Stereovision

Włodzimierz Kasprzak and Artur Wilkowski

5 November 2021

Outline of the stereovision problem

- 3D reconstruction from camera images is one of the main problems of computer vision
- Reconstruction from a single view is (in general) not possible . . .
- . . . but we use two or more views - we can triangulate points to obtain **depth**
- **Stereovision** is a set of methods for reconstruction of 3D scene structure from **two images**
- **Stereovision** can be generalized to handle **multiple images**.

Homographies

Image of a planar object in a perspective projection

Let us consider a situation where all points projected into image are located on a plane. In a selected coordinate system situated this plane, each point will then have coordinates $(X, Y, 0)^T$. Then the perspective projection equation takes form

$$k \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K_{int} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}$$

And since $Z = 0$ the equation simplifies to:

Image of a planar object in a perspective projection

$$k \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K_{int} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

Therefore planar points (X, Y) and their projections (x, y) are related by a **homography** transformation H.

$$k \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{22} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = H \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

where

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} = K_{int} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} = K_{int} H_{Rt}$$

Homography notes

- Note 1: perspective transformation from 3D to 2D now has a form of 2D invertible transform
- Note 2: A homography is defined up to a scale
- Note 3: If we know 3D-2D homography matrix H (and K_{int}), we can retrieve R and $\mathbf{t}$ directly using:

$$\begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix} = K_{int}^{-1} H$$

In an ideal (noiseless) case we obtain two column vectors of R and a full translation (however, with sign ambiguity)

Planar object in two views

Let assume that the planar object is observed from two views (or two cameras). We then have

$$kp_1 = K_{int1} H_{Rt1} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \text{ for the first view and}$$

$$lp_2 = K_{int2} H_{Rt2} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \text{ for the second.}$$

By eliminating $(X, Y, 1)$ we have

$$sp_2 = K_{int2} H_{Rt2} H_{Rt1}^{-1} K_{int1}^{-1} p_1$$

and therefore

$$sp_2 = K_{int2} H_{1\text{-}2} K_{int1}^{-1} p_1, \text{ where } H_{1\text{-}2} = H_{Rt2} H_{Rt1}^{-1}$$

It turns out, that images of points from the planar surface in two views are also related by a homography!

Planar object in two views

If we assume that projection matrices for both cameras are $A_1 = K_{int1}[I|0]$ and $A_2 = K_{int2}[R|\mathbf{t}]$, then the homography discussed can be related to extrinsic parameters R and $\mathbf{t}$, by

$$H_{1\text{-}2} = R - \mathbf{t}\mathbf{n}^T/d$$

where the normal vector $\mathbf{n}$ and distance to origin d are plane parameters:

$$\mathbf{n}^T\mathbf{P} + d = 0$$

From the estimated homography matrix, we can do a reverse - obtain R and $\mathbf{t}$ (as well as $\mathbf{n}$ and d) from $H_{1\text{-}2}$: (see: *Malis,Vargas: Deeper understanding of the homography decomposition for vision-based control*).

Remember: that $\mathbf{t}$ and d are recovered only up to their common scale!

Images from rotated cameras

A single camera observes an arbitrary 3D scene (planar or non-planar). Let us assume that its projection matrix is

$$A_1 = K_{int}[I|0]$$

Then we rotate the camera around its optical center and obtain

$$A_2 = K_{int}[R|0]$$

For the first camera configuration we have projection equation

$$kp_1 = K_{int}[I|0] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = K_{int} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Images from rotated cameras

For the second camera configuration we have

$$lp_2 = K_{int}(R|0)\begin{pmatrix}X\\Y\\Z\\1\end{pmatrix} = K_{int}R\begin{pmatrix}X\\Y\\Z\end{pmatrix}$$

Eliminating point coordinates $(X, Y, Z)^T$ we obtain

$$sp_2 = K_{int}RK_{int}^{-1}p_1$$

Hence points in both views are (again!) related by a homography $H_{R1\text{-}2} = K_{int}RK_{int}^{-1}$!

- Note 1: We can use homographies also for non-planar scenes whenever only camera rotation is involved.
- Note 2: Homography can be easily (and uniquely) obtained from intrinsic matrix K_{int} and rotation R and vice-versa.

Estimation of a homography matrix

- homography can be (and often is) estimated from point correspondences (either image-image or image-physical planar object)
- homography matrix has **9** paramters but only **8** degrees of freedom
- it is enough to use 4 point matchings (for each match we obtain 2 equations)
- solution is based on DLT transform (similar to camera calibration)
  - Solve E.1
- the system of equations is **homogenous** - so use appropriate methods!
- the linear system may require normalization for better numerical conditioning (see: *Hartley: In defense of an 8-point algorithm*)

Homography summary

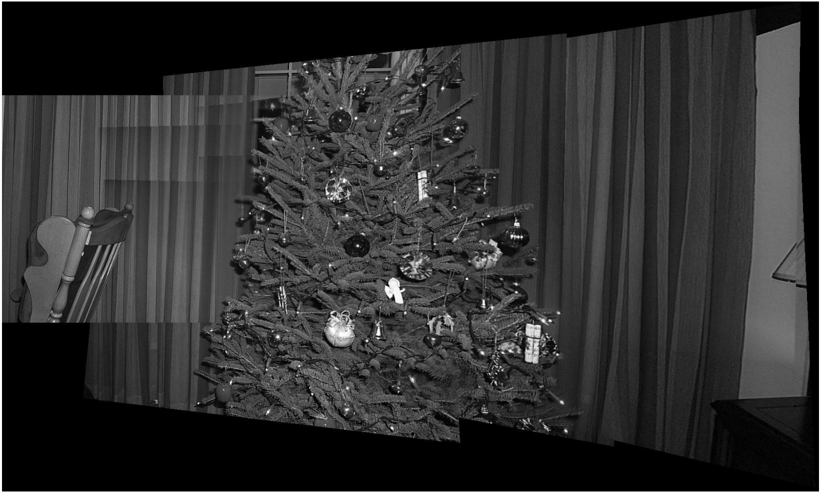
The following relations can be described using homography:

- relation between planar 3D points and their 2D images
- relation between 2D images of planar points in two camera views
- relation between arbitrary 3D points in two views, when only rotation is involved

Applications:

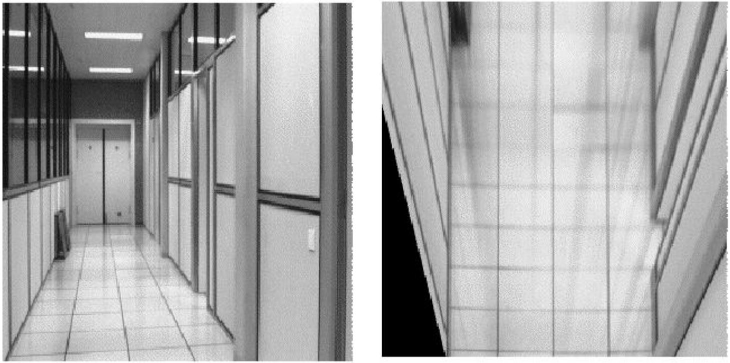
- calibration (e.g. Zhang algorithm)
- robust estimation of camera motion (another lecture...)
- image mosaicing, perspective distortion removal, stabilization, pattern augmentation

Homography applications



Source: Gerhard Roth

Homography applications



Source: Gerhard Roth

Stereovision

Stereovision problem

- Inverse transformation to perspective mapping is not unique, i.e. in general a single image point corresponds to multiple 3D points
- Depth information (3D point coordinates) can be obtained by observing a single point in images from several (at least 2) shifted cameras - the principle is identical to binocular vision of humans of animal predators
- A typical configuration of cameras are two cameras (hence stereo-) that are shifted horizontally
- By comparing positions of corresponding pixels in two images disparities, we can obtain information regarding image depth
- Image depth is inversely proportional to a measured disparity.

