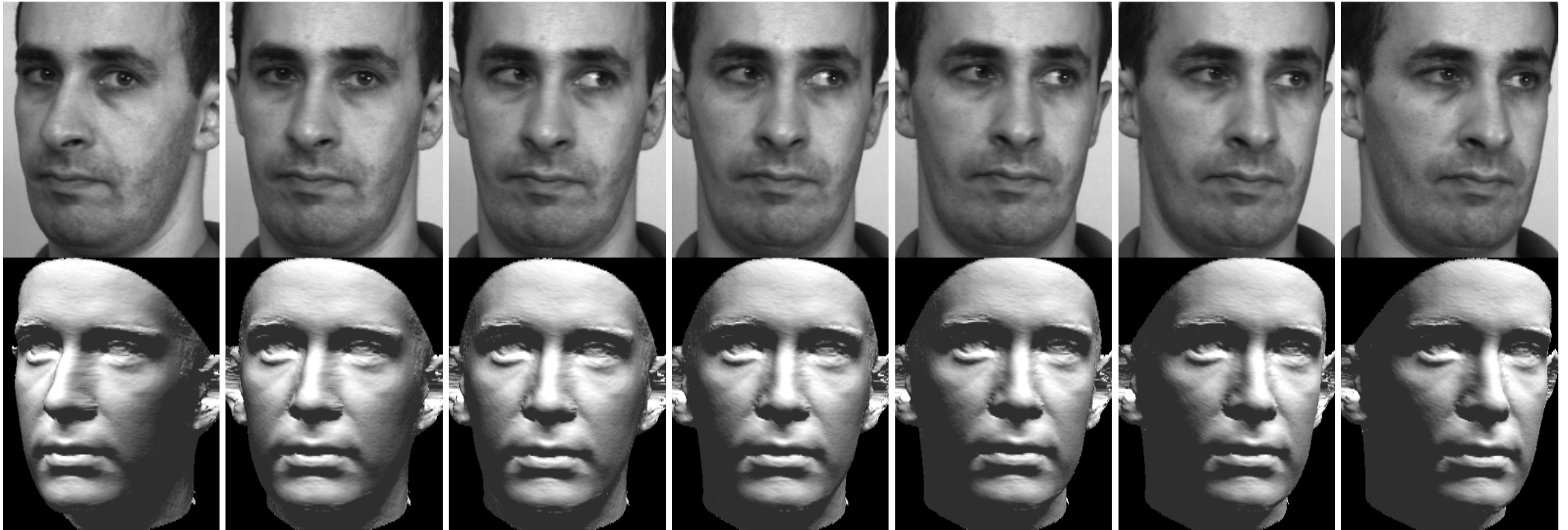


Transfer Function



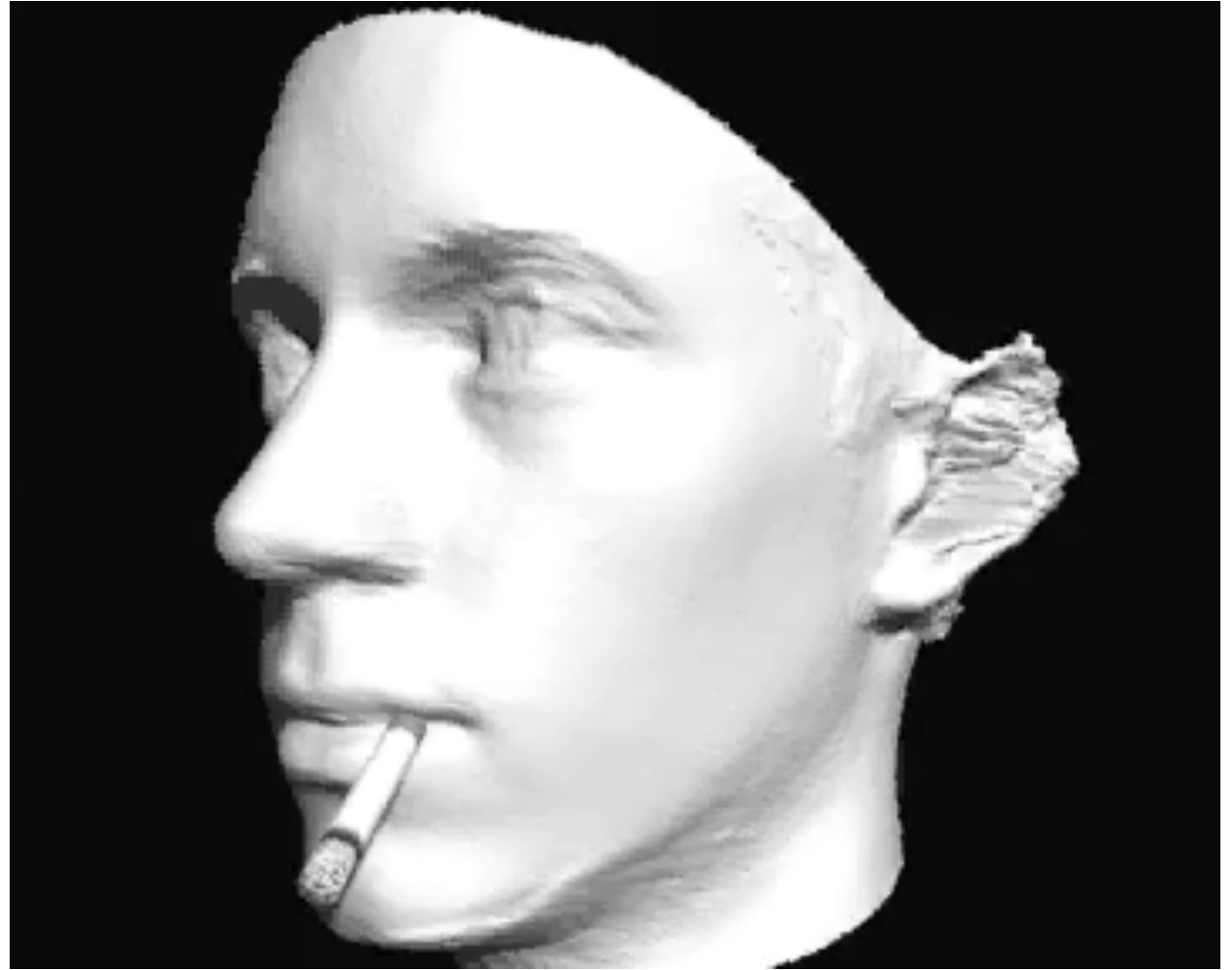
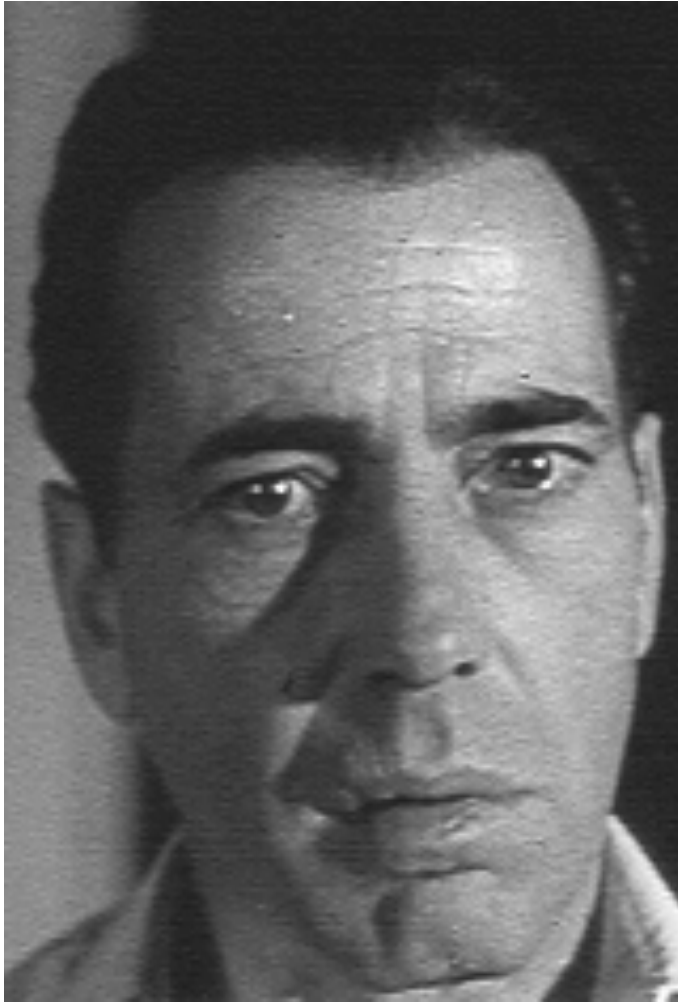
$$F_3(A, C_{i-1}, C_i, C_{i+1}) = \sum_{j \in Q_{i-1}} \|\Delta p_{i-1,i}^j\|^2 + \sum_{k \in Q_i} \|\Delta p_{i,i+1}^k\|^2$$

