Mobile Robotics, Perception: structure from stereo

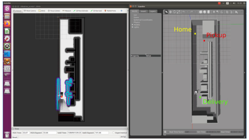
Material based on the book Autonomous Mobile Robot 2nd Ed. (Siegwart, Nourbakhsh, Scaramuzza) [AMR]; Chapter 4.2.5

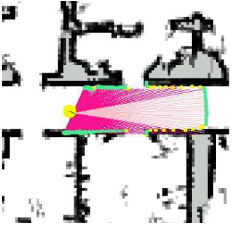
Summary

- Introduction to structure from stereo [4.2.5.1]
- Stereo vision [Chapter 4.2.5.2]
- Correspondence problem [Chapter 4.2.5.2]
- Epipolar geometry [Chapter 4.2.5.2]
- Epipolar rectification [Chapter 4.2.5.2]
- Disparity map [Chapter 4.2.5.2]

Introduction to structure from Stereo

- Range information are key for mobile robots
  - mapping, localization, obstacle avoidance....
- Single image **can not** provide information on depth
  - CCD/CMOS collapses 3D to 2D (homogeneous coordinates)
- Can recover depth by using several **different** images
  - camera geometry (e.g., depth from focus)
  - viewpoints
    - Structure from stereo
    - Structure from motion



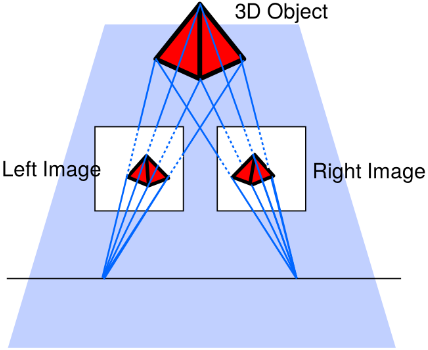


Stereo vision vs. Structure from motion

- Both techniques aim at recovering depth from different images
- **Stereo vision**
  - two images captured at the **same time** from two cameras in different positions
  - assume to know the relative positions of the two cameras
- **Structure from motion**
  - two images captured at **different times** in different positions
  - relative positions is **not known**
  - estimate motion (i.e., relative position) and structure

Stereo Vision: working principle

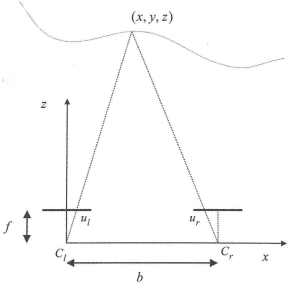
◇ Observes the same scene from two viewpoints, solve for the intersection of the rays to recover the 3D structure



Basic principle for stereo vision (Source [AMR])

Stereo Vision: simplified case

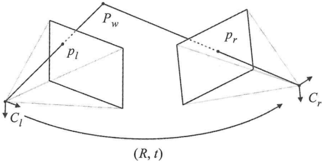
- ◇ both cameras are **identical** and **aligned** with the x-axis
- ◇ **Goal**: find an expression for the depth z of point $P = (x, y, z)$
- ◇ from similar triangles: $z = \frac{bf}{u_l - u_r}$
- ◇ **Disparity** difference in image location of the projection of p on the image plane: $u_l - u_r$
- ◇ **Baseline** distance between optical axes of two cameras



Idealized camera geometry for stereo vision (source [AMR])

Stereo Vision: general case

- two identical cameras do not exist!
- aligning both cameras on horizontal axis is very difficult (optical axes)
- to use stereo vision we need
  - **relative pose** between cameras (rotation, translation)
  - **focal lenght**, **optical center**, **radial distortion**
- use calibration!



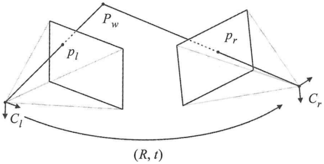
General case for stereo vision (source [AMR])

Commercial stereo camera (source [AMR])

Stereo Vision: triangulation

◇ to estimate the position of $P_w(X_w, Y_w, Z_w)$ we construct the system of equation for left and right camera

◇ **Triangulation**: problem of determining the 3D position of a point given a set of corresponding image location and known camera poses



General case for stereo vision (source [AMR])

