Stereo Vision – Correspondence

COMP4102A

Dr. Gerhard Roth

Winter 2013

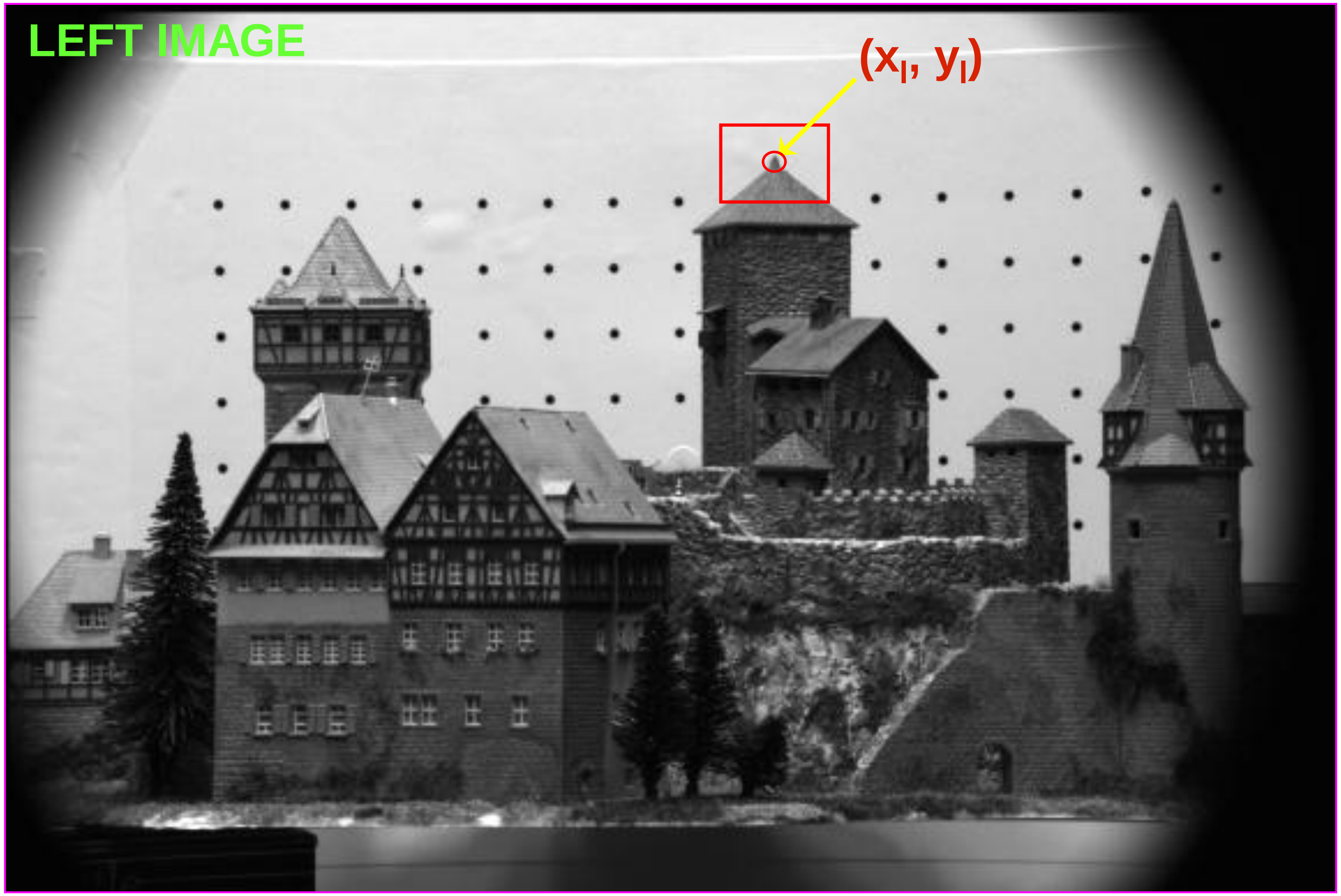
Problem Definition

- Correspondence problem
  - What parts of the left and right image are projections of the same point in the 3D scene
- Simple stereo configuration
  - Corresponding points are on same horizontal line
- Assumptions
  - Most scene points are visible from both regions
  - Corresponding image regions are similar (called similarity constraint)
- Search problem
  - Given scene element on left image search for
  - What parts of left and right images are parts of same object?
- Two decisions
  - Which element to match
  - Which similarity measure to adopt

