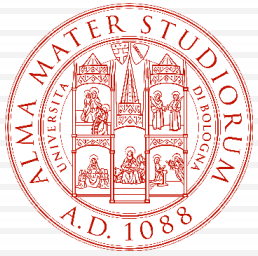


3D Computer Vision

Federico Tombari, Ph.D

federico.tombari@unibo.it

University of Bologna

Summary

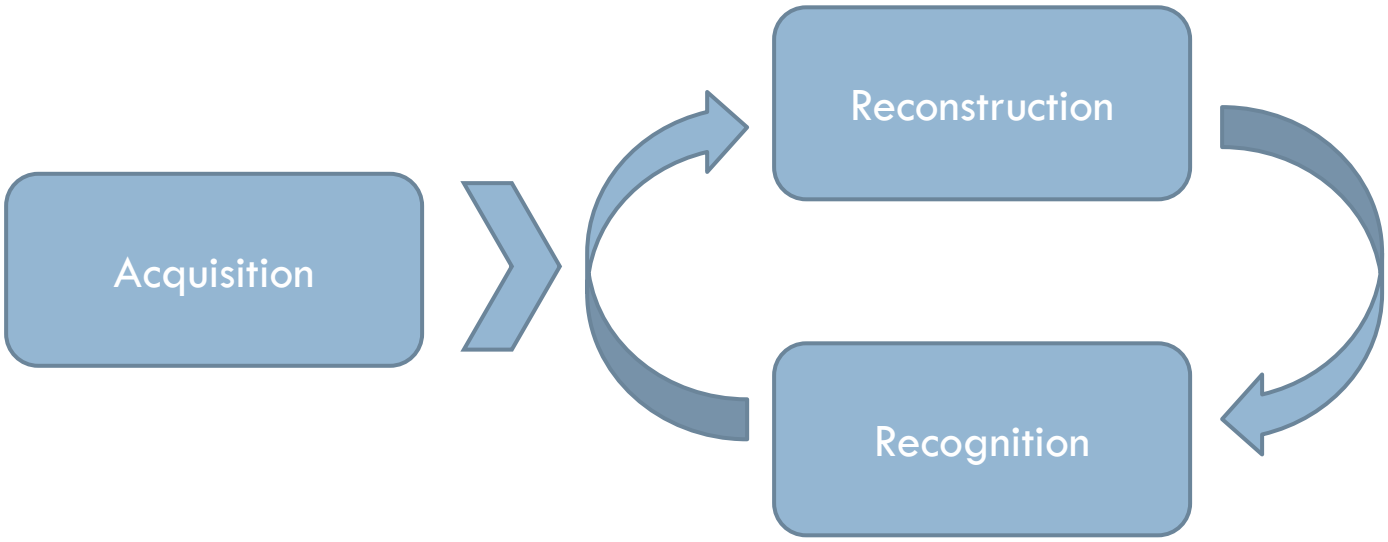
- Part 1: Sensors and Representations
- Part 2: Indepth stereo vision
- Part 3: Tasks and applications

Part 1 – Sensors and Representations

- 3D sensors
  - Active
  - Passive

- Data Representations
  - Unorganized
  - Organized
  - Range maps
  - Transformations

3D Computer Vision

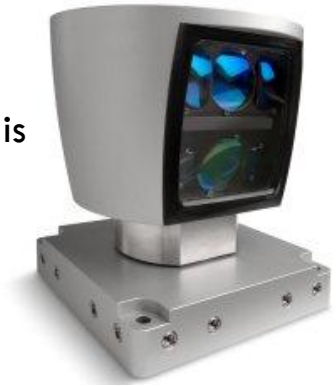


3D sensors

- Goal: create a **point cloud** of (samples from) the surface of an object/scene
  - Collection of distance measurements from the sensor to the surface
  - Distances are them transformed into 3D coordinates (x,y,z) by means of calibration information
  - Usually, 3D sensors acquire only a view of the object (**2.5D data**)
  - Some sensors also acquire information concerning color or light intensity (**RGB-D data**)
- Contact sensors
- Active sensors
  - LIDAR, rangefinders
  - Time-of-Flight cameras
  - Laser Triangulation
  - Structured light
  - Medical imaging (CT, MRI)
- Passive sensors
  - Stereo
  - Structure-from-motion
  - Shape-from-shading, shape-from-silhouette, shape-from-defocus, ..

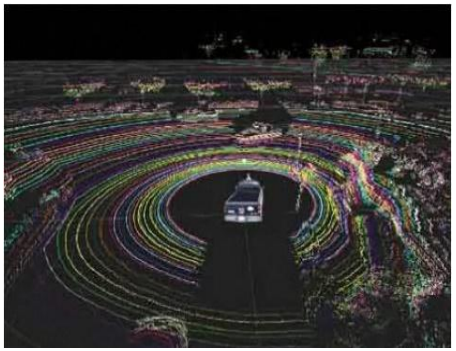
LIDAR

- **LIDAR**: Light Detection And Ranging
- A light pulse is emitted from the sensor and the round-trip time is computed as $\boldsymbol{d = \frac{ct}{2}}$
  - The higher the time, the further away the point from the sensor
- Usually visible or near-infrared light is used
- Arrays of emitters are employed together to yield a set of simultaneous range measureaments (3D slice)
- Slices can be swept using a **motor** to yield a slice array
- Pros:
  - High range (hundred meters / kms)
  - Works indoor/outdoor
  - Real-time (eg. 100 Hz slices)
- Cons
  - Accuracy is an issue due to speed of light (3 10^8 m/s -> 1 mm every 3.3 ps)
  - Cost
  - Color/intensity information is usually not provided



Velodyne

SICK LMS500



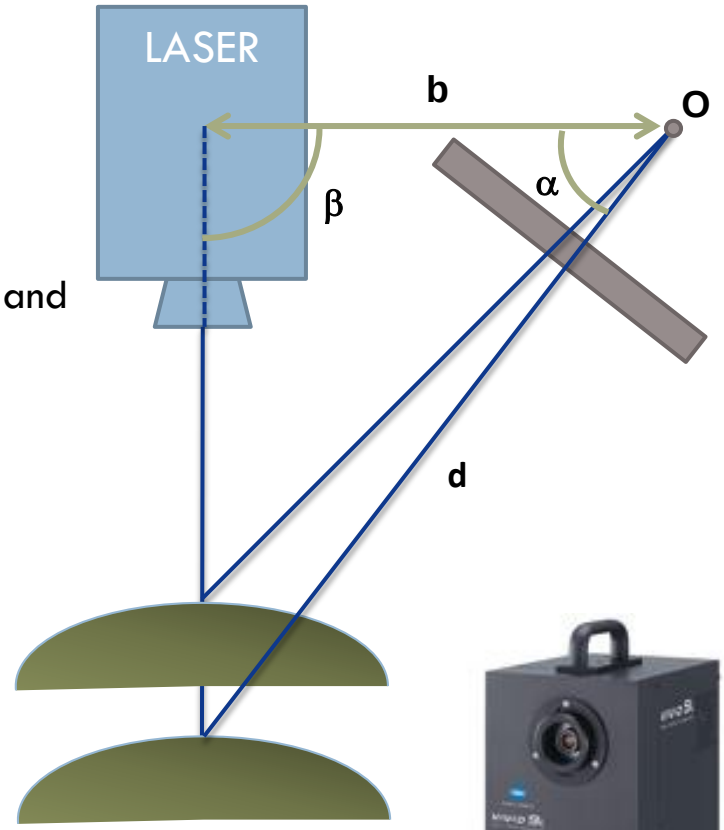
Time-of-Flight camera

- A particular LIDAR device yielding a full 2D array of range measurements
- A light pulse is emitted through an infrared illuminator; then:
  - Phase-shift measurament of the returning light pulse on each pixel (Photonic Mixed devies, Canesta Vision (now Microsoft), Swiss Ranger)
  - «range-gated imager»: each pixel has a shutter that starts closing when the light pulse is emitted; the less light received, the further away the point from the sensor (Zcam by 3DV Systems, now Microsoft)
- Pros
  - No motor needs to be employed
  - Real-time (30-100 fps)
  - Cost effective
- Cons
  - Low resolution
  - Dark, non-reflective objects are hard to be acquired
  - Hardly works under sunlight (no outdoors)
  - Multiple reflections can yield false measurements
  - Interference between different sensors

MESA SwissRanger 4000

Laser triangulation

- Laser + camera system
  - A laser dot or stripe is emitted on the scene
  - The camera locates the dot/stripe.
  - The distance is determined via triangulation of the position of the dot (emitting angle, receiving angle and baseline need to be known)



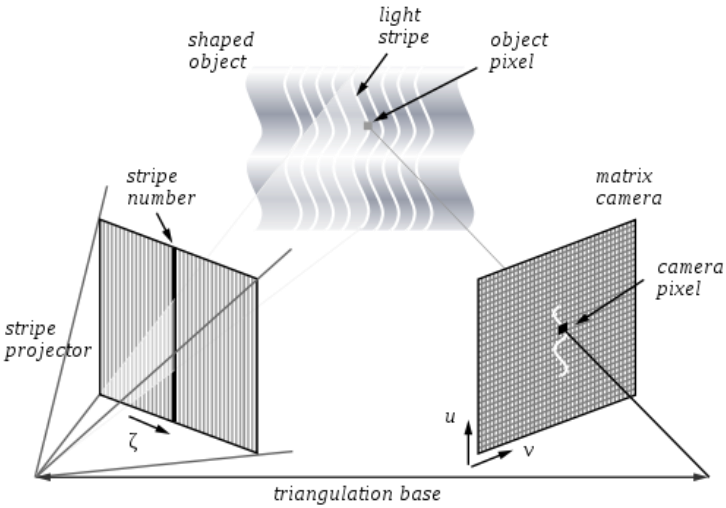
- Pros
  - Accuracy (tens of micrometers)

- Cons
  - Limited range
  - Slow scanning time (often requires static scene)
  - $$
  - No color information, needs to be paired with a color camera

Minolta Vivid 9I

Structured light

- Camera + projector system
- Similar to laser triangulation, where the laser stripe is replaced by a stripe projected by a light projector
- Projecting a set of stripes (2D pattern) allows for multiple sampling, hence a full 2D range image can be acquired at once (but problem of confusing different fringes)
- Using infrared projection and two cameras (one in infrared, one in the visible band) yields accurate RGB-D data
- Pros:
  - Relatively cheap
  - Sub-millimiter accuracy (down to tens of micrometers)
  - Real-time
- Cons:
  - Limited range
  - Hardly usable outdoor or in presence of other light sources
  - Highly dependent from the object surface characteristics (eg. reflective, translucent, ..)
  - Interference between different sensors

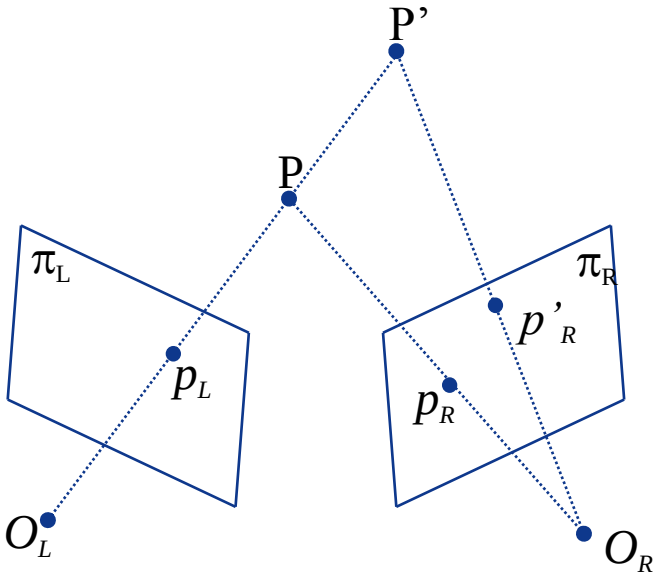


Microsoft Kinect

Stereo vision

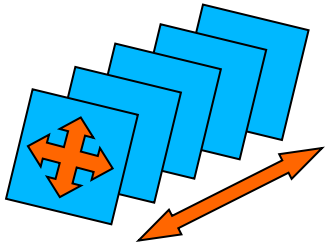
- Two (or more) cameras
- Cameras have to be sync-ed, especially in presence of non-static scenes
- Depth is retrieved via triangulation of the point projections on the two views
- **Correspondence** problem!
- Pros:
  - Cheap
  - Passive
  - Real-time
  - Color/intensity can be directly associated to range data (RGB-D)
- Cons
  - Low accuracy (tend to fail on low-textured regions, repetitive patterns and depth borders)
- A projector can help adding texture to the scene to improve accuracy on low-textured regions

Videre Design

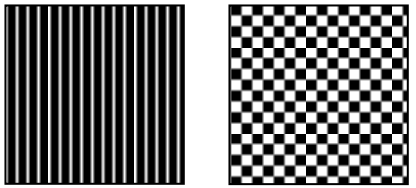


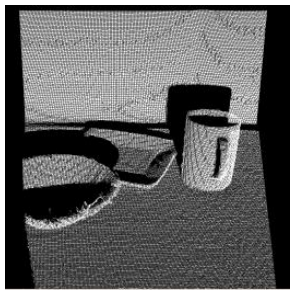
Spacetime stereo

- Stereo using **spatial** and **temporal** information[Davis 05][Zhang 03]
- To gather information, the appearance must change over time (but not the geometry!)
- A random pattern is projected to augment each frame with a different texture (no structured light, no interference)
- More accurate than standard stereo, but depth must be constant in time (static objects)



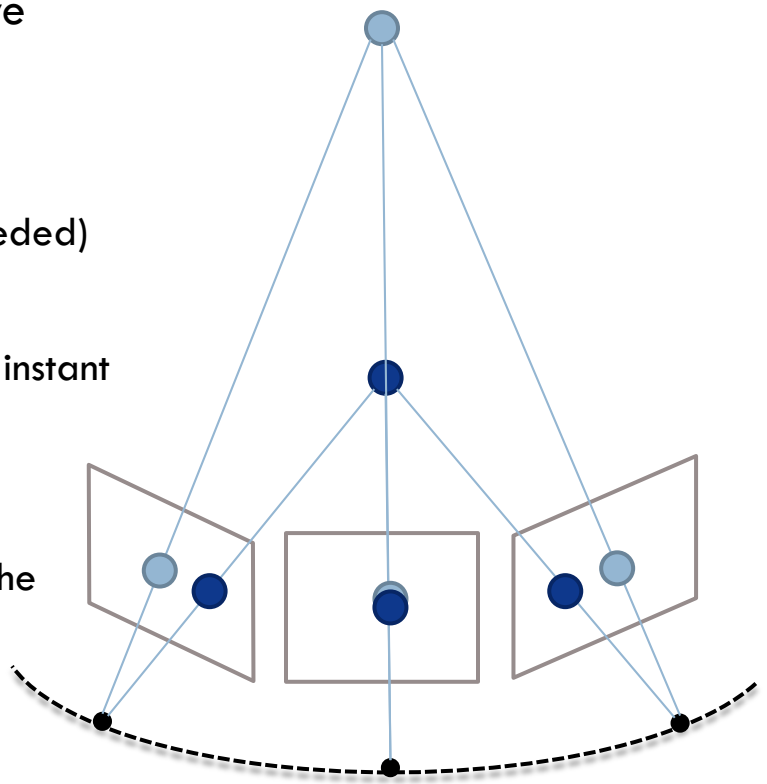
Joint spatio-temporal window





Structure-from-motion

- Monocular system
- Instead of spatially extending multiple views, they are temporally extended
- Requires either the surface or the camera to move
- Tracking and matching features
- Pros
  - Cheap and simple hardware (only one camera needed)
  - RGB-D data
  - Solving SfM also yields camera pose at each time instant
- Cons
  - Highly dependent from the available motion that the object/camera can undergo
  - Sparse depth information

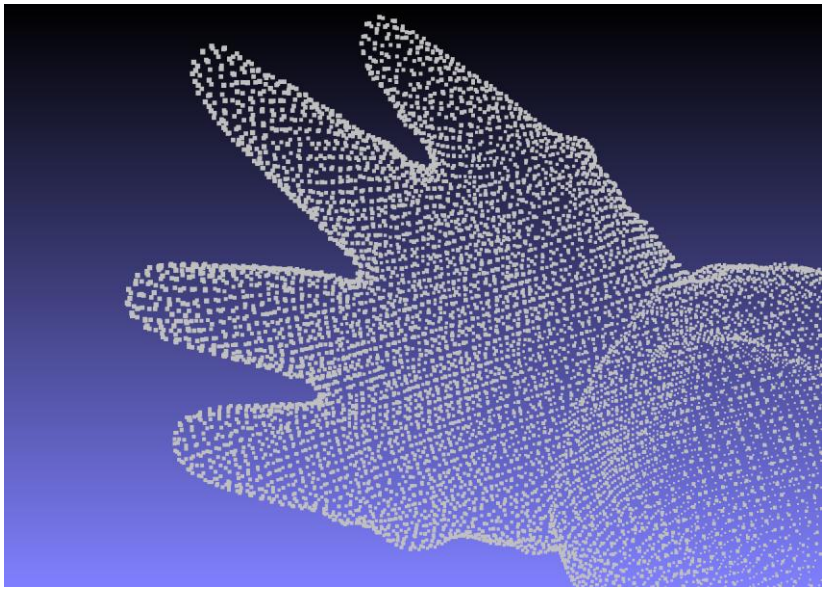


3D Data Representations

- 3D data may be represented in different formats
  - Unorganized
    - **Point cloud:** a set of vertices.
    - **Polygon mesh:** a set of vertices and their connections (topology).
  - Organized
    - **(Binary) voxelized cloud:** a 3D regular grid of (binary) density values.
    - **Range image:** a 2D regular grid (an image) of 3D coordinates.
- 3D data may be
  - **Pure 3D:** represent only the geometry of the scene
  - **RGB-D:** store the geometry of the scene as well as the intensity of the 3 color channels
- 3D data may represent
  - **Full 360° (3D) view** of an object /scene
  - **2.5D view** of an object /scene

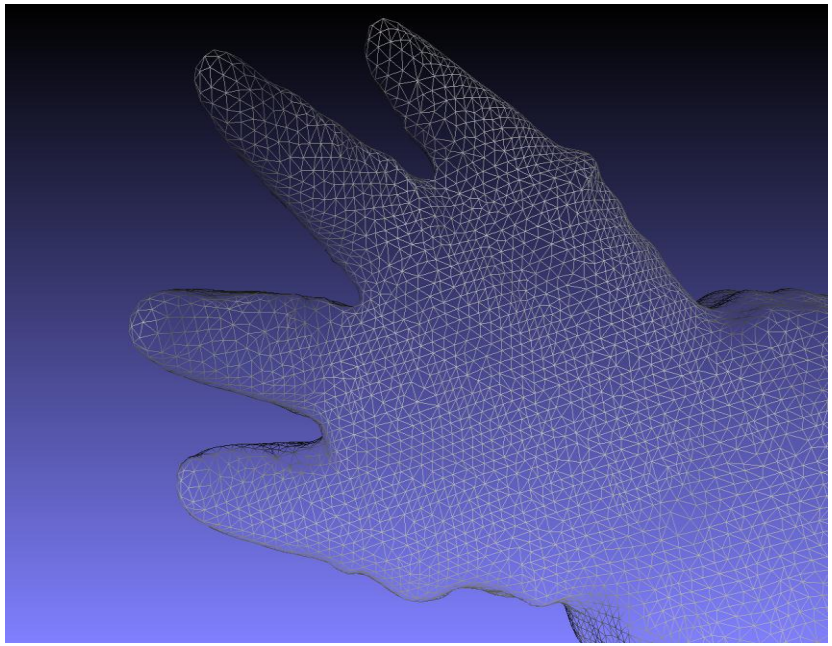
Point cloud

- Point cloud is the most common 3D data representation in computer vision.
- It is just a collection of 3D coordinates:
  - Unorganized: hence, nearest neighbor searches are costly
    - Must build an index structure (e.g. kD-tree or Oc-tree)
  - no topology
    - Inner and outer parts of the surface are hard to discriminate