Model Based Bundle Adjustment



Adjusting the PCA coefficients to minimize the objective function yields an accurate face reconstruction from low-resolution images.

Model from Old Movie

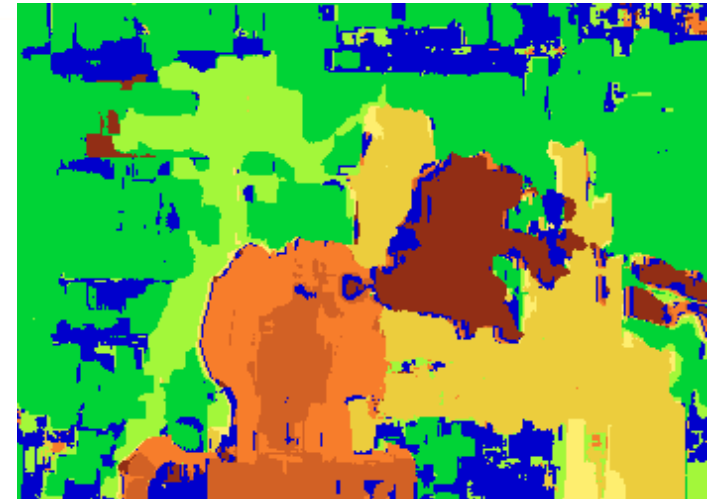


Adjusting the PCA coefficients to minimize the objective function yields an accurate face reconstruction from low-resolution images.