Correspondence and Feature Methods

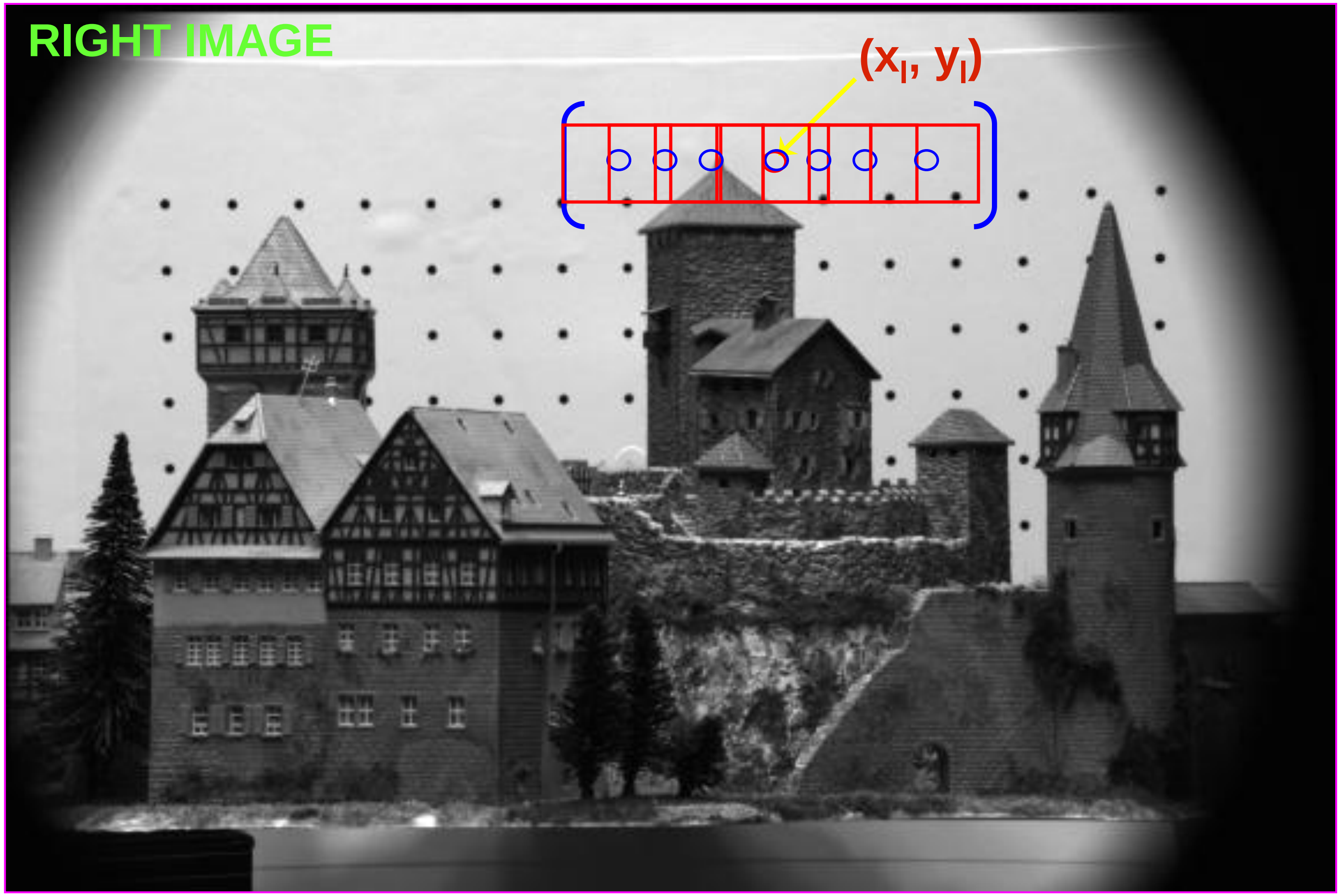
- Two basic approaches
- Correlation (Similarity) methods
  - Apply to all image points
  - Elements are image windows of fixed size
  - Similarity measure is the difference between two windows in the left and right images
  - Corresponding element is window that maximizes similarity criterion within a search window
- Feature methods
  - Apply only to a sparse set of feature points
  - Narrows down feasible matches by using constraints
  - Geometric constraints
  - Analytic constraints – uniqueness and continuity

