Master in
Computer Vision
Barcelona

Module: 3D Vision
Lecture 7: Triangulation methods.
Stereo and depth computation.
New view synthesis.
Lecturer: Gloria Haro

3D geometry from correspondences

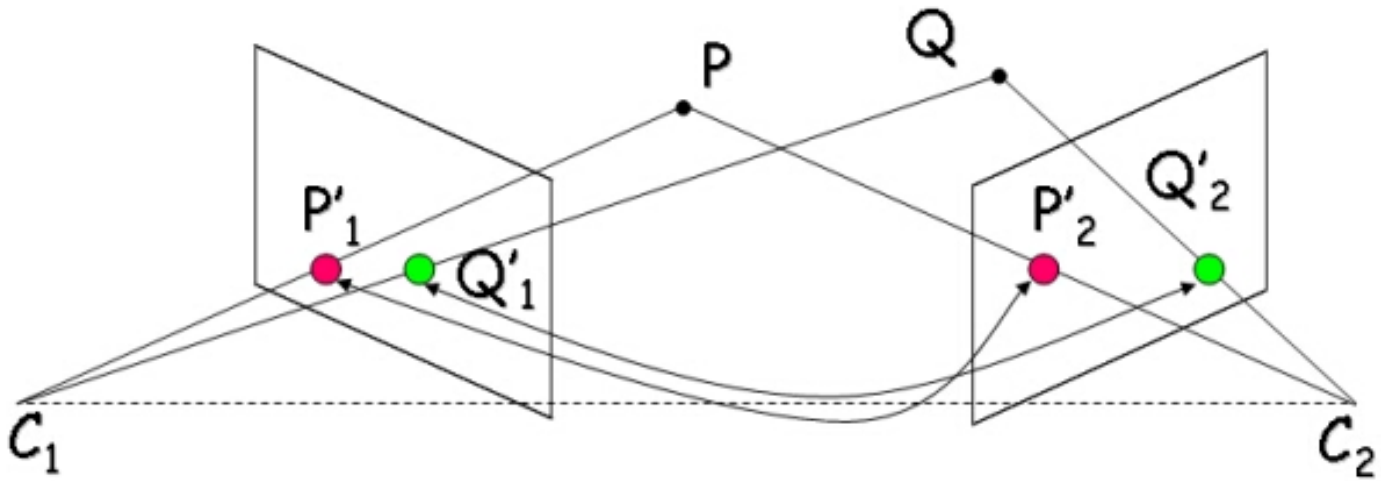
Problem	**Structure** (scene geometry)	**Motion** (camera geometry)	**Measurements** (data)
Pose estimation	known	???	3D to 2D correspondences
Triangulation, Stereo/depth, Multiview stereo	???	known	2D to 2D correspondences or images
Structure from motion	???	???	2D to 2D correspondences

Outline

- **Triangulation methods**
- Stereo and depth computation
- New view synthesis

Triangulation methods

Problem statement
Determining a point in 3D space given its projections onto two, or more, images. Called **structure computation**.



We know:

- the projection matrices,
- the point projections and their correspondences.

Image source: http://cvg.deis.unibo.it

Triangulation methods

Problem: Visual rays do not always intersect

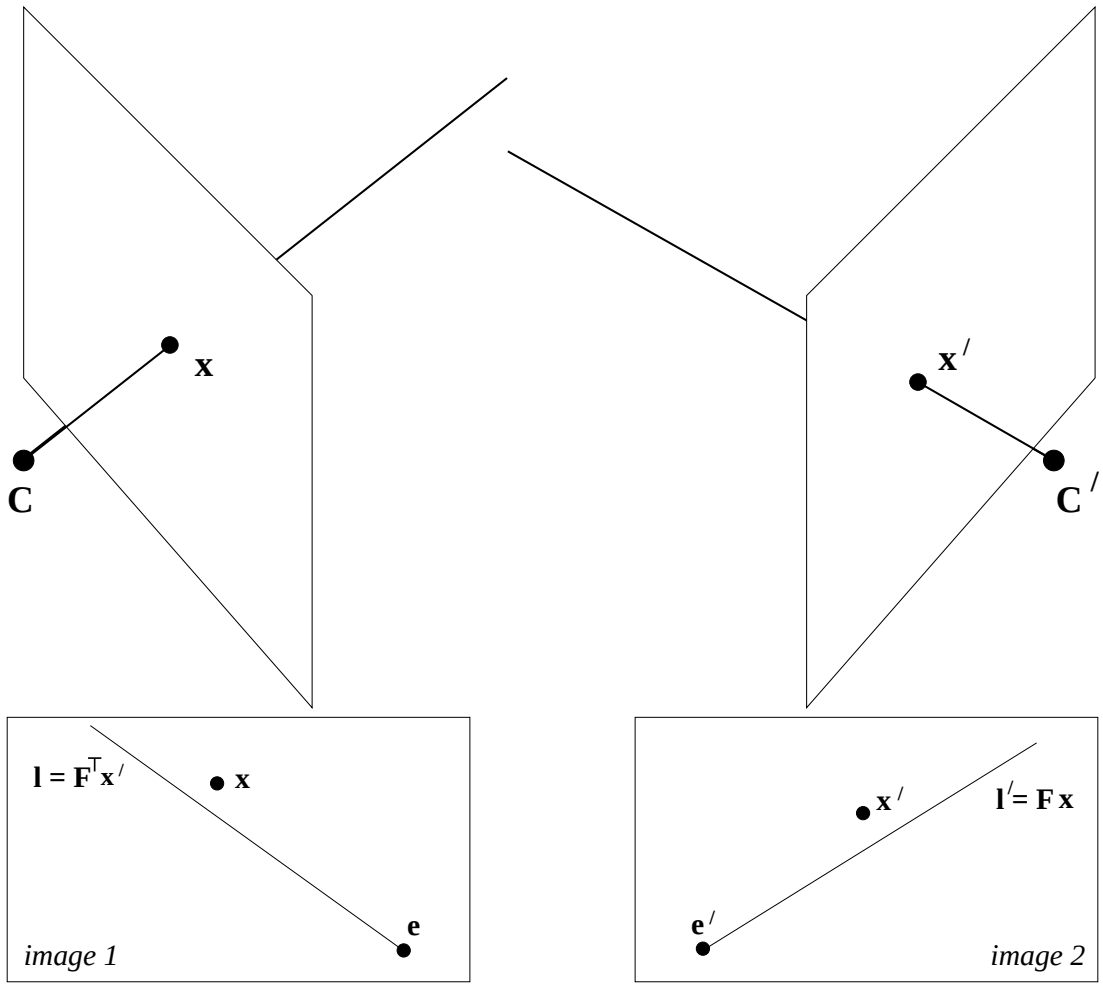
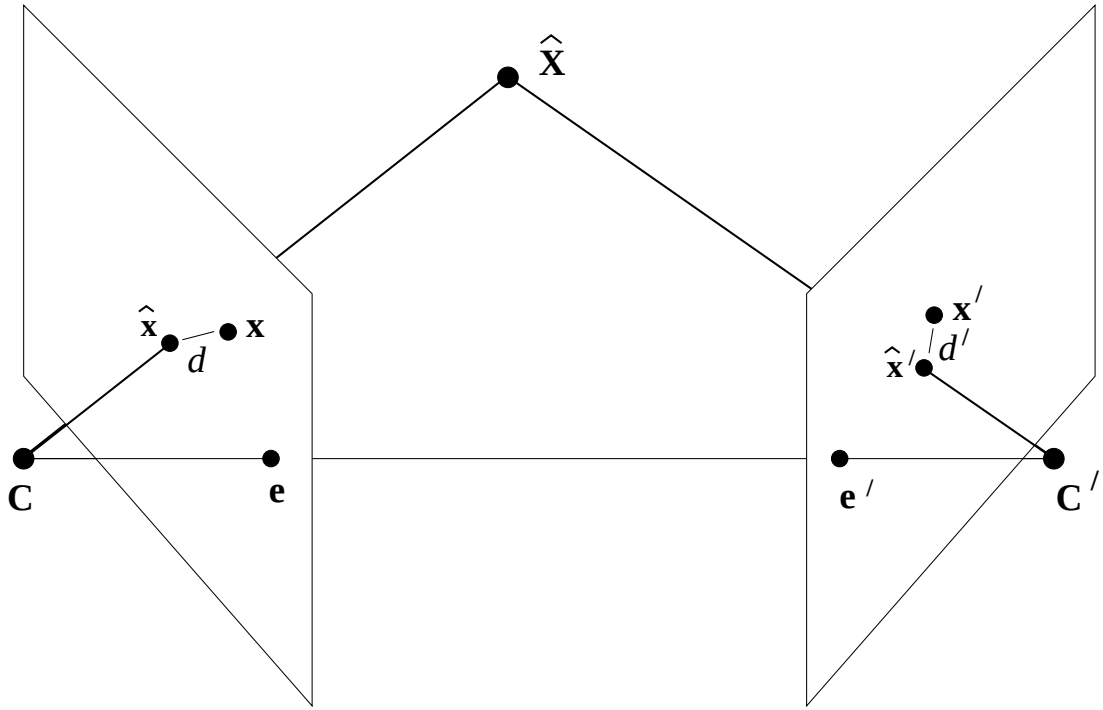


Image source: [Hartley and Zisserman 2004]

Triangulation methods

Problem formulation Two camera case



GOAL: Estimate a point $\widehat{\mathbf{X}} \in \mathbb{P}^3$ that satisfies $\widehat{\mathbf{x}} = P\widehat{\mathbf{X}}, \widehat{\mathbf{x}}' = P'\widehat{\mathbf{X}}$, for some points $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{x}}'$ in the images near the corresponding points $\mathbf{x}$, $\mathbf{x}'$.

Image source: [Hartley and Zisserman 2004]

Triangulation methods

Two kind of methods:

- **Algebraic (linear) methods**
  - **Homogeneous method (DLT)**
  - **Inhomogeneous method**
- **Geometric methods**

Triangulation methods

Linear triangulation methods

We have a correspondence $\mathbf{x} \longleftrightarrow \mathbf{x}'$, and the camera matrices P, and P' and we want to compute $\mathbf{X} \in \mathbf{R}^3$ such that,

$$\mathbf{x} \equiv P\mathbf{X} \text{ and } \mathbf{x}' \equiv P'\mathbf{X},$$

where $\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$, $\mathbf{x}' = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$ $P = \begin{pmatrix} \mathbf{p}^{1T} \\ \mathbf{p}^{2T} \\ \mathbf{p}^{3T} \end{pmatrix}$, and $P' = \begin{pmatrix} \mathbf{p}'^{1T} \\ \mathbf{p}'^{2T} \\ \mathbf{p}'^{3T} \end{pmatrix}$.

Triangulation methods

Linear triangulation methods

We have a correspondence $\mathbf{x} \longleftrightarrow \mathbf{x}'$, and the camera matrices P, and P' and we want to compute $\mathbf{X} \in \mathbf{R}^3$ such that,

$$\mathbf{x} \equiv P\mathbf{X} \text{ and } \mathbf{x}' \equiv P'\mathbf{X},$$

where $\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$, $\mathbf{x}' = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$ $P = \begin{pmatrix} \mathbf{p}^{1T} \\ \mathbf{p}^{2T} \\ \mathbf{p}^{3T} \end{pmatrix}$, and $P' = \begin{pmatrix} \mathbf{p}'^{1T} \\ \mathbf{p}'^{2T} \\ \mathbf{p}'^{3T} \end{pmatrix}$.

The problem can be written as:

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

where

$$\mathbf{A} = \begin{pmatrix} x\mathbf{p}^{3T} - \mathbf{p}^{1T} \\ y\mathbf{p}^{3T} - \mathbf{p}^{2T} \\ x'\mathbf{p}'^{3T} - \mathbf{p}'^{1T} \\ y'\mathbf{p}'^{3T} - \mathbf{p}'^{2T} \end{pmatrix}.$$

Triangulation methods

Homogeneous method (DLT)

$$\min_{\mathbf{X}} \|\mathbf{A}\mathbf{X}\|_2$$

$$\text{such that } \|\mathbf{X}\|_2 = 1.$$

Solution: The singular vector associated to the minimum singular value of **A**.

Normalization: Scaling and translation so that both pixel coordinates are in the interval $[-1, 1]$.

$$H = \begin{pmatrix} 2/\mathrm{nx} & 0 & -1 \\ 0 & 2/\mathrm{ny} & -1 \\ 0 & 0 & 1 \end{pmatrix}.$$

We build **A** from $\mathbf{x} = H\mathbf{x}$, $\mathbf{x}' = H\mathbf{x}'$, $P = HP$, and $P' = HP'$.

Triangulation methods

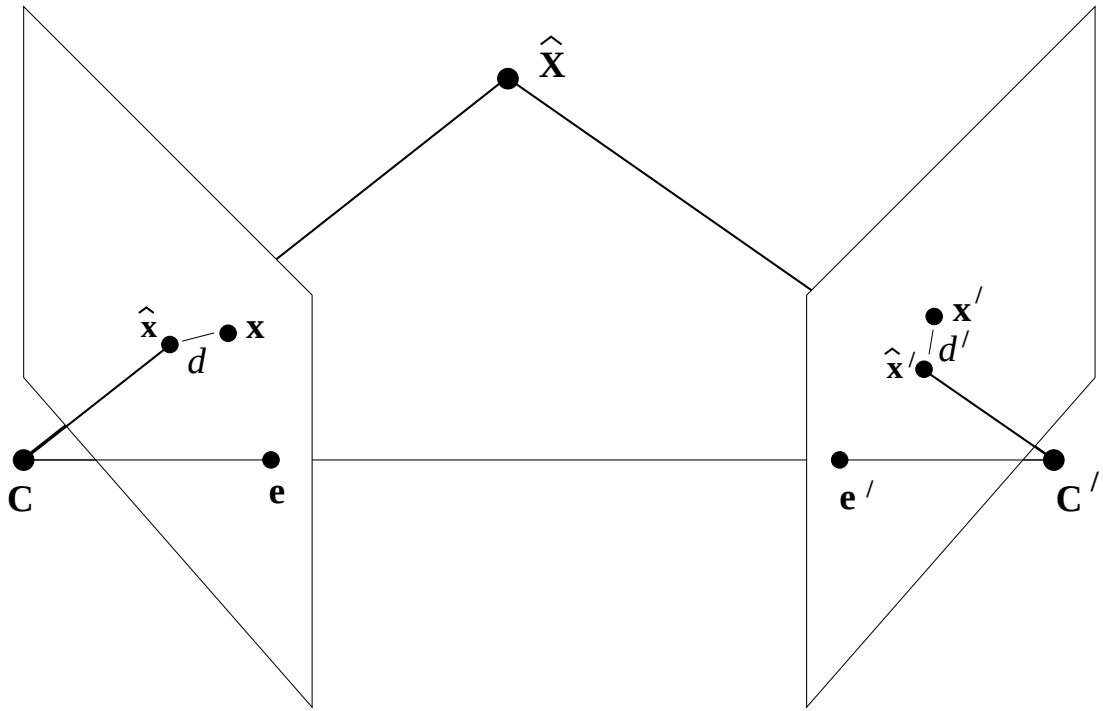
Advantages and disadvantages of linear homogeneous method

- ☺ Linear, non-iterative method
- ☺ It often provides acceptable results and serves as an initialization for the geometric methods.
- ☺ It generalizes easily to more than two views.
- ☹ It is not projective-invariant.
  (Because of the condition $\|\mathbf{X}\|_2 = 1$, the point $\mathbf{X}$ solving the original problem will not correspond to a solution $H\mathbf{X}$ for the transformed problem.)