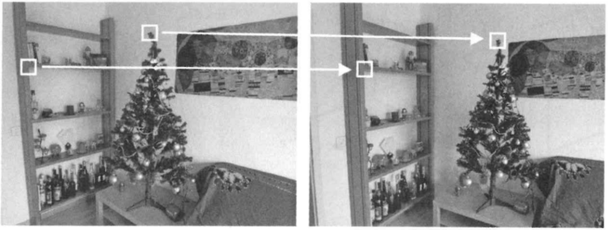
$$\tilde{p}_l = \lambda_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = K_l \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\tilde{p}_r = \lambda_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = K_r \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Stereo Vision: correspondence search

◇ **Goal**: identify image regions/patches in the left and right images that correspond to the same scene structure (Area Based)

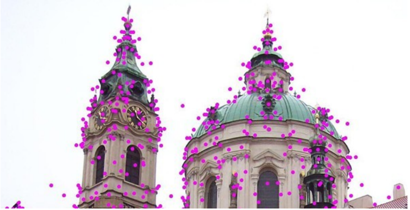
- Similarity measures: Normalized Cross-Correlation (NCC), Sum of Square Differences (SSD), Sum of Asbsolute Differences (SAD),...
- Exhaustive image search is computationally very expensive, need a more efficient approach



Corresponding points and re-projections of the same scene (source [AMR])

Correspondence search: feature based

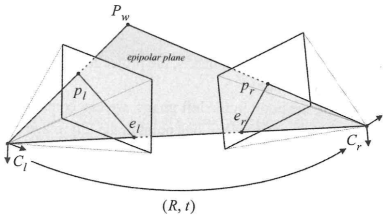
- **Features**: distinctive elements in the images that are easy to extract and that are invariant to small changes
  - Geometric: edges, corners, Line segments,...
  - Non geometric: SiFT, MSER, SURF
- Faster, more robust to illumination, scale, orientation
- Provide depth only for keypoints, need to interpolate.



Example of SIFT features (source [Lukas Mach via wikimedia])

Correspondence search: epipolar constraint

- ♢ The optical centers and an image point define the **epipolar plane**
- ♢ The intersection of the epipolar plane with the images define the **epipolar lines**
- ♢ The correspondent of a point in one image can only be found along the corresponding epipolar line in the other image
- ♢ We can restrict the search for correspondence along the epipolar line

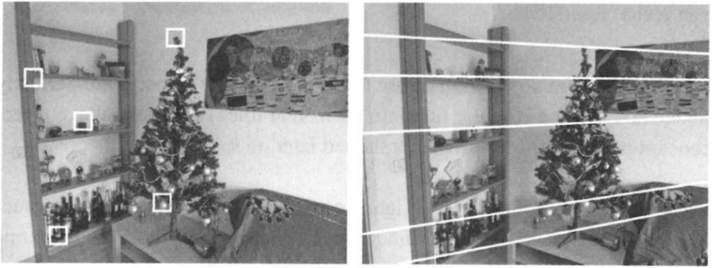


Epipolar plane and lines (source [AMR])

Correspondence search: using the epipolar lines

◇ Exploiting epipolar constraints corresponding points can be searched for only along epipolar lines.

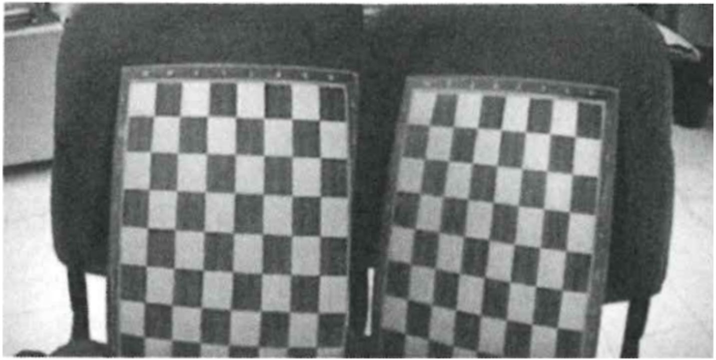
◇ Computational cost can be greatly reduced (search in 1D)



Epipolar lines for corresponding points (source [AMR])

Epipolar rectification

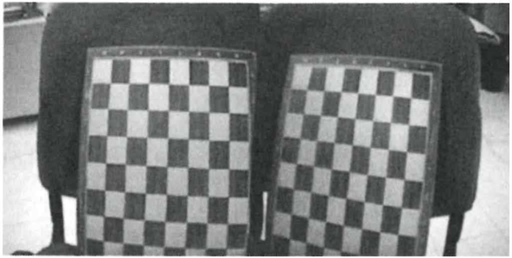
◊ **Goal**: transform left and right image so that epipolar lines are collinear and parallel to one of the two axes (usually the horizontal one)