Basic tasks: Having given **two views** of a given scene reconstruct 3D structure of the scene

Generalized scene reconstruction problem

Stereovision is a specific case of a generalized scene reconstruction problem

- Our input data: coordinates of 2D points in particular views of the 3D scene p_{ij} where p_{ij} is a projection of 3D point P_j into a view i
- We are looking for: coordinates of 3D points P_j, extrinsic camera parameters for particular views T_i and camera's intrinsic parameters K_i
- Optimal set of parameters minimizes a *reprojection error* given as:

$$rErr = \sum_{(i,j)} ||proj(P_j, T_i, K_i) - p_{ij}||^2$$

where $proj$ is a perspective projection into a view i

Generalized scene reconstruction problem

- For any two views the scene can be reconstructed only up to an unknown projective transformation
- For larger number of views the scale of a scene is ambiguous

Conclusions:

- Generalized scene reconstruction problem is considered mainly in the case of reconstructing 3D scene basing on random views of the scene (e.g. from photographs obtained from the Internet)
- In a controlled environment it is more convenient to divide the problem into simpler sub-problems and solve them independently.

Solution to the stereovision problem

Two sub-problems to solve:

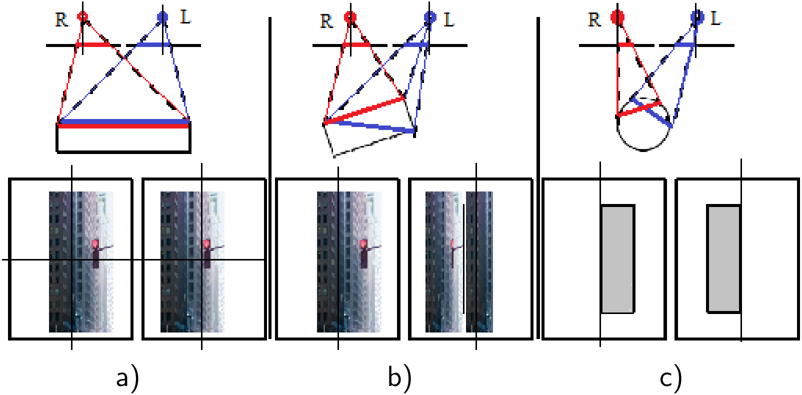
1. Calibration of the stereopair and normalization of stereo-images
2. Reconstruction of 3D scene

A common procedure is as follows

1. Independent calibration of to cameras and common optimization of parameters of both cameras
2. Normalization (rectification) of a pair of stereo-images for easier matching features in both images
3. Finding correspondences between image pixels (features) - extraction of a disparity map
4. Conversion of the disparity map into a depth map
5. Reconstruction of a 3D scene, e.g. by computation of global 3D coordinates of scene points

Solution to the stereovision problem

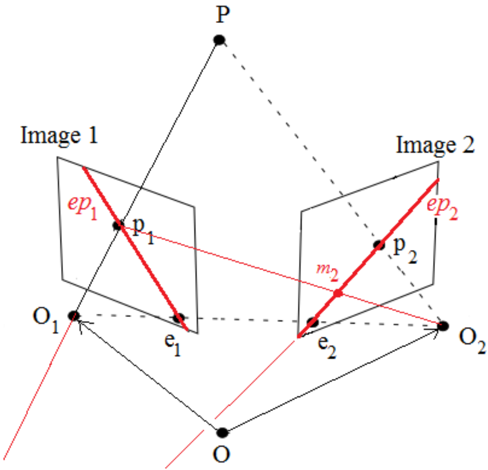
Typical stereovision procedure



a) optimal case b) different angle of observation in both cameras (harder matching) c) occlusions

Epipolar geometry

Epipolar geometry



Epipolar lines ep_1, ep_2

Epipolar geometry

- It can be verified, that that perspective projection of any 3D point is always situated on a plane specified by this point and optical centers of both cameras. Intersection of the plane with each image plane forms a line
- Thus, the set of points in the first image that can correspond to the point in the second image forms a line - called epipolar line
- Thus, in order to find a point in the first image corresponding to the point given in the second image we search only along the epipolar line, which greatly simplifies the procedure
- This observation form a basis for the epipolar geometry
- If we have a pair of normalized images
  - Epipolar lines are horizontal
  - Corresponding epipolar lines have the same y coordinate

Epipolar geometry

Let us cosider a situation from the last figure. We are given 3D point P and optical centers of cameras O_1 and O_2.

- Epipolar lines are defined as intersections of this plane and image planes.
- A line $O_1 O_2$ intersects image planes at points e_1 and e_2, these points are called epipoles.
- A line $O_1 O_2$ is called a baseline of a camera pair.

It can be observed that

- The line ep_1 is defined by points p_1 (projection of P w in the left image) and e_1, the line ep_2 is defined by points p_2 i e_2
- Each epipolar line goes through e_1 (in the left image) or e_2 (in the right image) if they are defined
- Image search for the point corresponding to p_1 is performed along ep_2 line
- Image search for the point corresponding to p_2 is performed along ep_1 line

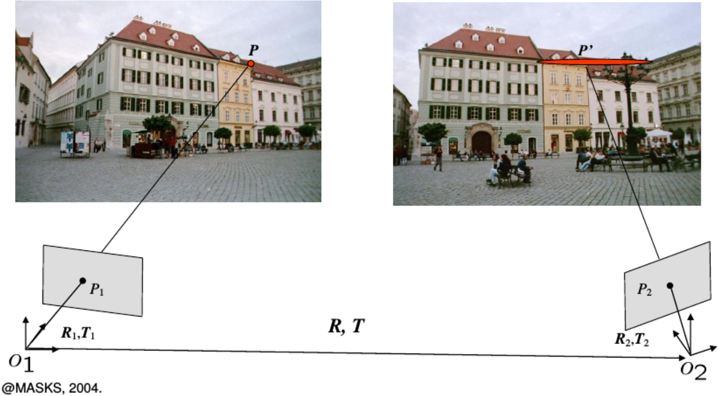
Epipolar geometry

Remark: If image plane is parallel to the baseline $O_1 O_2$ corresponding epipole is located in infinity, otherwise it has finite coordinates

Remark: Epipole may be situated outside visible image part

Remark: In the case of normalized images epipolar lines are parallel and both epipoles are located in infinity

Calibration of a stereo-pair



Coordinate systems of two cameras

Parameters of a stereo-pair

During the stere-pair calibration procedure thare are established the following camera parameters

1. First camera:
   - Shift (translation) of the world's system's origin to camera system's origin $\mathbf{t}_1$
   - Rotation transforming coordinates from the global coordinate system to the local coordinate system of the first camera R_1
   - Projective transformation F_1, and a transformation from a *normalized image coordinates* to image coordinates K_1 (the total intrinsic parameter matrix is $K_{int1} = K_1 F_1$
2. Second camera: Transformations $\mathbf{t}_2$, R_2, K_{int2} respectively

Remark: We can safely assume that our global coordinate system is the coordinate system of the first camera, then $R_1 = I$ $\mathbf{t}_1 = [0, 0, 0]^T$, and the second camera is described by t,R relating coordinates from the 1st to the 2nd camera frame.

Essential matrix

The basic stereo-pair calibration proces involves estimation of the rotation $\mathbf{R}$ and translation $\mathbf{t}$ between coordinate systems of both cameras. Thus we have

$$P_2 = \mathbf{R} P_1 + \mathbf{t}$$

where $P_1 = (X_1, Y_1, Z_1)^T$ oraz $P_2 = (X_2, Y_2, Z_2)^T$ are coordinates of the same point P expressed in coordinate systems of the 1st and 2nd camera correspondingly.