The recommended triangulation method would be one that is invariant to the projective frame of the cameras, and minimizes a geometric image error.

Triangulation methods

Geometric methods



$$\min_{\widehat{\mathbf{x}},\widehat{\mathbf{x}}'} d^2(\mathbf{x},\widehat{\mathbf{x}}) + d^2(\mathbf{x}',\widehat{\mathbf{x}}') = \min_{\widehat{\mathbf{x}},\widehat{\mathbf{x}}'} \|[\mathbf{x}] - [\widehat{\mathbf{x}}]\|_2^2 + \|[\mathbf{x}'] - [\widehat{\mathbf{x}}']\|_2^2$$

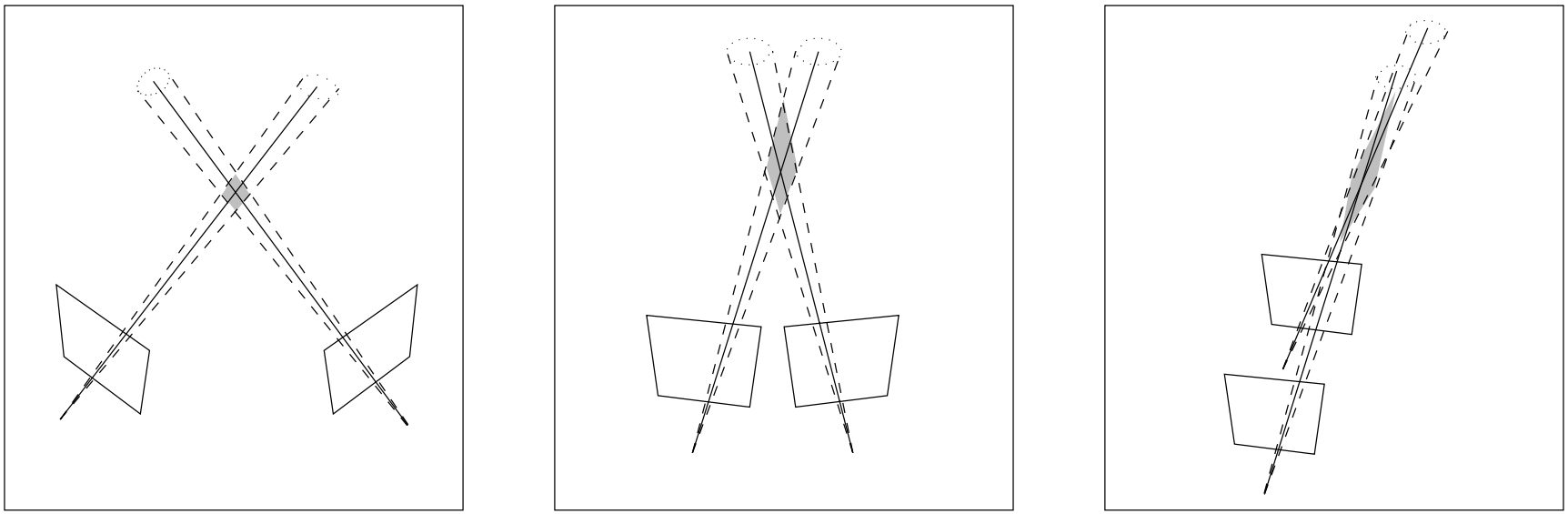
$$\text{such that } \widehat{\mathbf{x}}'^T F \widehat{\mathbf{x}} = 0.$$

Then, get $\widehat{\mathbf{X}}$ from $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{x}}'$ (the visual rays intersect).

Image source: [Hartley and Zisserman 2004]

Triangulation methods

Uncertainty in the 3D point localization / depth estimation



Small angle between rays/baseline: large triangulation/depth error
Large angle between rays/baseline: difficult search problem

Image source: [Hartley and Zisserman 2004]

Outline

- Triangulation methods
- **Stereo and depth computation**
- New view synthesis

Monocular depth perception

Monocular cues

size

perspective

aerial perspective

defocus blur

occlusions

lighting & shading

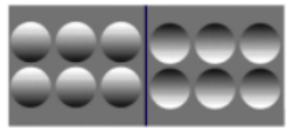
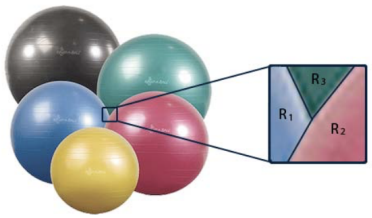
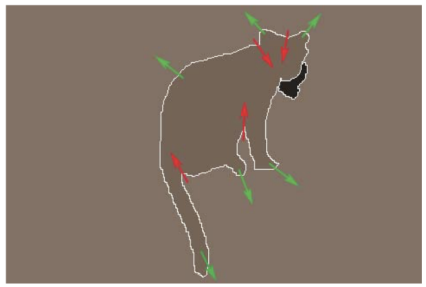


Image source: [J. Ruiz]

Monocular depth estimation

Works based on low level cues

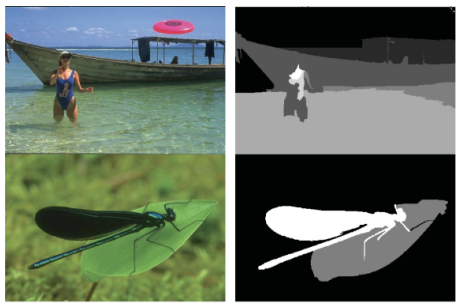


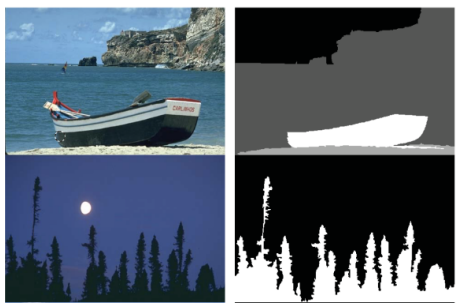


See e.g. (and references therein/to them):

G. Palou and P. Salembier. Monocular Depth Ordering Using T-Junctions and Convexity Occlusion Cues. IEEE Trans. on Image Processing, 22(5), 2013.

F. Calderero and V. Caselles. Recovering relative depth from low-level features without explicit T-junction detection and interpretation. International Journal of Computer Vision, 104. 2013.





Images source: [Palou and Salembier]

Monocular depth estimation

Works based on learning

See e.g. (and references therein/to them):

A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. NIPS, 2005. **MRF supervised**

A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3-d scene structure from a single still image. TPAMI, 2008. **MRF supervised**

D. Eigen, C. Puhrsch, R. Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. NIPS, 2014. **DL supervised**

C. Godard, O. Mac Aodha, G.J. Brostow. Unsupervised monocular depth estimation with left-right consistency, CVPR, 2017. **DL unsupervised**

A. Gordon, H. Li, R. Jonschkowski, A. Angelova. Depth From Videos in the Wild: Unsupervised Monocular Depth Learning From Unknown Cameras, ICCV, 2019. **DL unsupervised (moving camera)**

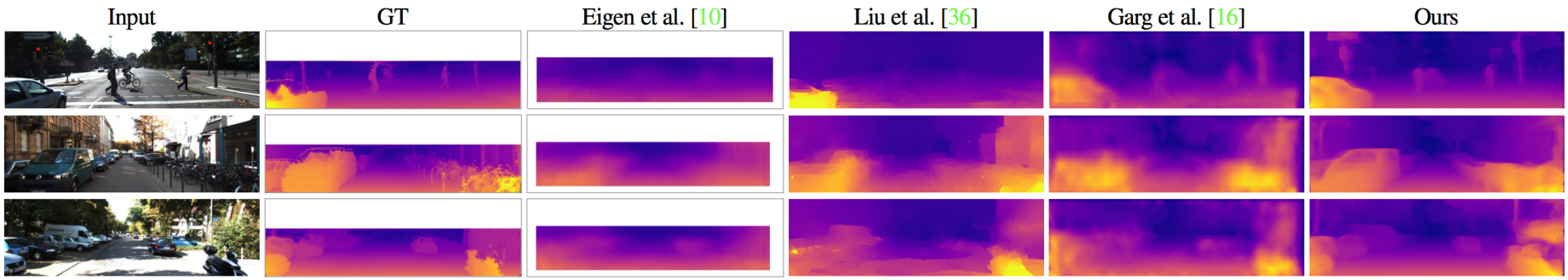
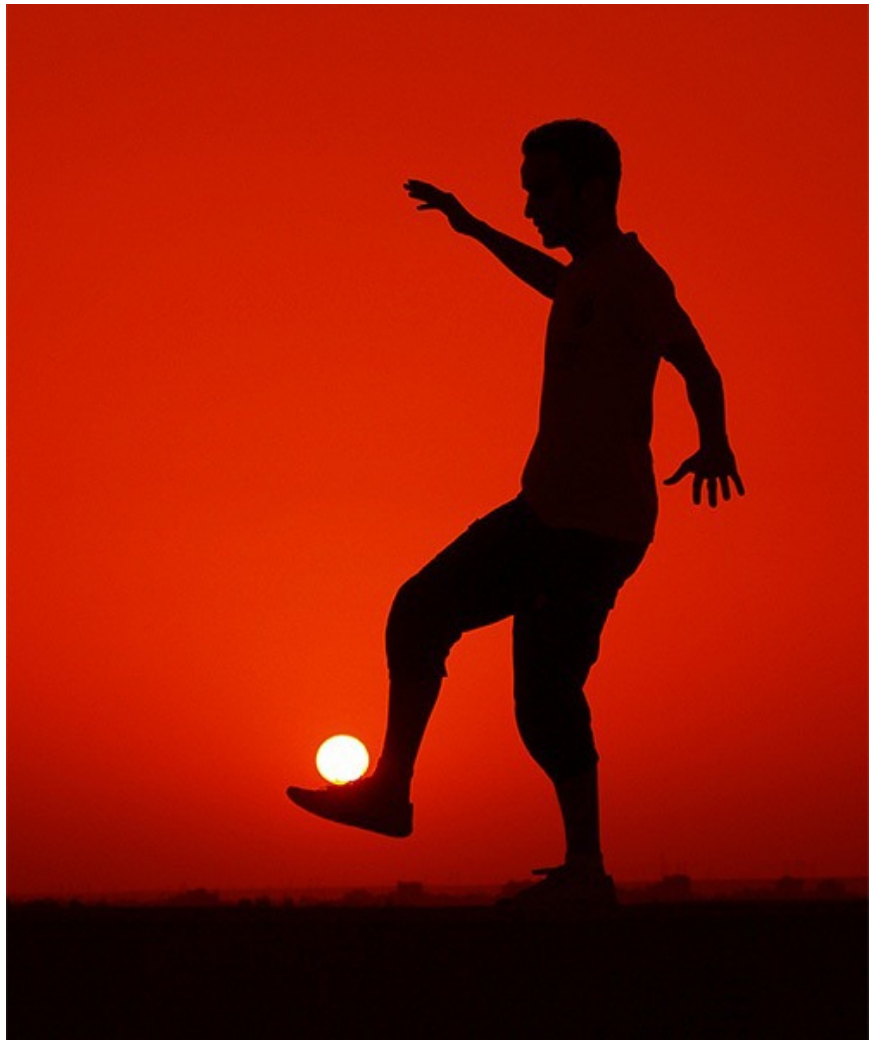


Image source: [Godard et al.]

Why stereo is useful?

3D structure and depth are ambiguous from a single view



Stereo and depth computation

Stereoscope

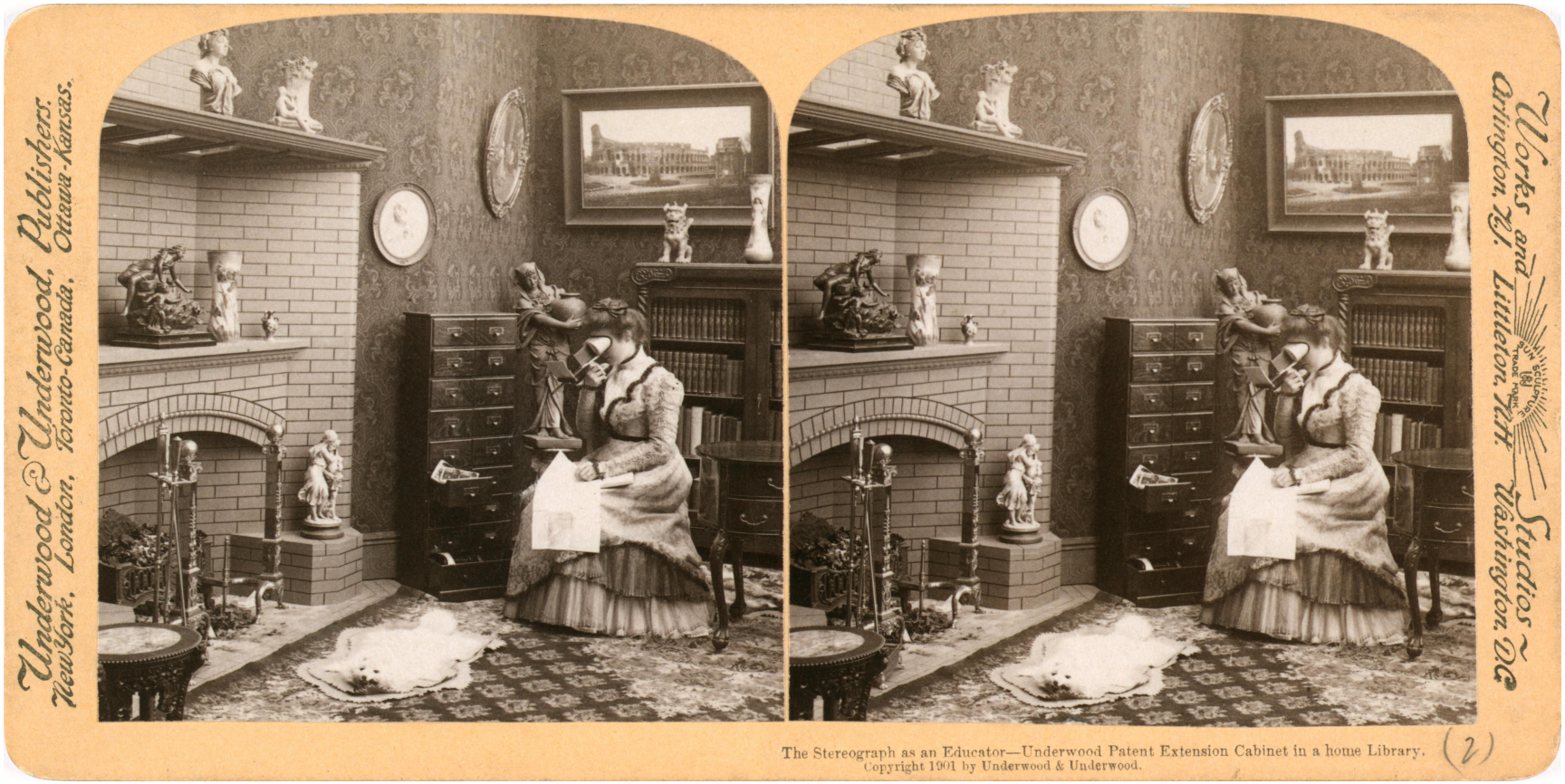


Image source: [Wikipedia]

Stereo and depth computation

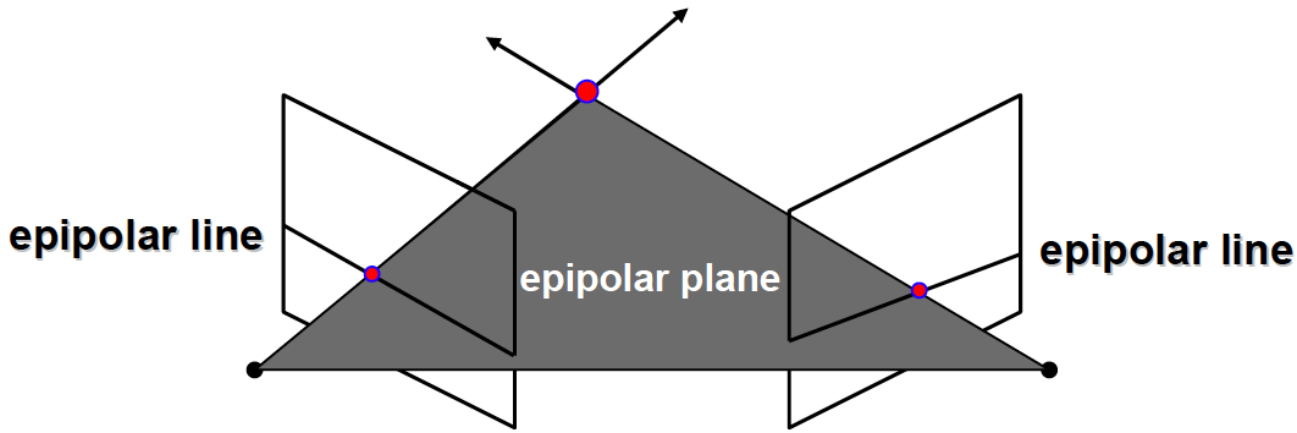
Anaglyph images

Image source: [S. Seitz]

Stereo and depth computation

Stereo correspondence

Determine pixel correspondences, i.e. pairs of pixels that correspond to the same scene point.