Limitations of Window Based Methods



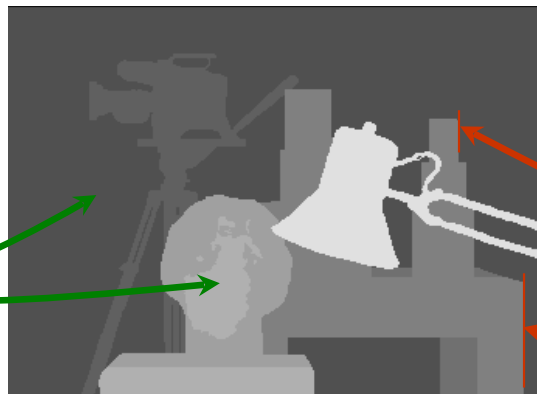
Ground truth



Correlation result

Energy Minimization

Disparity
continuous in
most places,



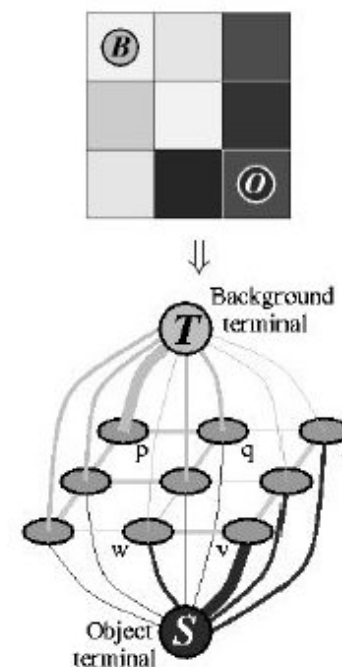
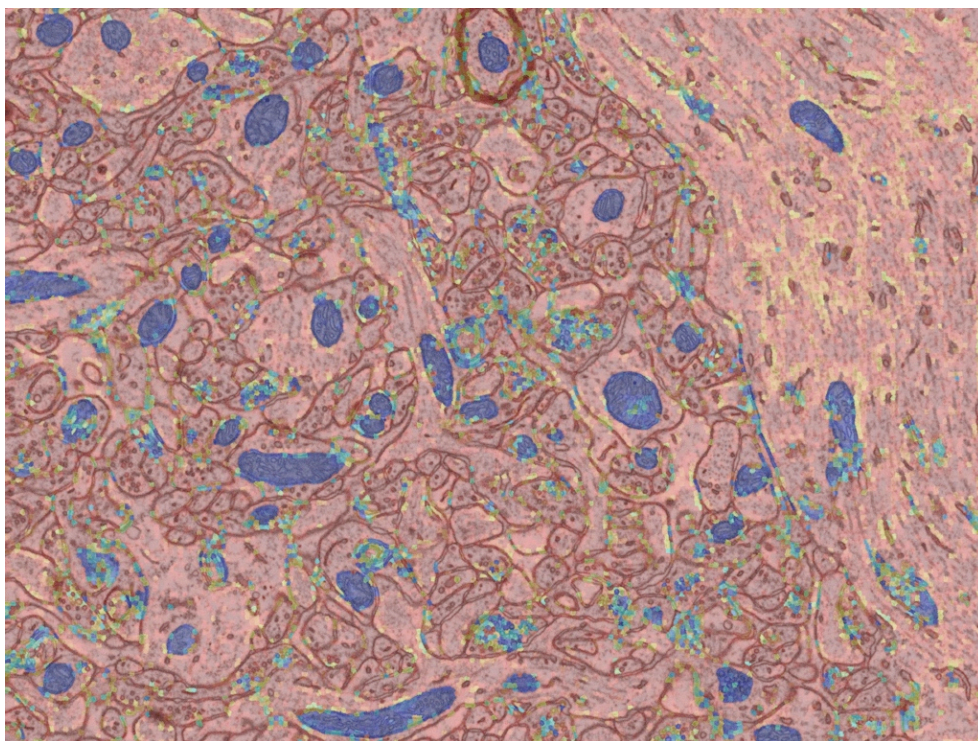
except at
depth
discontinuities

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

→ Minimize

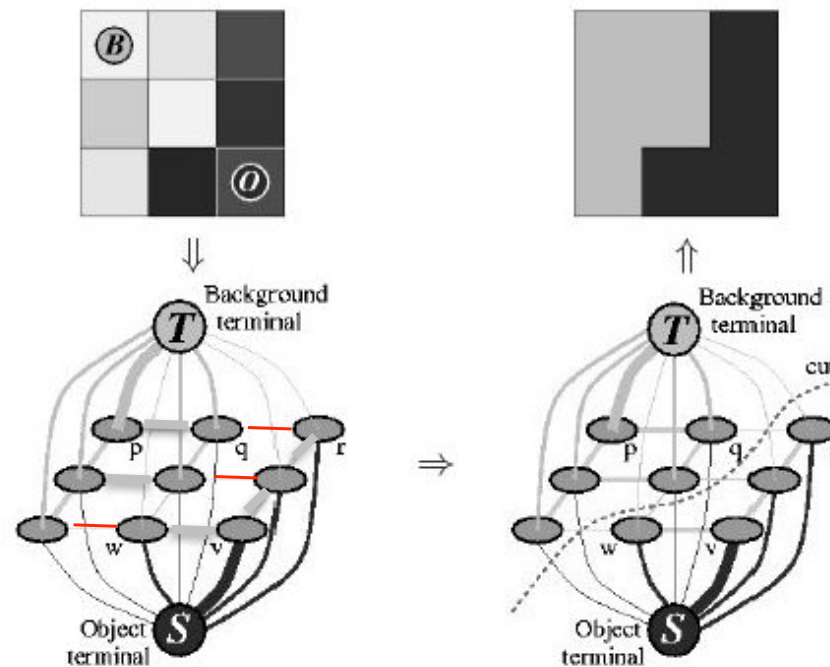
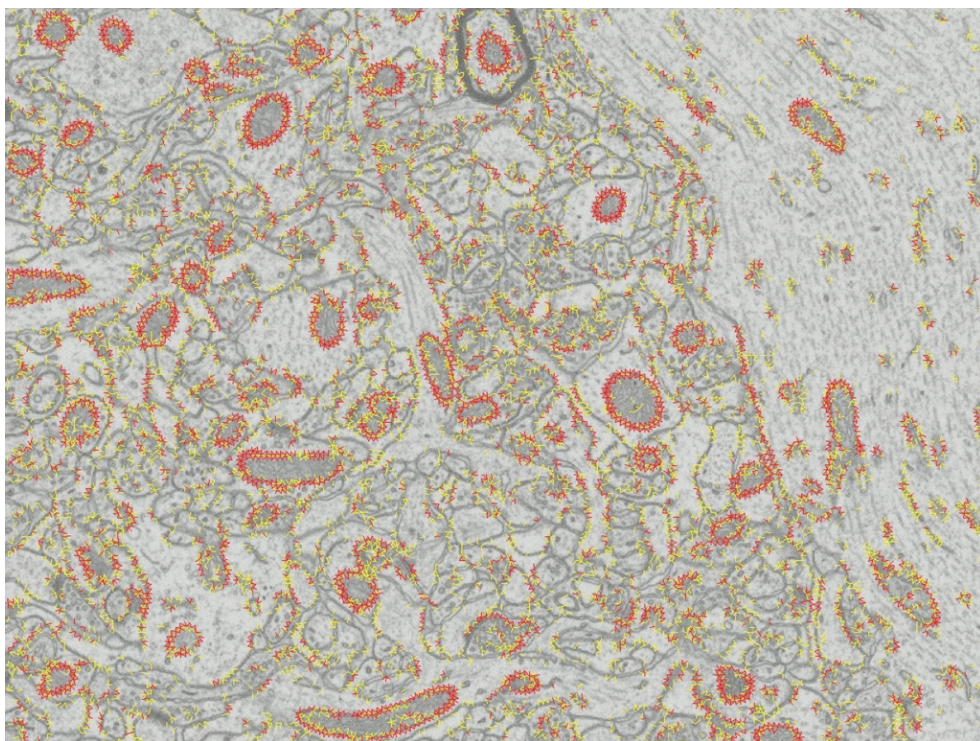
$$\sum [I_2(x+D(x,y), y) - I_1(x,y)]^2 + \lambda \sum [D(x+1,y) - D(x,y)]^2 + \mu \sum [D(x,y+1) - D(x,y)]^2$$