It can be verified, that coordinates P_1 i P_2 satisfy the epipolar constraint

$$P_2^T \hat{\mathbf{T}} \mathbf{R} P_1 = 0$$

Where $\hat{\mathbf{T}}$ is a matrix representation of a cross-product

$$\hat{\mathbf{T}} p = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} p = \mathbf{t} \times p$$

Essential matrix

Epipolar constraints define a direct relation between points in coordinate systems of both cameras. Thie relation can be conveniently represented by an essential matrix

$$\mathbf{E} = \hat{\mathbf{T}}\mathbf{R}$$

Thus, the epipolar constraint is given as

$$P_2^T \mathbf{E} P_1 = 0$$

Essential matrix

Now instead of 3D coordinates let us used a *normalized image coordinates*.

$$P_{norm1} = \begin{pmatrix} X_1/Z_1 \\ Y_1/Z_1 \\ 1 \end{pmatrix}, P_{norm2} = \begin{pmatrix} X_2/Z_2 \\ Y_2/Z_2 \\ 1 \end{pmatrix}$$

(normalized image coordinates can be regarded as a perspective projections of points onto a virtual screen located 1 unit from the optical center, with the optical axis intersecting $(0,0,1)^T$).
In such situation the epipolar constraints equation takes similar form:

$$P_{norm2}^T \mathbf{E} P_{norm1} = 0$$

Conversion between *normalized image coordinates* and real image coordinates is trivial

$$p = \mathbf{K_{int}} P_{norm}, \; P_{norm} = \mathbf{K_{int}}^{-1} p$$

Fundamental Matrix

Let's now use a substitution $P_{norm} = \mathbf{K_{int}}^{-1} p$

$$\begin{aligned} p_2^T (K_{int2}^{-1})^T \mathbf{E} (K_{int1}^{-1}) p_1 &= 0 \\ p_2^T \mathbf{F_m} p_1 &= 0 \end{aligned}$$

where

$$\begin{aligned} (K_{int2}^{-1})^T \mathbf{E} (K_{int1}^{-1}) &= \mathbf{F_m} \text{ or simpler} \\ K_{int2}^T \mathbf{F_m} K_{int1} &= \mathbf{E} \end{aligned}$$

A matrix $\mathbf{F_m}$ is a fundamental matrix and directly relates projections of a 3D points int two stereo-images

Essential matrix properties

Essential matrix $(\mathbf{E})$

- constraints the transformation of points on a normalized image plane in both cameras
- does not depend on intrinsic camera parameters, in the case of cameras with calibrated intrinsic parameters the real image coordinates can be computed directly from normalized image
- is a singular matrix or rank 2
- have 5 independent parameters or 6 six independent parameters if it is normalized (i.e. equals to $\hat{\mathbf{T}}\mathbf{R}$ exactly)

Essential matrix properties

Essential matrix ($\mathbf{E}$)

- can be estimated basing on corresponding point pairs in stereo-images
  - the basisc estimation algorithm is based on DLT and is called 8-point algorithm
  - more advanced algorithm requires only 5 points (hence 5-points algorithm) - utilizes additional constraints on E
  - estimation yields ambiguous for some point configurations
- can be computed from relative $\mathbf{R}$ i $\mathbf{t}$ of camera pair
- the other way round $\mathbf{R}$ i $\mathbf{t}$ can be computed basing on $\mathbf{E}$ (with scale ambiguity in $\mathbf{t}$) - estimation of $\mathbf{R}$ i $\mathbf{t}$ can have up to 4 solutions - cheirlarity constraint (reprojection of points) is used to disambiguate
- most often enables to establish relative camera positions (up to a scale) and thus to reconstruct a structure of the observed 3D scene from point correspondences

Fundamental matrix properties

Fundamental matrix ($\mathbf{F_m}$)

- relates coordinates of corresponding points in stereo-images (after distortion removal)
- depends on the intrinsic camera parameters
- is a singular matrix of rank 2
- can be estimated from correspondences of points in stereo-images, the basic estimation algorithm is the 8-point algorithm
- can be easily computed from $\mathbf{R}$, $\mathbf{t}$, $\mathbf{K_{int1}}$, $\mathbf{K_{int2}}$ of camera stereo-pair
- typically does not allow to reconstruct 3D scene structure (only up to an unknown projective transformation)

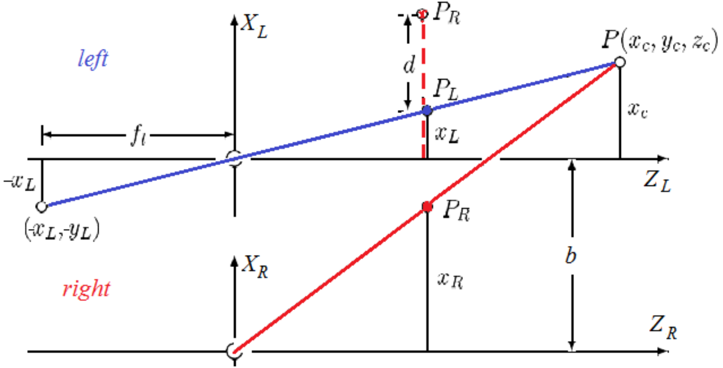
Stereo-images normalization

Stereo-images normalization

Let us assume that camera 1 is the left camera (L), and camera 2 is the right camera (R). Goal of normalization - transformation of stereo images in a way that:

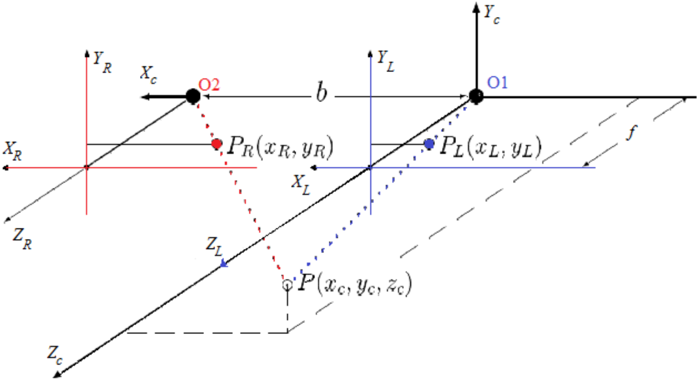
1. look like captured from 2 identical cameras (having identical K_{int}
2. be epiplanar i.e.
   - depths and heights of each scene point meet a condition $Z_L = Z_R$ oraz $Y_L = Y_R$, alternatively
   - planes and axes $X_L Y_L$ and $X_R Y_R$ were parallel and axes X_L i Y_R coincide, alternatively
   - there is no between-camera rotation $\mathbf{R}$, and the translation vector $\mathbf{t}$ coincides with the baseline (connecting optical centers of cameras)

Stereo-images normalization



Normalization of stereo images (2D projection)

Stereo-images normalization



Normalization of stereo images

Stereo-images normalization

In normalized stereo-images

- Epipoles are in infinity
- Epipolar lines are horizontal
- Corresponding epipolar lines have identical y coordinate

General stereo-normalization procedure

- Project points of both images to a new (common) image plane L
- For each image pixel p_i $(i = 1, 2)$ in i-th image its projection p_i' $(i = 1, 2)$ is the intersection of of a line $O_i p_i$ and the plane L

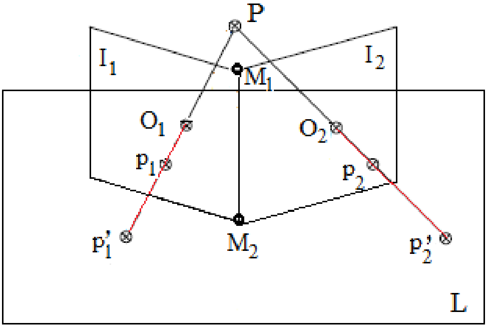
How to establish parameters of L?