The **epipolar constraint** reduces the problem to 1D search along the epipolar lines.

Image source: [S. Seitz]

Stereo and depth computation

Stereo image rectification

Images are reprojected onto a common plane parallel to the baseline
Simplifies the correspondence problem: pixel motion is horizontal

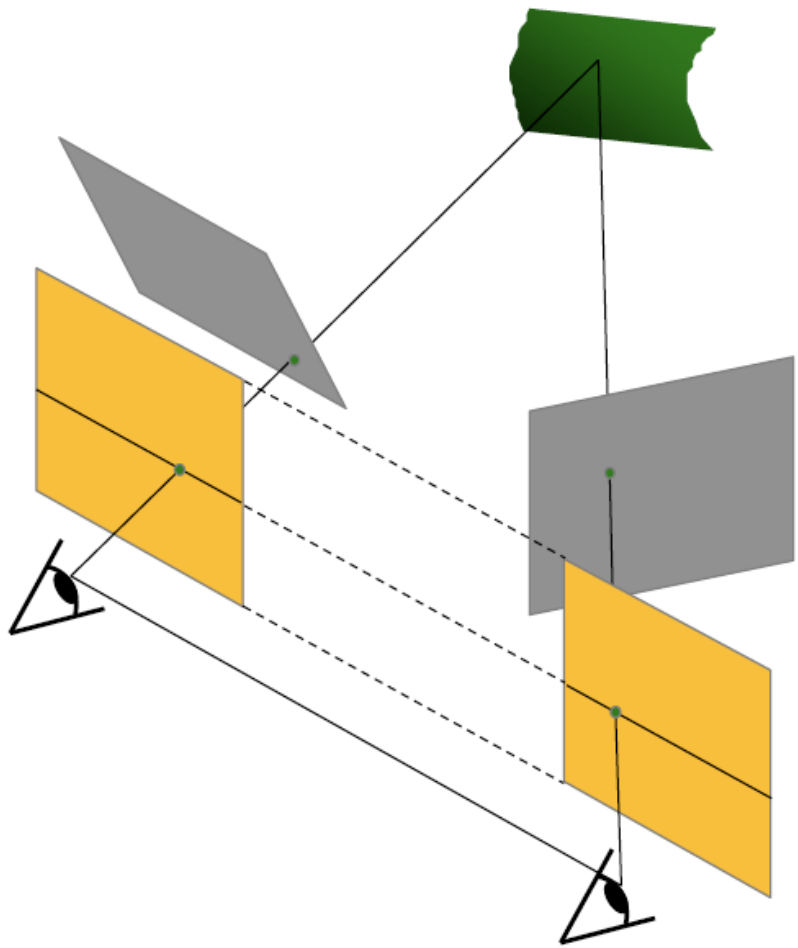
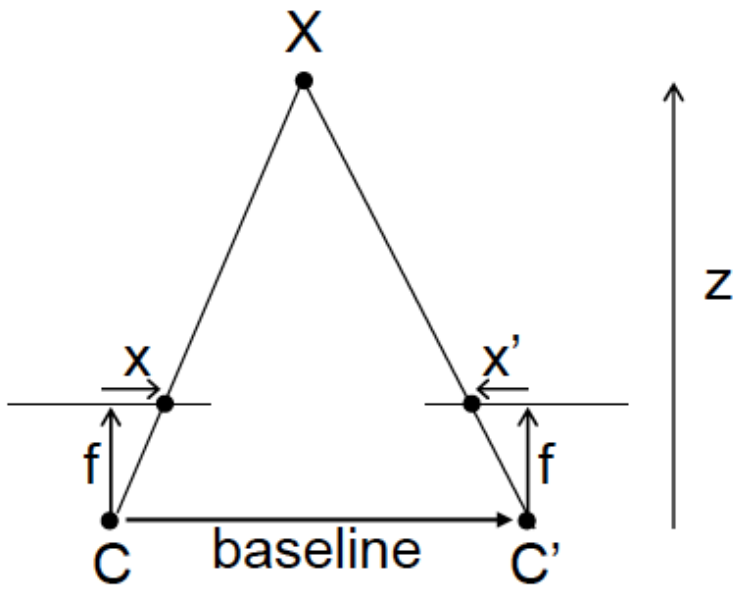


Image source: [S. Seitz]

Stereo and depth computation

Depth from disparity

Pair of recitified stereo images



$$disparity = x - x' = \frac{baseline * f}{z}$$

Image source: [S. Seitz]

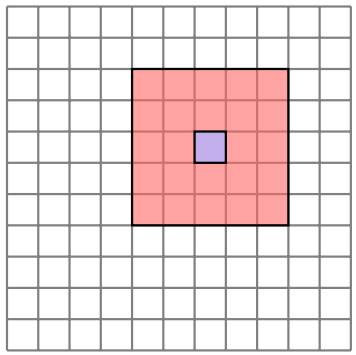
Stereo and depth computation

Stereo matching algorithm

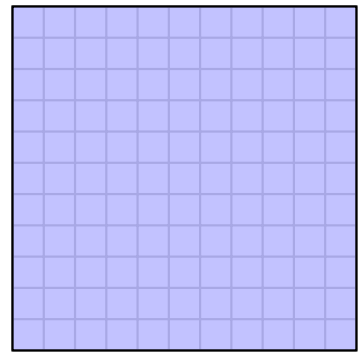
GOAL: Match pixels in conjugate epipolar lines

Several approaches
`http://www.middlebury.edu/stereo`
`http://www.cvlibs.net/datasets/kitti/`

Classification in two main families of approaches:



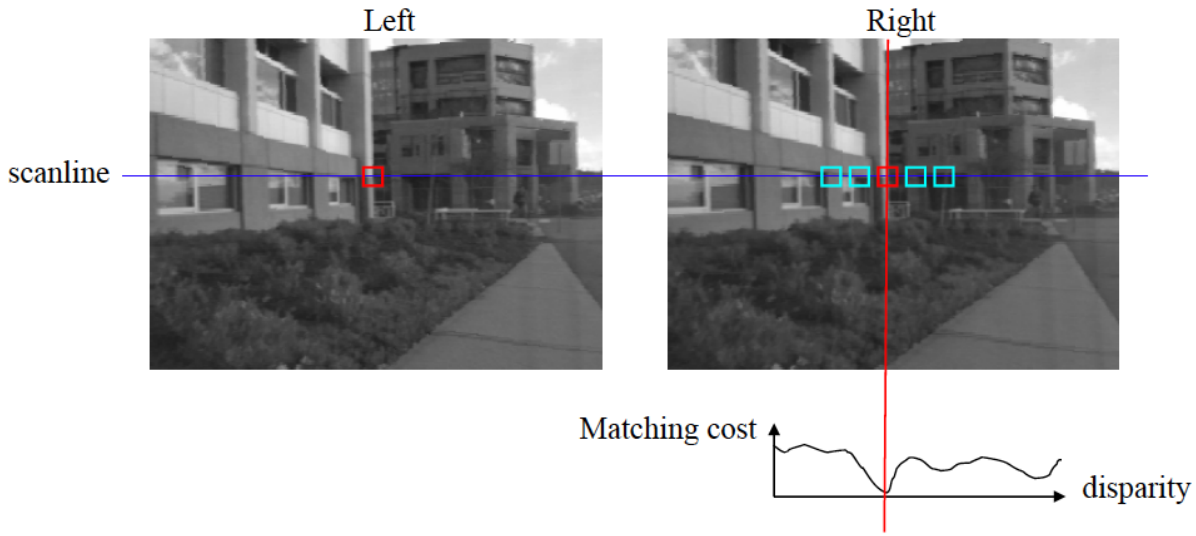
Local methods



Global methods

Stereo and depth computation

Local methods for stereo matching



BASIC IDEA:

For every pixel in the left image

1. Slide a window along the same line in the right image and compare its content to that of the reference window in the left image.
2. Pick pixel with minimum matching cost (WTA: Winner-Takes-All).

Image source: [S. Lazebnik]

Stereo and depth computation

Matching costs

- SSD: Sum of Squared Differences $(m = 2)$
- SAD: Sum of Absolute Differences $(m = 1)$

$$C(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q}) \, |I_1(\mathbf{q}) - I_2(\mathbf{q} + \mathbf{d})|^m, \quad \sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q}) = 1$$

$$N_\mathbf{p} = \{\mathbf{q} = (q_1, q_2)^T \mid p_1 - \tfrac{n}{2} \leq q_1 \leq p_1 + \tfrac{n}{2}, \; p_2 - \tfrac{n}{2} \leq q_2 \leq p_2 + \tfrac{n}{2}\}$$

- NCC: Normalized Cross Correlation

$$NCC(\mathbf{p}, \mathbf{d}) = \frac{\sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q})(I_1(\mathbf{q}) - \bar{I}_1)(I_2(\mathbf{q} + \mathbf{d}) - \bar{I}_2)}{\sigma_{I_1} \sigma_{I_2}}$$

$$\bar{I}_1 = \sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q}) I_1(\mathbf{q}); \quad \bar{I}_2 = \sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q}) I_2(\mathbf{q} + \mathbf{d});$$

$$\sigma_{I_1} = \sqrt{\sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q})(I_1(\mathbf{q}) - \bar{I}_1)^2};$$

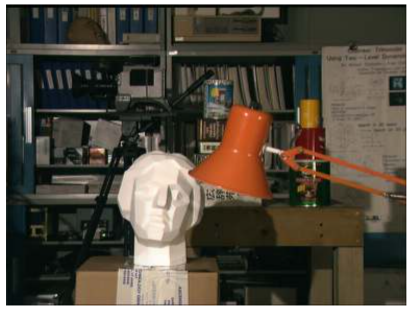
$$\sigma_{I_2} = \sqrt{\sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q})(I_2(\mathbf{q} + \mathbf{d}) - \bar{I}_2)^2}$$

- Others: Census Transform, Rank Filters, Mutual Information, CNNs-based

Stereo and depth computation

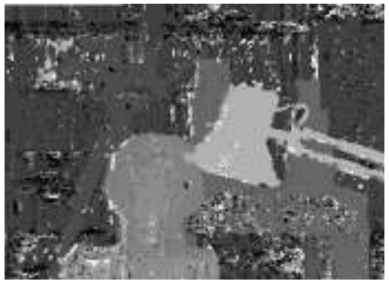
Effect of the window size

(a) Reference image.

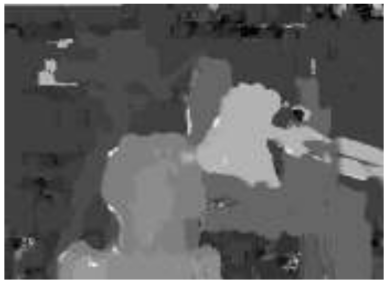


(b) Target image.

(c) Ground-truth disparity map.

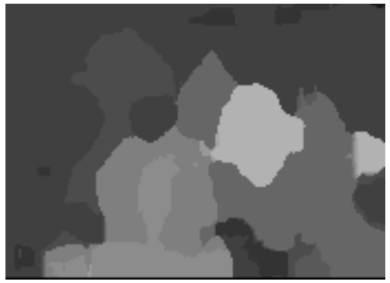


(a) 3×3



(b) 9×9

(c) 20×20



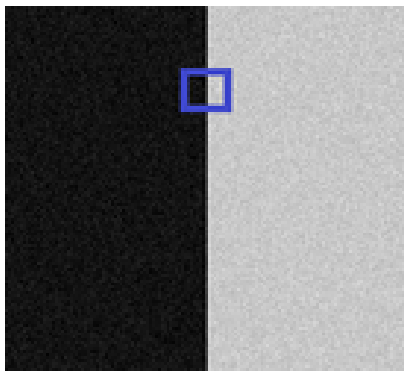
(d) 35×35

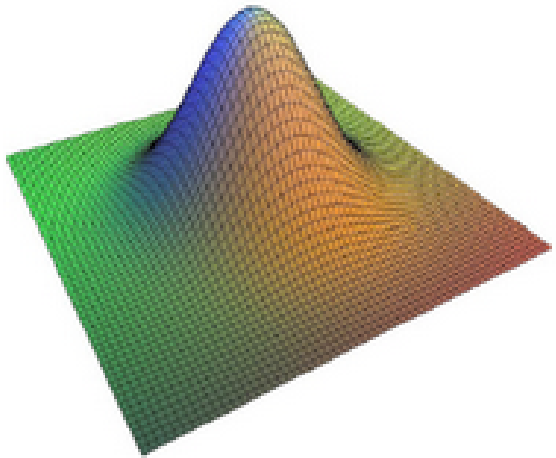
Smaller window	Larger window
+ more detail	− less detail
− more noise	+ smoother disparity maps

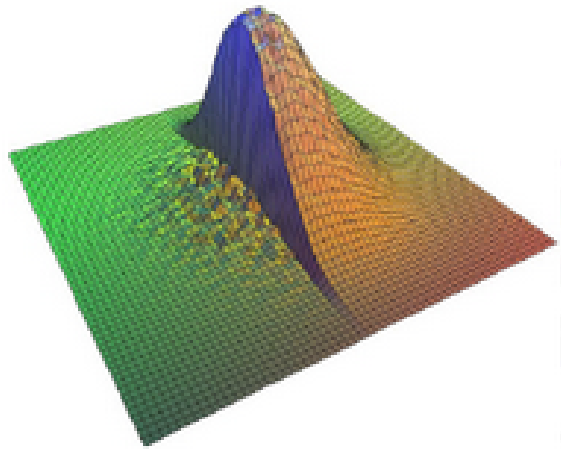
Image source:
L. F. Julià, P. Monasse. Bilaterally Weighted Patches for Disparity Map Computation. Image Processing On Line, 2015.