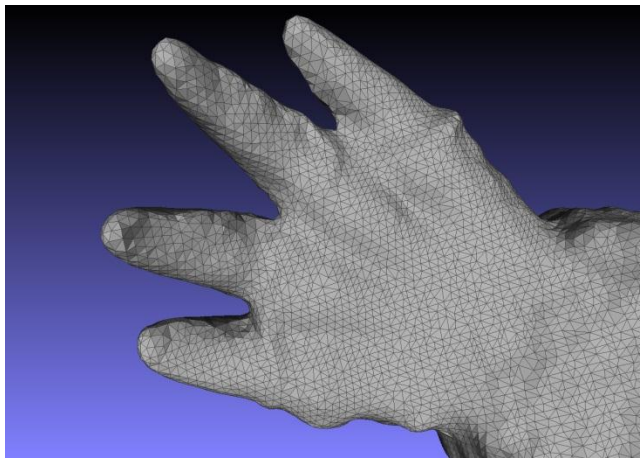


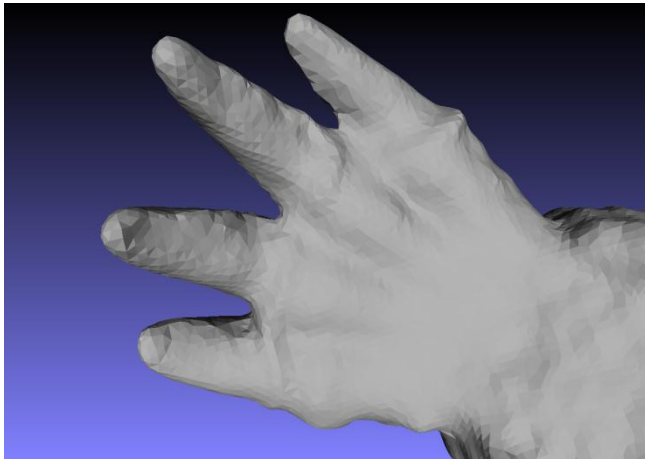
Polygon mesh

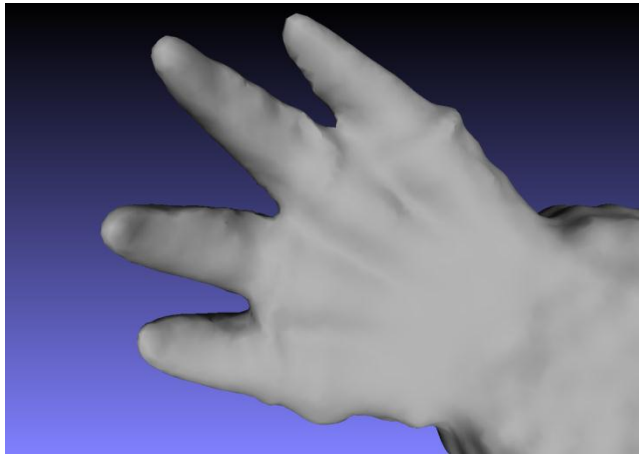
- Polygon mesh is the most common 3D data representation in computer graphics.
- It is a collection of 3D vertices and of faces connecting them:
  - Unorganized: hence, nearest neighbor searches are costly
    - Must build an index structure (e.g. kD-tree or Octree)
  - Has topology
    - Essential for rendering
    - Provides surface orientation



Polygon Mesh Rendering





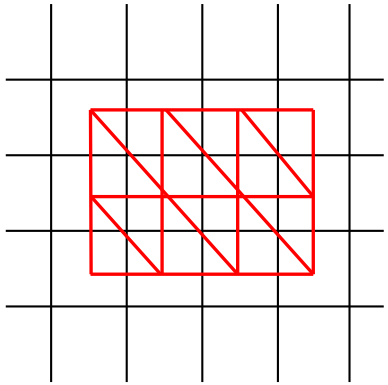


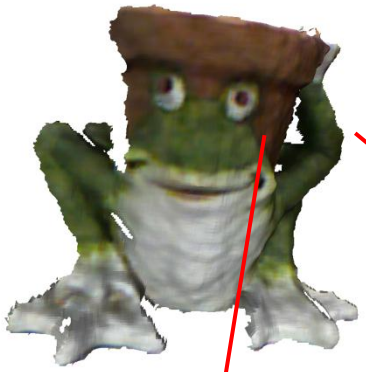
Range image

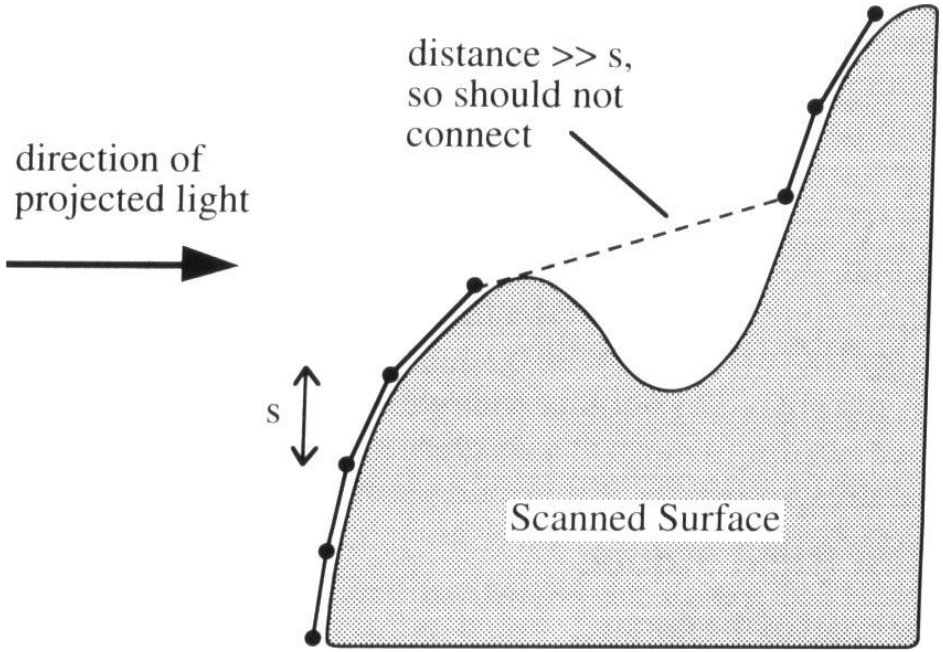
- Range image is a useful representation for efficient 3D data processing
- The term is a bit ambiguous as it is used to denote
  - a single channel image whose pixels encode the distance of the scene point from the sensor
  - **a three channels image whose pixels encode the coordinates of the scene points**
- It is an organized representation
  - Nearest neighbor searches can be carried out efficiently by exploiting the image lattice
- No topology
  - But it is easy to create it from the lattice
- **Only 2.5D views**: it provides also the sensor position (point (0,0,0)).

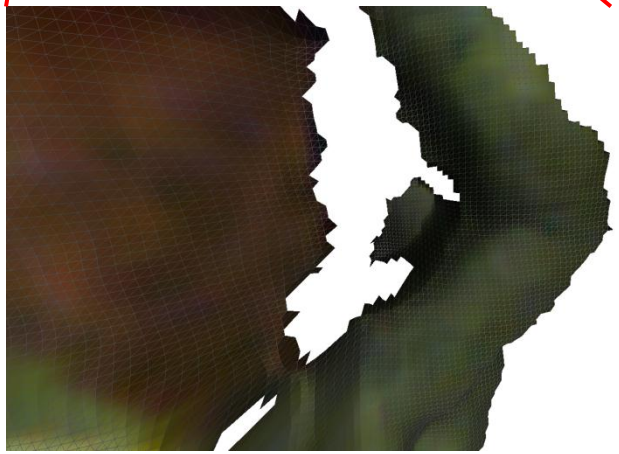
Image from Stuttgart Range Image Database (http://range.informatik.uni-stuttgart.de/)

From range images to meshes





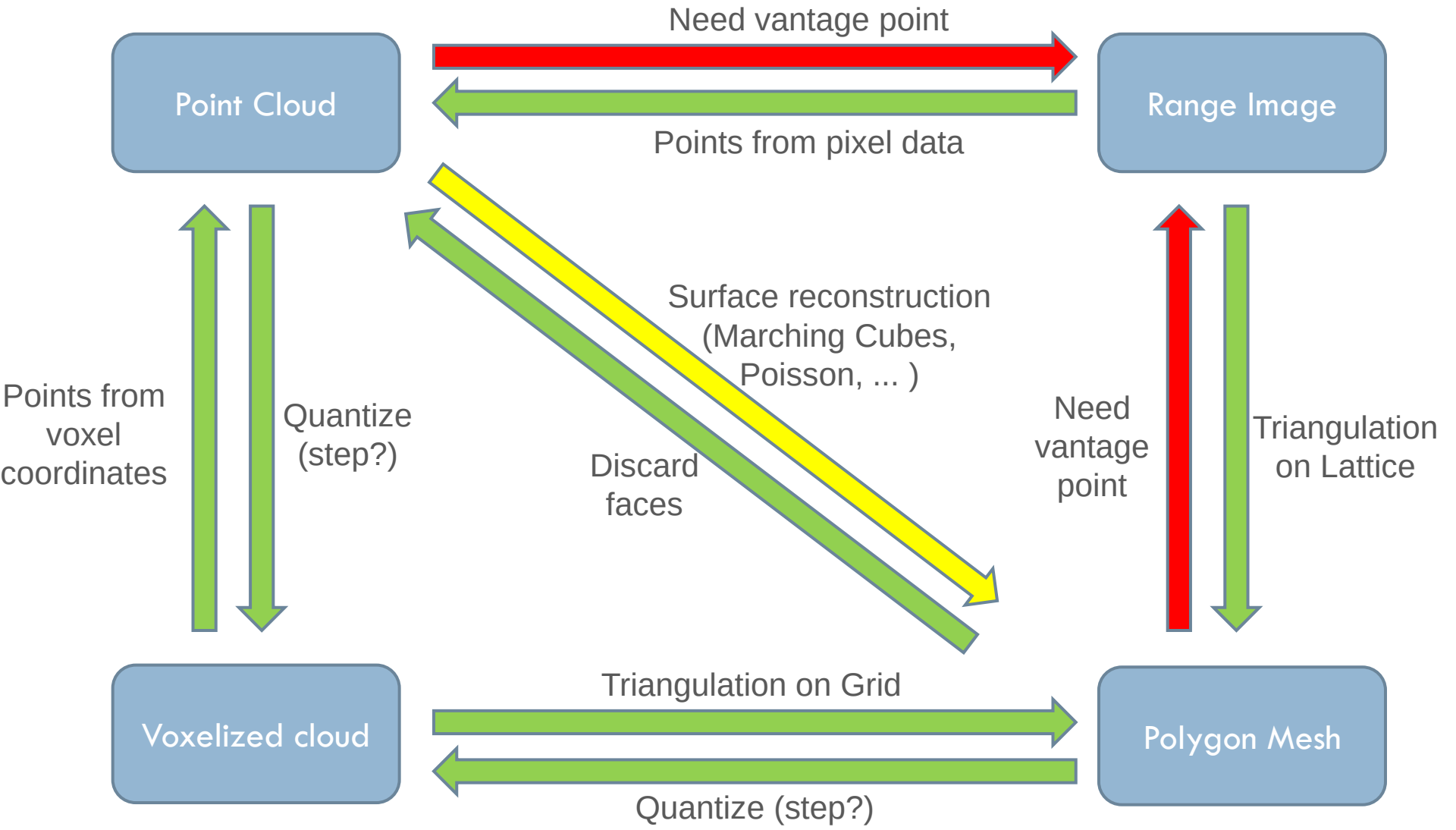




Voxelized clouds

- Voxelization is a useful representation for efficient 3D data processing
  - It is also the output of some sensors (e.g. metal detectors, body scanners)
- It represents surfaces with a regular grid of voxels (volumetric pixels)
  - 3D coordinates are implicitly defined by the index in the grid.
- It is an organized representation
  - Nearest neighbor searches can be carried out efficiently by exploiting the regular structure
- No topology
  - But it is easy to create it from the grid
- Suitable also for full 3D views.

Change representation



Bibliography

- **[Davis05]** J. Davis, D. Nehab, R. Ramamoorthi, S. Rusinkiewicz, "*Spacetime stereo: a unifying framework for depth from triangulation*," *Trans. Pattern Analysis and Machine Intelligence*, vol. 27(2), 2005
- **[Zhang03]** L. Zhang, B. Curless, S. Seitz, "*Spacetime stereo: shape recovery for dynamic scenes*" Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2003

Part 2 – Indepth stereo vision

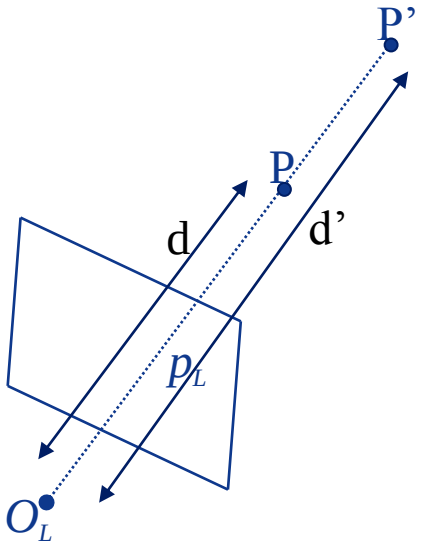
- Why stereo?
- Epipolar geometry
- Algorithms
- Real-time stereo

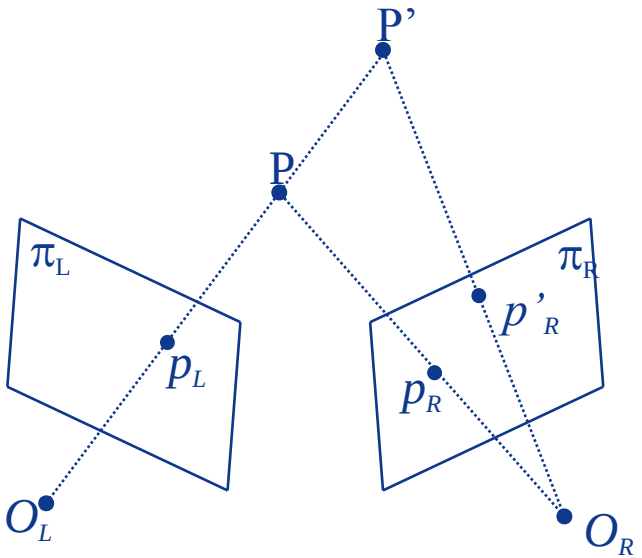
The single-view case

- By means of only one viewpoint, an observer can not determine univocally the distance between him and visible points in real world.
- Human vision relies in this case on «monocular cues» (paralellism, similarity, relative movement, ..) to estimate distances
- Doesn't work always – lots of ambiguities (see picture)

Using multiple viewpoints

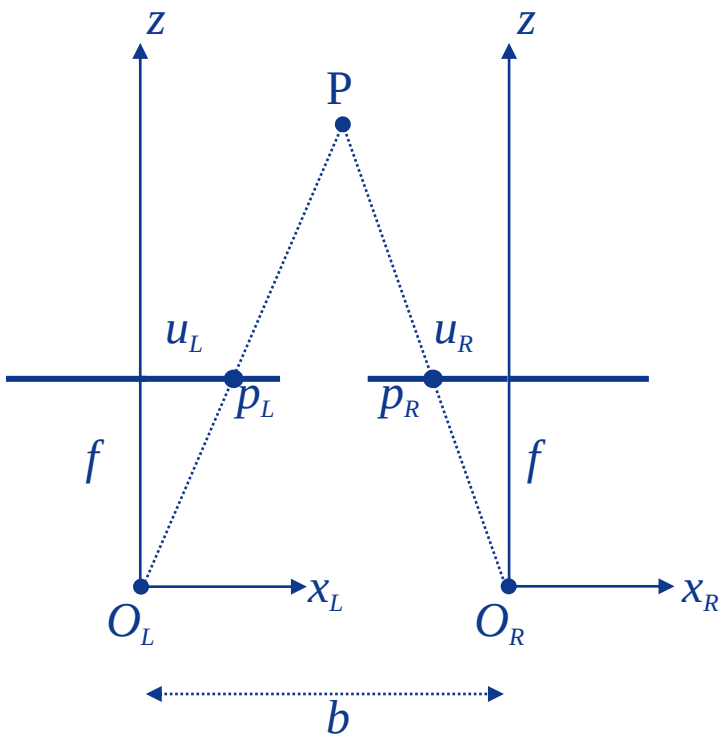
- On the other hand, using multiple views (e.g. two) ambiguities can be solved, so that it is possible to reconstruct the 3D geometry of the observed scene
- Problem solved? Not so easy! There's still the correspondence problem!





Ideal stereo setup

- Ideal stereo setup:
  - Optical axes are parallel
  - Same focal length
  - Image planes are coplanar
  - The two Reference Frames have parallel axes



- Under these conditions, the two Reference Frames only differ for a translation along the x direction equal to **b**.
- **b** stands for **baseline** and it is a characteristic parameter of a stereo setup.