Stereo-images normalization procedure (1)

Method 1

Plane L is defined as parallel to

- line O_1O_2 (base line)
- line M_1M_2 (intersection of original image planes)



Some problems: e.g. if O_1O_2 and M_1M_2 are parallel (cameras are rotated around baseline)

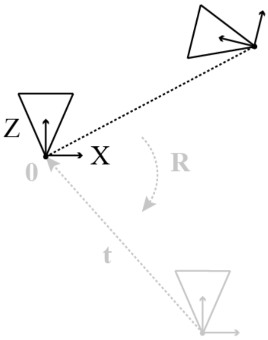
Stereo-images normalization procedure (2)

Method 2

Bouget's algorithm

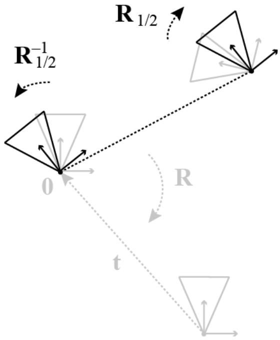
(images: Michael Hornacek, Vienna Univ. Of Techn.)

Initial camera setup $\mathbf{R}$ and $\mathbf{t}$



Stereo-images normalization procedure (2)

'Division' of rotation $\mathbf{R}$ between two cameras $\mathbf{R}^{1/2}$



Cameras now 'look' in the same direction (their coordinate frames have parallel axes)

Stereo-images normalization procedure (3)

Alignment of axis X of cameras to the baseline

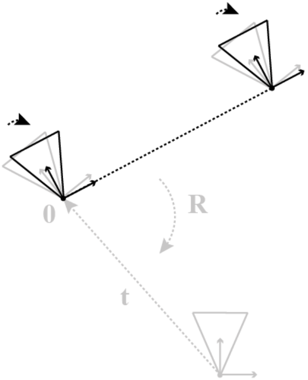
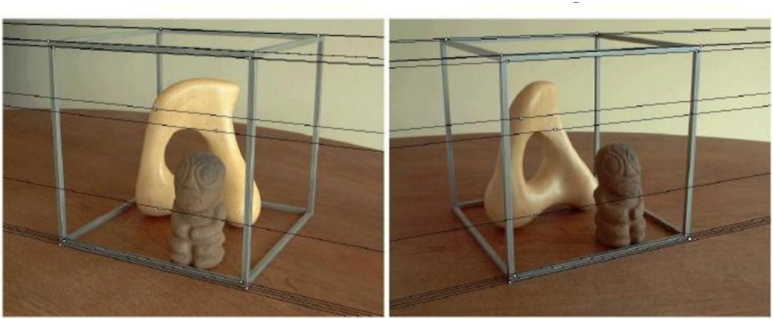


Image normalization - examples

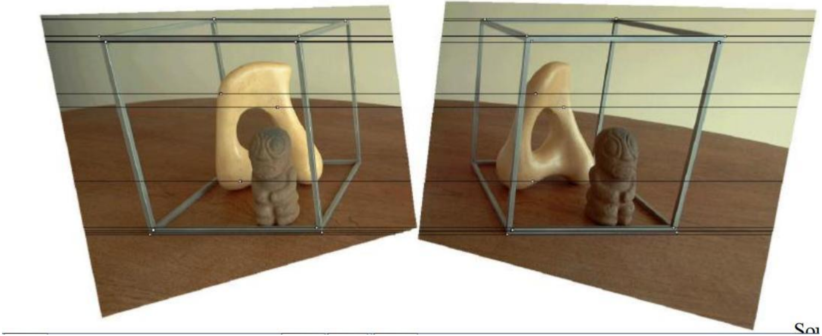
Unnormalized pair of mages



source: Alosha Eyfros

Image normalization - examples

Normalized pair of images



source: Alosha Eyfros

Image normalization - examples

Well aligned cameras but unnormalized images

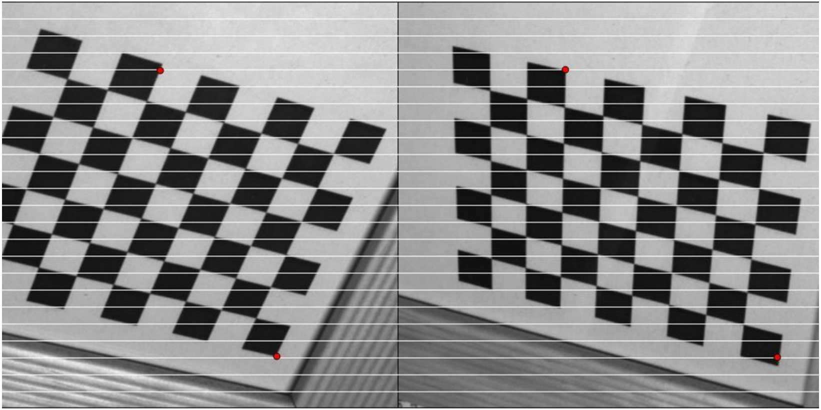
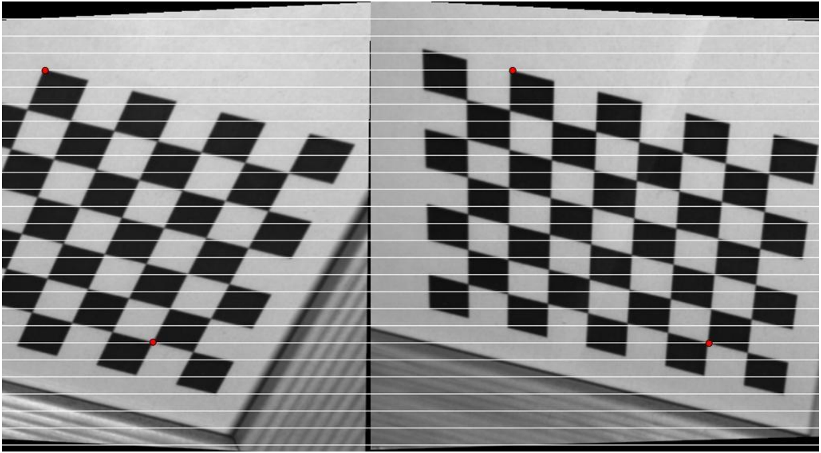


Image normalization - examples

Normalized pair of images



Disparity map

Disparity map

In normalized stereo-images, the vector of translation between optical centers of two cameras is

$$O_2 = O_1 + (b, 0, 0)^T$$

and b is the baseline distance Any scene points $P = (X_c, Y_c, Z_c)^T$ expressed in coordinate frames of two cameras is

$$P_L = (X_c, Y_c, Z_c)^T, P_R = (X_c - b, Y_c, Z_c)^T,$$

Disparity d of corresponding points in stereoimages p_r i p_l (projections of a single point P), is defined as

$$d = x_L - x_R$$

Disparity map and the depth

Value of disparity of each point is related to the point depth (Z-value), i.e.

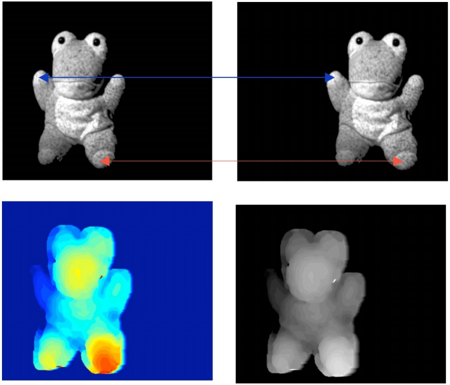
$$x_L = X_c \frac{f}{Z_c} + c_x \Rightarrow X_c = x_L \frac{Z_c}{f} - Z_c c_x,$$
$$x_R = (X_c - b)\frac{f}{Z_c} + c_x \Rightarrow X_c = x_R \frac{Z_c}{f} + b - Z_c c_x.$$

After elimination of X_c we obtain

$$x_L \frac{Z_c}{f} = x_R \frac{Z_c}{f} + b$$
$$d = x_L - x_R = \frac{f \cdot b}{Z_c}$$
$$Z_c = \frac{f \cdot b}{x_L - x_R} = \frac{f \cdot b}{d}$$