Stereo and depth computation

Gaussian vs. Bilateral weights







Stereo and depth computation

Bilateral weights

$$w(\mathbf{p},\mathbf{q}) = w_{col}(\mathbf{p},\mathbf{q})\, w_{pos}(\mathbf{p},\mathbf{q}) = exp\left(-\frac{\Delta c(\mathbf{p},\mathbf{q})}{\gamma_{col}}\right) exp\left(-\frac{\Delta g(\mathbf{p},\mathbf{q})}{\gamma_{pos}}\right)$$

$$\Delta c(\mathbf{p},\mathbf{q}) = \frac{1}{3}\|I(\mathbf{p}) - I(\mathbf{q})\|_1 = \frac{1}{3}\sum_{c\in\{r,g,b\}} |I_c(\mathbf{p}) - I_c(\mathbf{q})|$$

$$\Delta g(\mathbf{p},\mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_2 = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

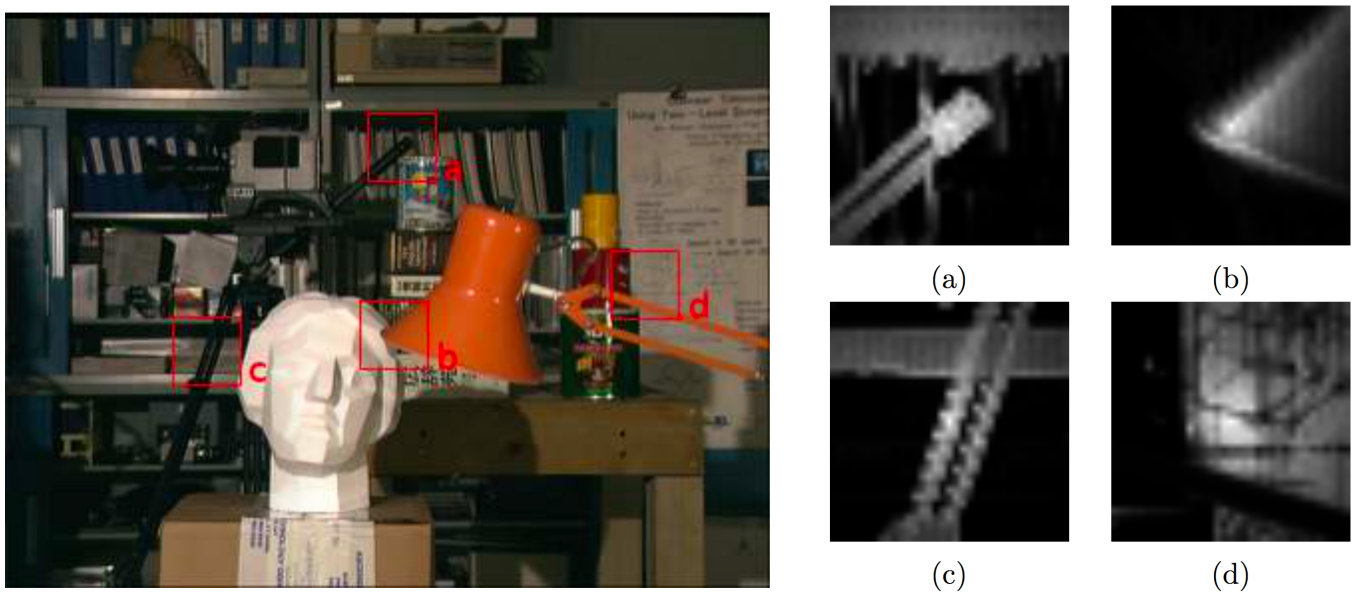


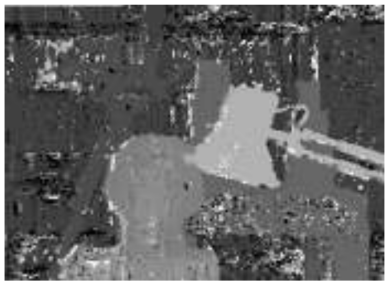
Image source:
L. F. Julià, P. Monasse. Bilaterally Weighted Patches for Disparity Map Computation. Image Processing On Line, 2015.

Stereo and depth computation

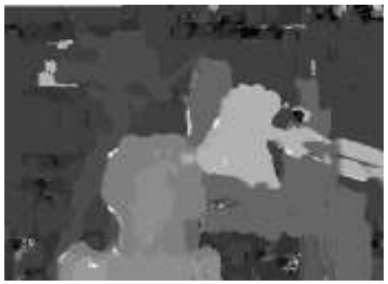
(a) Reference image.

(b) Target image.

(c) Ground-truth disparity map.



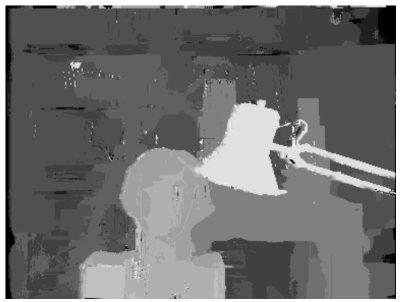
(a) 3×3



(b) 9×9

(c) 20×20

(d) 35×35



35×35 bilateral weights

Image source:
L. F. Julià, P. Monasse. Bilaterally Weighted Patches for Disparity Map Computation. Image Processing On Line, 2015.

Stereo and depth computation

Failures of correspondence search

- Textureless areas
- Occlusions
- Repetitions (self-similarity)
- Non-Lambertian surfaces, specularities

Image source: [T.V. Haavardsholm]

Stereo and depth computation

Failures of correspondence search

- Textureless areas
- Occlusions
- Repetitions (self-similarity)
- Non-Lambertian surfaces, specularities

Some solutions:

- Use more views:
  3rd view,
  multiple baseline stereo,
  multiview stereo
- Add regularity:
  global methods

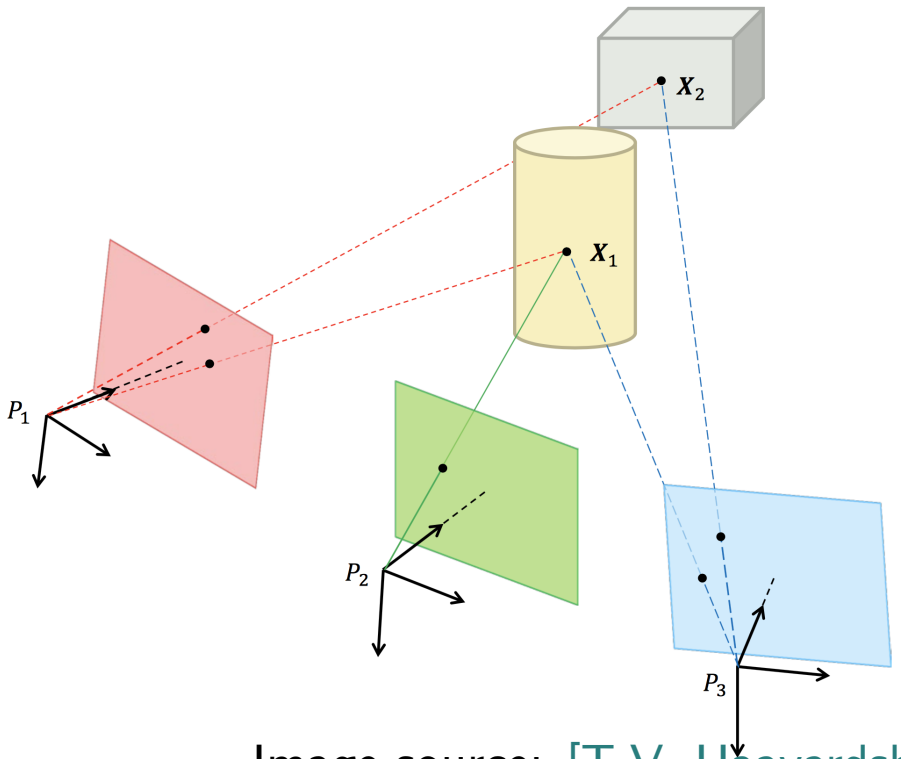
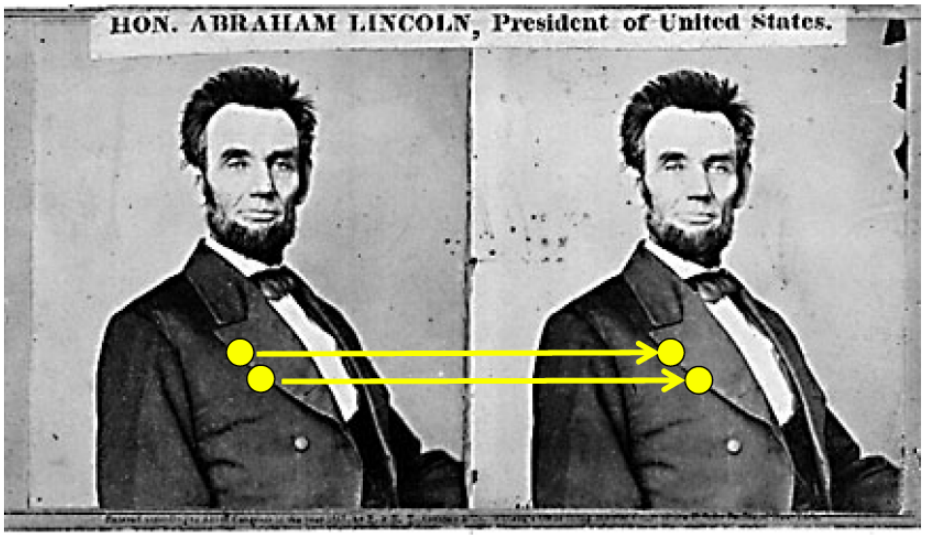


Image source: [T.V. Haavardsholm]

Stereo and depth computation

Global methods: based on energy minimization



$$\min_d E_{\text{data}}(I_1, I_2, d) + E_{\text{regularity}}(d)$$

- **Data term:** d finds the good match.
  E.g. $L2$ data term (others exist)

$$E_{\text{data}}(I_1, I_2, d) = \sum_{(x,y)} |I_1(x,y) - I_2(x+d,y)|^2$$

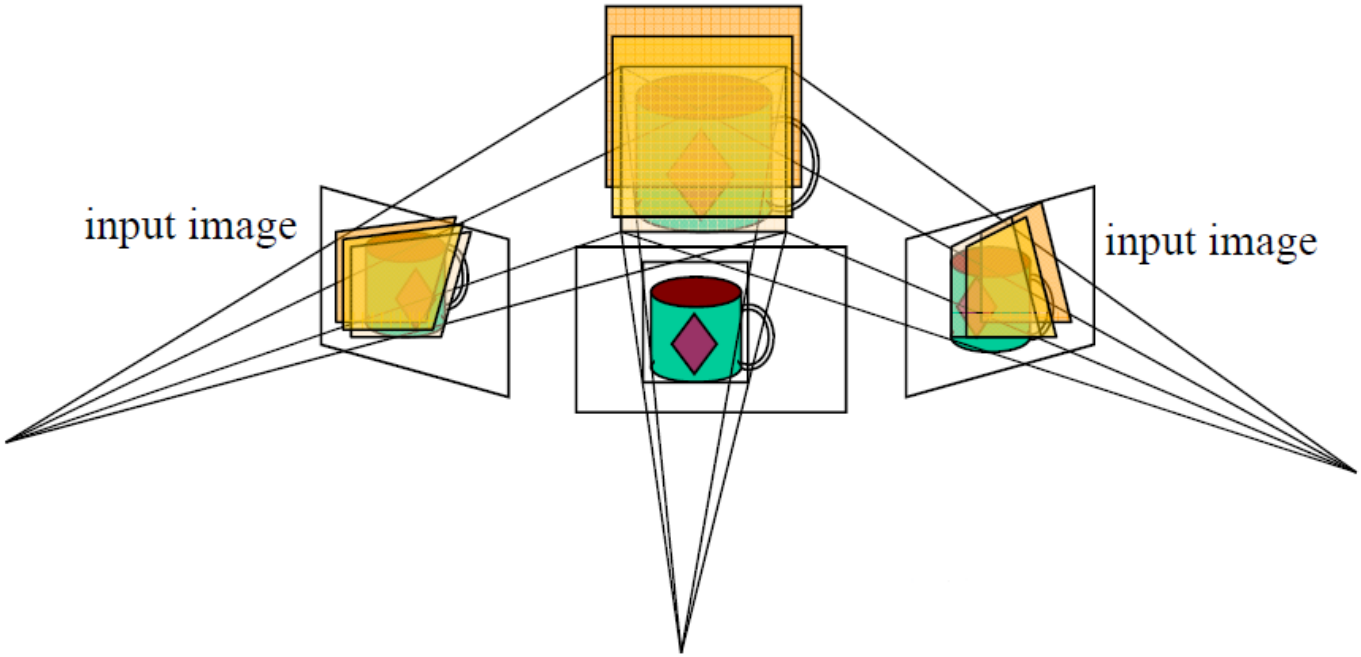
- **Regularity term:** close pixels should (usually) have similar d.

$$E_{\text{regularity}}(d) = \sum_{(x,y)} \sum_{\text{neighbor pixels}(x_n,y_n)} \rho(|d(x,y) - d(x_n,y_n)|)$$

Image source: [S. Seitz]

Stereo and depth computation

Plane sweep algorithm



Cameras in general position (not necessarily rectified).

BASIC IDEA: For every pixel in the first image, travel its viewing ray, project it onto the second image, and compare the image similarities.

STRATEGY: Travel the viewing ray of all pixels at once by projecting the image onto fronto parallel planes that move forward.

Image source: [S. Seitz]

Stereo and depth computation

Plane sweep algorithm

For each sampling depth:

- Compute the fronto-parallel plane at the corresponding depth
- Compute the image to image homography given a plane
- Warp the second image according to the homography
- Compute the matching score (SSD, NCC, ...)
- For each pixel keep the depth with best score

R. Collins. A Space-Sweep Approach to True Multi-Image Matching, CVPR 1996.

Stereo and depth computation

Plane sweep algorithm

Stereo and depth computation

Plane sweep algorithm

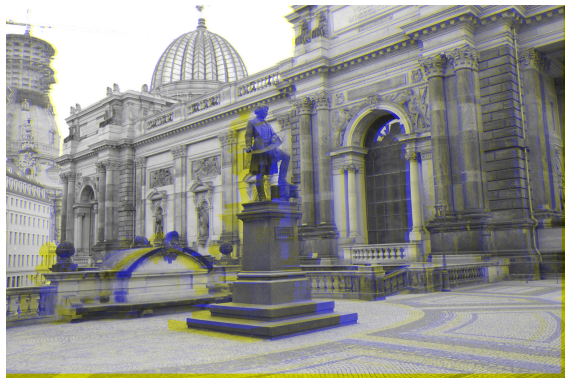
d_2

d_4

d_{12}

d_{20}

d_{54}



d_{104}

Stereo and depth computation

Plane sweep algorithm

Stereo and depth computation

Plane sweep algorithm: fronto-parallel planes

A plane that is parallel to the image plane at a certain depth.
All points of the plane are at the same depth (distance to the image plane).

OBSERVATION: The depth is the 3rd coordinate of the point when expressed in the camera frame $\mathbf{x} = K[I|0]\mathbf{X}_{\text{cam}}$
(assuming last row of K: $(0, 0, 1)$).