Ideal stereo setup (2)

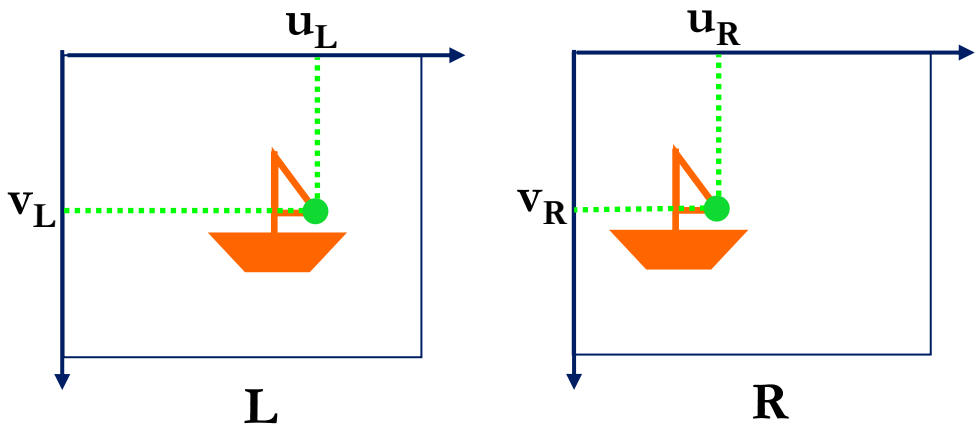
$$v_L = v_R = y \cdot f/z$$
$$u_L = x_L \cdot f/z$$
$$u_R = x_R \cdot f/z$$

$$u_L - u_R = \mathbf{d}, \textit{ disparity}$$

$$d = (x_L - x_R)\frac{f}{z} = b \cdot \frac{f}{z}$$

$$z = \frac{b \cdot f}{d}$$

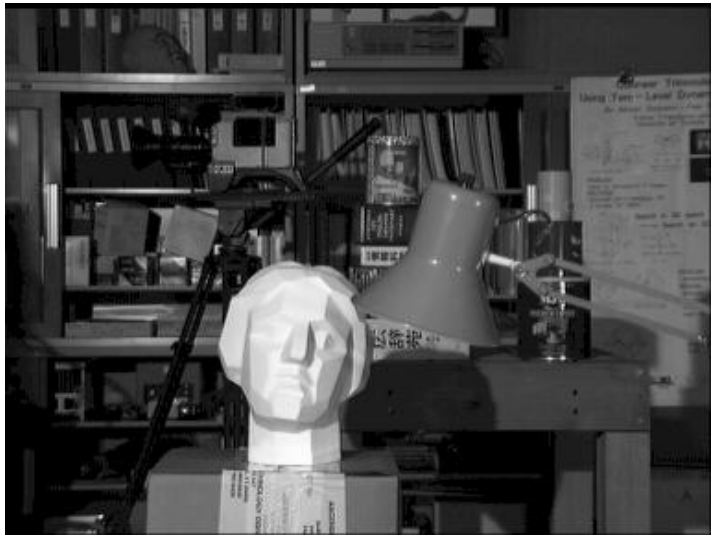
- Big disparity: close point
- Small disparity: far away point
- d=0 : infinite depth (horizon)

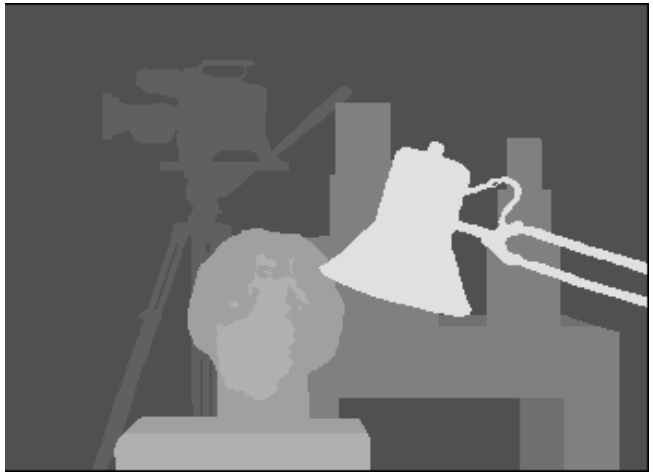


- Given two *homologous* (i.e. corresponding) points (p_L, p_R), each being the projection of the same point P on the scene, and knowing the parameters b and f, we can directly compute the depth of P
- Given p_L, how can we determine its homologous p_R (and viceversa)? **Correspondence problem**

Disparity map

- Grayscale image displaying at each pixel the associated disparity value.
- For visualization purposes, disparity values are remapped in the range [0,255].
- Color mapping is also effective in conveying scene depth understanding.





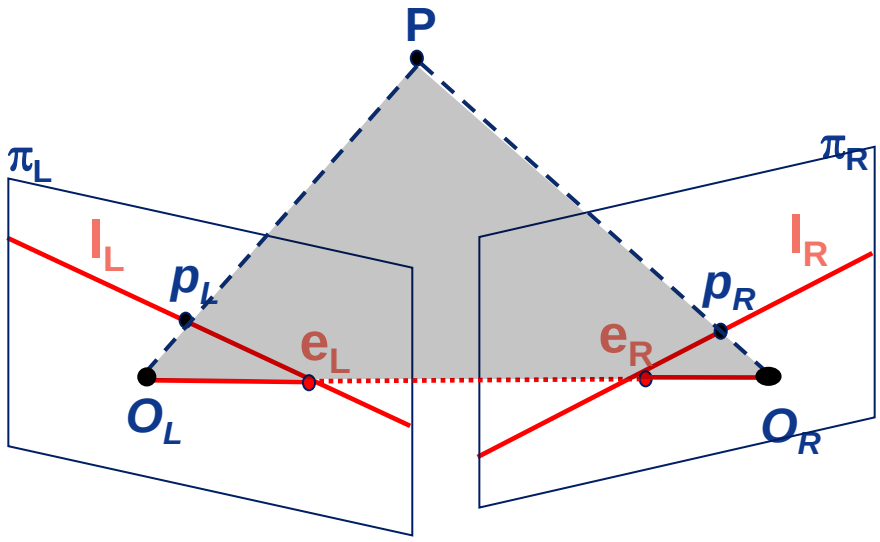
Tsukuba dataset: Left image and ground-truth disparity map.
Max Disp. 15. Scale factor: 16.

Correspondence search constraints

- To retrieve depths via stereo, a point on one image has to be associated to its homologous in the other image (***correspondence problem***)
- This problem is hard, since theoretically each point could match to all points in the other view.
- Correspondence («stereo matching») algorithms exploits constraints aimed at reducing the number of potential candidates
  - **epipolar constraint**
  - **disparity range constraint**
  - smoothness constraint
  - uniqueness constraint
  - ordering constraint

Epipolar geometry

- Epipoles are determined by the intersection of the segment lying on the optical centers with the image planes
- For different 3D points, epipolar lines rotate around the epipole
- P lies on $O_L P$, and the projection of $O_L P$ over p_R belongs to l_R
- Thus, corresponding points must lie on conjugate epipolar lines



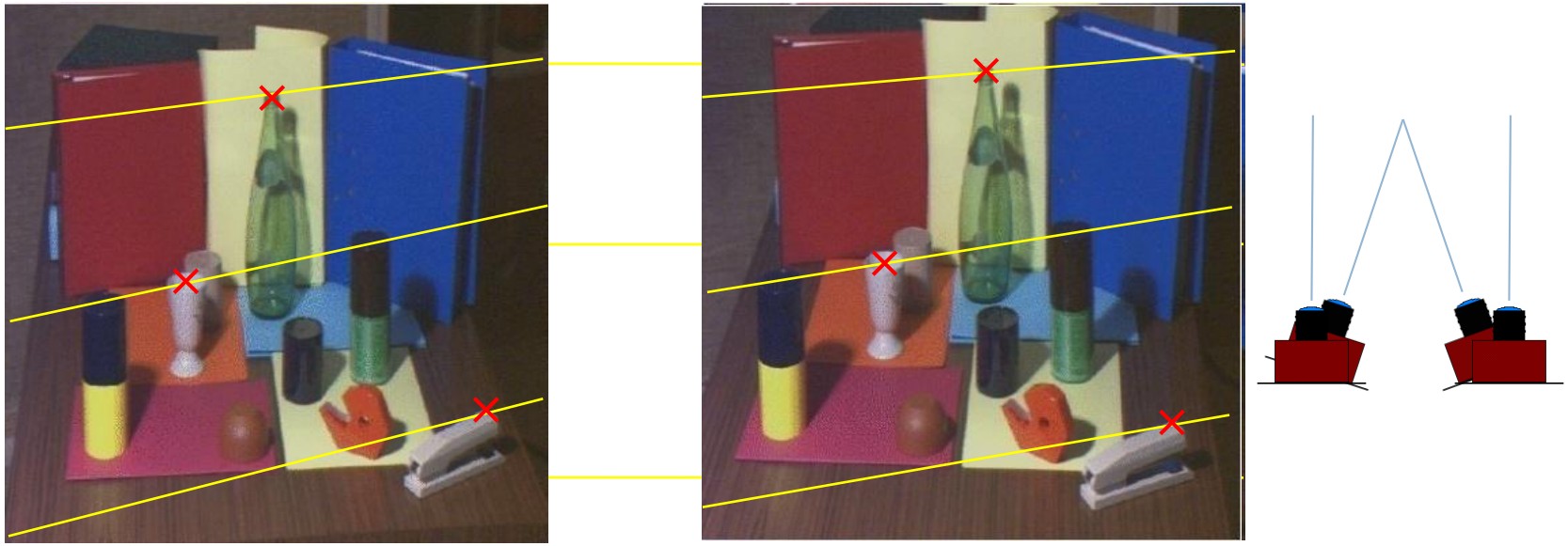
e_L, e_R: epipoles

l_L, l_R: epipolar lines

PO_LO_R : epipolar plane

Epipolar constraint

- The epipolar constraint reduces the correspondence problem from 2D to 1D (search along epipolar lines only)
- If conjugate epipolar lines are collinear and parallel to the horizontal image axis, the candidates to search are easy to determine (e.g. $p_L(i^*,j^*) \rightarrow p_R(i^*,j)$)
- This happens in the ideal stereo setup; while, in real systems, it is impossible to recreate these conditions via mechanical alignment
- We need to obtain it via software: stereo rectification [Trucco 98]



Stereo calibration and rectification

- Stereo camera calibration:
  1. Calibration of each view:
     a) Estimate the **K matrix** (intrinsic parameter matrix, or camera calibration matrix):

$$K = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

- $f_u = f / pw$
- $f_v = f / ph$
- u_0, v_0: pixel coord. of image center
- s = skew factor

     b) Estimate the **lens distortion parameters:**

        radial distortion (k_1, k_2, k_3, ..)

        tangential distortion (τ_1, τ_2, ..)

  2. Estimate the **extrinsic parameters** related to both views:

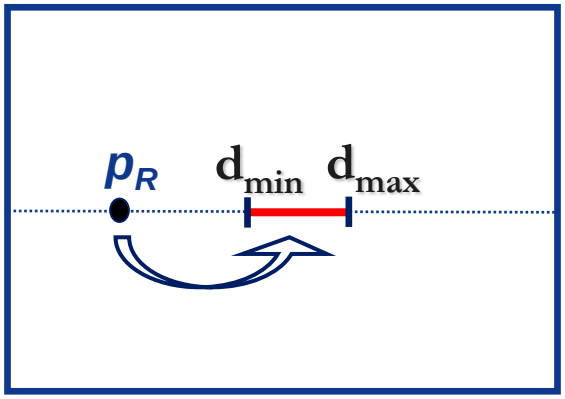
     Rotation matrix **R** (3x3) and translation vector **T** (1x3) which define the transformation with respect to an absolute 3D Reference Frame

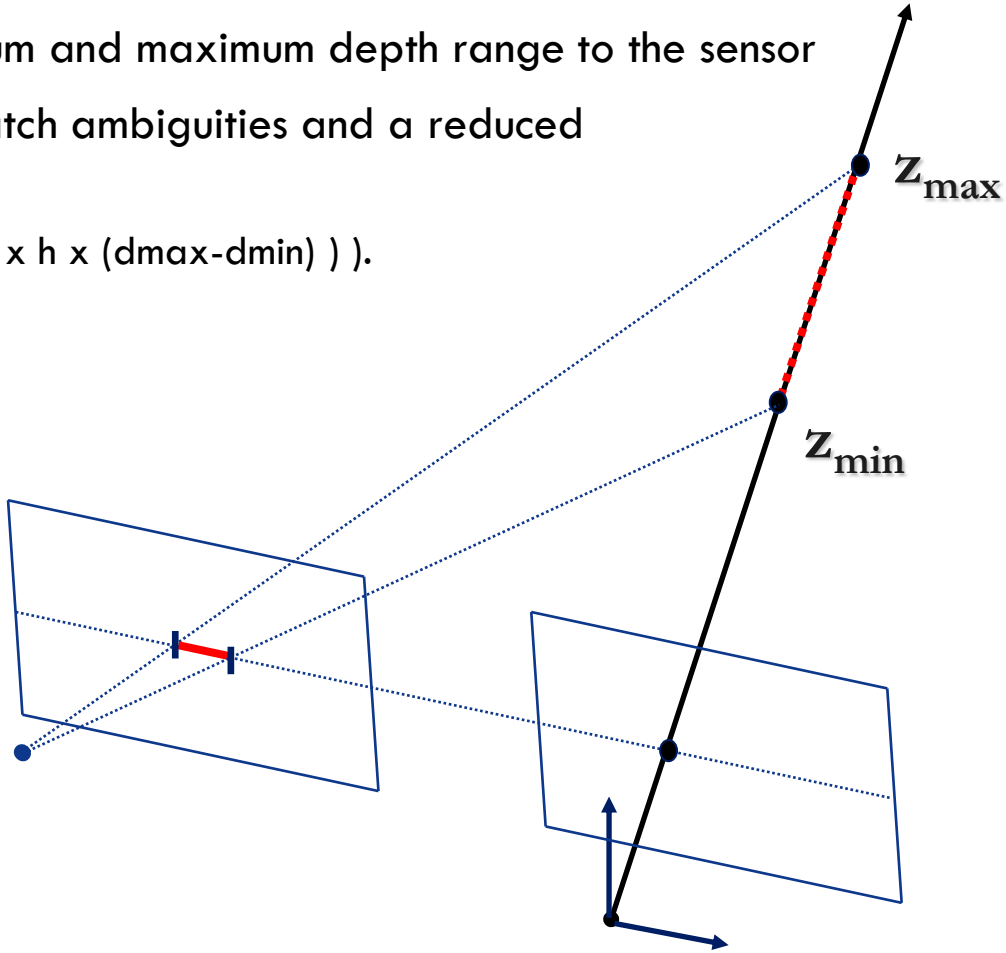
This way, we are able to obtain:

- Perspective projection matrices (**ppm**): $P = K \cdot [R;T]$
- **Rectification homographies:** 3x3 matrices which transform each view to a new 3D space where each view are **rectified** with respect to each other(*warping*).

Disparity range constraint

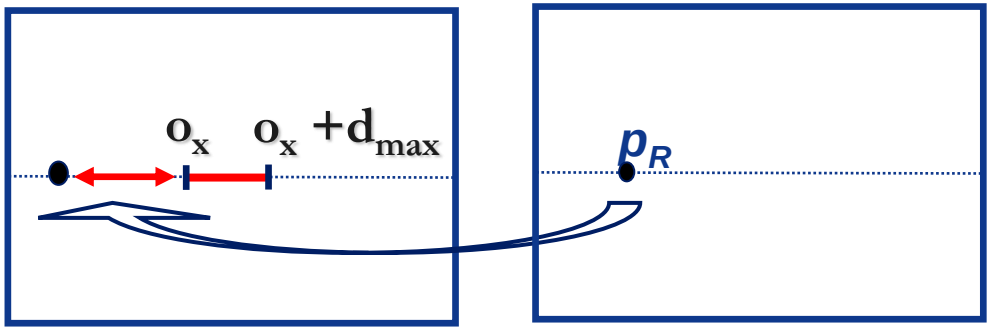
- The possible disparity values are reduced from all possible candidates in the scanline to [dmin dmax].
- This is equivalent to setting a minimum and maximum depth range to the sensor
- As a consequence, there are less match ambiguities and a reduced computational burden
  - (complexity of most algorithms: O(w x h x (dmax-dmin))).
- Disparity range:
  - [dmin, dmax] $\Leftrightarrow$ [zmax, zmin]





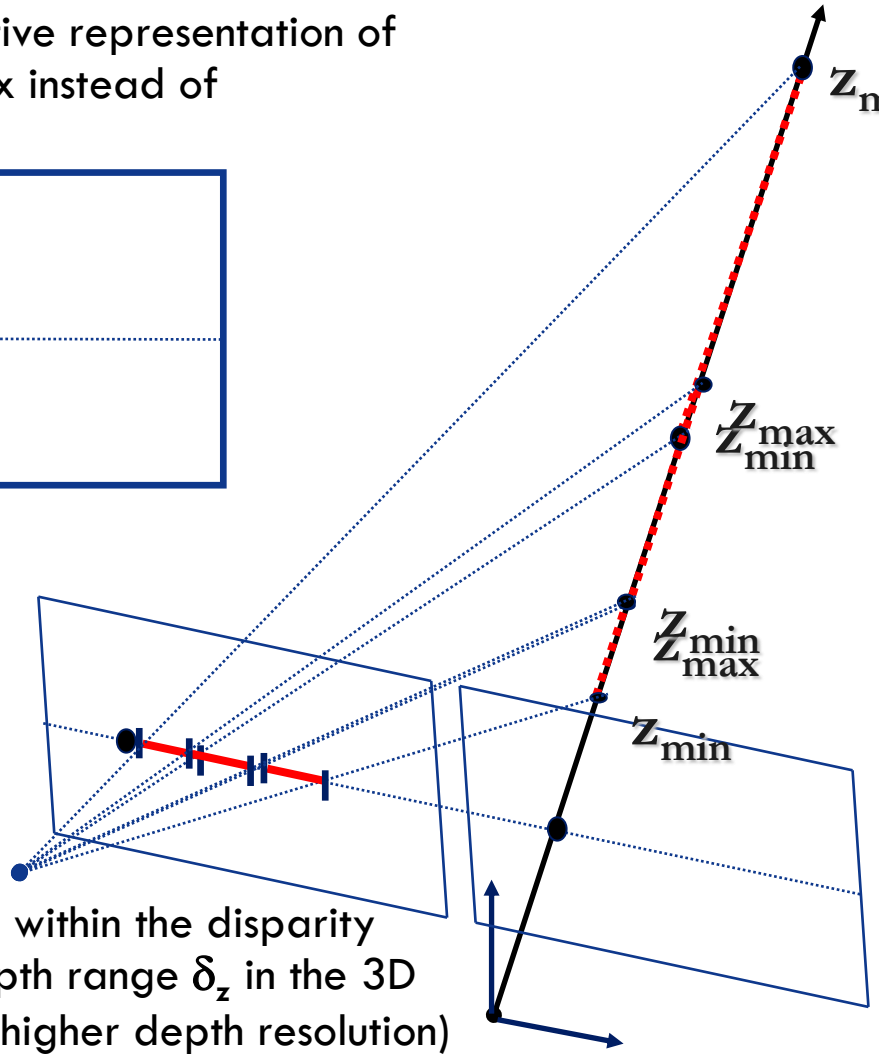
Horizontal offset

- The horizontal («x») offset is an alternative representation of the disparity range constraints (ox/dmax instead of dmin/dmax)



- $[o_x, o_x + d_{max}] \Leftrightarrow [z_{max}, z_{min}]$

- **Horopter (stereo depth of field):**
  is defined as the range $[z_{min}, z_{max}]$



- **Note:** with the same number of elements within the disparity range and by varying the offset, the depth range δ_z in the 3D scene is modified (higher o_x, smaller δ_z, higher depth resolution)

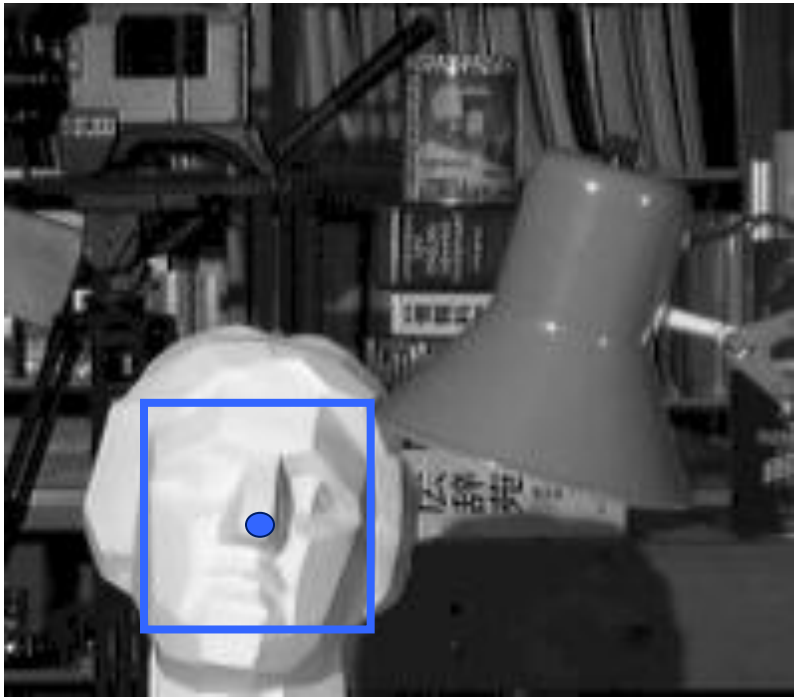
Stereo taxonomy [Scharstein02]