Disparity map and the depth

Right view



Left view

hot=close

light=close

Disparity

source: Andrea Fusiello Lecture Notes (corrected)

Disparity map and the depth

Conclusion: For every non-zero disparity d value, assuming that the baseline distance b and focal length f are known we can establish Z coordinate of a point.
Baseline length is a compromise between the accuracy and ease of finding point correspondences in a stereo-pair of images:

- large values of the baseline length b make it more difficult to match points in both images (slanted surfaces are observed from different angles in both views, but also
- large values of the baseline length b increase depth resolution

Depth estimation error

Let us imagine that the point is moving along Z axis giving a small change of depth $\Delta Z = z_1 - z_2$ and a small change of disparity $\Delta d = d_2 - d_1$.:

$$d = \frac{fb}{Z} \Rightarrow d_2 - d_1 = \frac{fb}{z_2} - \frac{fb}{z_1} \Rightarrow d_2 - d_1 = fb\frac{z_1 - z_2}{z_1 z_2}$$

Hence

$$\Delta d = fb\frac{\Delta Z}{(Z_2 + \Delta Z)Z_2}$$

If $|\Delta Z| << |Z_2|$, then the equation (after dropping Z index), can be approximated as

$$\Delta d \approx fb\frac{\Delta Z}{Z^2} \Rightarrow \Delta Z \approx \frac{Z^2}{fb}\Delta d$$

Observation: error of depth estimation is proportional to the **square** of disparity error (Solve E.2)

Matching pixel blocks

Finding correspondences in stereo-images

In order to obtain disparity map, to points in the left image should be matched with points in the right image. This is a point correspondence problem

With respect to the method of finding correspondences stere-vision algorithms can be divided into categories:

- 'sparse' stereovision - correspondences are established of a selected 'sparse' set of characteristic points)
- 'dense' stereovision - correspondences are established for all image points

In the case of 'dense' stereovision correspondence finding can be implemented by:

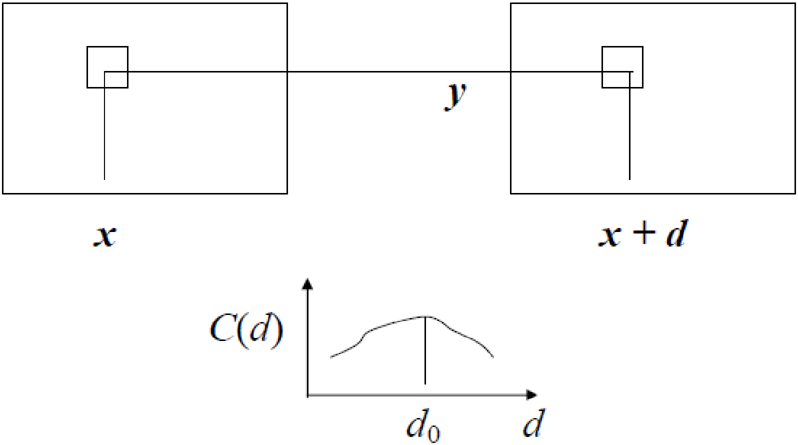
- comparing of single pixels (not so effective without global optimization)
- comparing of pixel blocks
- comparing of pixels or pixel blocks supported by a global or semi-global optimization

Matching pixel blocks

Problem statement: For arbitrary block $B_{x,y}$ in the first image centered around (x, y) our task is find the best matching block $B_{x+d,y}$ in the second image
The best match can be understood two-fold

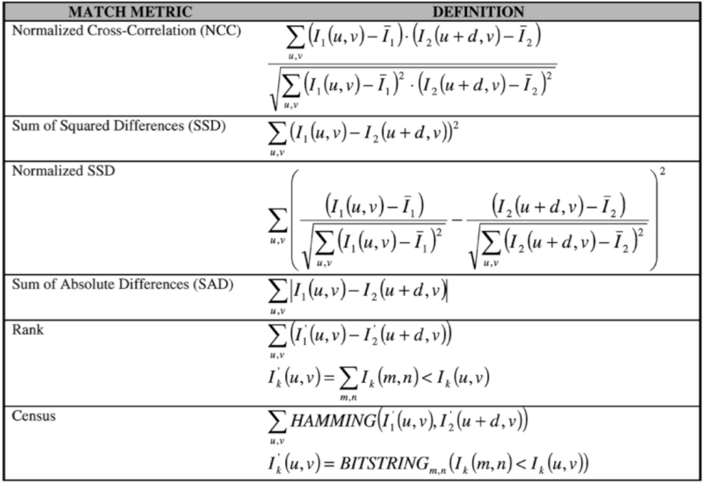
- $d_0 = \text{argmax}_d C_{x,y}(d)$, where $C(\cdot)$ is a measure of correlation between blocks
- $d_1 = \text{argmin}_d E_{x,y}(d)$, where $E(\cdot)$ is a measure of difference between blocks

Matching pixel blocks



Matching pixel blocks

Measures of match quality:



MATCH METRIC	DEFINITION
Normalized Cross-Correlation (NCC)	$\dfrac{\sum_{u,v}(I_1(u,v)-\bar{I}_1)\cdot(I_2(u+d,v)-\bar{I}_2)}{\sqrt{\sum_{u,v}(I_1(u,v)-\bar{I}_1)^2\cdot(I_2(u+d,v)-\bar{I}_2)^2}}$
Sum of Squared Differences (SSD)	$\sum_{u,v}(I_1(u,v)-I_2(u+d,v))^2$
Normalized SSD	$\sum_{u,v}\left(\dfrac{(I_1(u,v)-\bar{I}_1)}{\sqrt{\sum_{u,v}(I_1(u,v)-\bar{I}_1)^2}}-\dfrac{(I_2(u+d,v)-\bar{I}_2)}{\sqrt{\sum_{u,v}(I_2(u+d,v)-\bar{I}_2)^2}}\right)^2$
Sum of Absolute Differences (SAD)	$\sum_{u,v}\lvert I_1(u,v)-I_2(u+d,v)\rvert$
Rank	$\sum_{u,v}(I_1'(u,v)-I_2'(u+d,v))$ $I_k'(u,v)=\sum_{m,n}I_k(m,n)<I_k(u,v)$
Census	$\sum_{u,v}HAMMING(I_1'(u,v),I_2'(u+d,v))$ $I_k'(u,v)=BITSTRING_{m,n}(I_k(m,n)<I_k(u,v))$

source: Brown, M. Z., Burschka, D., and Hager , Adv. In comp. stereo

Matching pixel blocks

Typical procedure of block matching

1. For each pixel (x, y) from the **first** stero-image and acceptable d compute a normalized correlation of blocks $C_{x,y}^{1\rightarrow 2}(d)$, select the best value d_1
2. For each pixel (x, y) from the **second** stereo-image and acceptable d compute symmetrically a normalized correlation of blocks $C_{x,y}^{2\rightarrow 1}(d)$, select the best value d_2
3. Compute disparity map $d(x, y)$ only for pixels (x, y), for which both matches are compliant , i.e. $d_1 = -d_2$ (this is a cross-valiation)
4. Optionally: fill 'empty spaces' by interpolating disparity map
5. Optionally: find a compromise between smoothness of the disparity map and computed original disparities

Matching pixel blocks

A compromise between smoothness of the disparity map and computed origina disparities can be obtained by minimization of the function:

$$\text{argmin}_S \left(\sum_{(x,y) \text{for avail.} d(x,y)} (S(x,y) - d(x,y))^2 + \lambda \sum_{(x,y)} (\Delta S(x,y))^2 \right)$$

where $\Delta S(x,y) = \frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2}$ - is the Laplace operator