Correlation Approach



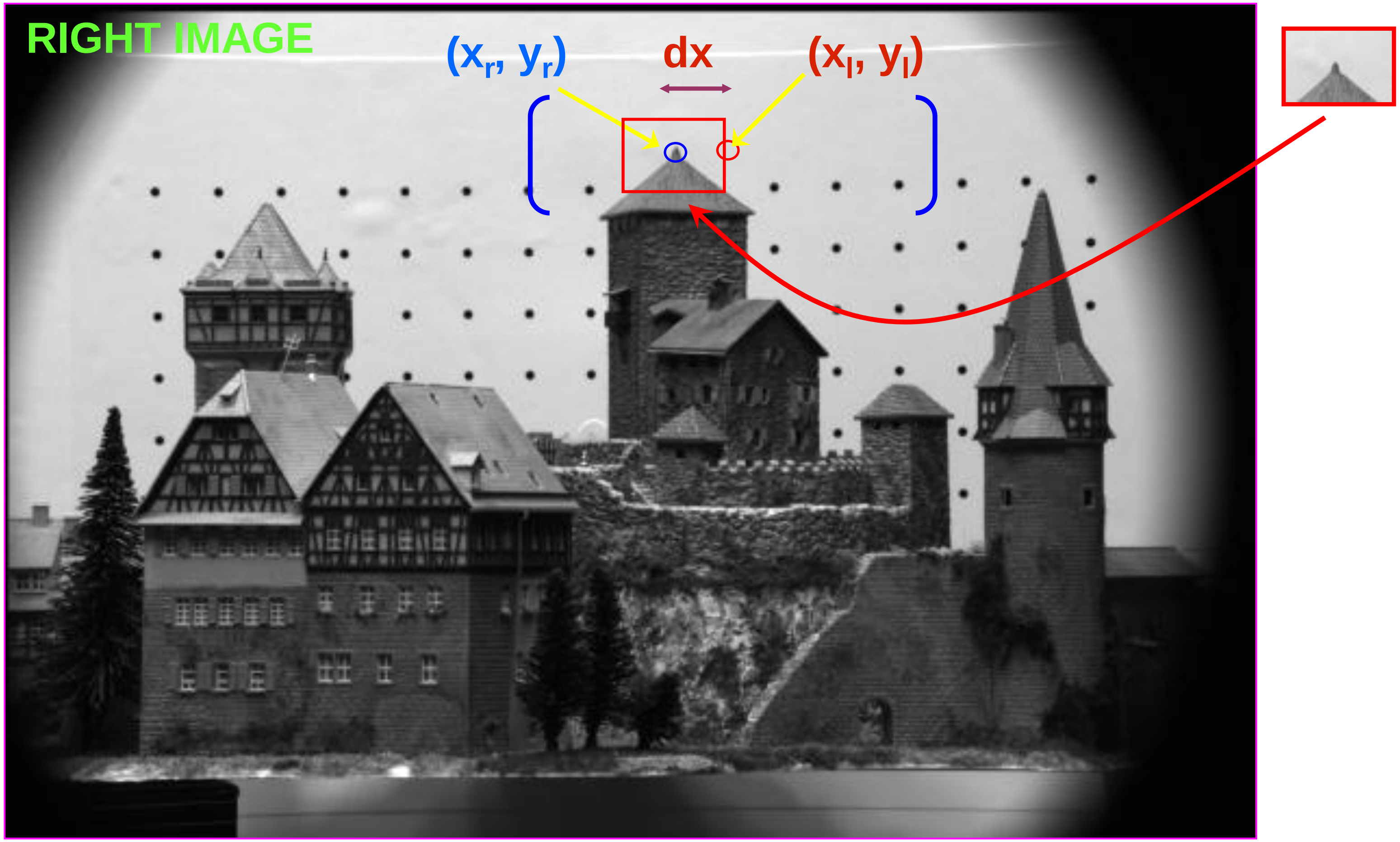
For Each point (x_l, y_l) in the left image, define a window centered at the point