In the general case: $\mathbf{x} = P\mathbf{X} = K[R|t]\mathbf{X}$, where $\mathbf{X}_{\text{cam}} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \mathbf{X}$

Then, the depth of $\mathbf{x}$: $d(\mathbf{x}) = (P\mathbf{X})_3 = p_3^T \mathbf{X}$

Fronto-parallel plane: $\mathbf{\Pi} = p_3^T - (0, 0, 0, d)$
(plane equation: $\mathbf{\Pi}^T \mathbf{X} = 0$)

Stereo and depth computation

Plane sweep algorithm: image to image homography

Let $\mathbf{x}$ be a pixel of the first image in homogeneous coordinates.
We want to back-project the pixel into the 3D plane $\mathbf{\Pi}$ and reproject it to the second image.

We have:

$$\begin{pmatrix} P \\ \mathbf{\Pi}^T \end{pmatrix} \mathbf{X} = \begin{pmatrix} \mathbf{x} \\ 0 \end{pmatrix} \rightarrow \mathbf{X} = \begin{pmatrix} P \\ \mathbf{\Pi}^T \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{x} \\ 0 \end{pmatrix} = A \begin{pmatrix} \mathbf{x} \\ 0 \end{pmatrix}$$

Project $\mathbf{X}$ to the second image:

$$\mathbf{x}' = P' A \begin{pmatrix} \mathbf{x} \\ 0 \end{pmatrix} = P' \bar{A} \mathbf{x} = H \mathbf{x}$$

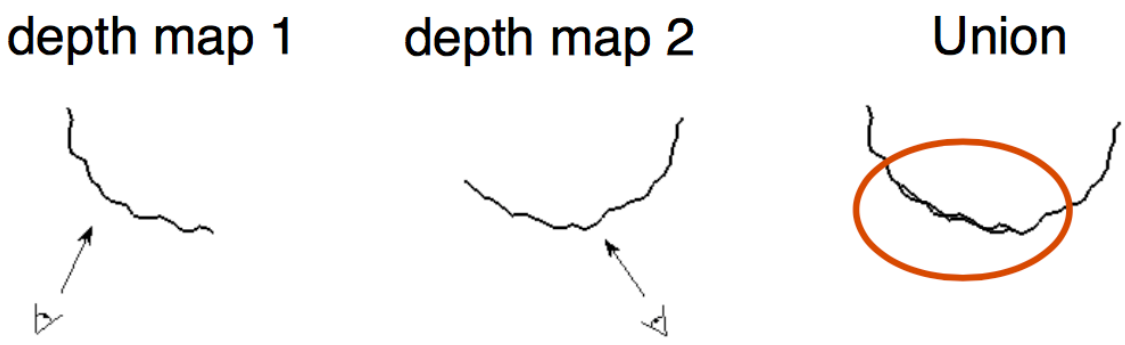
where $\bar{A}$ is the 4×3 submatrix of A.

We get $\boxed{H = P' \bar{A}}$ where $\bar{A}$ depends on P and d

Depth map fusion

3D reconstruction

A depth map allows to create a 3D point cloud

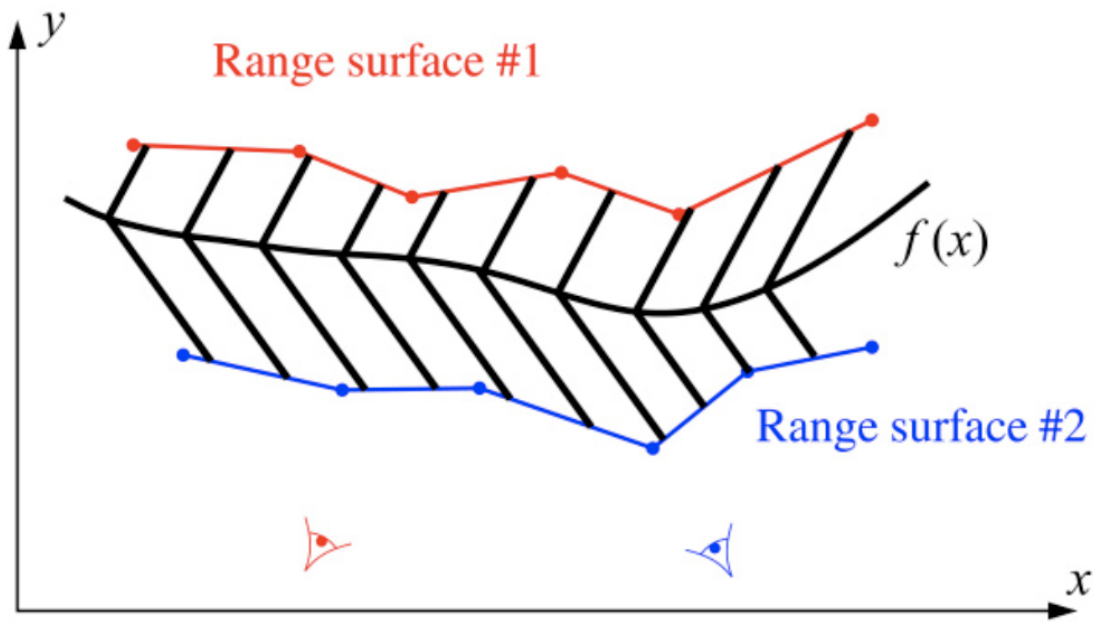


Naïve combination produces artifacts.

Image source: [S. Seitz]

Depth map fusion

Better solution: find average surface



e.g. surface that minimizes sum of distances to the given surfaces

Image source: [S. Seitz]

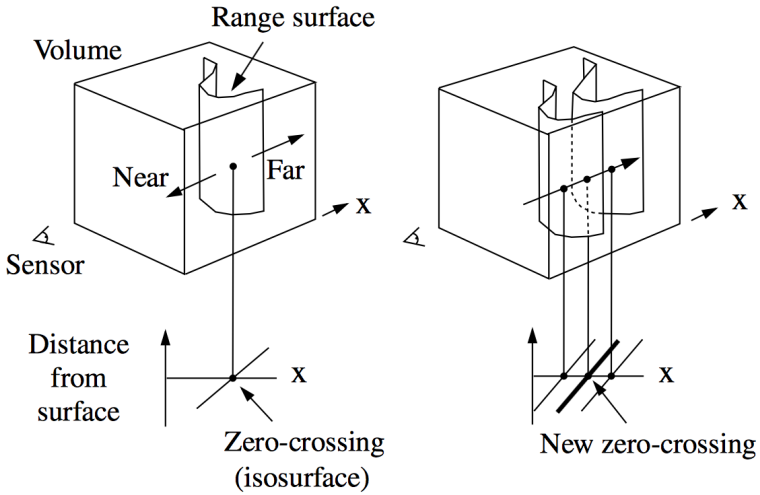
Depth map fusion

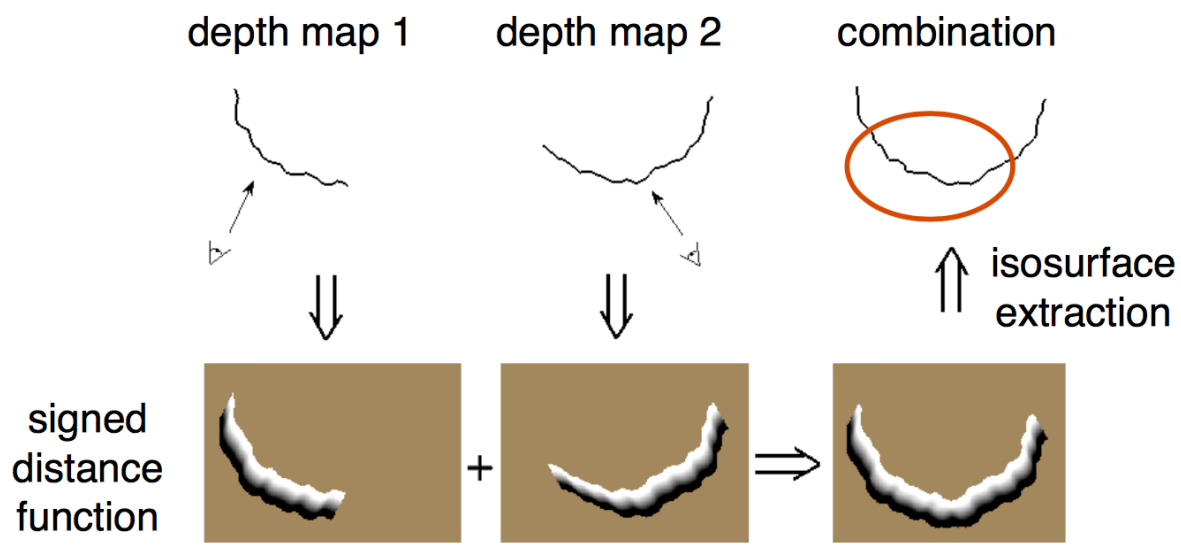
PRACTICAL SOLUTION:

1. Create Signed Distance Function (SDF) for every depth map d_i

$$sdf_i(\mathbf{z} = [\mathbf{X}]) = (P_i\mathbf{X})_3 - d_i([P_i\mathbf{X}])$$

2. Average different signed distance functions





Isosurface extraction: W.E. Lorensen, H.E. Chile. Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH, 1987.
Fusion method:B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. In Proc. SIGGRAPH, 1996.

Image source: [S. Seitz]

Depth map fusion

Weights w_i may be used in the average

$$D(\mathbf{z}) = \frac{\sum_i w_i(\mathbf{z}) sdf_i(\mathbf{z})}{\sum_i w_i(\mathbf{z})}$$

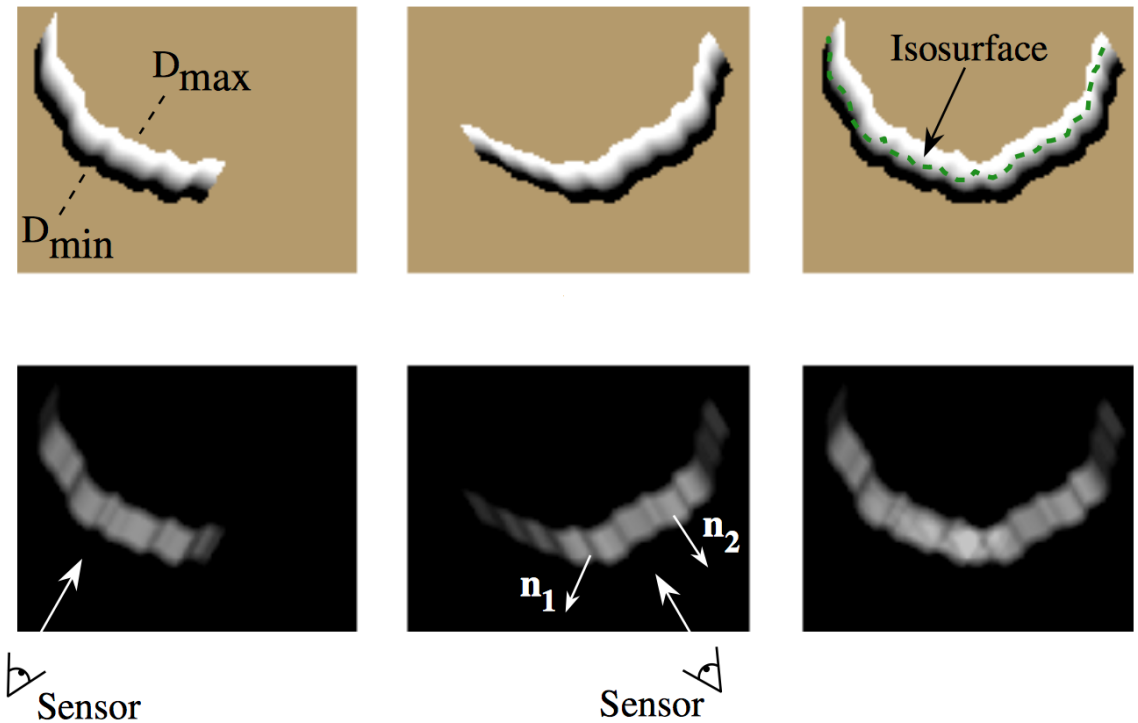


Image source and proposed method:
B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. In Proc. SIGGRAPH, 1996.

Depth map fusion

3D reconstruction

1. Depth map estimation: plane sweep approach with NCC
2. Depth map fusion: Curless and Levoy method

input image

317 images (hemisphere)

ground truth model

Image source and proposed method:
M. Goesele, B. Curless, S. Seitz. Multi-view stereo revisited. In Computer Vision and Pattern Recognition, 2006.

Depth map fusion

Other approaches:

Variational methods: *TV-*L_1

C. Zach, T. Pock, H. Bischof. A globally optimal algorithm for robust TV-L1 range image integration. In International Conference on Computer Vision, 2007.

Deep learning methods: OctNetFusion

G. Riegler, A. O. Ulusoy, H. Bischof, A. Geiger. OctNetFusion: Learning depth fusion from data. In International Conference on 3D Vision, 2017.

`https://github.com/griegler/octnetfusion`

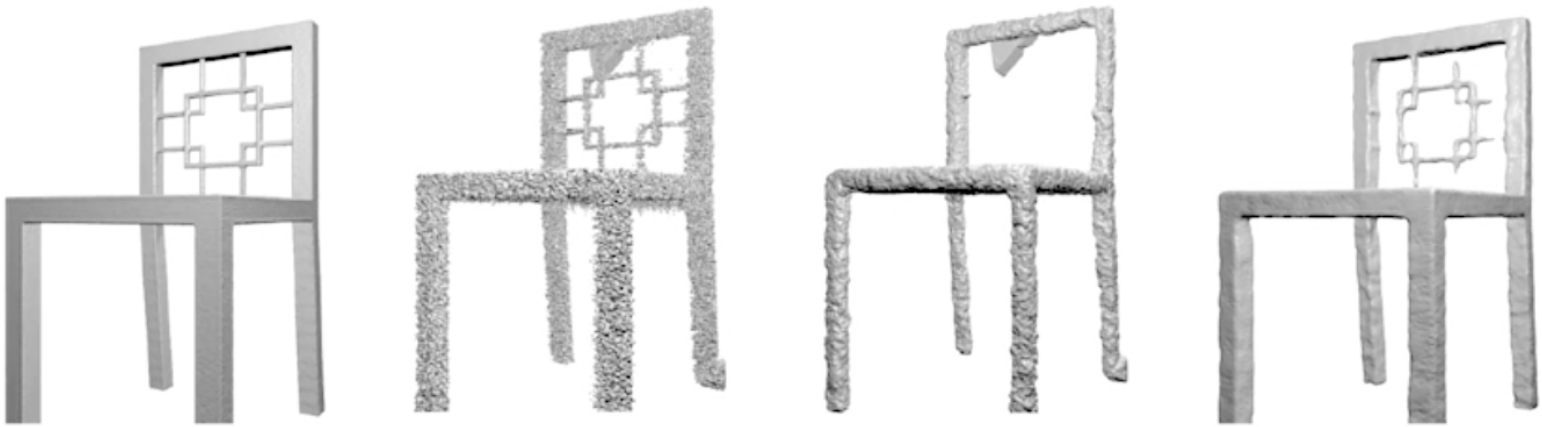


Image source: [A. Geiger]

Depth map fusion

Deep learning methods: Neural Depth Fusion

S. Weder, J. Schonberger, M. Pollefeys, M.R. Oswald. NeuralFusion: Online Depth Fusion in Latent Space. In Int. Conf. on Computer Vision and Pattern Recognition, 2021.
`https://www.silvanweder.com/publications/neural-fusion/`

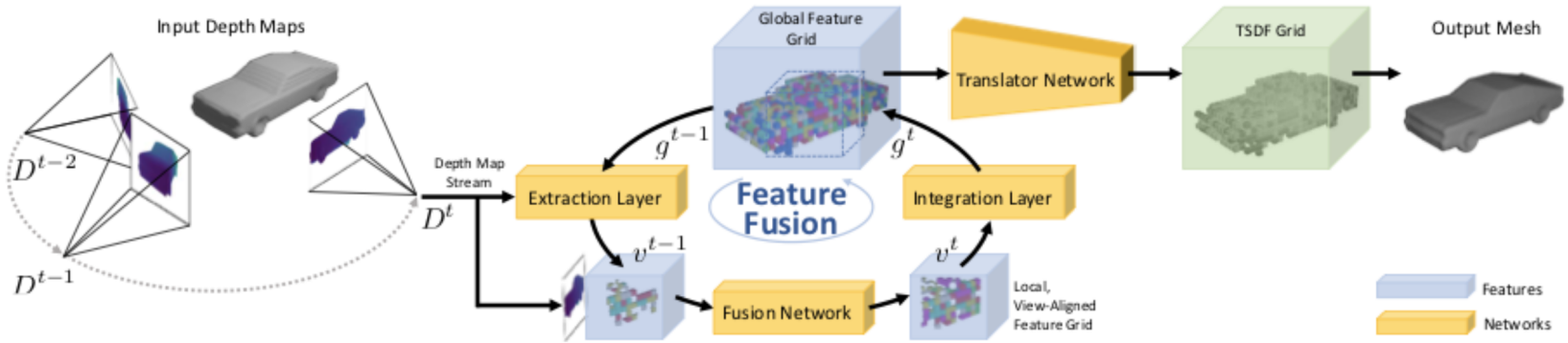


Image source: [Weder et al 2021]

Outline

- Triangulation methods
- Stereo and depth computation
- **New view synthesis**

New view synthesis

GOAL: Synthetize a new view from a set of available images taken from different angles.