Reminder: Graph-Based Segmentation



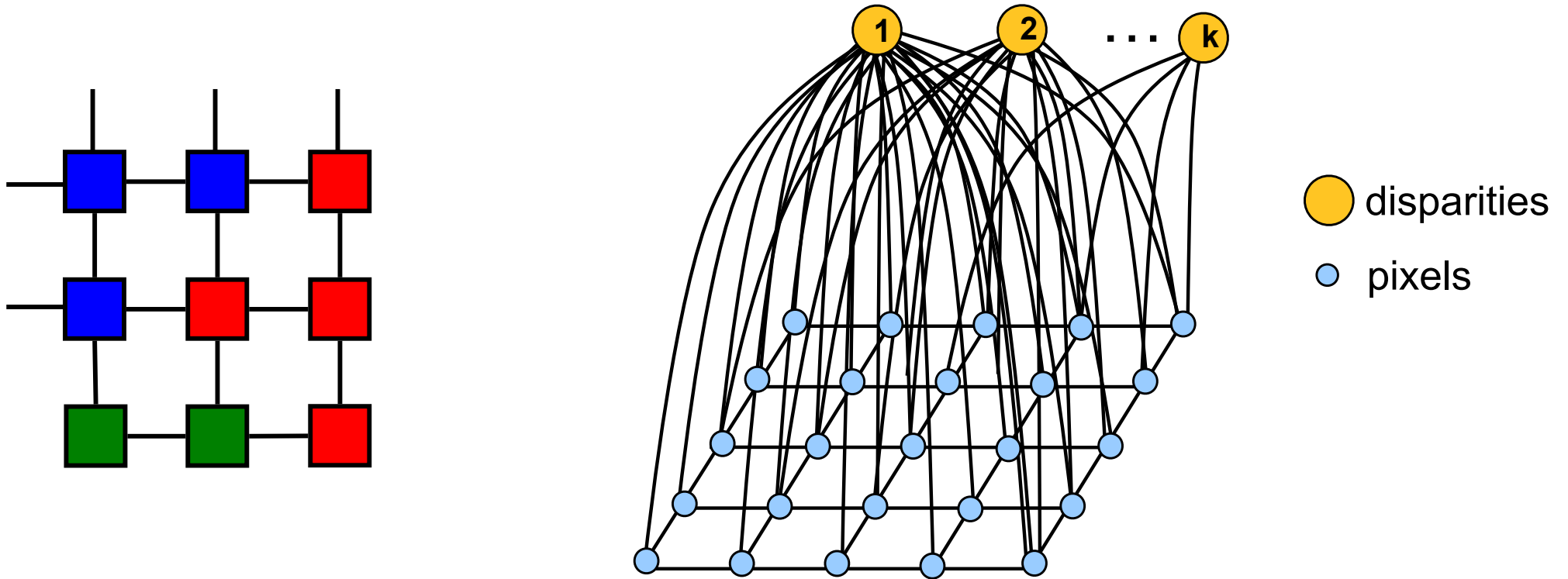
- A high probability of being a mitochondria can be represented by a strong edge connecting a supervoxel to the source and a weak one to the sink.
- And conversely for a low probability.

Reminder: Graph-Based Segmentation



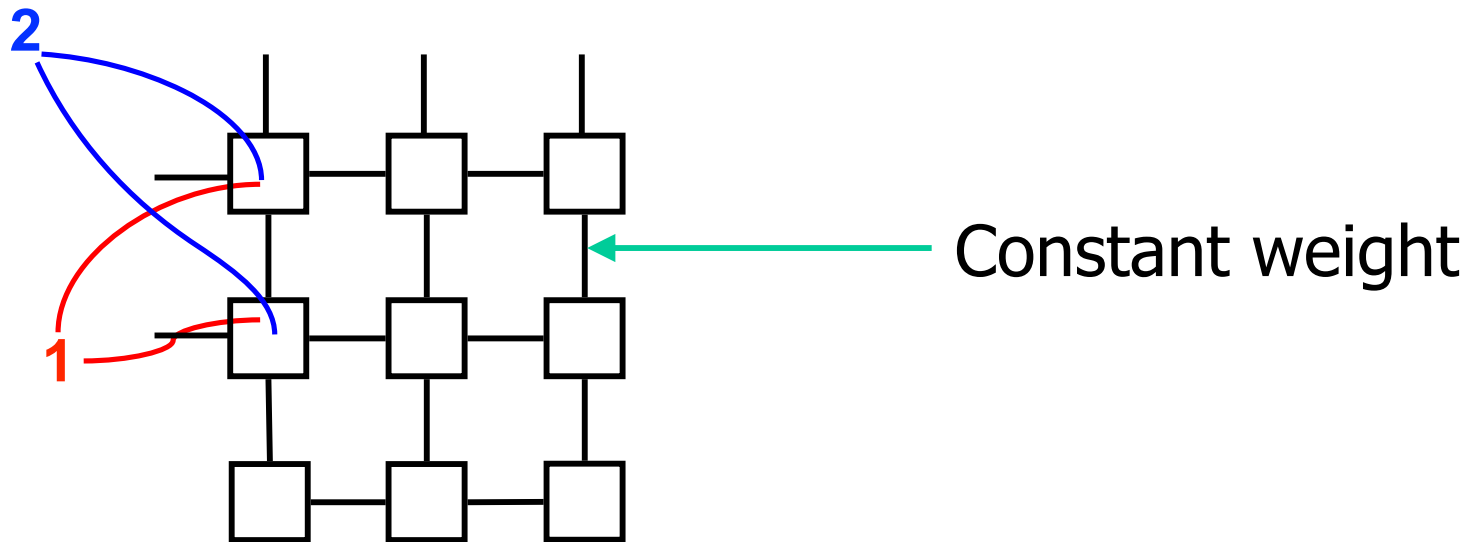
- Another classifier can be trained to assign a high-weight to edges connecting supervoxels belonging to the same class and a low one to others.
- Graph-cut can then be used to partition the pixels into separate regions.