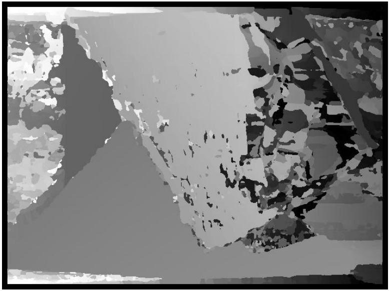
In general: Block matching is efective for fronto-parallel surfaces or near fronto-parallel surfaces. For close slanted surfeace or in the case of weak texture - matching results may be much worse

Selection of block size

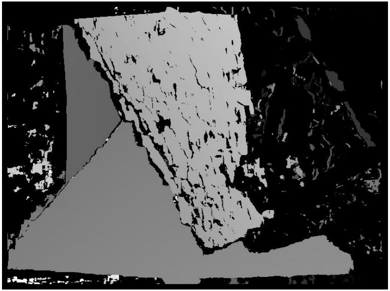
- Small block - better depth accuracy and details (no averaging), but greater chance for a bad match
- Large block - worse accuracy and details (averaging), but less chance for a bad match (especially if cross-validation is applied)

Matching pixel blocks

Matching pixel blocks

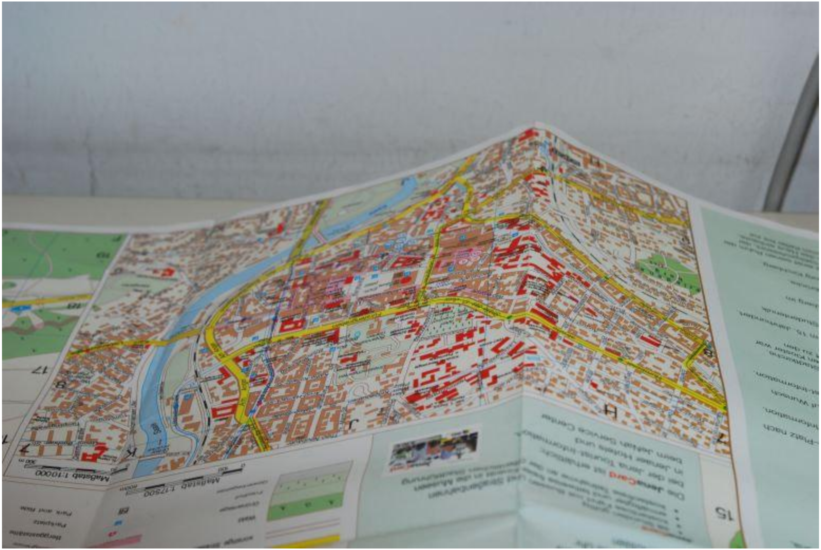


no cross-validation

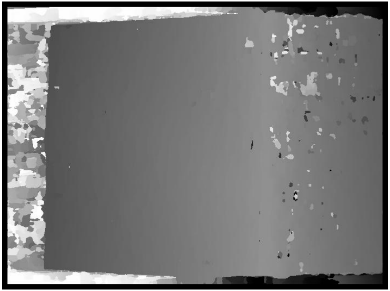


cross-validation

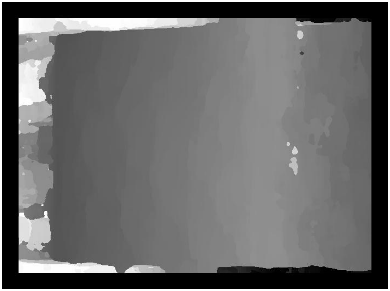
Matching pixel blocks



Matching pixel blocks



block 10 pix.



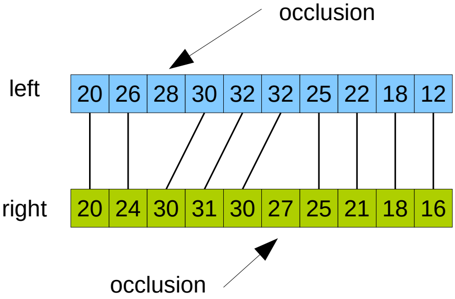
block 30 pix.

Matching by a dynamic programming

Dynamic programming matching

Dynamic programming matching:

- sub-global method - we try to optimally match two lines



left	20	26	28	30	32	32	25	22	18	12
right	20	24	30	31	30	27	25	21	18	16

- the optimal matching consider both pixel similarity and disparity smoothness
- the dynamic programming is used as a solution
- we assume the pixel ordering constraint

Dynamic programming matching

Dynamic programming matching - more formally:

- left scan-line pixel values $L = \{l_i | i = 1, 2, \dots, N\}$
- right scan-line pixel values $R = \{r_j | j = 1, 2, \dots, M\}$
- $C(i, j)$ - the minimum cost of matching the first i pixels from L and j first pixels from R

A goal is to find a matching path for pixels from both scanlines

$$(i_0, 0) \vee (0, j_0) \rightarrow (i_1, j_1) \rightarrow (i_2, j_2) \rightarrow \cdots \rightarrow (i_K, j_K) = (N, M)$$

that is optimal with respect to:

- the total cost of matching of full scan-lines $C(N, M)$

Dynamic programming matching

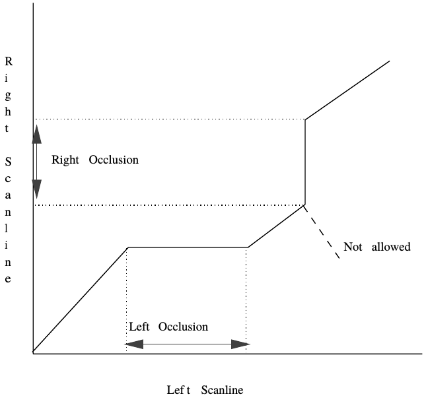
Before starting the algorithm we need to define:

- The cost of pixel $\mathrm{Occlusion}$ (independent of image data)
- The cost of pixel pixel match $c(l, r)$ (dependent on image data)

Acceptable transitions:

- Transition $(i_k, j_k) \rightarrow (i_k + 1, j_k + 1)$ - induces only a cost of pixel (block) matching $c(l_{i_k+1}, r_{j_k+1})$
- Transition $(i_k, j_k) \rightarrow (i_k + 1, j_k)$ - induces only a cost of occlusion of a left-scanline pixel: $\mathrm{Occlusion}$
- Transition $(i_k, j_k) \rightarrow (i_k, j_k + 1)$ - induces only a cost of occlusion of a right-scanline pixel: $\mathrm{Occlusion}$
- Due to our pixel ordering constraint other directions are not allowed, so $i_k \leq i_{k+1}$ i $j_k \leq j_{k+1}$

Dynamic programming matching



Source: Cox, Hingorani, Rao, Maggs A Maximum Likelihood Stereo Algorithm

Dynamic programming matching

Optimal cost estimation - algorithm:

```
C(0,0) = 0 ;
for (i=1;i <= N;i++) { C(i,0) = i*Occlusion ; }
for (j=1;j <= M;j++) { C(0,j) = j*Occlusion ; }
for (i=1;i <= N;i++)
   for (j=1;j <= M;j++) {
      min1 = C(i-1,j-1)+c(i,j) ;
      min2 = C(i-1,j)+Occlusion ;
      min3 = C(i,j-1)+Occlusion ;
      C(i,j) = cmin = min(min1,min2,min3) ;
      if(min1=cmin) M(i,j)=1 ;
      if(min2=cmin) M(i,j)=2 ;
      if(min3=cmin) M(i,j)=3 ;
   }
```

Dynamic programming matching

Reconstruction of the optimal path:

```
p = N ;
q = M ;
while(p!=0 && q!=0) {
   switch(M(p,q)) {
      case 1:
        p matches q ;
        p--;q--;
        break ;
      case 2:
        p have no match
        p--;
        break ;
      case 3:
        q have no match
        q--;
        break ;
      }
   }
```

Dynamic programming matching

Dynamic programming matching - match cost:

- the cost function $c(i,j)$ can be e.g. a normalized SSD matching windows centered at i and j
- the optimal occlusion cost $\mathrm{Occlusion}$ is a constant and can be estimated theretically [Cox, Hingorani, Rao, Maggs A Maximum Likelihood Stereo Algorithm] or established experimentally

Dynamic programming matching

Additional constraints: In order to speed-up the algorithm and/or to eliminate improbable results we can introduce limitations concerning acceptable values of disparity

Then in the cost-computation algorithm we consider only pairs (i, j) meeting the condition $min_j \leq j \leq max_j$, where

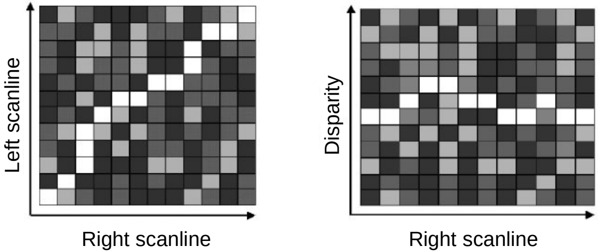
$$min_j = i + d^-$$
$$max_j = i + d^+$$

i d^- i d^+ - task-specific minimal and maximal disparity values

Dynamic programming matching

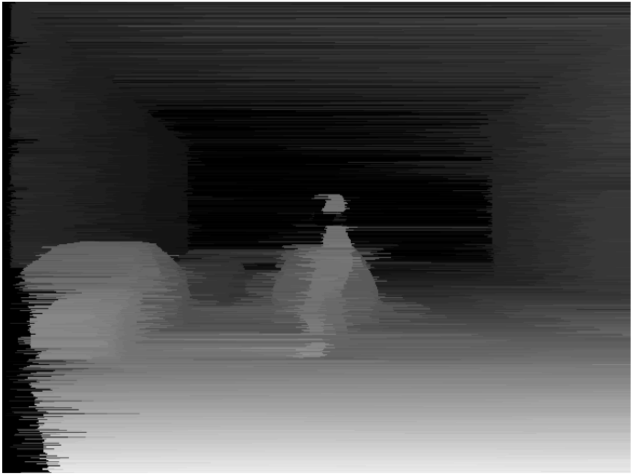
The search for the optimal path can be performed in

- the right scanline -left scanline graph
- the right scanline - disparity graph



source: W. Mark and D. M. Gavrila, Real-Time Dense Stereo for Intelligent Vehicles

Dynamic programming matching - streaking effects



source: M. Brown, Z. Burschka, i G.D. Hager (colors are not representative)

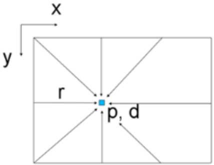
Semi-global matching via DP

Method outline:

- The optimized energy (cost) function is similar to the one used in DP method, however bigger disparity jumps are penalized more than smaller ones (moderately slanted surfaces are accepted more easily).

$$E(D) = \sum_{p} \left(C(p, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + P_2 T[|D_p - D_q| > 1] \right)$$

- Optimization of disparity smoothness is performed in several directions at the same time. There is chosen a direction with the smallest cost



source: Heiko Hirschmuller

Global optimization

Graph-cut method

Method outline

The result of pixel matching is the function f given as:

$$f(p,q) = \begin{cases} 1 & \text{if there is a match between } p \text{ and } q \\ 0 & \text{otherwise} \end{cases}$$

A goal is to select the function f, that minimizes the following energy function

$$E = E_{data} + E_{smooth} + E_{unique} + E_{occl}$$

- E_{data} - measures quality of pixel (block) match
- E_{smooth} - measures the disparity smoothness (with an accent on areas of similar color)
- E_{unique} - measures the uniqueness of matches (ideally there should be no double matches)
- E_{occl} - measures the amount of occluded pixels (so we have the maximum possible number of matches)

Graph-cut method

Optimization problems: local minima and prohibitive computation cost

The solution: The energy function can be expressed in a way that enables using **a global optimization** by using **graph-cut algorithms**

Article: *Kolmogorov and Zabih's Graph Cuts Stereo Matching Algorithm*

Sparse stereovision

Sparse stereovision

Sparse stereovision; requires initial extraction of image features

Typical features: lines, corners, blobs, vertices, polygons and other characteristic shapes and objects

Pros

- much lower cost of computation
- smaller sensivitiy to lighting conditions
- easier implementation of sub-pixel accuracy

Cons

- we have a sparse point cloud as an output
- a result depends on the quality of feature extraction
- in the case of a 'dull' areas we may have only a small number of features

Examples

E.1 Homography estimation - general case

Construct an appropriate system of equations for homography estimation from point correspondences using DLT.

E.2 Depth estimation error

Estimate an error of depth estimimation for stereo images obtained from identical cameras with focal length $f = 530pix$, baseline length 30cm for an object located 1,2 or 10 meters from the camera.
Assume that the estimation error of point position in an image is a) 0.5 pix. b) 0.1 pix.

E.2 Depth estimation error - hints & results

	$Z = 1$	$Z = 2$	$Z = 10$
$\Delta d = 0.5$	0.0031	0.0126	0.3145
$\Delta d = 0.1$	0.0006	0.0025	0.0629

(values given in meters)

E.3 Homography of 3D planar object

A camera with a projection matrix $I[R|t]$ (R,$\mathbf{t}$ – unknown) observes a planar points with local coordinates $(0, 0)^T$, $(1, 0)^T$, $(0, 1)^T$, $(1, 1)^T$. The projections of respective points are $(-0.4, -0.6)^T$,$(-0.3, -0.4268)^T$,$(-0.5732, -0.5)^T$, $(-0.4732, -03268)^T$.

1. Compute homography matrix relating points and their projections
2. Recover R and $\mathbf{t}$ from the homography matrix.

E.3 Homography of 3D planar object - hints & results I

1. Utilize zero elements of input vectors e.g.

$$\begin{pmatrix} \mathbf{h}_{\cdot 1} & \mathbf{h}_{\cdot 2} & \mathbf{h}_{\cdot 3} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ 1 \end{pmatrix} = s \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

where $h_{\cdot i}$ is the i-th column of H, X_1, X_2 are coordinates on a plane and x_1, x_2 are image coordinates. We will number the (augmented) points pairs consecutively (e.g. $\mathbf{X}^1 \rightarrow \mathbf{x}^1$ ... are point coordinates agumented with 1). Taking e.g. $\mathbf{X}^1 = (0, 0, 1)^T$ we have

$$\begin{pmatrix} \mathbf{h}_{\cdot 1} & \mathbf{h}_{\cdot 2} & \mathbf{h}_{\cdot 3} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{h}_{\cdot 3} \end{pmatrix} = s \begin{pmatrix} x_1^1 \\ x_2^1 \\ 1 \end{pmatrix} \text{ so } \mathbf{h}_{\cdot 3} = s\mathbf{x}^1$$

E.3 Homography of 3D planar object - hints & results II

Note that we may fix scaling of the homography matrix by assuming the first scaling factor to be 1, so $\mathbf{h}_{.3} = \mathbf{x}^1$ (we have the first vector of H). Processing further points we end up with a system of equations

$$\begin{cases} \mathbf{h}_{.3} = \mathbf{x}^1 \\ \mathbf{h}_{.1} + \mathbf{h}_{.3} = \alpha \mathbf{x}^2 \\ \mathbf{h}_{.2} + \mathbf{h}_{.3} = \beta \mathbf{x}^3 \\ \mathbf{h}_{.1} + \mathbf{h}_{.2} + \mathbf{h}_{.3} = \gamma \mathbf{x}^4 \end{cases}$$

Which can be further transformed into

$$\mathbf{h}_{.3} = \alpha \mathbf{x}^2 + \beta \mathbf{x}^3 - \gamma \mathbf{x}^4$$

E.3 Homography of 3D planar object - hints & results III

By solving the equation we have $\alpha = \beta = \gamma = 1$. The resulting homography matrix is then

$$H = \begin{pmatrix} 0.1 & -0.1732 & -0.4 \\ 0.1732 & 0.1 & -0.6 \\ 0 & 0 & 1 \end{pmatrix}$$

2. Normalize homography matrix to obtain rotation and translation parameters. In our task the first two vectors of our matrix are rotation vectors, so their norm should be 1. After normalization of the matrix we have:

$$H_{norm} = \begin{pmatrix} 0.5 & -0.866 & -2 \\ 0.866 & 0.5 & -3 \\ 0 & 0 & 5 \end{pmatrix}$$

E.3 Homography of 3D planar object - hints & results IV

The recovered R and $\mathbf{t}$ are

$$R = \begin{pmatrix} 0.5 & -0.866 & 0 \\ 0.866 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{t} = \begin{pmatrix} -2 \\ -3 \\ 5 \end{pmatrix}$$

What if our normalization coefficient was negative? Do we obtain a second equivalent solution in this way?