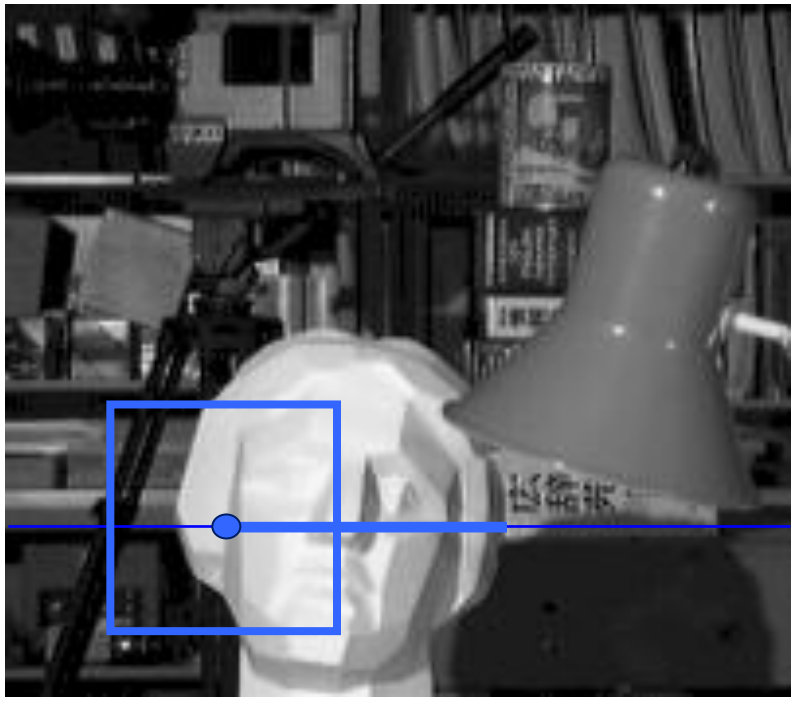
Area-based (dense)

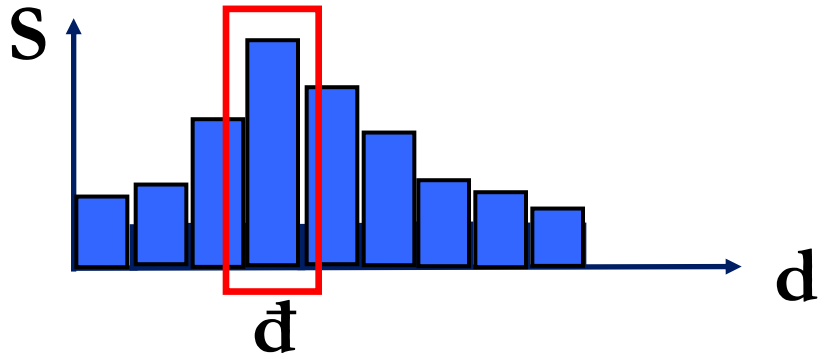
- **Local methods:** the correspondence problem is solved at each point by employing only pixels in the same neighborhood of the point.
  - fast
  - Problems along low-textured areas/ depth borders
- **Global methods:** the correspondence problem is interpreted as a cost minimization problem computed on a graph, which takes into account all pixels of the image. Each disparity thus depends from all other pixels in the image
  - Better results in terms of accuracy of the retrieved disparity map
  - Higher computational cost
- **Semi-global methods:** the cost problem is defined on a graph computed not on all image pixels (typically loopy graph) but on each scanline (acyclic graph)
  - Good accuracy/speed trade-off

Feature-based (sparse, only computed on features)

Block-based stereo

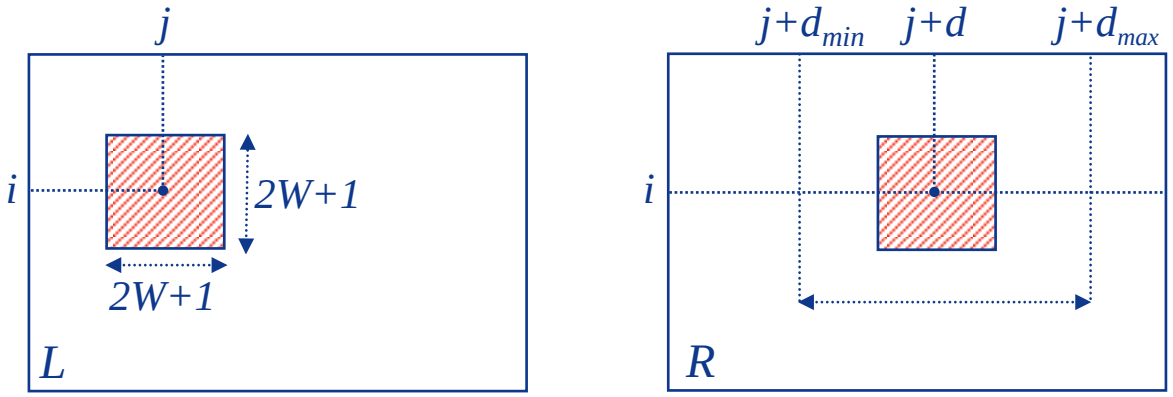






WTA approach
(Winner-Take-All)

Block-based stereo



- Uses squared windows of size 2W+1
- At each point of coordinates (i,j) of the reference image L, a matching function is computed among all pairs formed by the candidates belonging to the disparity range

$$\forall (i, j) \in L, \forall d \in [d_{\min}, d_{\max}]:$$

$$S(i, j, d) = \sum_{m=-W}^{W} \sum_{n=-W}^{W} \psi\big(L(i+m, j+n), R(i+m+d, j+n)\big)$$

Matching measures

- Matching measures are typically either similarity (affinity)-based or dissimilarity (distoriont)-based. The former are generally based on cross-correlation, while the latter are derived from the L_p -norm distance:
- One of the most used similarity measures is the **NCC (Normalised Cross-Correlation):**

$$NCC(i,j,d) = \frac{L(i,j) \circ R(i,j,d)}{\|L(i,j)\|_2 \cdot \|R(i,j,d)\|_2} = \frac{\sum_{m=-W}^{W} \sum_{n=-W}^{W} L(i+m, j+n) \cdot R(i+m+d, j+n)}{\sqrt{\sum_{m=-W}^{W} \sum_{n=-W}^{W} L(i+m, j+n)^2} \cdot \sqrt{\sum_{m=-W}^{W} \sum_{n=-W}^{W} R(i+m+d, j+n)^2}}$$

- Being a similarity measure, the disparity value to choose for the current point corresponds to the maximum of the NCC values computed on the points within the disparity range:

$$d(i,j) = \underset{d \in [d_{\min}, d_{\max}]}{\arg\max} \{NCC(i,j,d)\}$$

- It holds the property of being invariant to constant linear transformations between the two images

$$L = \alpha \cdot R$$

Matching measures (2)

- Traditional dissimilarity measures are derived from the L_p norm of the difference between the two vectors L(i,j) and R(i,j,d), representing the two squared window on which the disparity is being evaluted:

$$\delta_p(i,j,d) = \left\| L(i,j) - R(i,j,d) \right\|_p^p = \sum_{m=-W}^{W} \sum_{n=-W}^{W} \left| L(i+m, j+n) - R(i+m+d, j+n) \right|^p$$

- The most commonly used dissimilarity measures are the **SSD (Sum of Squared Differences)**, choosing p=2:

$$\delta_2(i,j,d) = \left\| L(i,j) - R(i,j,d) \right\|_2^2 = \sum_{m=-W}^{W} \sum_{n=-W}^{W} \left(L(i+m, j+n) - R(i+m+d, j+n) \right)^2$$

- and the **SAD (Sum of Absolute Differences)**, with p=1:

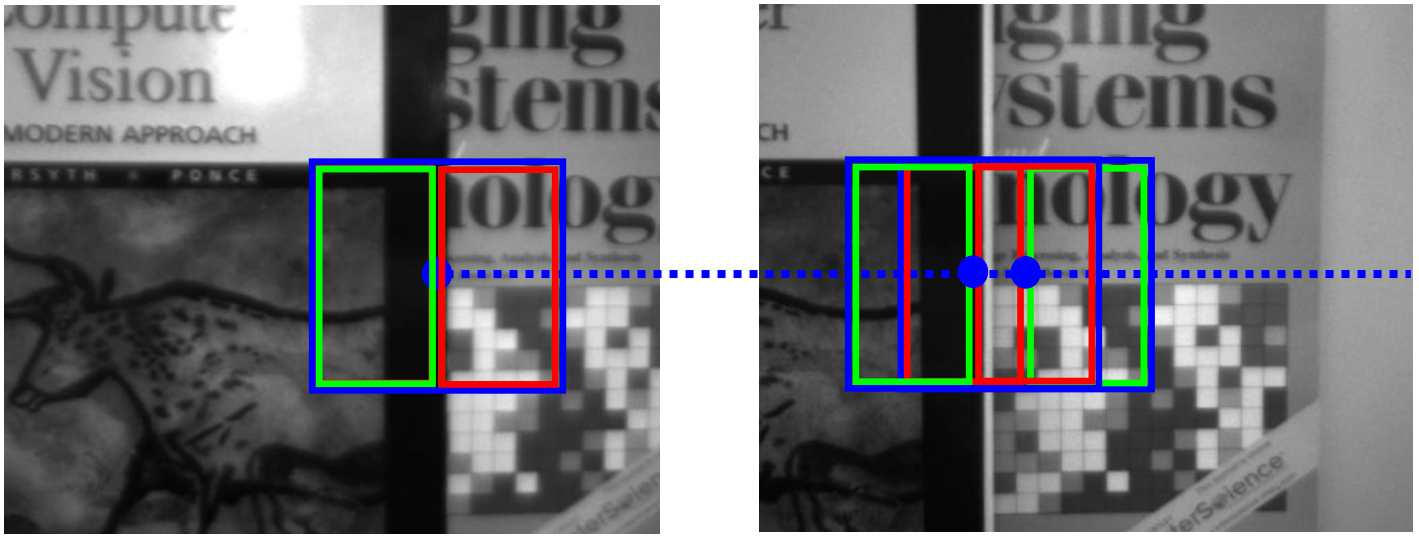
$$\delta_1(i,j,d) = \left\| L(i,j) - R(i,j,d) \right\|_1^1 = \sum_{m=-W}^{W} \sum_{n=-W}^{W} \left| L(i+m, j+n) - R(i+m+d, j+n) \right|$$

- Being dissimilarity measures, the disparity value associated to the current point corresponds to the minimum of the SSD/SAD values computed on the points within the disparity range

$$d(i,j) = \arg\min_{d \in [d_{\min}, d_{\max}]} \left\{ \delta_p(i,j,d) \right\}$$

Main problems

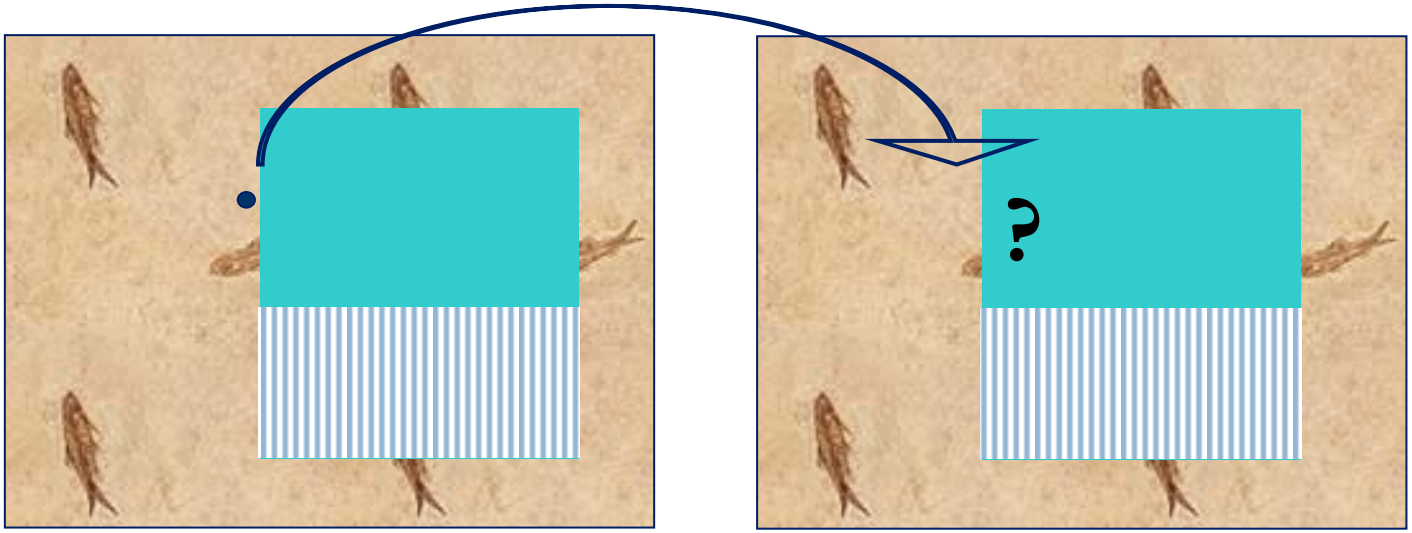
- **Disparity variations within the squared window:** the block-based algorithm relies on the assumption that disparity is constant within the window (fronto-parallel surfaces). This is not true in real scenes, anytime we are considering points along depth borders (occluding boundaries).

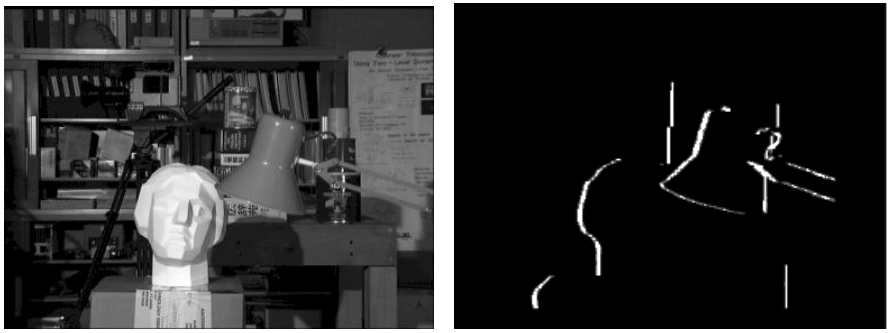


- A window covering regions at different depths doesn't have a corresponding window in the other image since these regions will either be non-contiguous or occluding each other.
- This implies higher ambiguity in determining the maximum/minimum of the matching measure being used and, thus, inaccuracies in localizing the occluding boundaries (smearing/fattening problem).

Main problems (2)

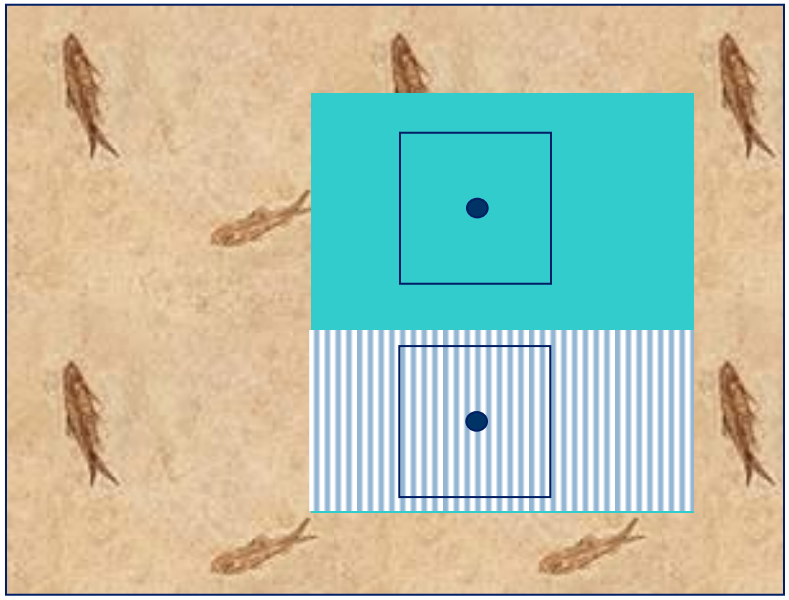
- **Occlusions:** areas of the scene visible only in the refence image (occluded by foreground objects)

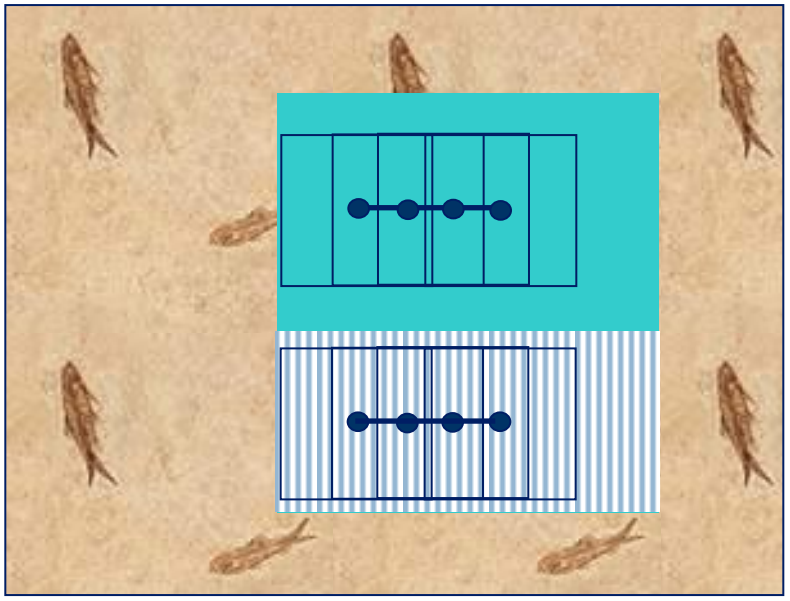




Main problems (3)

- **Low textured regions** and **repetitive patterns along epipolar lines**





- Choice of the window size: a bigger window has a higher SNR (the window «captures» more useful appearance information especially on low-textured regions), but lower spatial resolution (details are not retrieved).
- The higher the window size, the more probable the constant disparity assumption will be violated.