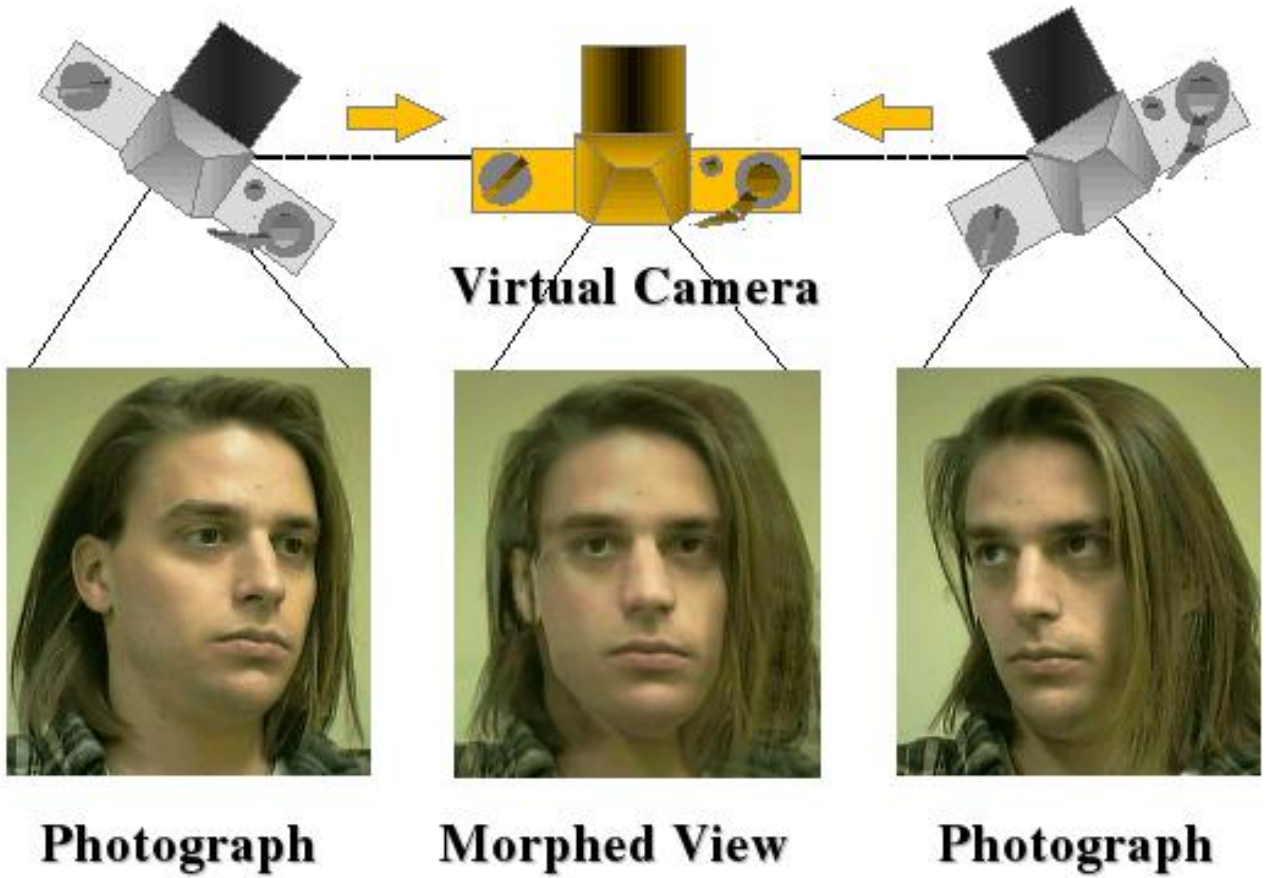


Image source: [Seitz and Dyer 1996]

New view synthesis

Classification of methods:

- **MBR: Model-Based Rendering**
  A 3D reconstruction of the scene is performed.
  ☺ Easily solves the occlusion problem
  ☹ Needs (several) accurately calibrated views (computationally expensive)
  ☹ Due to errors in the 3D model, the texture mapping may create blur in the generated view.
- **IBR: Image-Based Rendering**
  ☺ No need of an explicit 3D model, based on a proper warping of the existing views. The epipolar geometry has to be known.
  ☹ Fails at occlusions.
- **Hybrid models**
  Combination of both previous methods. Use IBR basically and the 3D model when ambigous situations arise (e.g. occlusions).

New view synthesis

Image-Based Rendering

Two different approaches:

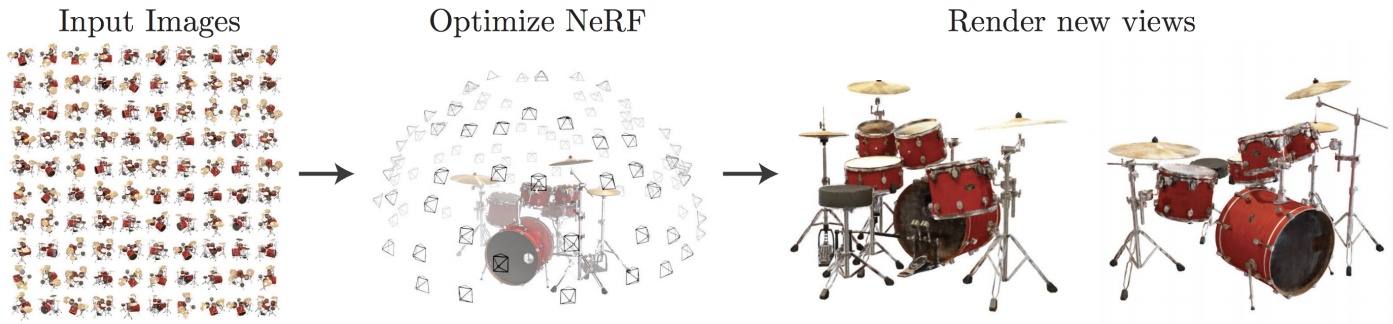
- **Interpolation based on a dense set of samples**
  Samples of the plenoptic function $F(\phi, \theta, x, y, z)$ are acquired from different images and stored in a table. Then, to synthetise a new view the values of the table are interpolated.
  Fast method BUT needs many views as well as the knowledge of camera position and principal direction.

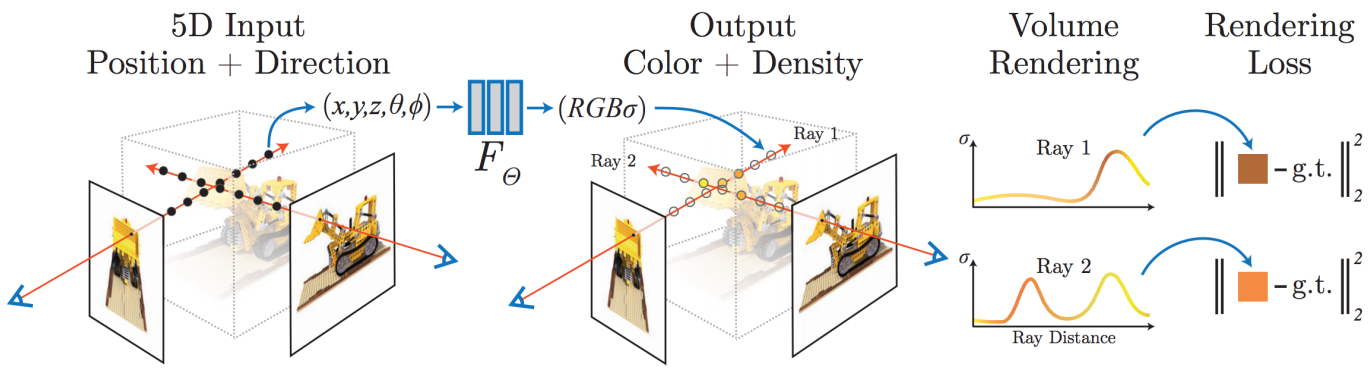
  M. Levoy, and P. Hanrahan, Light field rendering, Proc. ACM SIGGRAPH 1996.
  S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, The lumigraph, Proc. ACM SIGGRAPH 1996.

- **Interpolation based on projective reconstruction and morphing**
  Based on the geometrical relationship between images (fundamental matrix or trifocal tensor) and morphing techniques.
  They can be applied with a few images.

New view synthesis - NeRF

Interpolation based on radiance field
NeRF: Neural Radiance Field

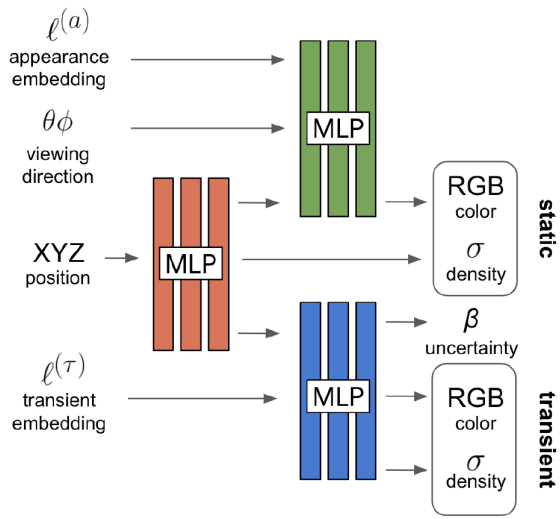




https://www.matthewtancik.com/nerf

B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T Barron, R. Ramamoorthi, R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. ECCV 2020

New view synthesis - NeRF in the Wild



(a) Photos (b) Renderings

https://nerf-w.github.io/

R. Martin-Brualla, N. Radwan, M.S.M. Sajjadi, J. T. Barron, A. Dosovitskiy, D. Duckworth. NeRF in the Wild. CVPR, 2020

New view synthesis - View morphing

Interpolation based on projective reconstruction and morphing

[Seitz and Dyer 1996] S. Seitz, and C. Dyer, View morphing, Proc. ACM SIGGRAPH 1996.

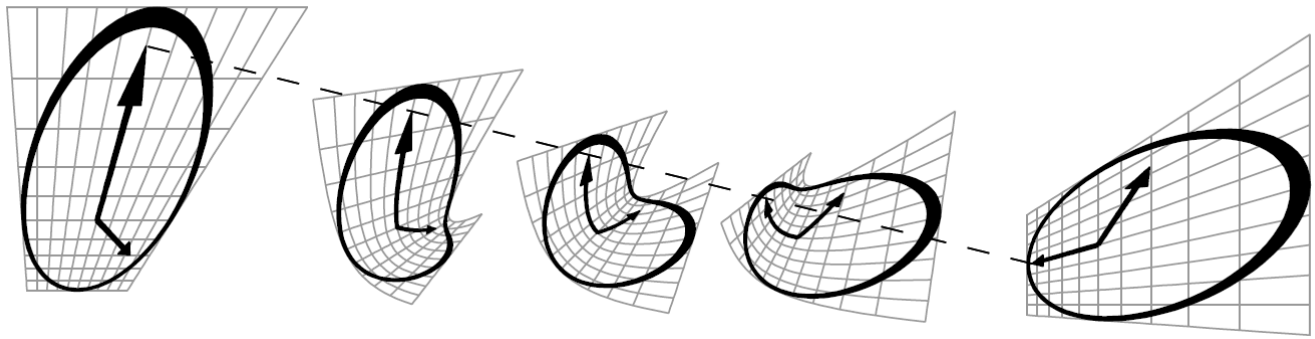


image morphing vs view morphing

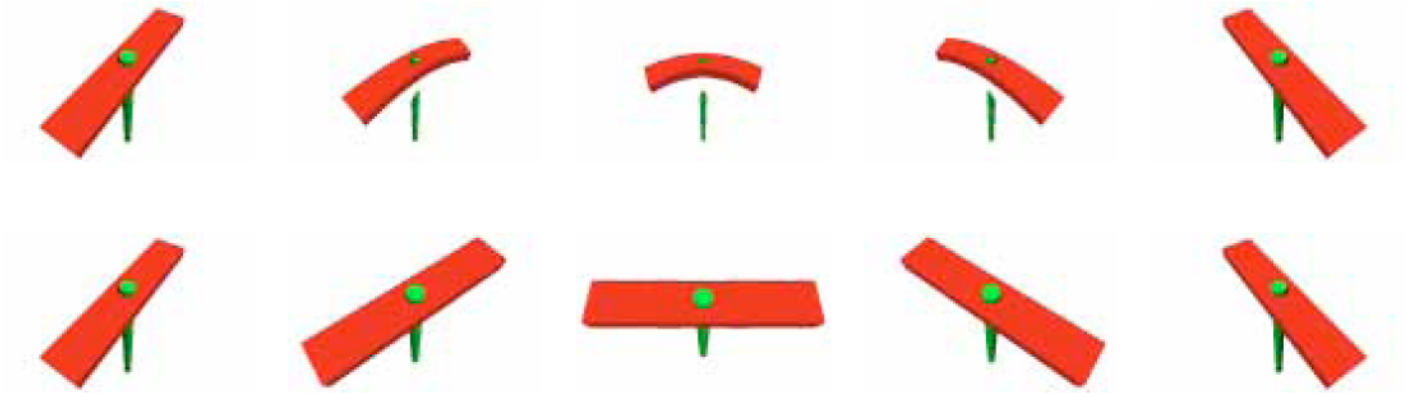


Image source: [Seitz and Dyer 1996]

New view synthesis - View morphing

Interpolation based on projective reconstruction and morphing
[Seitz and Dyer 1996]