Graph Cut for Stereo



1. Stereo is a labeling problem. \rightarrow Use graph cut.
2. 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

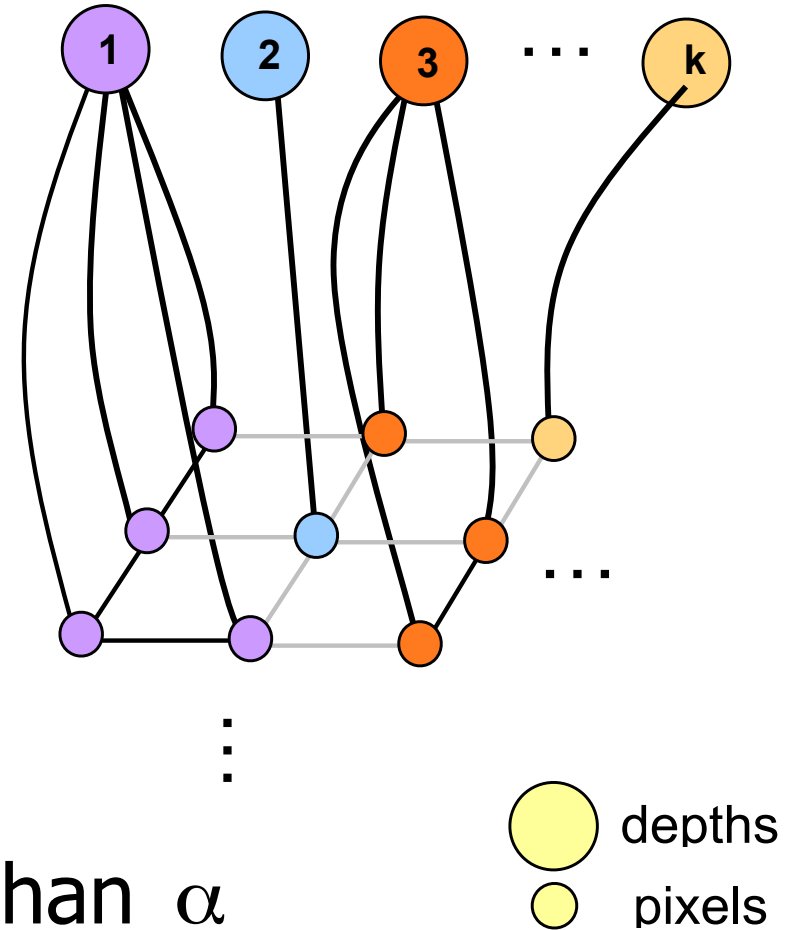
Minimize:

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

Graph cut algorithm:

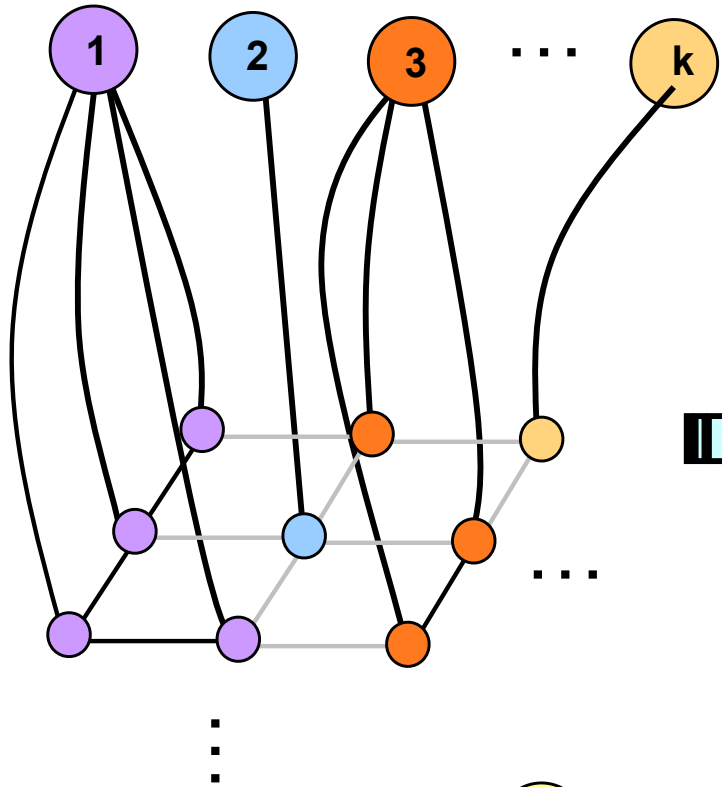
- Guarantees an absolute minimum only when there are only two possible disparities.
- Effective heuristics (a-expansion, a-b swap) otherwise.