Facing photometric distortions

- Photometric distortions are common due to illumination differences in the two images (due to non-Lambertian surfaces, specularities..) and different intrinsic camera parameters (gain, exposure, ..)
- To compensate for these differences, typical solutions are:
  - Band-pass filters (eg. by means of the **LOG** – Laplacian of Gaussian operator) applied on both images. This is typically done as a pre-processing step before stereo matching
  - **Subtraction of the mean value** computed on a squared window (e.g. [Di Stefano04, Faugeras03]) (also a pre-processing step)
  - More **robust matching measures** towards photometric distortions

ZNCC, ZSSD, ZSAD

- A transformation that can be applied to image with the aim of increasing the robustness of stereo matching in presence of photometric distortions is subtracting to each point P the average intensity value computed on a squared window centered in P.
- Also this approach compensates constant intensity offset variations.
- In practice, this transformation can be either done as a pre-processing step prior to stereo matching, or can be included in the matching measures typically employed by stereo algorithms.
- When applied to Lp-norm based measures, we obtain the *Zero-mean Sum of Squared Differences* (ZSSD) and the *Zero-mean Sum of Absolute Differences* (ZSAD):

$$ZSSD(i,j,d) = \sum_{m=-W}^{W}\sum_{n=-W}^{W}\left(\left(L(i+m,j+n)-\mu_L(i,j)\right)-\left(R(i+m+d,j+n)-\mu_R(i+d,j)\right)\right)^2$$

$$ZSAD(i,j,d) = \sum_{m=-W}^{W}\sum_{n=-W}^{W}\left|\left(L(i+m,j+n)-\mu_L(i,j)\right)-\left(R(i+m+d,j+n)-\mu_R(i+d,j)\right)\right|$$

ZNCC, ZSSD, ZSAD (2)

- Applying the same approach to the NCC measure, we obtain the *Zero-mean Normalised Cross-Correlation (ZNCC)*:

$$ZNCC(i,j,d) = \frac{\sum_{m=-W}^{W}\sum_{n=-W}^{W}\left(L(i+m,j+n)-\mu_L(i,j)\right)\cdot\left(R(i+m+d,j+n)-\mu_R(i+d,j)\right)}{\sqrt{\sum_{m=-W}^{W}\sum_{n=-W}^{W}\left(L(i+m,j+n)-\mu_L(i,j)\right)^2}\cdot\sqrt{\sum_{m=-W}^{W}\sum_{n=-W}^{W}\left(R(i+m+d,j+n)-\mu_R(i+d,j)\right)^2}}$$

- ZNCC is thus invariant to affine intensity transformations between the two images:

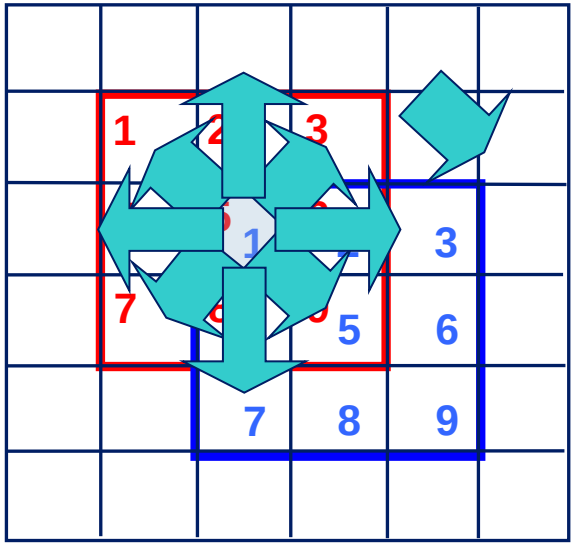
$$L = \alpha \cdot R + \beta$$

“Interest operators”

- Only certain «interest points» are selected (the others are discarded) on which to compute stereo matching
- These points are selected *a-priori* as reliable for correspondence matching (e.g. characterized by enough texture)
- E.g.: Moravec operator [Hanna85,Moravec79]: based on the intensity variation of a pixel P over a neigborhood N(P) (3x3..11x11).
- 8 directional variations are being computed as the sum of the squared differences among adjacent pixel along 8 directions

$$\sigma_1(P) = \sum_{(i,j) \in N(P)} \left(I(i,j)-I(i+1,j+1)\right)^2$$

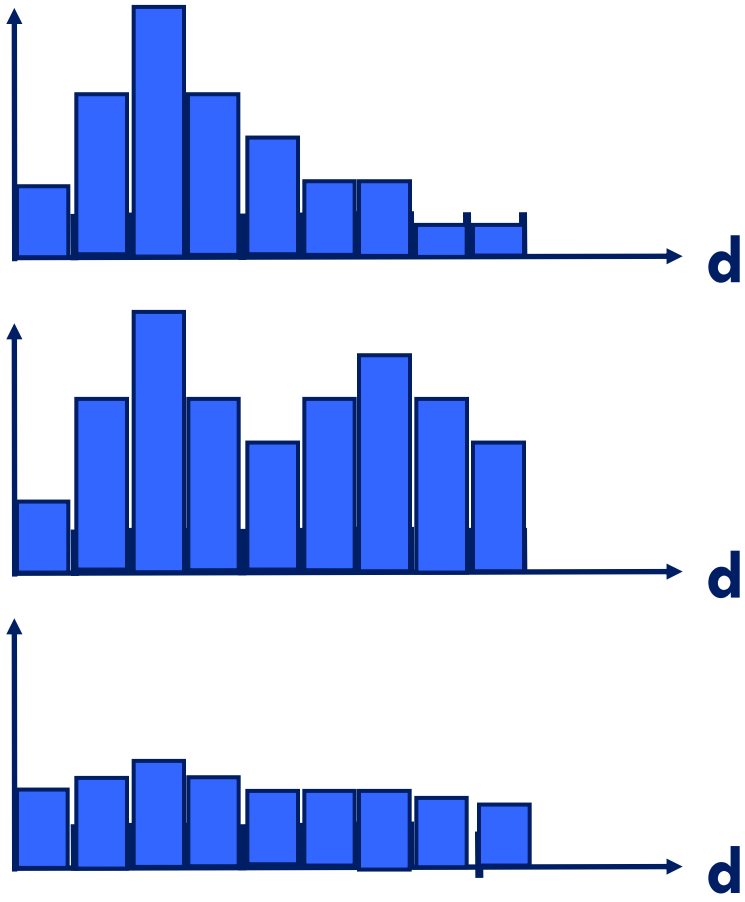
$$\sigma_1(P), \ldots, \sigma_8(P)$$



- Intensity variation: $\sigma = \min (\sigma_1(P), \ldots, \sigma_8(P))$
- Interest points are those above a threshold

Disparity filtering methods

- Analysis of the disparity surface at each stereo correspondence



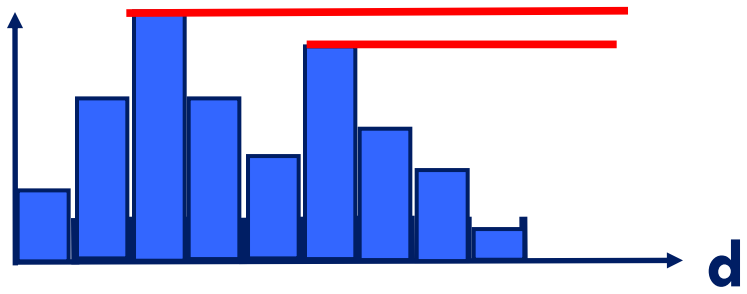
Reliable correspondence

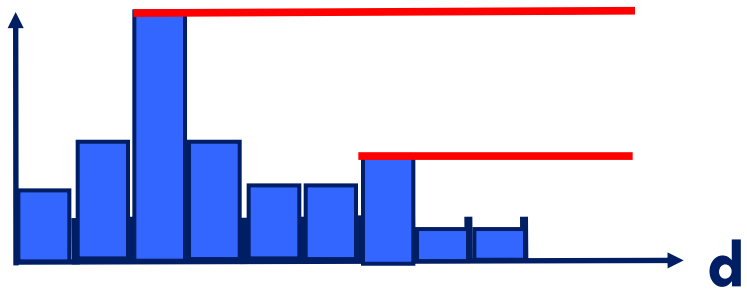
Depth border

Low-textured region

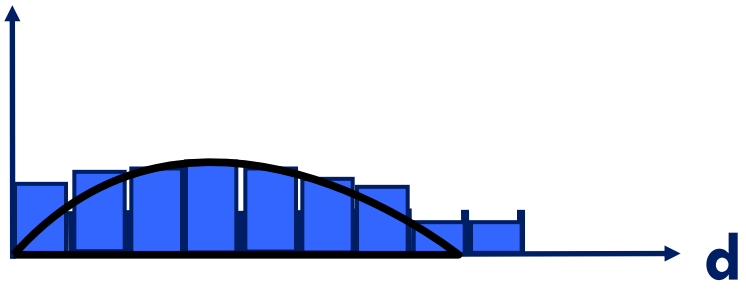
Disparity filtering methods (2)

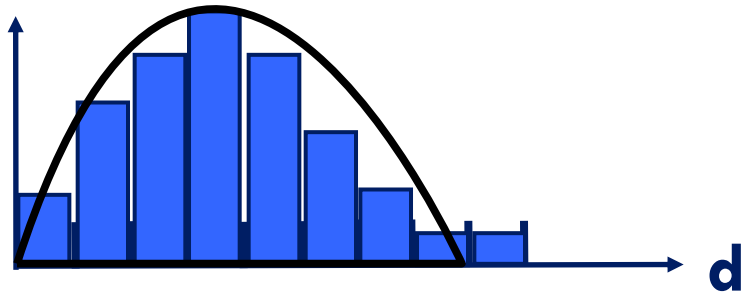
- **Ratio between best and second-best maxima**





- **Analysis of the peak spread**

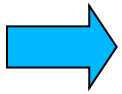




Disparity filtering methods (3)

- **Ratio between best and second-best maxima**

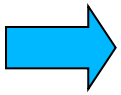
$$\frac{f\left(x_{loc-\max}\right)}{f\left(x_{\max}\right)} > TH_{RATIO}\left(\in [0,1]\right)$$



The correspondence is invalid

- **Analysis of the peak spread**

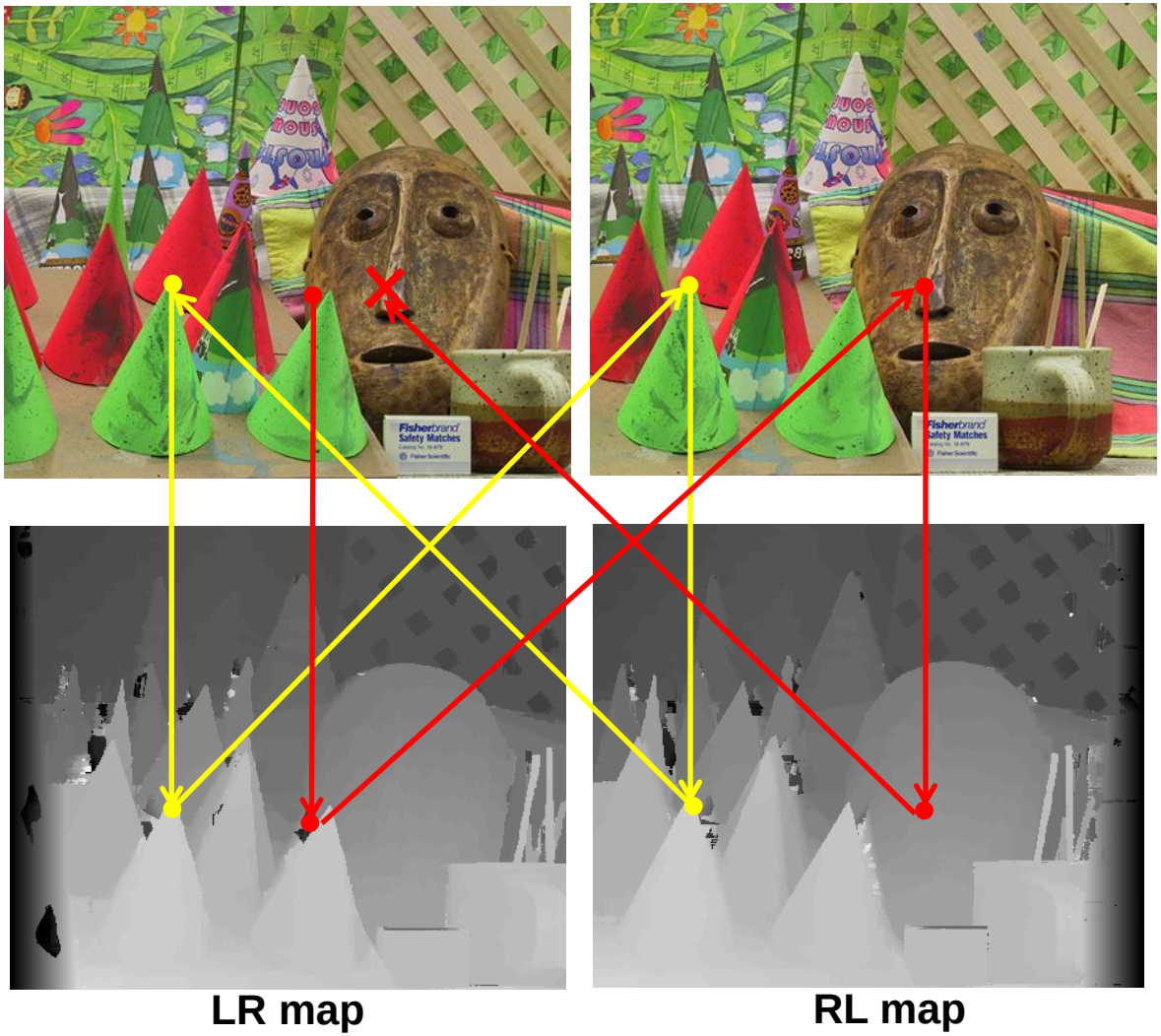
$$\frac{f_{-}^{'}(x_{\max}) + f_{+}^{'}(x_{\max})}{2} < TH_{PEAK}$$



The correspondence is invalid

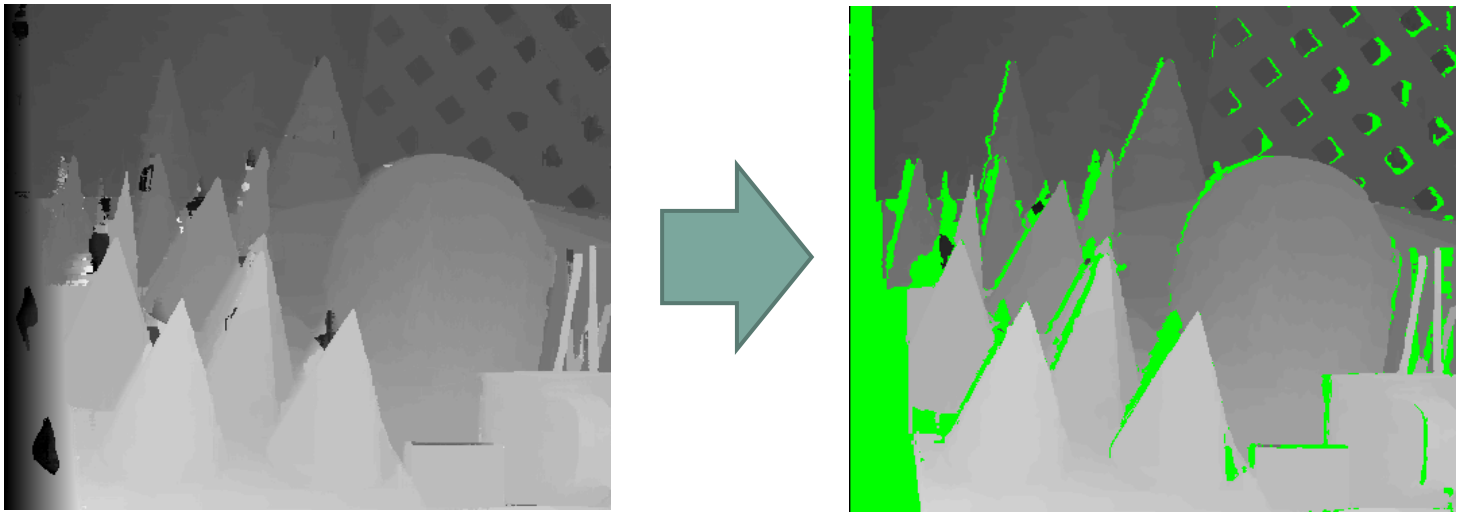
Left-Right consistency check

- Compute **two** disparity maps using, as reference image, both the Left Image (***LR map***) and the Right image (***RL map***)
- Validate correspondences only if coherent over the both views:
- ***If, according to the LR map, p_R is the best match for p_L, then for the RL map, p_L must be the best match for p_R [Fua93].***

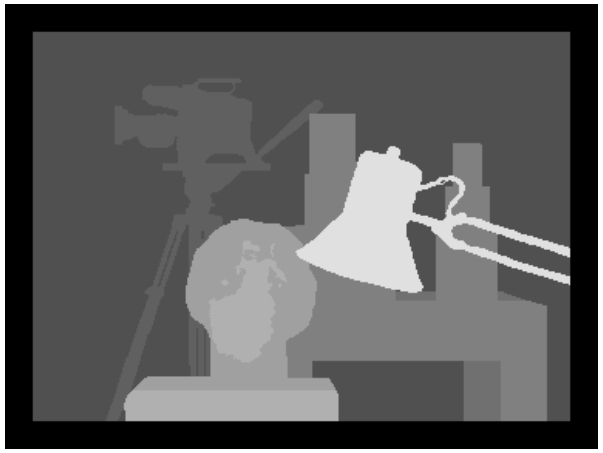


Left-Right consistency check (2)

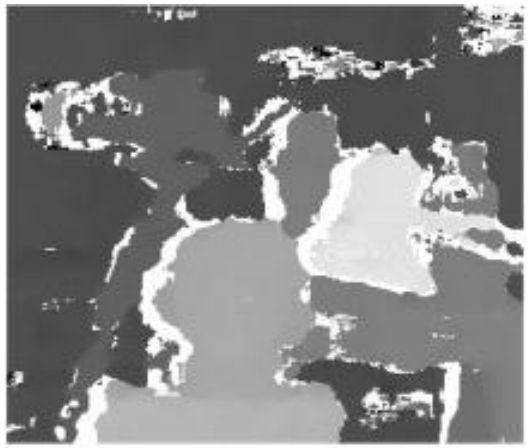
- Useful to filter out some correspondence errors due to **occlusions**: if p_L is not visible on R, during Stage 1 it will be wrongly matched with a point on R, which in turn, if visible on L, will have its own homologous p'_L != p_L



Some results (local algorithms)



Ground truth

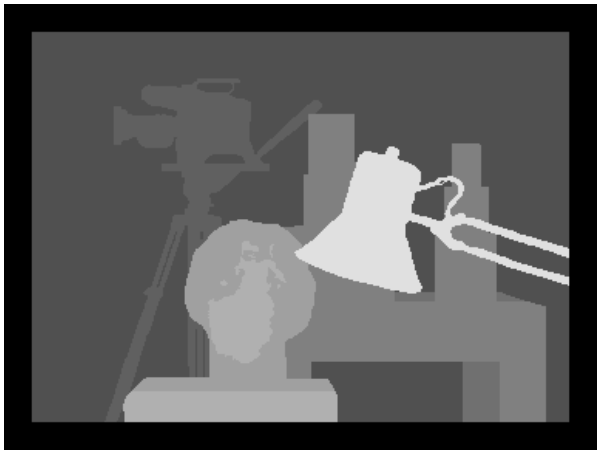


Block-based

Variable support
(Shiftable Windows [Bobick99])

Advanced local
(Segment support [Tombari07])

Some results (global algorithms)