1. Prewarp
   - *align views*

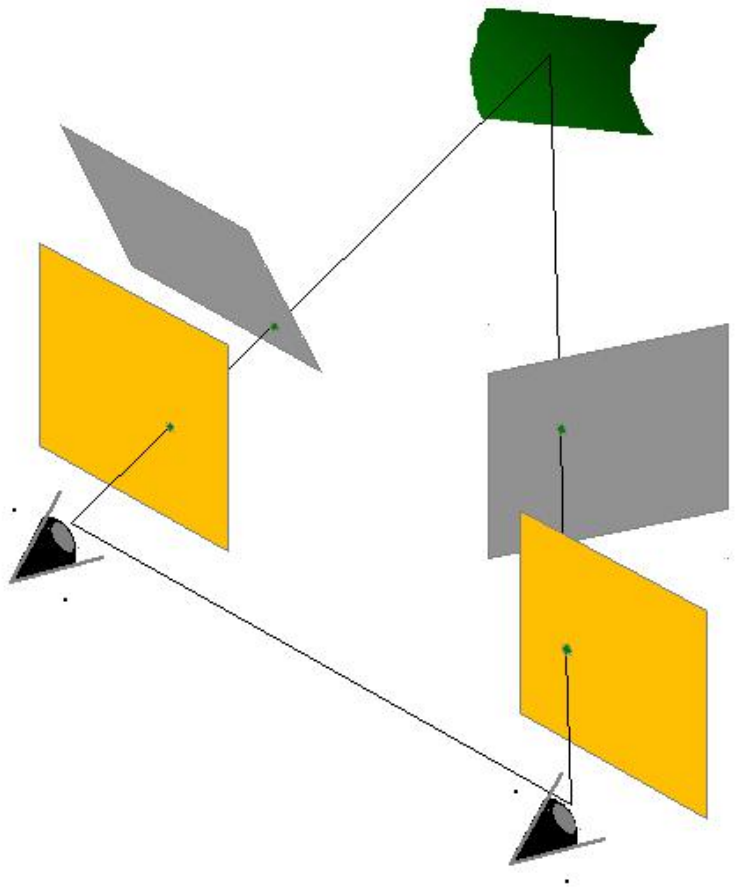


Image source: [Seitz and Dyer 1996]

New view synthesis - View morphing

Interpolation based on projective reconstruction and morphing
View morphing [Seitz and Dyer 1996]

1. Prewarp
   ⇨ *align views*

2. Morph
   ➡ *move camera*

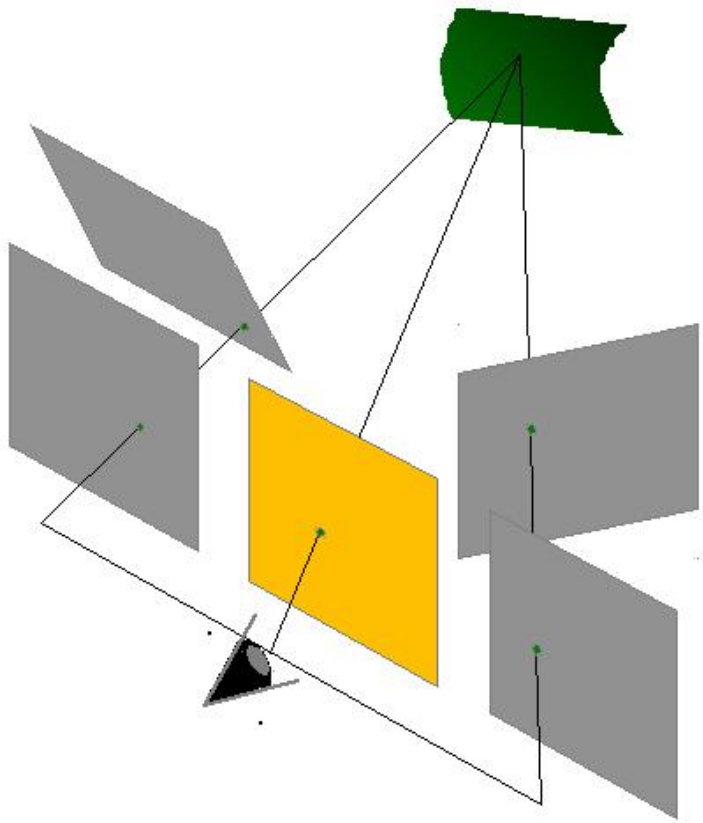


Image source: [Seitz and Dyer 1996]

New view synthesis - View morphing

Interpolation based on projective reconstruction and morphing
[Seitz and Dyer 1996]

1. Prewarp
   ⇨ *align views*
2. Morph
   ⇨ *move camera*
3. Postwarp
   ⇨ *point camera*

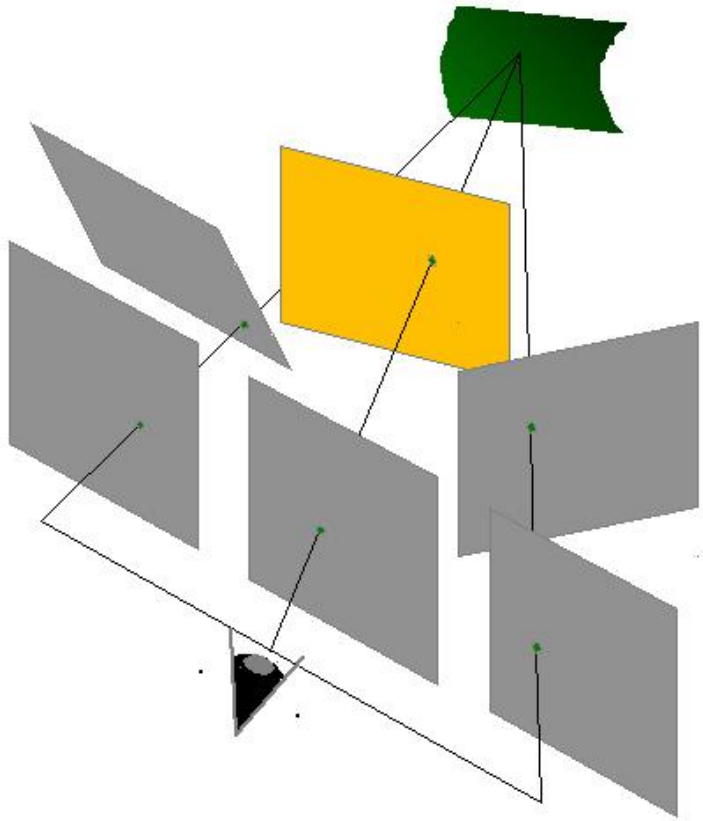


Image source: [Seitz and Dyer 1996]

New view synthesis - View morphing

Special case of two parallel views:
The camera moves parallel to the image plane

$$P_0 = \begin{pmatrix} f_0 & 0 & 0 & 0 \\ 0 & f_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, P_1 = \begin{pmatrix} f_1 & 0 & 0 & -f_1 C_X \\ 0 & f_1 & 0 & -f_1 C_Y \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$\mathbf{X}$ projects to $\mathbf{x}_0$ and $\mathbf{x}_1$ in images I_0 and I_1 respectively.
We have:

$$p = (1-s)\mathbf{x}_0 + s\mathbf{x}_1 = (1-s)\frac{1}{Z}P_0\mathbf{X} + s\frac{1}{Z}P_1\mathbf{X}$$
$$= \frac{1}{Z}P_s\mathbf{X}$$

where $P_s = (1-s)P_0 + sP_1$

$$I(p) = (1-s)I_0(\mathbf{x}_0) + sI_1(\mathbf{x}_1)$$

$$C_s = (sC_x, sC_Y, 0)$$

$$f = (1-s)f_0 + sf_1$$

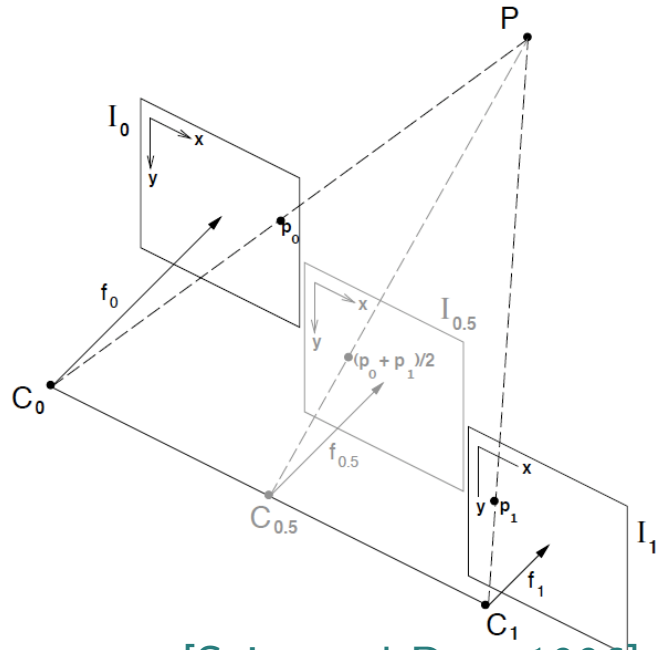


Image source: [Seitz and Dyer 1996]

New view synthesis - View morphing

Interpolation based on projective reconstruction and morphing

Non-parallel views

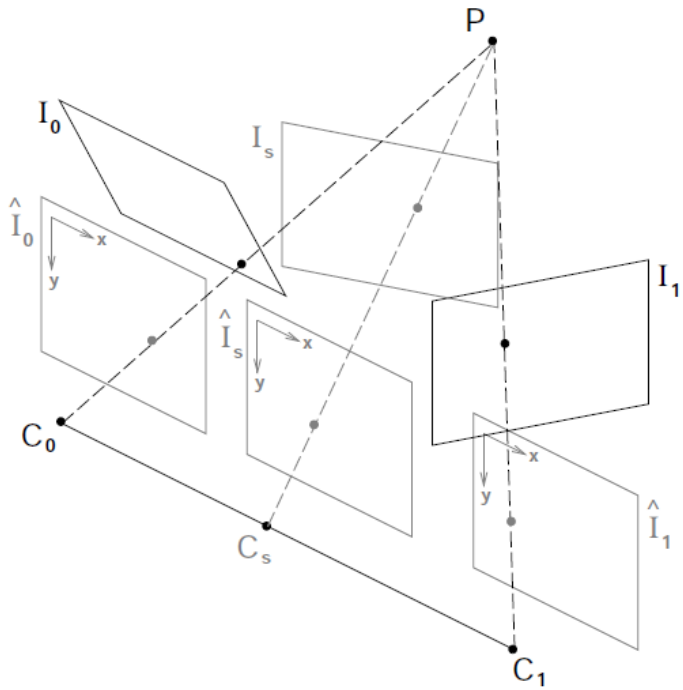
Property: Any two views that share the optical center are related by a planar projective transformation (homography).
Let I and $\hat{I}$ be two images with projection matrices $P = [H| - HC]$ and $\hat{P} = [\hat{H}| - \hat{H}C]$. The projections $\mathbf{x}$ and $\hat{\mathbf{x}}$ of any scene point $\mathbf{X}$ are related by:

$$\hat{H}H^{-1}\mathbf{x} = \hat{H}H^{-1}P\mathbf{X} = \hat{P}\mathbf{X} = \hat{\mathbf{x}}$$

New view synthesis - View morphing

Interpolation based on projective reconstruction and morphing
[Seitz and Dyer 1996] algorithm steps:

- **Prewarp:** obtain images $\hat{I}_0$ and $\hat{I}_1$ by applying H_0^{-1} and H_1^{-1} to I_0 and I_1 respectively.
- **Morph:** form $\hat{I}_s$ by linear interpolation of position and colors of corresponding points in $\hat{I}_0$ and $\hat{I}_1$.
- **Postwarp:** obtain I_s by applying H_s to $\hat{I}_s$.



We choose: $\hat{P}_0 = [I| - C_0]$ and $\hat{P}_1 = [I| - C_1]$.
We know: $P_0 = [H_0| - H_0 C_0]$ and $P_1 = [H_1| - H_1 C_1]$.
$P_s = [H_s| - H_s C_s]$, where $C_s = (1 - s)C_0 + sc_1$

Image source: [Seitz and Dyer 1996]

New view synthesis - View morphing

Interpolation based on projective reconstruction and morphing
If the camera matrices are not known, then use image rectification.

Algorithm steps:

- Image rectification
- Computation of dense correspondences
- Linear interpolation of corresponding points
- Post-warping of the interpolated view

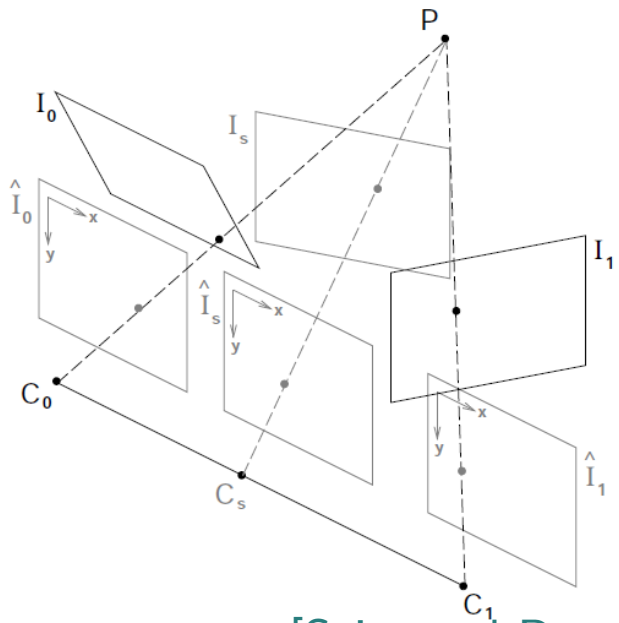


Image source: [Seitz and Dyer 1996]

New view synthesis - View morphing

Interpolation based on projective reconstruction and morphing
[Seitz and Dyer 1996]