Correlation Approach



… search its corresponding point within a search region in the right image

Correlation Approach



… the disparity (dx, dy) is the displacement when the correlation is maximum

Correlation (Similarity) Approach

Elements to be matched

- Image window of fixed size centered at each pixel in the left image

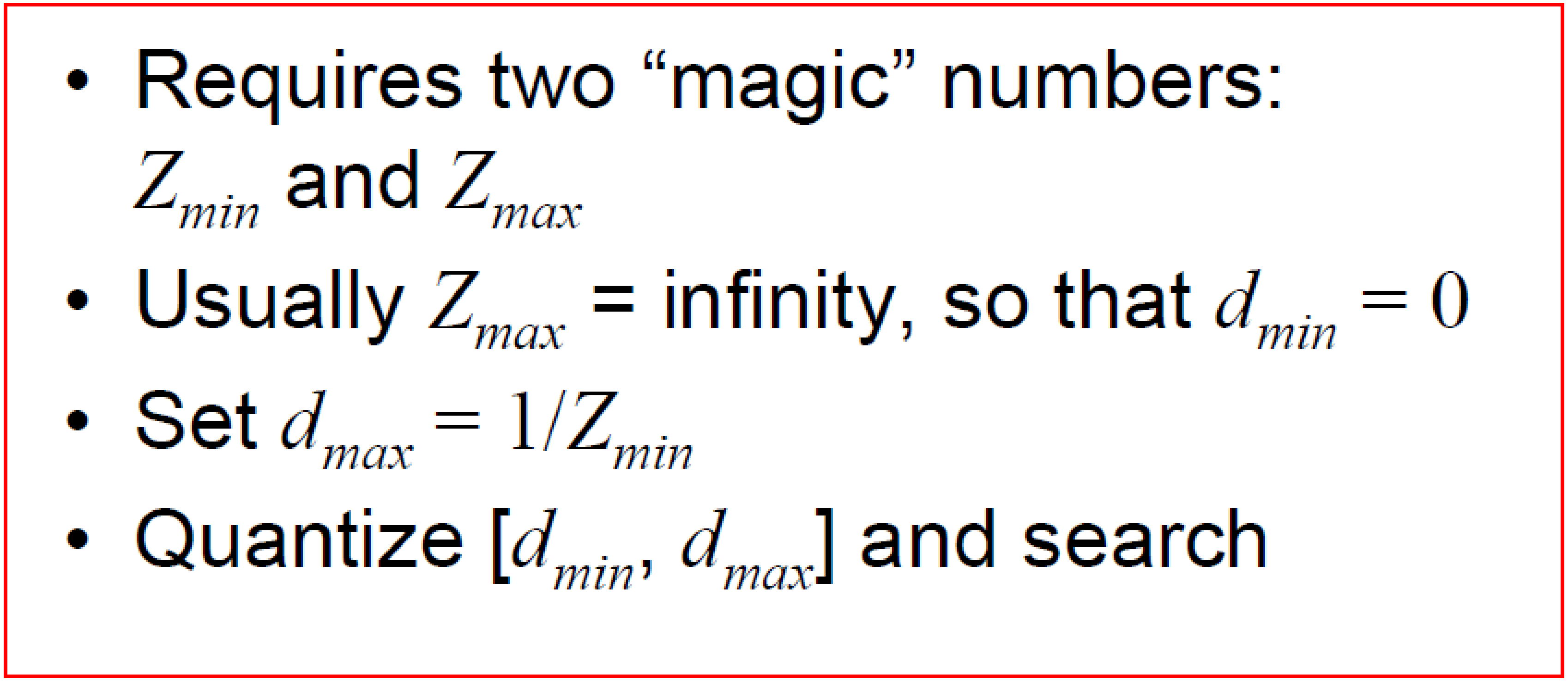
Similarity criterion

- A measure of similarity between windows in the two images
- The corresponding element is given by window that maximizes the similarity criterion within a search region

Search regions

- Theoretically, search region can be reduced to a 1-D segment, along the horizontal line (in future we will use term epipolar line), and within the disparity range.
- In practice, search a slightly larger region due to errors in calibration

Search Region Size (Disparity Range)



- Requires two “magic” numbers: Z_{min} and Z_{max}
- Usually Z_{max} = infinity, so that $d_{min} = 0$
- Set $d_{max} = 1/Z_{min}$
- Quantize [d_{min}, d_{max}] and search

Correlation Approach (general disparity)

Equations = w is the window size

$$c(dx,dy)=\sum_{k=-W}^{W}\sum_{l=-W}^{W}\psi(I_l(x_l+k,y_l+l),I_r(x_l+dx+k,y_l+dy+l))$$