α -Expansion

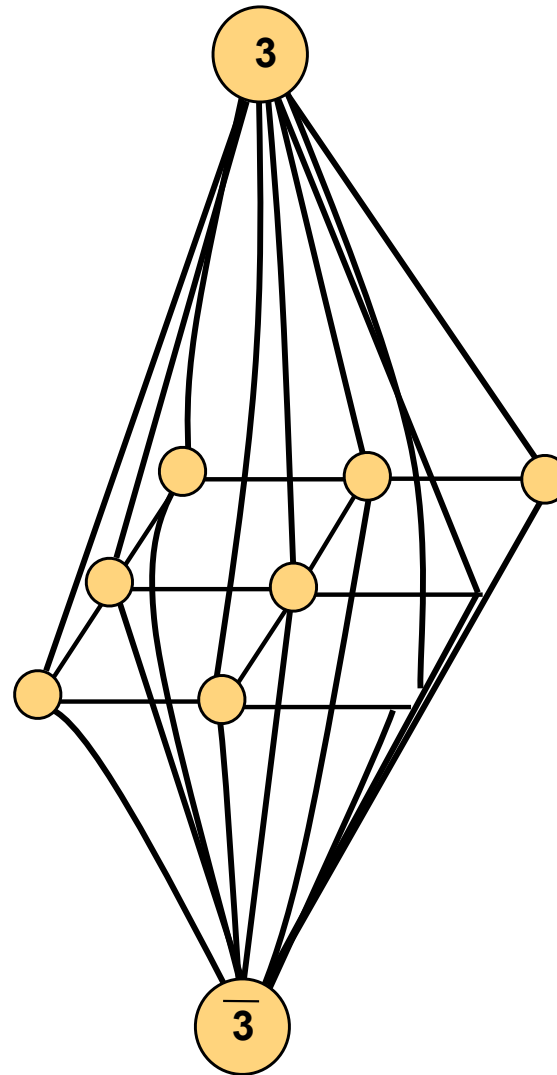


- Nodes having a label different than α can either keep it or switch to α .
- Edges between neighbors are updated according to the new labeling.
- Other edges remain unchanged.