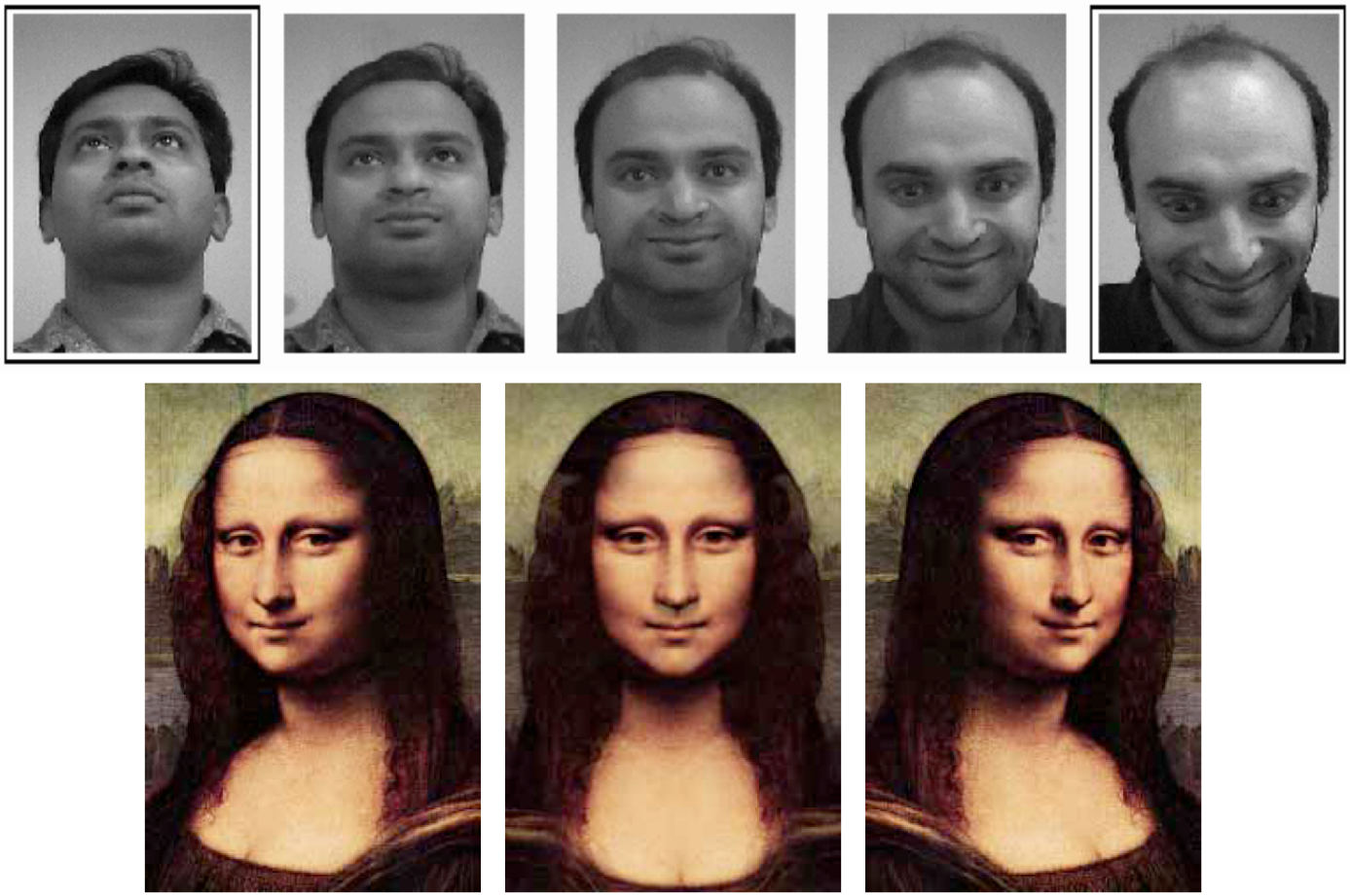
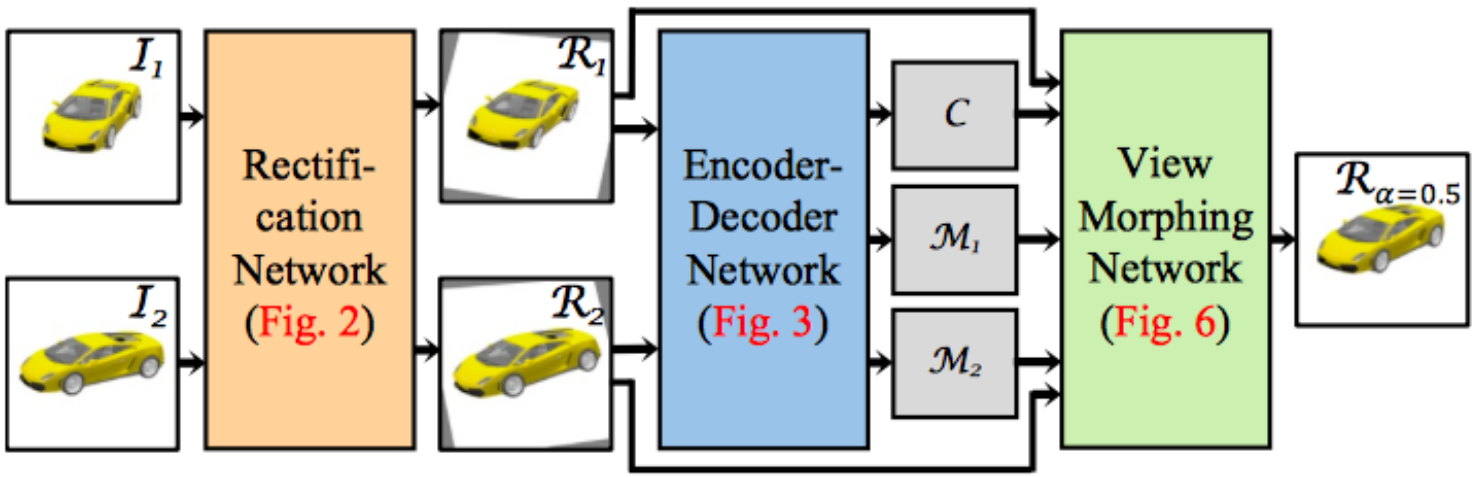


Image source: [Seitz and Dyer 1996]

New view synthesis

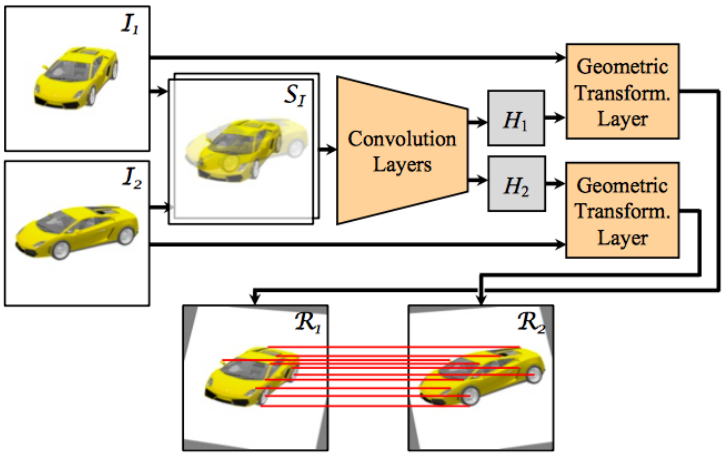
IBR with Deep Learning: Deep View Morphing



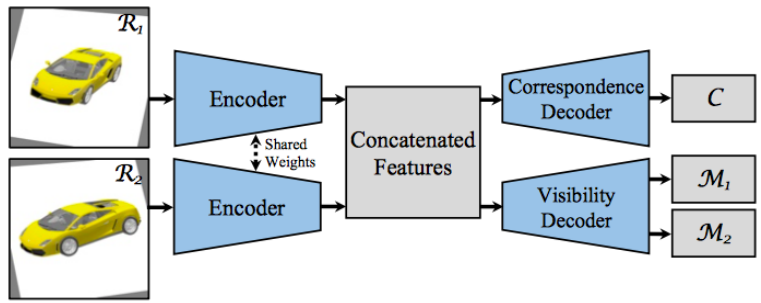
D. Ji, J. Kwon, M. McFarland, and S. Savarese, Deep View Morphing, Proc. CVPR 2017.

New view synthesis

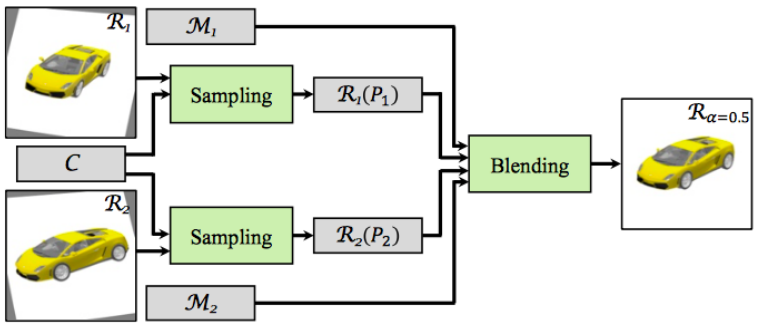
IBR with Deep Learning: Deep View Morphing



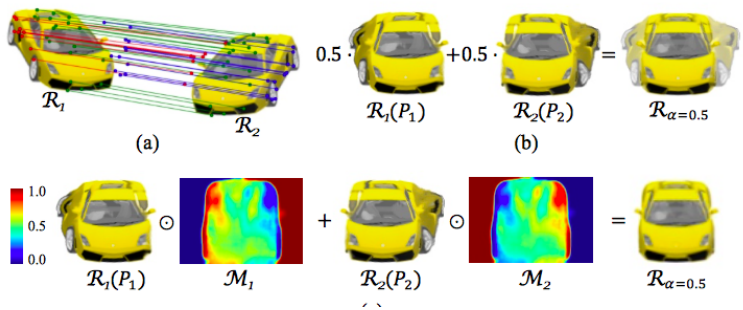
Rectificaction network



Encoder-Decoder network



View morphing (blending) network

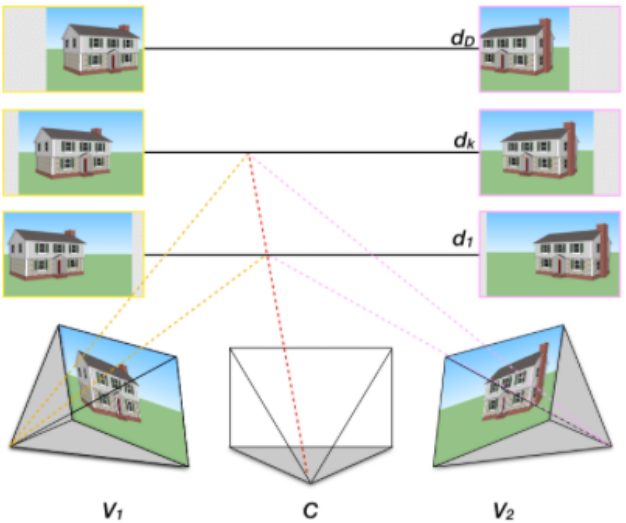


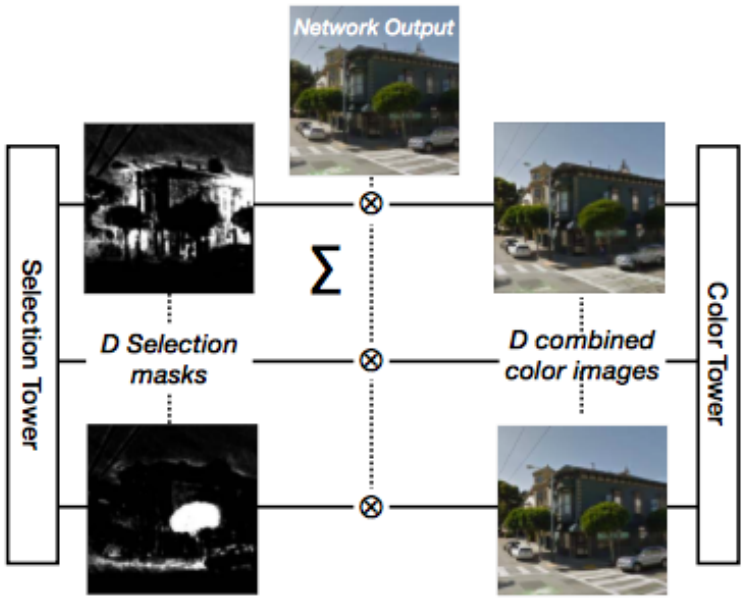
Blending procedure

D. Ji, J. Kwon, M. McFarland, and S. Savarese, Deep View Morphing, Proc. CVPR 2017.

New view synthesis

Image-Based Rendering with Deep Learning: DeepStereo

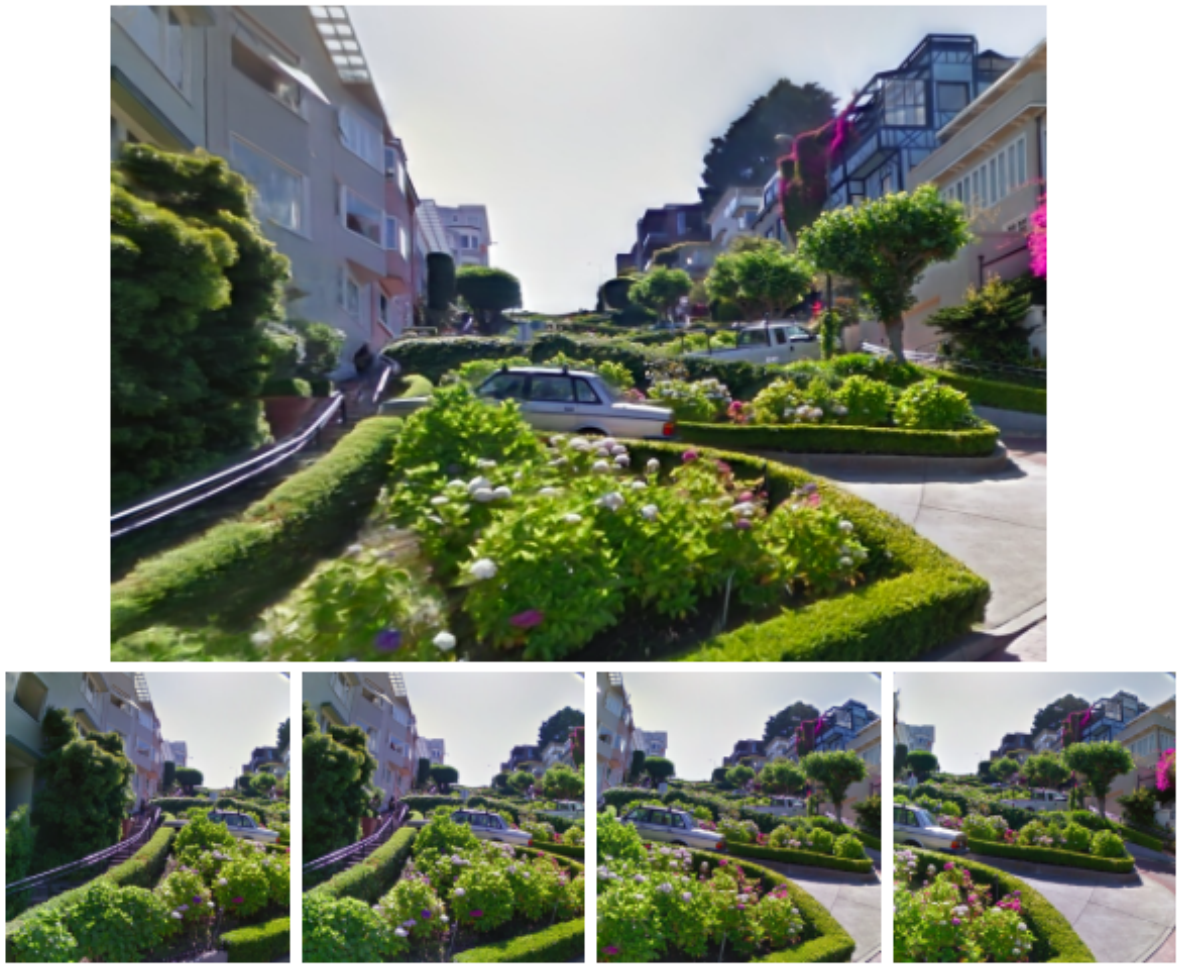




J. Flynn, I. Neulander, J. Philbin, N. Snavely, DeepStereo: Learning to Predict New Views from the World's Imagery, Proc. CVPR 2016.

New view synthesis

Image-Based Rendering with Deep Learning: DeepStereo



J. Flynn, I. Neulander, J. Philbin, N. Snavely, DeepStereo: Learning to Predict New Views from the World's Imagery, Proc. CVPR 2016.

Summary

We have seen:

- Triangulation methods
- Stereo and depth computation
- New view synthesis

Next class:

- Multi-view stereo
- Structure from motion