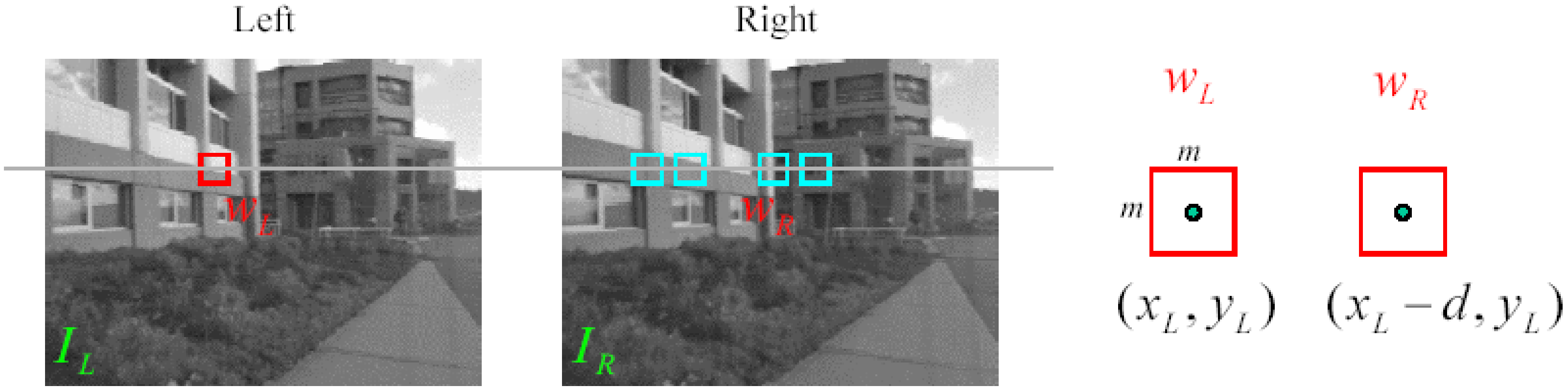
disparity

$$\bar{\mathbf{d}}=(\bar{d}x,\bar{d}y)=\arg\max_{\mathbf{d}\in R}\{c(dx,dy)\}$$

Similarity criterion

- Cross-Correlation $\Psi(u,v)=uv$
- Sum of Square Difference (SSD) $\Psi(u,v)=-(u-v)^2$
- Sum of Absolute Difference(SAD) $\Psi(u,v)=-|u-v|$

Sum of Squared Differences (SSD)



w_L and w_R are corresponding m by m windows of pixels.

We define the window function :

$$W_m(x, y) = \{u, v \mid x - \tfrac{m}{2} \le u \le x + \tfrac{m}{2}, y - \tfrac{m}{2} \le v \le y + \tfrac{m}{2}\}$$

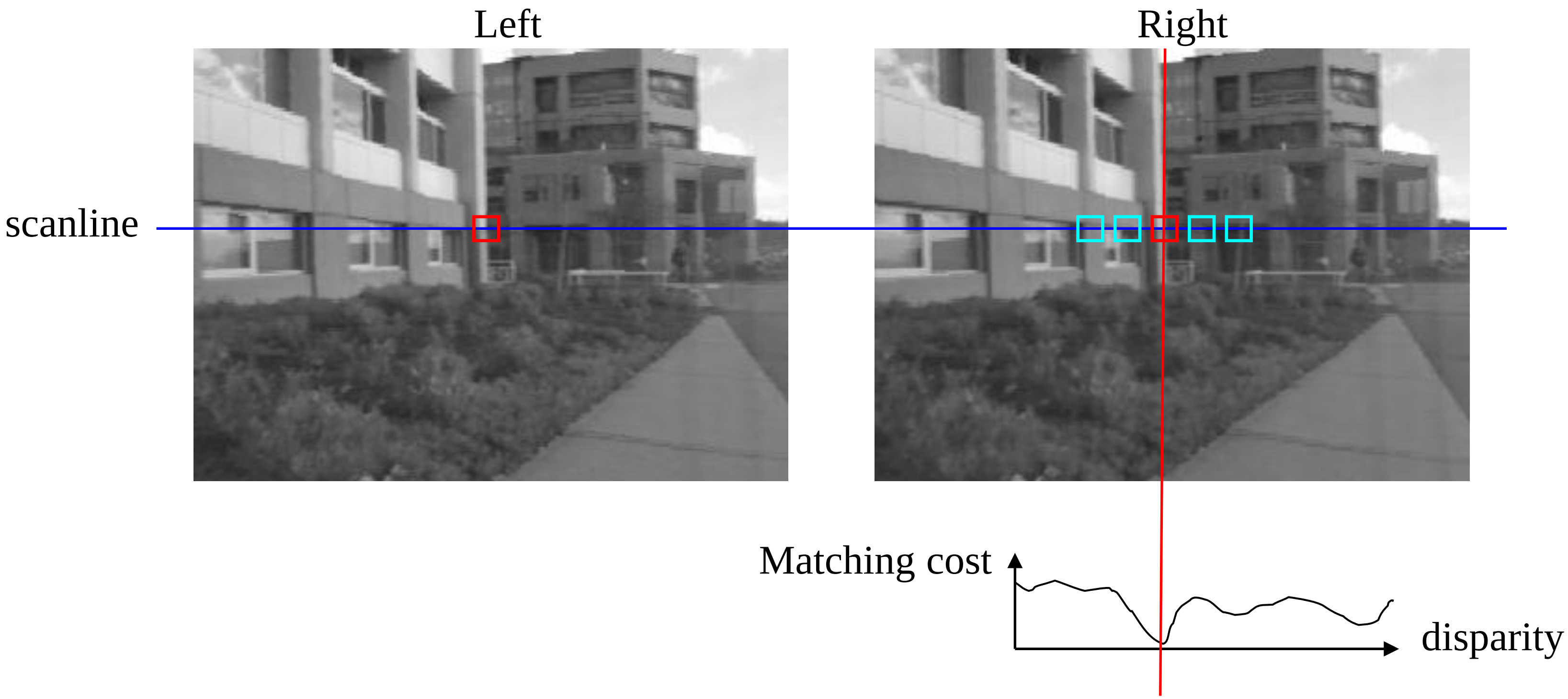
The SSD cost measures the intensity difference as a function of disparity :

$$C_r(x, y, d) = \sum_{(u,v) \in W_m(x,y)} [I_L(u, v) - I_R(u - d, v)]^2$$

For simple stereo need only move to left in rightmost image.