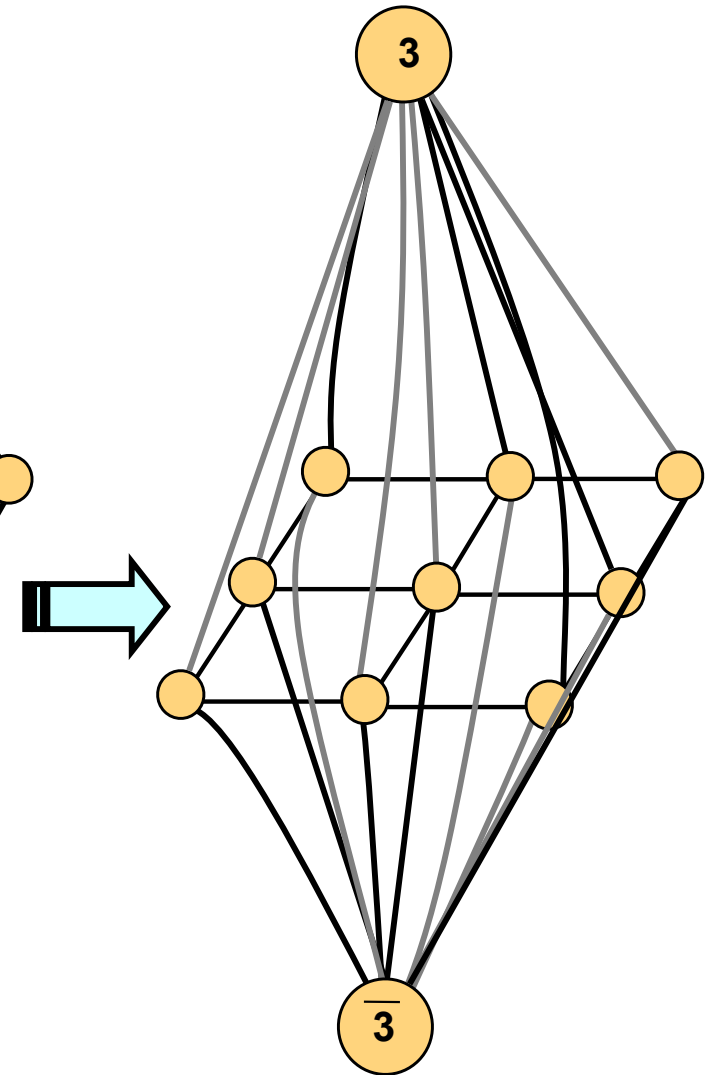
Example: 3 Expansion



● depths
● pixels

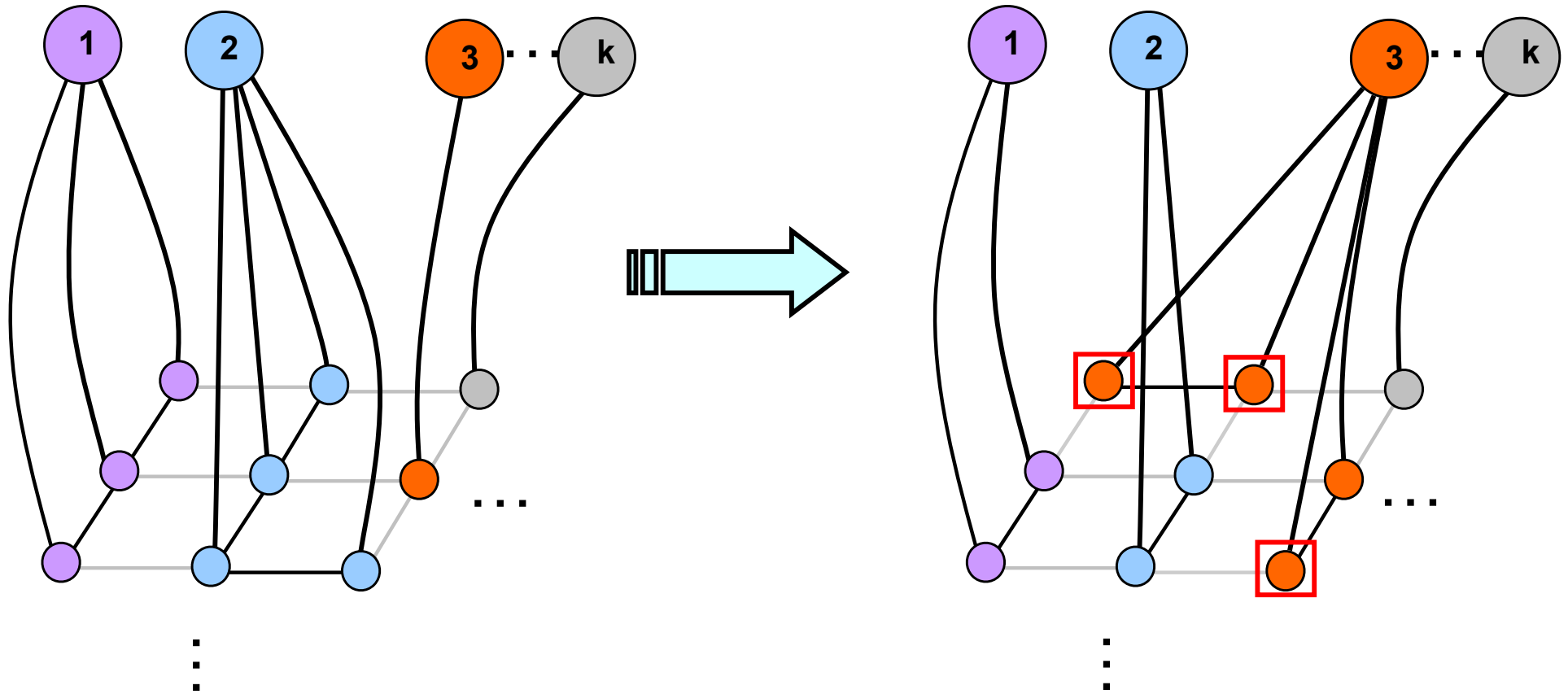


Connect all nodes to both 3 and $\bar{3}$



Find minimal cut

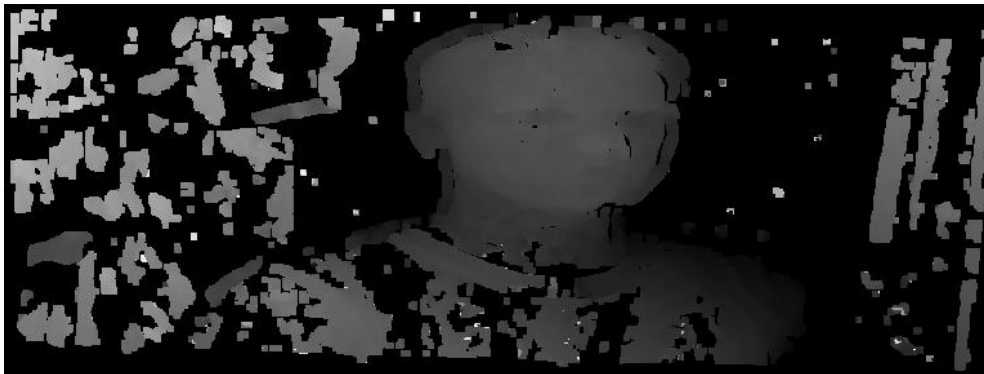
Example: 3 Expansion



Graph Cut Algorithm

1. Start with an arbitrary labeling
2. For every label α in $\{1, \dots, L\}$
 - Find the α -Expansion that minimizes the function
 - Update the graph by adding and erasing edges
3. Quit when no expansion improves the cost
4. Induce pixel labels