E.4 Homography of the rotated camera I

The camera rotates around an unknown axis, the rotation angle is also unknown. There were established point correspondences between the original (first) and rotated (second) views. Basing on these correspondences, the following homography matrix was estimated

$$H = \begin{pmatrix} 0.273 & 0 & 176.2 \\ -0.12 & 0.5 & 22.323 \\ -0.0005 & 0 & 0.593 \end{pmatrix} \text{ and } sp_2 = Hp_1$$

p_1 belongs to the original view and p_2 belongs to the rotated view. The intrinsic camera matrix is also given

$$K_{int} = \begin{pmatrix} 500 & 0 & 320 \\ 0 & 500 & 240 \\ 0 & 0 & 1 \end{pmatrix}$$

E.4 Homography of the rotated camera II

1. Find rotated view point coordinates corresponding to the point $(300, 300, 1)^T$ in the original view
2. Find a rotation matrix for the second view knowing that the rotation for the first view is identity.

E.4 Homography of the rotated camera - hints & results

1. $p_2 = (582.6104, 307.7181)^T$
2. Use relation $R = K_{int}^{-1} H K_{int}$ and normalize the rotation matrix

$$R = \begin{pmatrix} 0.866 & 0 & 0.5 \\ 0 & 1 & 0 \\ -0.5 & 0 & 0.866 \end{pmatrix}$$

E.5 Fundamental matrix I

There is given a fundamental matrix relating two views obtained from a single but re-positioned camera

$$F_M = \begin{pmatrix} 0 & 0 & 0 \\ -0.001 & 0 & 1.1860 \\ 0.24 & -1 & -44.6461 \end{pmatrix}, \; p_2^T F_m p_1 = 0$$

1. Find epipolar line in the 2nd image corresponding to the point $p_1 = (300, 300, 1)^T$ from the first image
2. Find epipolar line in the 1st image corresponding to the point $p_2 = (300, 300, 1)^T$ from the second image
3. Find epipoles in both images (if they exist)

E.5 Fundamental matrix II

4. Does the given fundamental matrix correspond to the following projection matrices?:

$$P_1 = K_{int}(\mathbf{I}|0),\ P_2 = K_{int}(\mathbf{R}|\mathbf{t})$$

with

$$\mathbf{R} = \begin{pmatrix} 0.866 & 0 & 0.5 \\ 0 & 1 & 0 \\ -0.5 & 0 & 0.866 \end{pmatrix},\ \mathbf{K_{int}} = \begin{pmatrix} 500 & 0 & 320 \\ 0 & 500 & 240 \\ 0 & 0 & 1 \end{pmatrix},\ \mathbf{t} = (-5, 0, 0)^T$$

E.5 Fundamental matrix - hints & answers I

1. $0.8861y - 272.6461 = 0$
2. $-0.06x - y + 311.1615 = 0$
3. 1st: $\approx (1186.025, 240)^T$, 2nd: does not exist
4. YES

E.6 Triangulation

The projection matrix of the 1st camera is $P_1 = K_{int}[\mathbf{I}|0]$, and the projection matrix of the second camera is $P_2 = K_{int}[\mathbf{R}|\mathbf{t}]$ (all the parameters $K_{int}, R, \mathbf{t}$) are given.
Find equations of rays for both cameras corresponding to the image point p_1 in camera 1 and the point p_2 in camera 2, that both are projections of an unknown 3D point P. Propose a method to find coordinates of P (assume ideal error-free data).

E.6 Triangulation - hints & answers I

1. Hint: reverse a general projection equation

$$s\begin{pmatrix}x\\y\\1\end{pmatrix} = K_{int}[R|\mathbf{t}]\begin{pmatrix}X\\Y\\Z\\1\end{pmatrix}$$

and express world coordinates in a parametric form

$$\begin{pmatrix}X\\Y\\Z\end{pmatrix} = s\mathbf{v} + \mathbf{w}$$

E.g. for the first camera we have $\mathbf{v} = K_{int}^{-1}\begin{pmatrix}x\\y\\1\end{pmatrix}$ and $\mathbf{w} = 0$

2. Hint: compare ray equations

E.7 Stereovision via dynamic programming

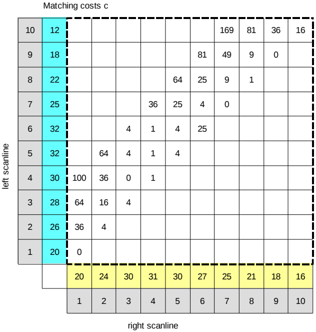
There are given to corresponding scanlines of the stereopair of images

left	20	26	28	30	32	32	25	22	18	12
right	20	24	30	31	30	27	25	21	18	16

1. Find correspondences between scanlines using *dynamic programming*. Assume:
   - single pixel matching
   - cost function for matching corresponding pixels $c(i, j) = (l_i - r_i)^2$
   - cost of occlusion $\text{Occlusion}{=}10$
   - border disparity values $d^- = 0$ i $d^+ = 3$
2. Assuming baseline length $b = 0.1m$ and focal length value $f = 500pix$, find z coordinates of points that were successfuly matched

E.7 Stereovision via dynamic programming - hints & answers

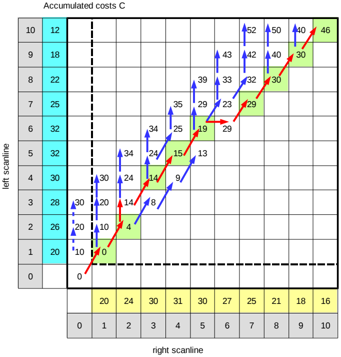
1. DP table



Matching costs c

left scanline											
10	12							169	81	36	16
9	18						81	49	9	0	
8	22					64	25	9	1		
7	25				36	25	4	0			
6	32			4	1	4	25				
5	32		64	4	1	4					
4	30	100	36	0	1						
3	28	64	16	4							
2	26	36	4								
1	20	0									
		20	24	30	31	30	27	25	21	18	16
		1	2	3	4	5	6	7	8	9	10

right scanline



Accumulated costs C

left scanline												
10	12								52	50	40	46
9	18							43	42	40	30	
8	22						39	33	32	30		
7	25					35	29	23	29			
6	32				34	25	19	29				
5	32			34	24	15	13					
4	30		30	24	14	9						
3	28	30	20	14	8							
2	26	20	10	4								
1	20	10	0									
0		0										
			20	24	30	31	30	27	25	21	18	16
		0	1	2	3	4	5	6	7	8	9	10

right scanline

E.7 Stereovision via dynamic programming - hints & answers

2. Use the standard formula $z = \frac{fb}{d}$

left	20	26	28	30	32	32	-	25	22	18	12
right	20	24	-	30	31	30	27	25	21	18	16
disparity	0	0	-	1	1	1	-	0	0	0	0
depth (m)	inf	inf	-	50	50	50	-	inf	inf	inf	inf