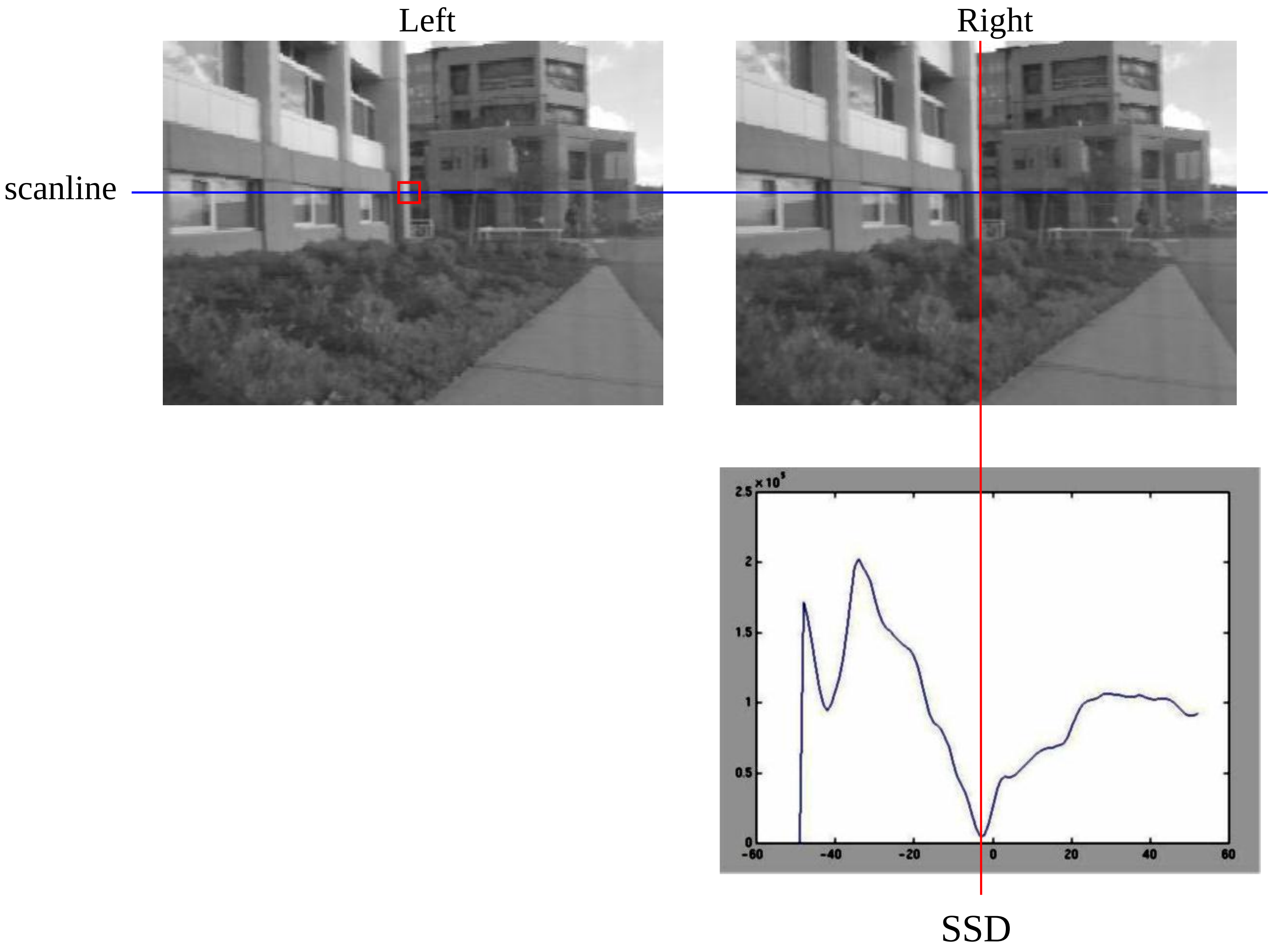
If x is zero is leftmost part of image, only look at (x_L – d)