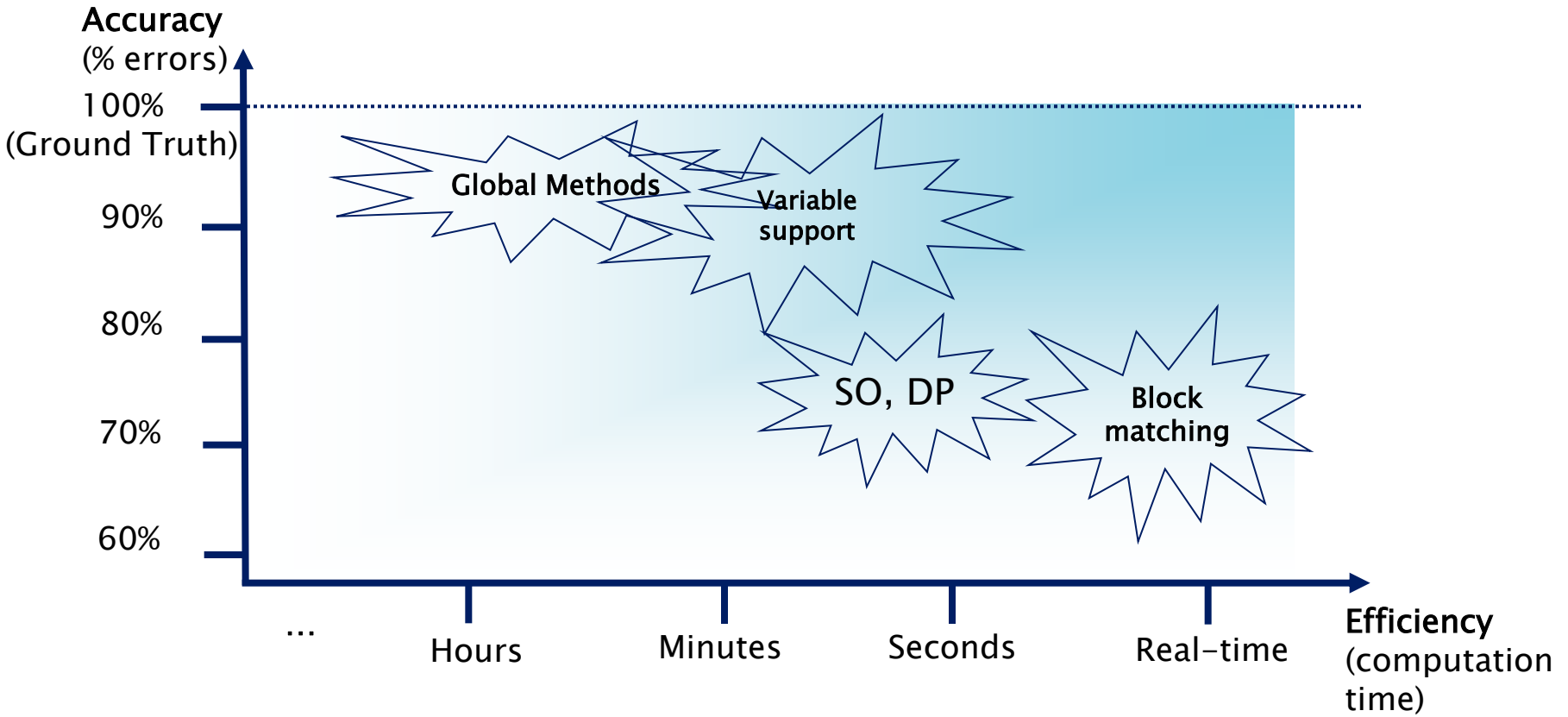
Ground truth

Belief Propagation [Klaus06]

Graph Cuts [Kolmogorov07]

The stereo dilemma

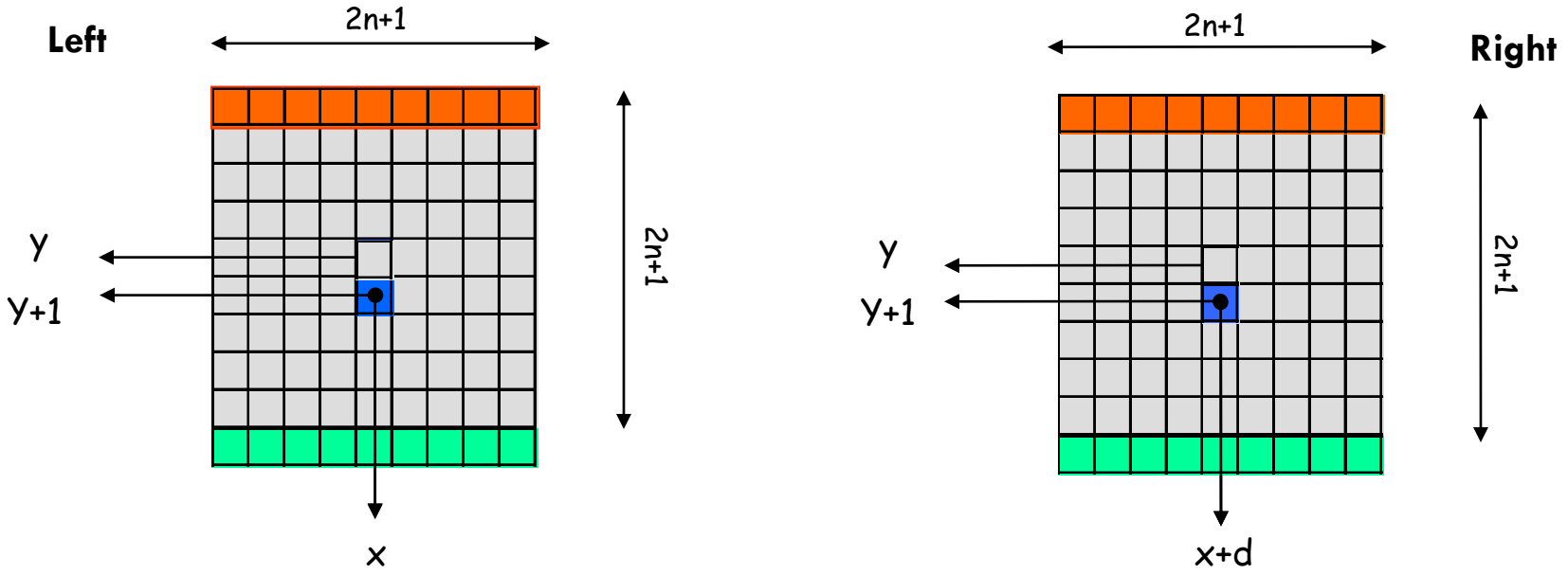
- Accurate algorithms exists...
- But for slight improvements in accuracy, we currently pay a high price



Real-time stereo

- Even considering the simplest algorithm (i.e. block-matching) the complexity of the stereo algorithm is quite high: $O(M^2 \times N^2 \times D)$
- For real-time applications, we need **computational optimizations**
- coarse-to-fine search
  - the disparity is estimated at low resolution, and refined (but exploring only a small disparity range) at high resolution
- Hardware acceleration: DSP [Faugeras93b], FPGA [Woodfill98, Corke99, Jia03], GPU or dedicated chips [Vanderval01].
- Incremental schemes to avoid redundant operations involved in the computation of function C(i,j,d) [Faugeras93b, DiStefano04, Fua93].
- Parallellization via SIMD instruction sets (Single Instruction Multiple Data) for multimedial data (e.g. MMX) available on current general-purpose CPUs [DiStefano04].

Box-Filtering [McDonnell81]



$$SAD(x,y,d) = \sum_{i,j=-n}^{n} \left| L(x+j, y+i) - R(x+d+j, y+i) \right|$$

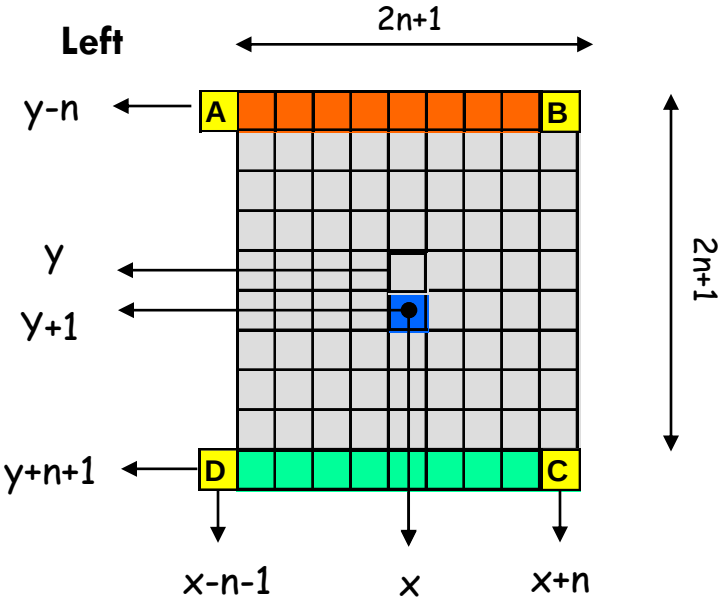
$$SAD(x,y+1,d) = SAD(x,y,d) + U(x,y+1,d)$$

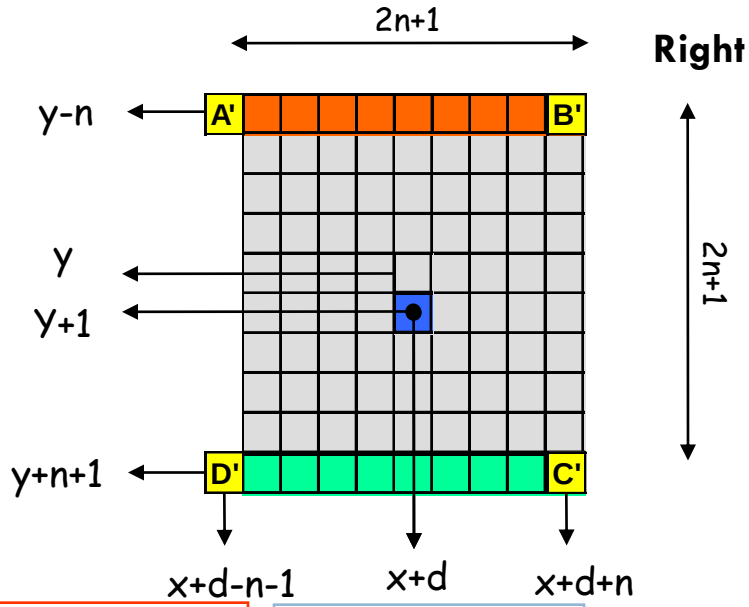
$$U(x,y+1,d) = \sum_{j=-n}^{n} \left| L(x+j, y+n+1) - R(x+d+j, y+n+1) \right| - \sum_{j=-n}^{n} \left| L(x+j, y-n) - R(x+d+j, y-n) \right|$$

Box-Filtering (2)

$$SAD(x, y+1, d) = SAD(x, y, d) + U(x, y+1, d)$$

$$U(x, y+1, d) = \sum_{j=-n}^{n} \left| L(x+j, y+n+1) - R(x+d+j, y+n+1) \right| - \sum_{j=-n}^{n} \left| L(x+j, y-n) - R(x+d+j, y-n) \right|$$





$$U(x, y+1, d) = U(x-1, y+1, d) + \left| A - A' \right| - \left| B - B' \right| + \left| C - C' \right| - \left| D - D' \right|$$

$$SAD(x, y+1, d) = SAD(x, y, d) + U(x-1, y+1, d) + \left| A - A' \right| - \left| B - B' \right| + \left| C - C' \right| - \left| D - D' \right|$$

Bibliography – Part 2

[Bobick99] A. Bobick, S. Intille, "Large occlusion stereo", International Journal of Computer Vision, 33(3):181–200, 1999.

[Corke99] P. Corke, P. Dunn, J. Banks "Frame-rate stereopsis using non parametric transforms and programmable logic", Proc. IEEE Conf. On Robotics and Automation, 1999.

[Di Stefano04] L. Di Stefano, M. Marchionni, S. Mattoccia "A Fast Area-Based Stereo Matching Algorithm", Image And Vision Computing, Vol. 22, No. 12, Oct. 2004.

[Faugeras93] O. Faugeras et. al. "Real time correlation-based stereo: algorithm, implementations and applications", INRIA Rapport de recherche N. 2013, 1993.

[Faugeras93b] O. Faugeras et. al. "Real time correlation-based stereo: algorithm, implementations and applications", INRIA Rapport de recherche N. 2013, 1993.

[Fua93] P. Fua "A parallel stereo algorithm that produces dense depth maps and preserves image features", Machine Vision and Applications, 1993.

[Hanna85] M.J. Hanna "SRI's Baseline Stereo System", Proc. Image Understanding Workshop, 1985.

[Jia03] Y. Jia, Y. Xu, W. Liu, C. Yang, Y. Zhu, X. Zhang, L. An "A Miniature Stereo Vision Machine for Real-Time Dense Depth Mapping", 3th Int. Conf. Computer Vision Systems, 2003.

[Klaus06] A. Klaus, M. Sormann and K. Karner, "segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure", Int'l Conf. Pattern Recognition 2006.

Bibliography – Part 2

[Kolmogorov01] V. Kolmogorov, R. Zabih "Computing visual correspondence with occlusions using graph cuts", In Eighth Intern. Conf. on Computer Vision, 2001.

[Moravec79] H.P. Moravec "Visual mapping by a robot rover", Proc. of. 6th Int. Joint Conf. on Artificial Intelligence, 1979.

[McDonnell81] M. Mc Donnell, "Box-Filtering techniques, Computer Graphics and Image Processing, 17:65-70, 1981

[Scharstein02] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", International Journal of Computer Vision, 47(1/2/3):7–42, 2002.

[Tombari07] F. Tombari, S. Mattoccia, L. Di Stefano, "Segmentation-based adaptive support for accurate stereo correspondence", IEEE Pacific-Rim Symposium on Image and Video Technology, 2007

[Trucco98] E. Trucco, A. Verri "Introductory Techniques for 3-D Computer Vision", Prentice Hall, 1998.

[Vanderval01] G. Van der Val, M. Hansen, M. Piacentino "The ACADIA Vision Processor", 5th Int. Work. on Computer Architecture for Machine Perception, 2001.

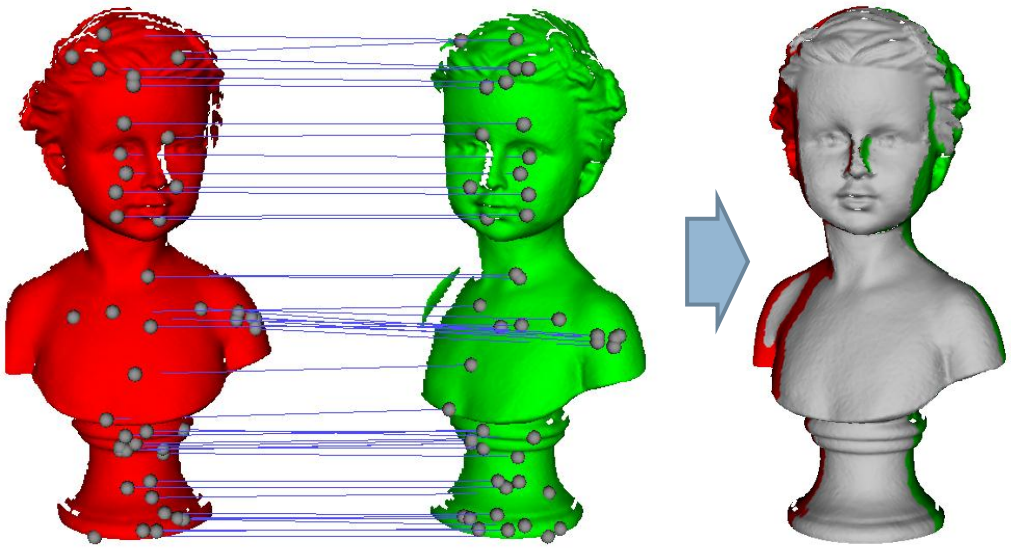
[Woodfill98] J. Woodfill, B. Von Herzen "Real-time stereo vision on the PARTS reconfigurable computer", Proc. IEEE Symp. On FPGA for Custom Computing Machines, 1998.

Part 3 – tasks and applications

- 3D Computer Vision tasks
  - Registration
  - SLAM
  - Retrieval
  - Recognition
  - Semantic Segmentation
- Applications

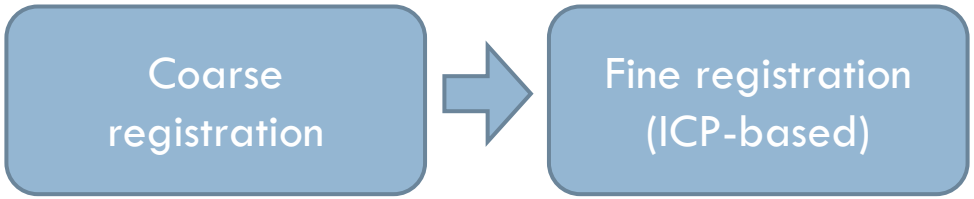
3D registration

- Alignment of partially-overlapping 2.5D views
- Useful to yield a high-def, fully-3D reconstruction of an object from views acquired from different view points
  - Pairwise registration
  - Registration from multiple views



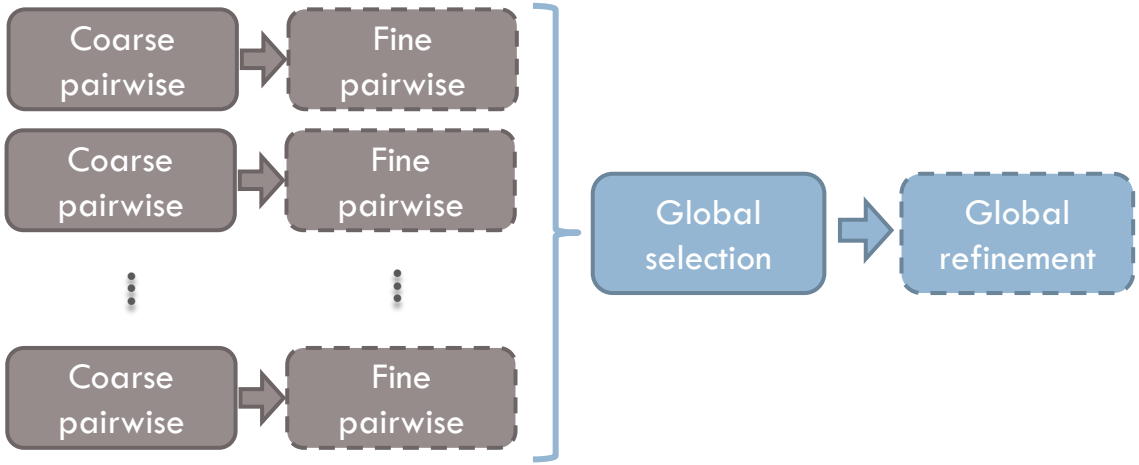
Coarse-to-fine approach

- Coarse registration provides an initial guess for the set of views that need to be registered
  - By hand
  - By matching features
- Fine registration is generally based on Iterative Closest Points (**ICP** [Besl92])
  - Will diverge if initial guess is not reliable enough or data is noisy



Registration from multiple views

- Global selection and registration (unordered input views) [Huber03]