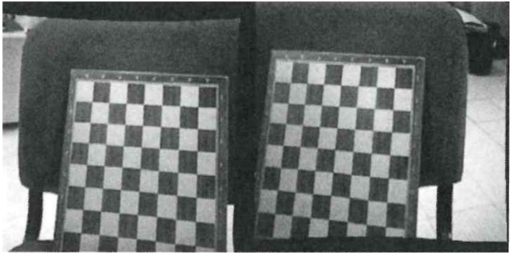
Original images (source [AMR])

Epipolar rectification: step 1

◇ Remove radial distortion



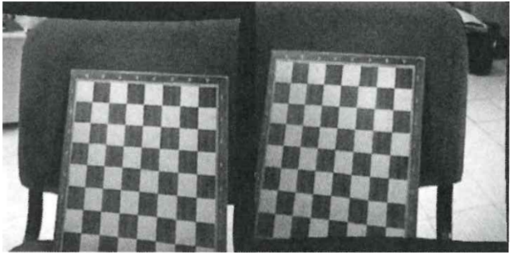
Original images (source [AMR])



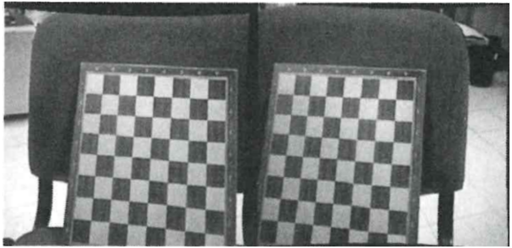
Radial distortion removed (source [AMR])

Epipolar rectification: step 2

◇ Compensation of rotation and translation



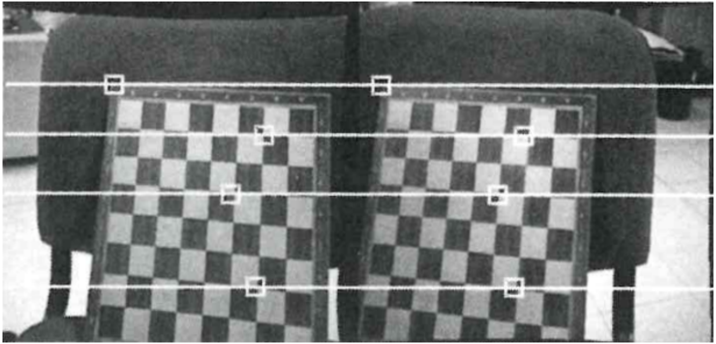
Radial distortion removed (source [AMR])



Compensation of rotation and translation (source [AMR])

Epipolar rectification: final result

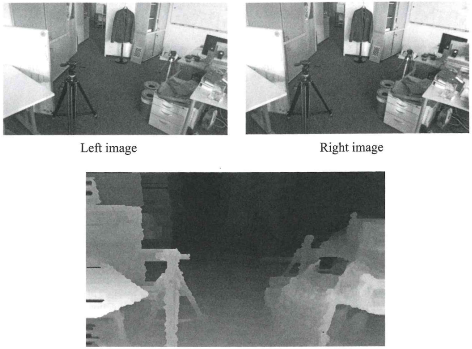
◇ Epipolar lines are collinear and parallel to horizontal axis



Rectified images (source [AMR])

Disparity maps

◇ Each pixels show disparity $(u_l - u_r)$ for corresponding points

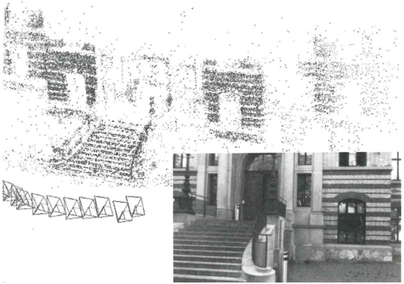


Disparity map

Disparity map: closer objects have higher disparity hence they appear lighter (source [AMR])

Structure from motion

- Unknown R and t $\Rightarrow$ estimate given correspondences
- Simultaneously estimate both 3D geometry (**structure**) and camera pose (**motion**)
- Usually non linear minimization of reprojection error



Example of structure from motion (source [AMR])