Correspondence search with similarity constraint

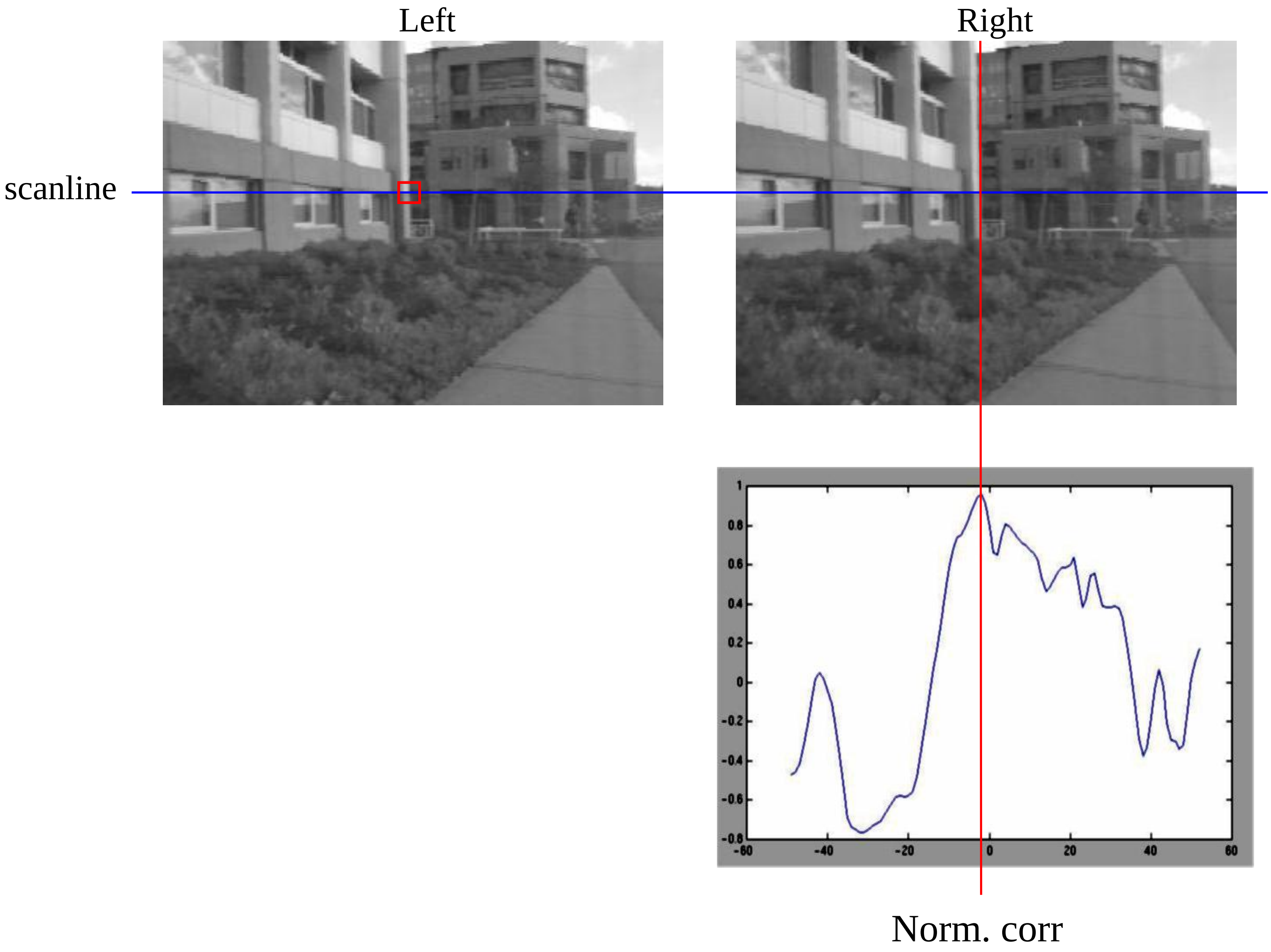


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search with similarity constraint