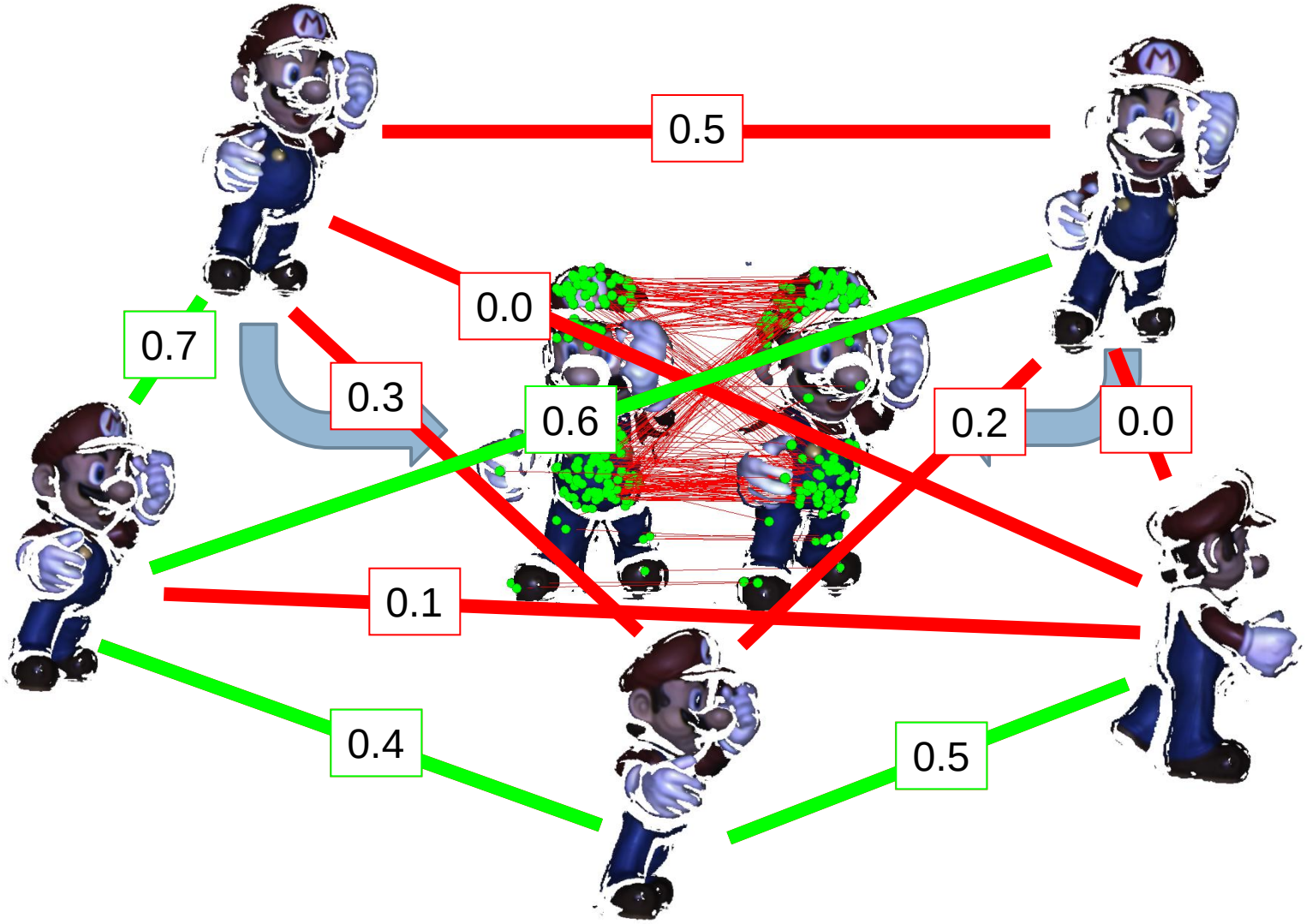


unordered input views

After global selection

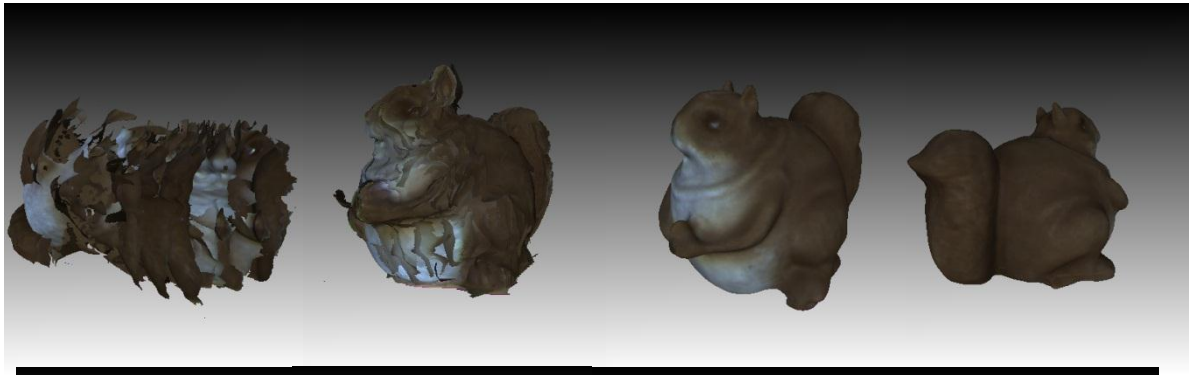
After global refinement (Scanalyze)

Global selection



Results

Spacetime Stereo



Kinect sensor

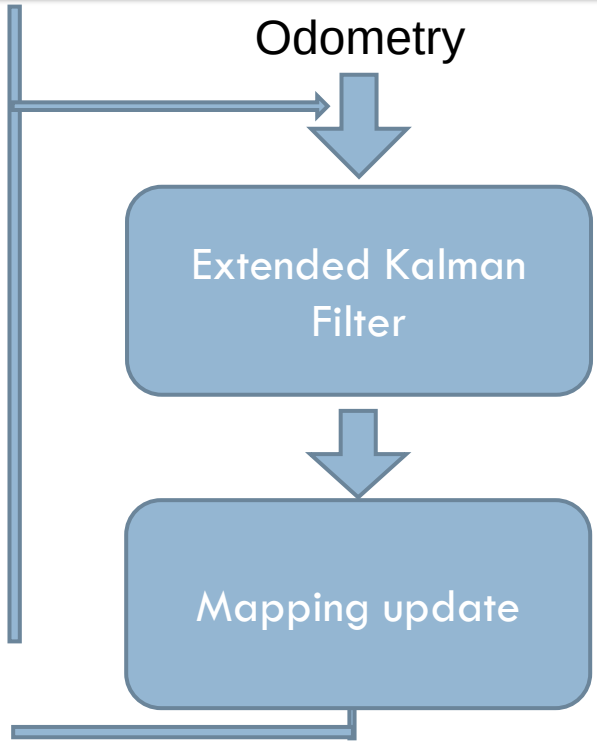
SLAM (1)

- Simultaneous Localization and Mapping
  - incrementally build a map of the agent's surroundings (**mapping**)
  - Localize itself within that map
- Odometry, inertial sensing
  - Measurement drifts
  - Visual odometry [Nistèr 04] [Konolige 06]
- 3D / photometric sensors
  - Laser scanner
  - Sonar
  - Stereo [Sim 06]
  - Visual sensors (vSLAM) [Karlsson 05][Folkesson 05]
    - Landmark initialization?
  - 6DOF SLAM
- monoSLAM [Davison 03] [Eade 06][Clemente 07]
  - Visual odometry + single camera

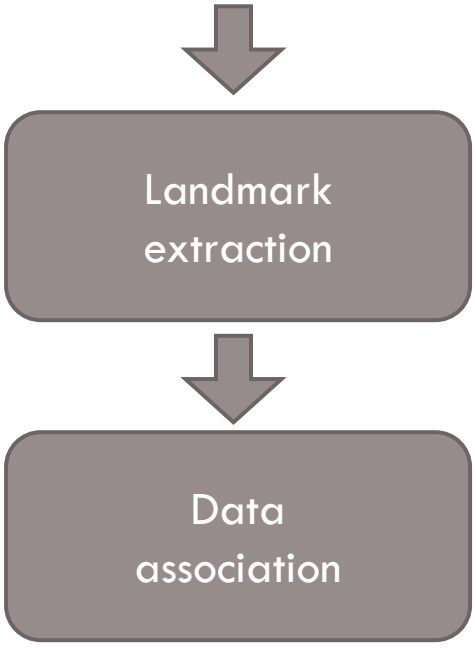
Credits: J.B.Hayet

SLAM (2)

EKF:
- Update via odometry
- Update via landmark re-observation



Geometric/photometric data

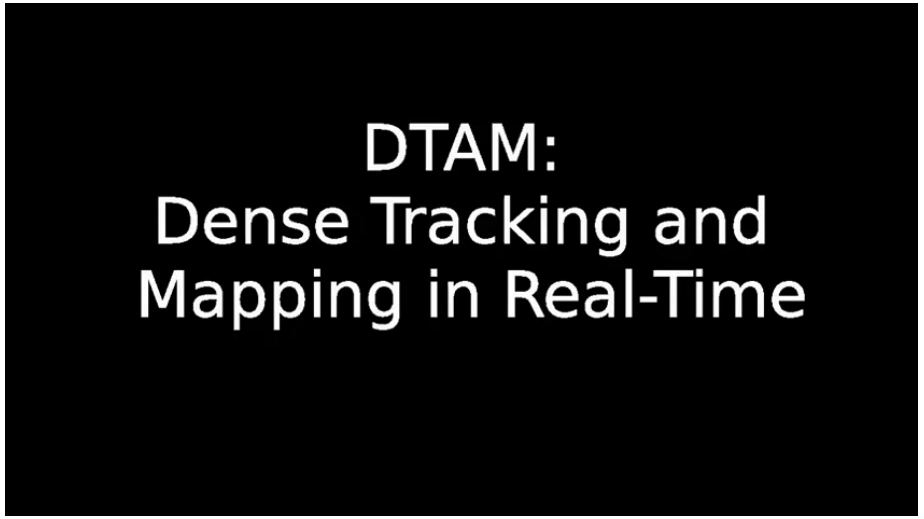


Landmarks:
- Re-observable
- Distinctive
- Stationary

Local vs. Global consistency
(loop closure, bundle adjustment)

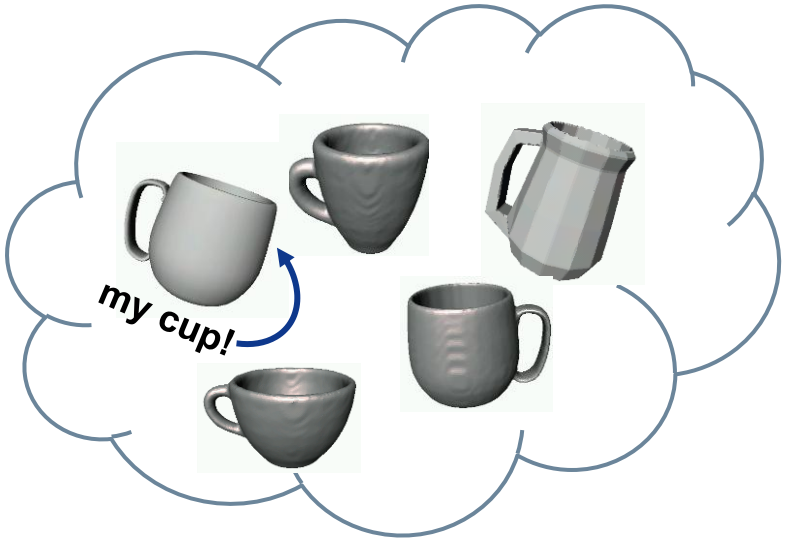
SLAM (3)

- MonoSLAM converging to Structure-from-Motion [Strasdat 10]
  - PTAM [Klein 07], DTAM [Newcombe 11])
- 6DOF SLAM with RGB-D sensors
  - Kinect Fusion [Newcombe 11b]
  - RGB-D dense point cloud mapping [Henry 11]

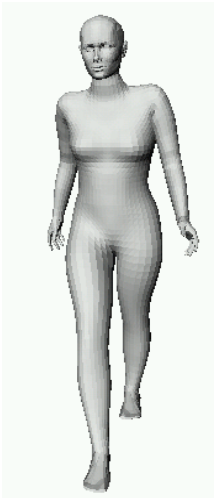


3D Shape Retrieval / Categorization

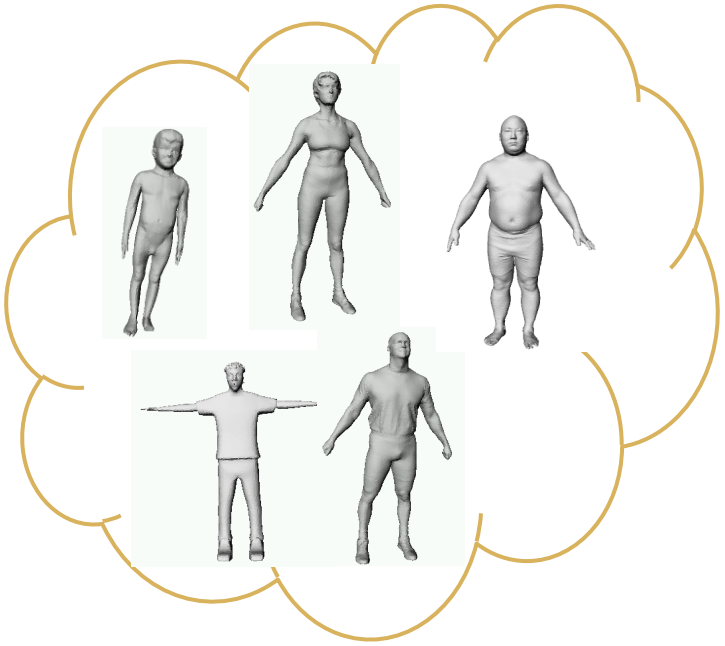
- Differently from recognition, which recognizes specific object instances (**my** cup, **that** teddy bear), shape retrieval/categorization associates a category label to a given query model
- Typically no clutter and occlusion, but high **intra-class variance**



CUPS

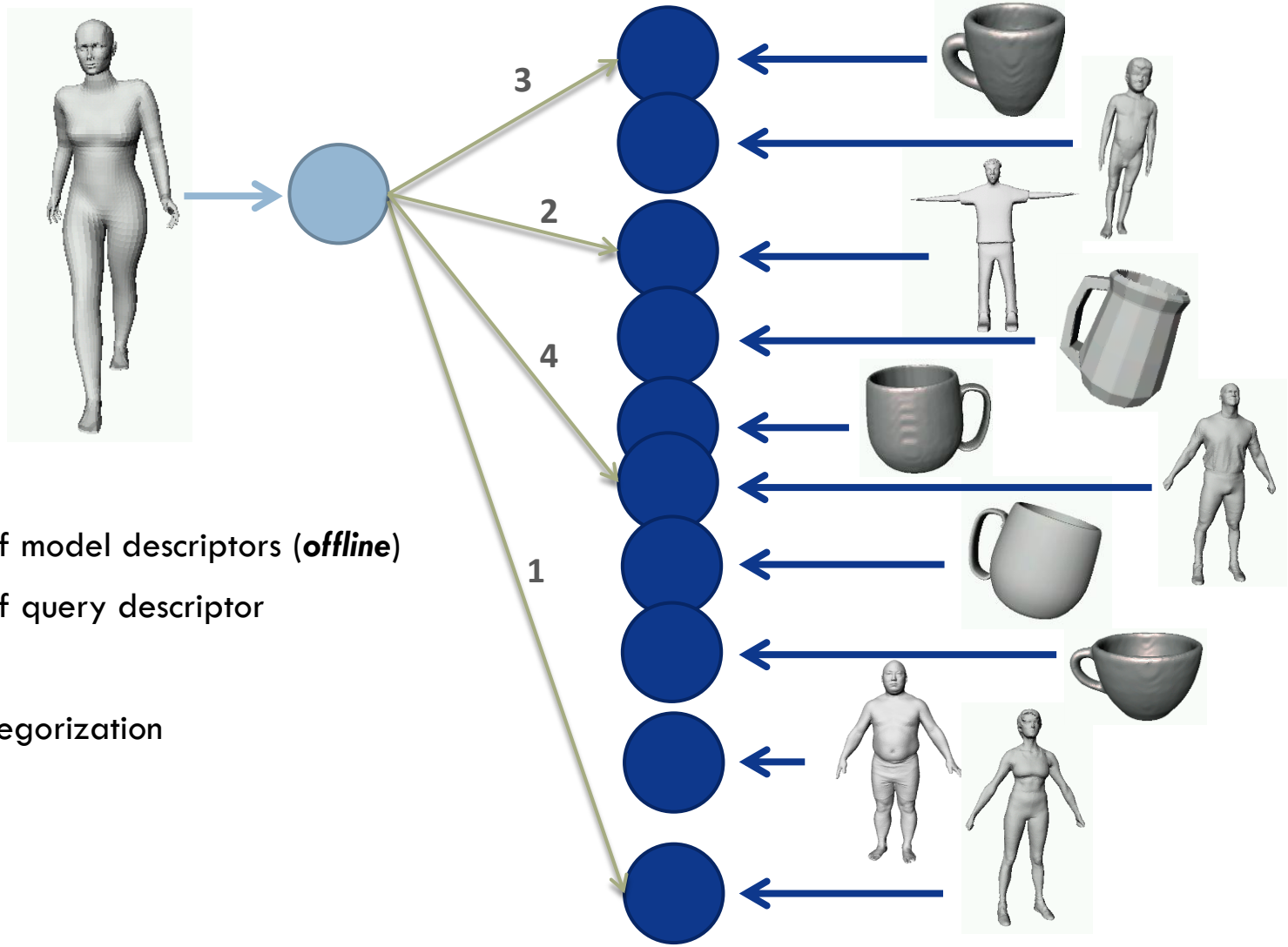


A human..or a cup?