NCC vs Graph-Cut

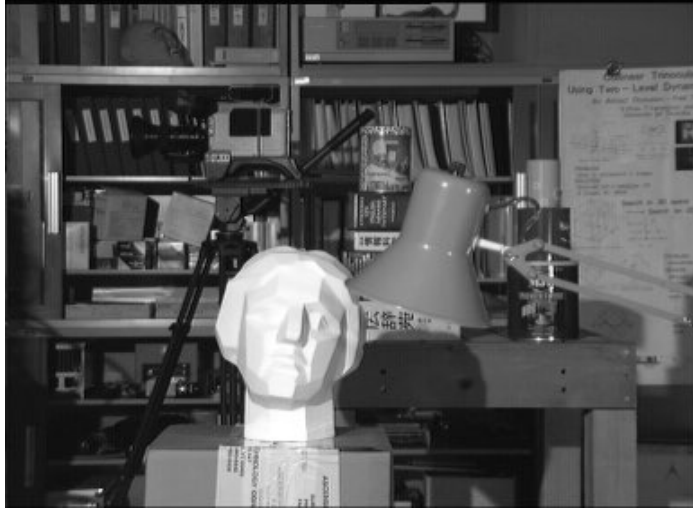


Normalized correlation

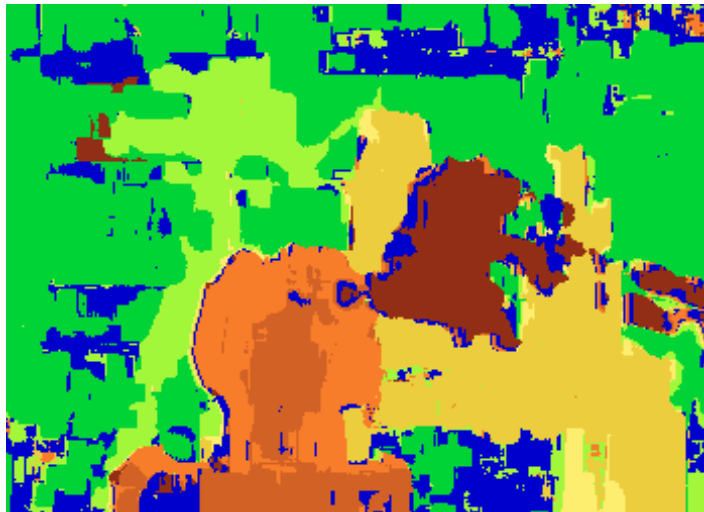
Graph Cut

NCC vs Graph Cut

left image



Normalized correlation



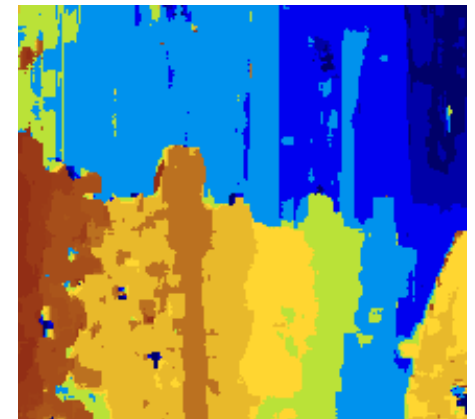
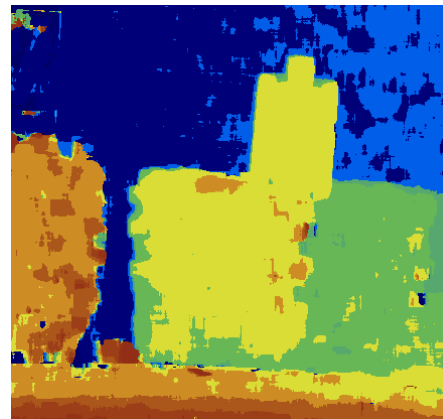
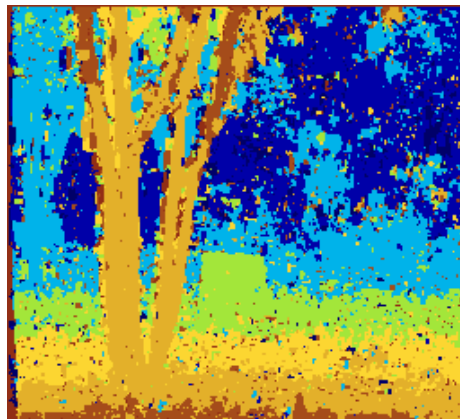
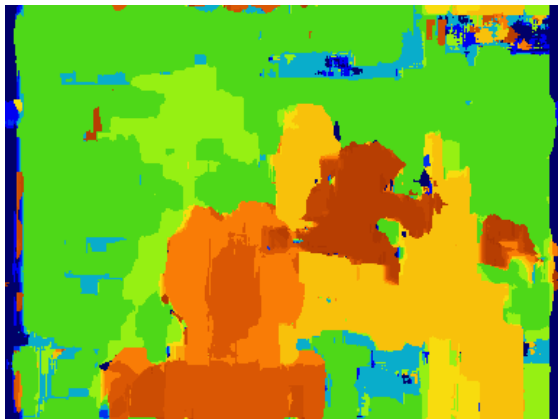
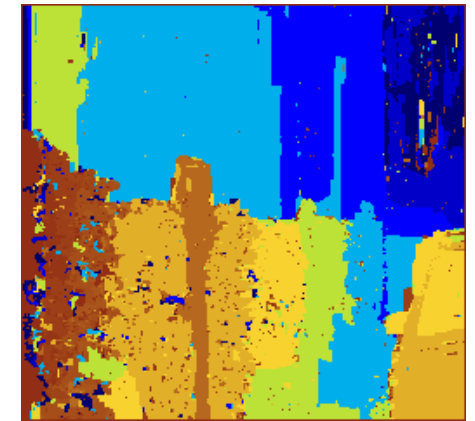
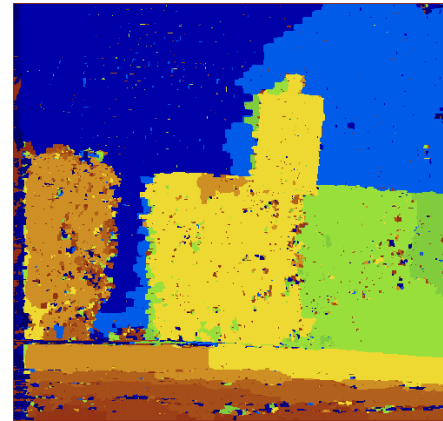
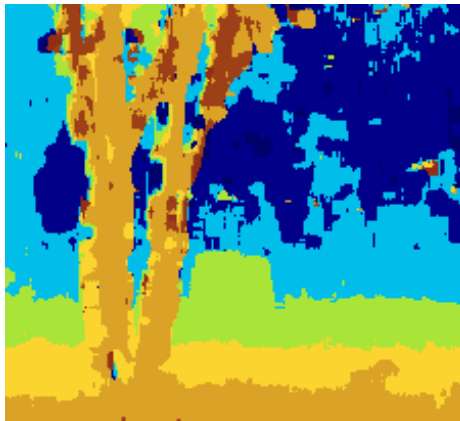
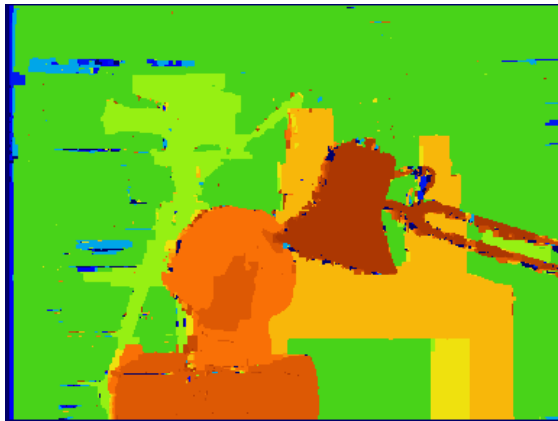
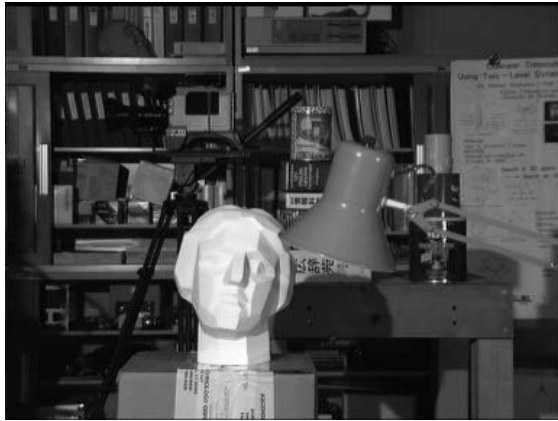
true disparities



Graph Cuts



NCC vs Graph Cut



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.