HUMANS

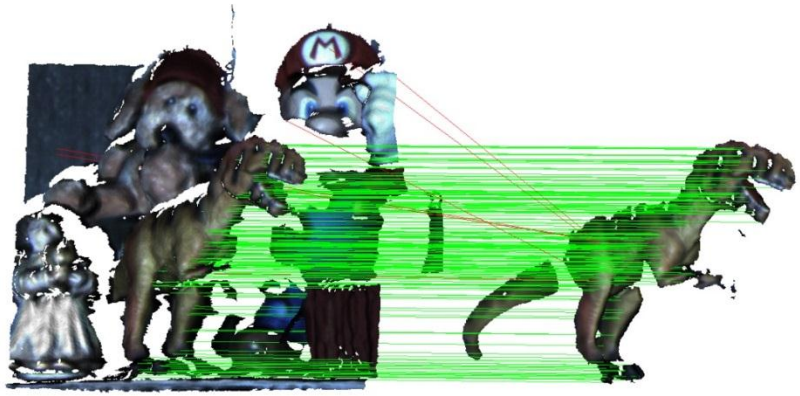
Shape retrieval via global descriptors

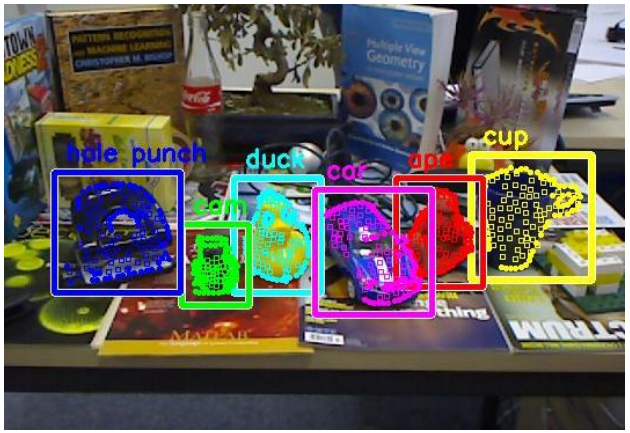


- Computation of model descriptors (***offline***)
- Computation of query descriptor
- kNN matching
- Retrieval / categorization

3D Object Recognition

- Determine the presence of a model in a scene and estimate its 6DOF pose
- Challenges
  - Clutter, occlusions
  - Sensor noise: missing parts, holes (transparent/dark objects), artifacts
  - Dealing with large model libraries
- To deal with clutter and occlusions, object descriptors are «shrinked» to a small region around interest points (**local descriptors**) [Tombari10]
- Otherwise, area-based (*template matching*) [Hinterstoisser12]



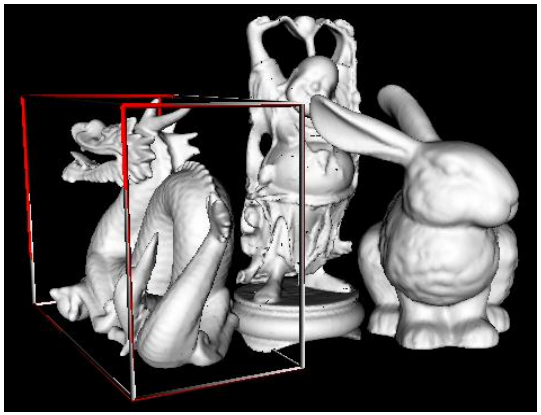


Courtesy of S. Hinterstoisser

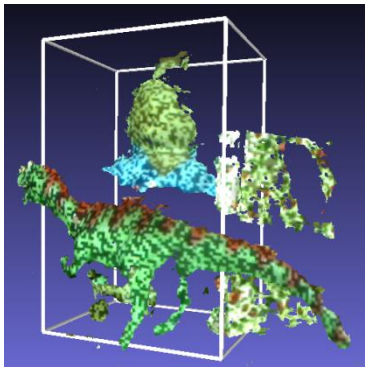
Results

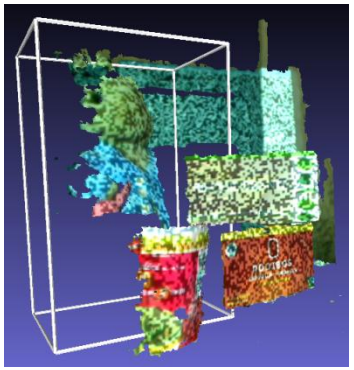
Kinect

Spacetime Stereo



Syntethic data



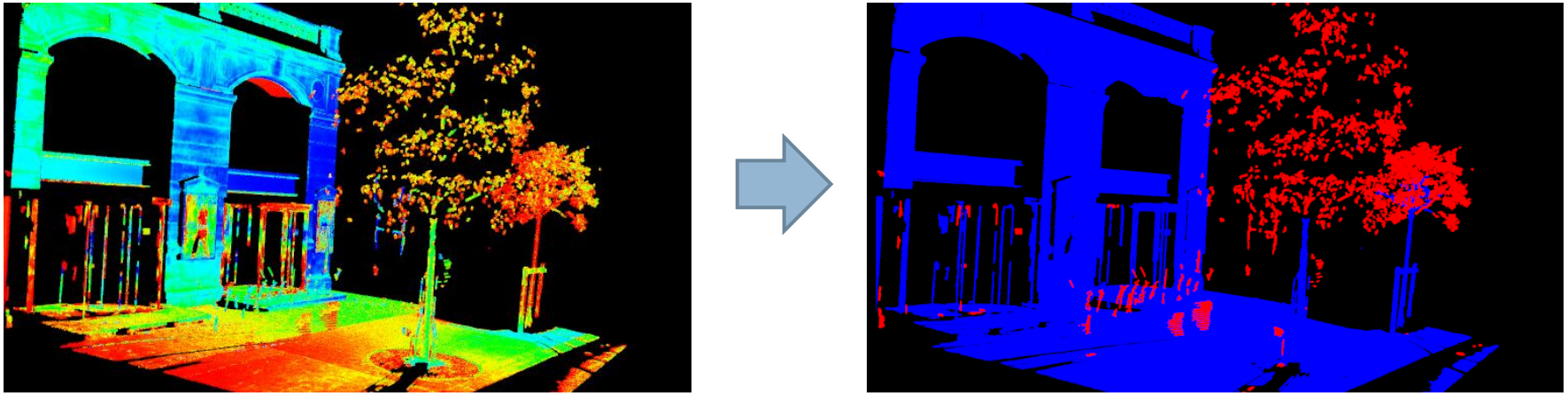


Real-time stereo

Semantic segmentation

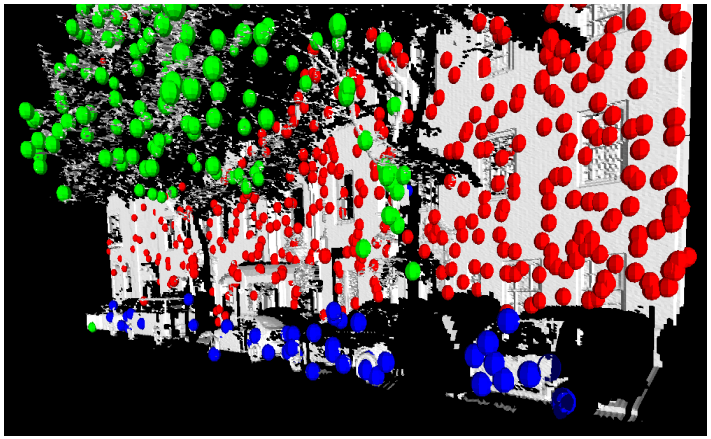
- Goal: Determine 3D connected components with specific properties or belonging to a particular semantic category

- Applications: urban/indoor scene understanding, robot localization and navigation
- Also: useful as the first step of an object categorization/recognition algorithm



Approaches

1. Segmentation + segment classification
   - Euclidean clustering
   - Smooth region growing (clustering neighboring points on smooth surfaces)
   - Exploit prior knowledge (e.g. dominant plane(s))
2. Pointwise feature classification + grouping
   - Inference on a loopy graph (MRF) [Tombari11]



- **Associative Markov Networks** (AMN) [Anguelov05][Triebel07][Munoz09]

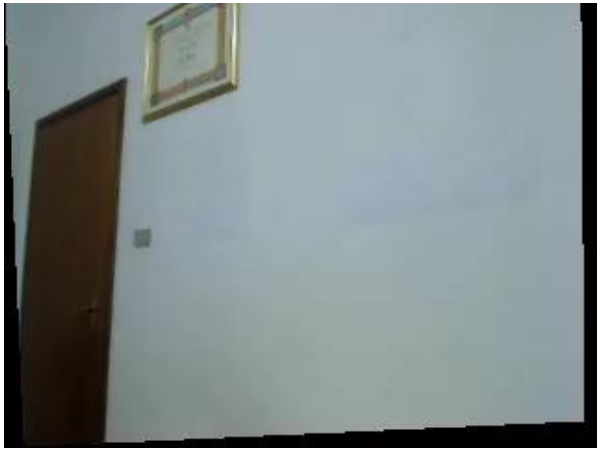
Applications - robotics

- Autonomous mobile robots – AMR (*navigation*)

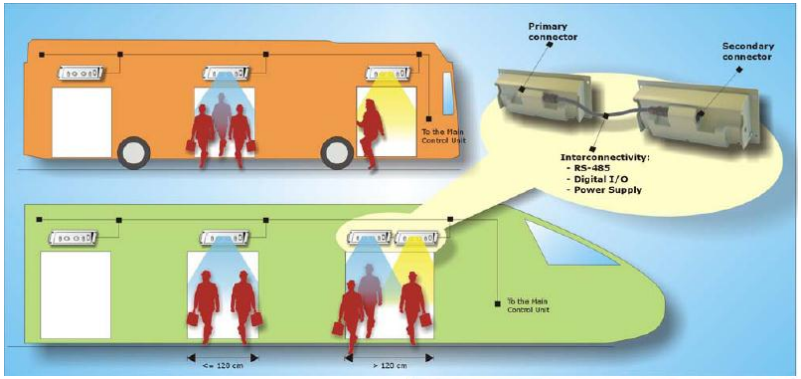
- Object recognition, grasping and manipulation (*social robotics*)

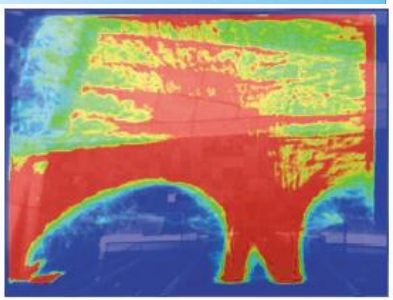
Applications - video surveillance

Tracking and motion detection

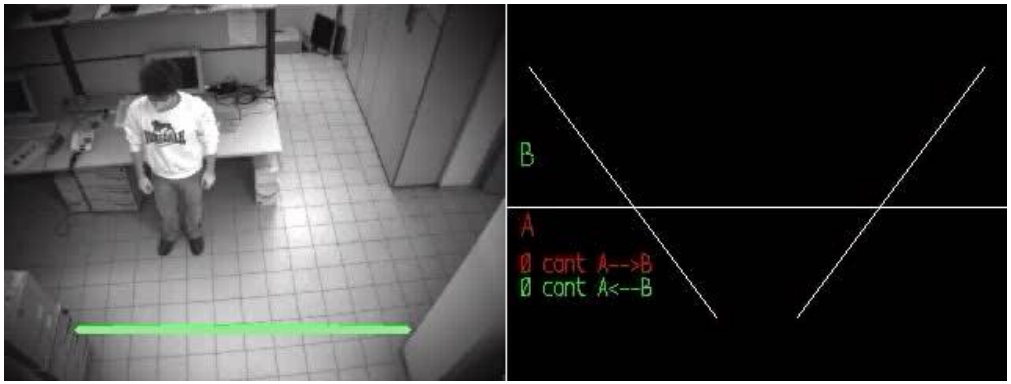


Behavior analysis





Retail intelligence
Crowd monitoring



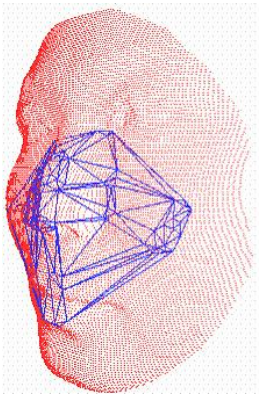
People counting

Other applications

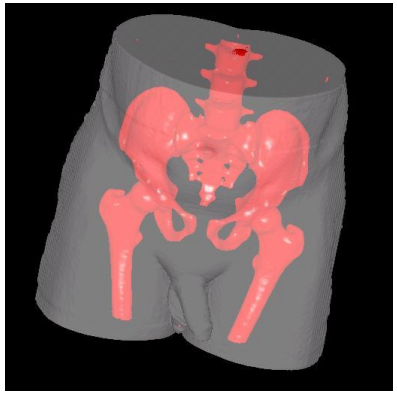
- Shape retrieval (*www*)
- High-def 3D model acquisition (*computer graphics*)
- Biometrical systems (eg. face recognition)
- Medical imaging (MRI, CT, PET, x-ray, ultrasound, ..)

Michelangelo project

Google warehouse



3D face recognition



3D medical imaging

And yet..

- Autonomous vehicle navigation (AVN)
- Augmented reality
- Human computer interaction (HRI)
- Videogaming, entertainment
- ..

Augmented reality by Lego and Intel

DARPA GRAND CHALLENGE

Autonomous Vehicle Navigation

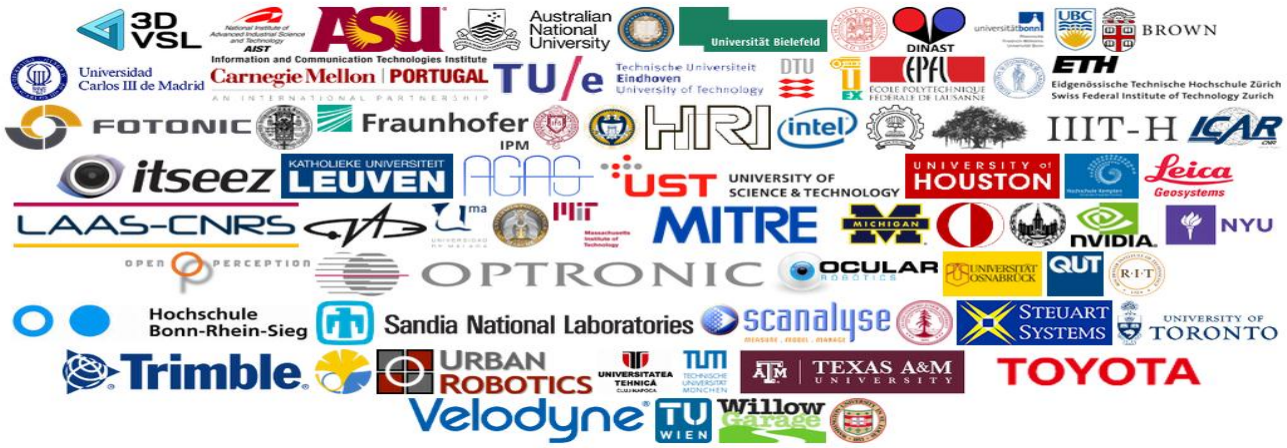
Microsoft Xbox

Point Cloud Library

- Reference open source community for 3D computer vision and robotic perception
- Includes modules for
  - Keypoint extraction (pcl_keypoint)
  - Global/local descriptors (pcl_features)
  - Object Recognition in clutter (pcl_recognition)
  - Surface registration (pcl_registration)
  - Cloud segmentation (pcl_segmentation)
- And many more .. (machine learning, stereo, I/O, ...)

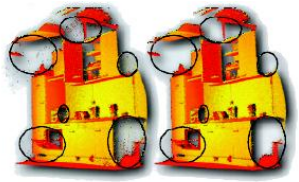
pcl

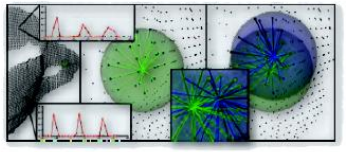
www.pointclouds.org

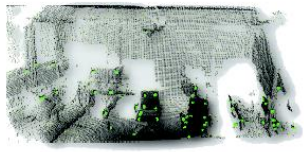


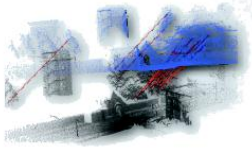
PCL module

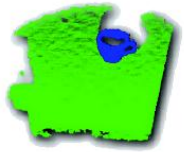
- 22 Code libraries + separate CUDA/GPU/Mobile modules


Features

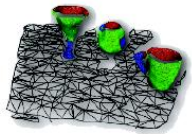

Filters

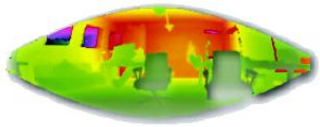

Keypoints


Registration


Segmentation


Sample Consensus


Surface


Range Image

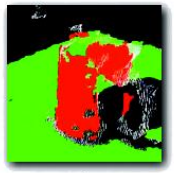

I/O


People


Simulation


Out-of-core

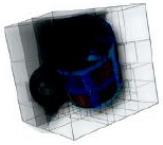

Visualization


Segmentation


2D


ML


Recognition


Kdtree


Octree

Bibliography – part 3a

- [**Anguelov 05**] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, A. Ng, "*Discriminative learning of markov random fields for segmentation of 3-d scan data*", *Proc. CVPR*, 2005
- [**Clemente 07**] L. Clemente, A. J. Davison, I. Reid, J. Neira, J. Tardòs, «*Mapping large loops with a single hand-held camera*», Proc. Conf. Robotics: Science and Systems (RSS), 2007
- [**Davison 03**] A. J. Davison, "*Real-time simultaneous localisation and mapping with a single camera*", Proc. ICCV, 2003
- [**Eade 06**] E. Eade, T. Drummond, "*Scalable monocular SLAM*", Proc. *Conf. on Computer Vision and Pattern Recognition,* 2006
- [**Folkesson 05**] J. Folkesson, P. Jensfelt, H. Christensen, "*Vision SLAM in the Measurement Subspace*," IEEE Int. Conf. Robotics and Automation (ICRA), 2005
- [**Henry 11**] P. Henry, M. Krainin, E. Herbst, X. Ren, D. Fox, "*RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments*", Proc. Int. Symp. on Experimental Robotics, 2010
- [**Hinterstoisser 12**] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm N. Navab, P. Fua, V. Lepetit, "Gradient Response Maps for Real-Time Detection of Texture-Less Objects", IEEE Trans. on Pattern Analysis and Machine Intelligence, 2012
- [**Huber03**] D.F. Huber, M. Hebert, "Fully automatic registration of multiple 3D data sets", Image and Vision Computing, 21:637-650, 2003

Bibliography – part 3b

- **[Karlsson 05]** N. Karlsson, E. D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, M. E. Munich, "*The vSLAM algorithm for robust localization and mapping*," Proc. *Int. Conf. on Robotics and Automation ICRA)*, 2005
- **[Klein 07]** G. Klein, D. W. Murray, "*Parallel tracking and mapping for small AR workspaces*", Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR), 2007
- **[Konolige 06]** K. Konolige, M. Agrawal, R.C. Bolles, C. Cowan, M. Fischler, B. Gerkey, *"Outdoor mapping and navigation using stereo vision"*, Proc. Int. Symp. on Experimental Robotics (ISER),2006
- **[Munoz 09]** D. Munoz, J. A. Bagnell, N. Vandapel, M. Hebert, "*Contextual classification with functional max-margin markov networks*", Proc. CVPR, 2009.
- **[Newcombe 11]** R.A. Newcombe, S.J. Lovegrove, A.J. Davison, "*DTAM: Dense Tracking and Mapping in Real-Time*", IEEE International Conference on Computer Vision (ICCV), 2011
- **[Newcombe 11b]** R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, A. Fitzgibbon, «*KinectFusion: Real-Time Dense Surface Mapping and Tracking*", Proc. Int. Symp. on Mixed and Augmented Reality (ISMAR), 2011
- **[Nistèr 04]** D. Nistèr, O. Naroditsky, J. Bergen, "Visual odometry", Proc. *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004
- **[Strasdat 10]** H. Strasdat, J.M.M. Montiel, A. J. Davison, «*Real-time Monocular SLAM: Why Filter?*", Proc. ICRA, 2010

Bibliography – part 3c

- [**Sim 06**] R. Sim, J. J. Little, "*Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters*," Proc. *Conf. on Intelligent Robots and Systems (IROS)*, 2006
- [**Triebel 07**] R. Triebel, R. Schmidt, O. M. Mozos, W. Burgard, "*Instance-based AMN classification for improved object recognition in 2d and 3d laser range data*", Proc. Int. Conf. on Art. Intelligence, 2007
- [**Tombari et al., 10**] Tombari F, Salti S, Di Stefano L. "Unique signatures of histograms for local surface description" In: *Proc. Europ. Conf. on Computer Vision (ECCV)*, Springer-Verlag, Berlin, Heidelberg, pp 356-369, 2010.
- [**Tombari 11**] F. Tombari, L. Di Stefano, "*3D Data Segmentation by Local Classification and Markov Random Fields*", Proc. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011