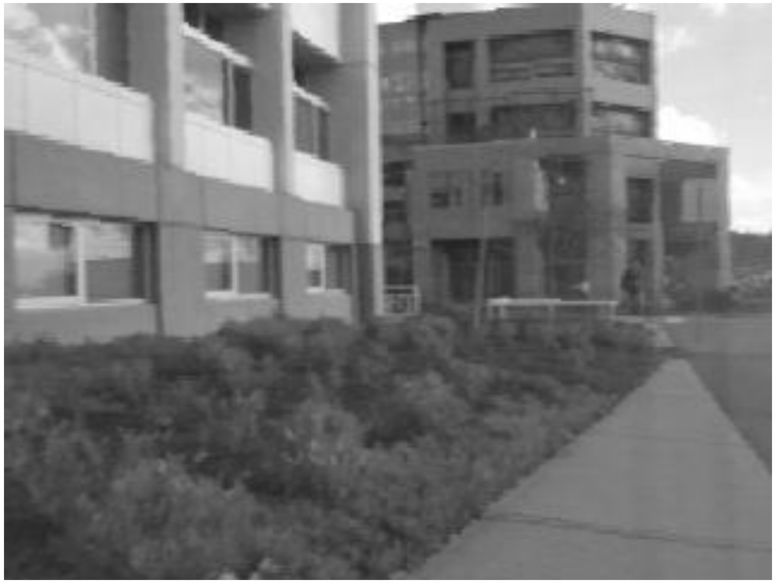


Correspondence search with similarity constraint

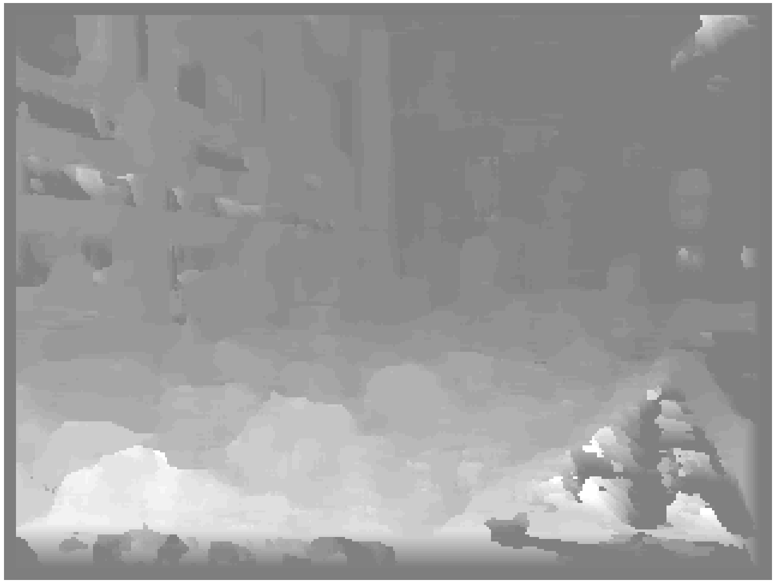


Correspondence Using Correlation

Left



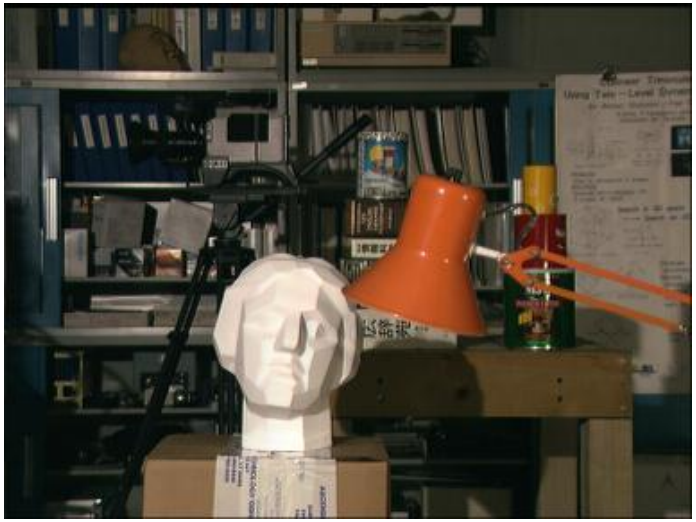
Disparity Map



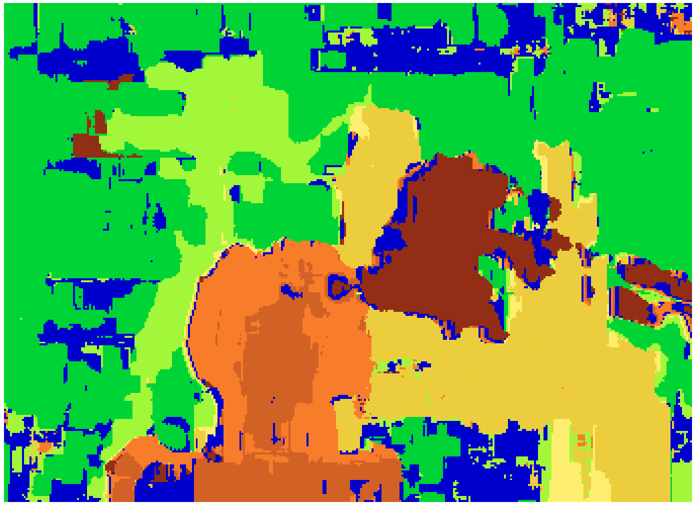
Images courtesy of Point Grey Research

Correspondence Using Correlation

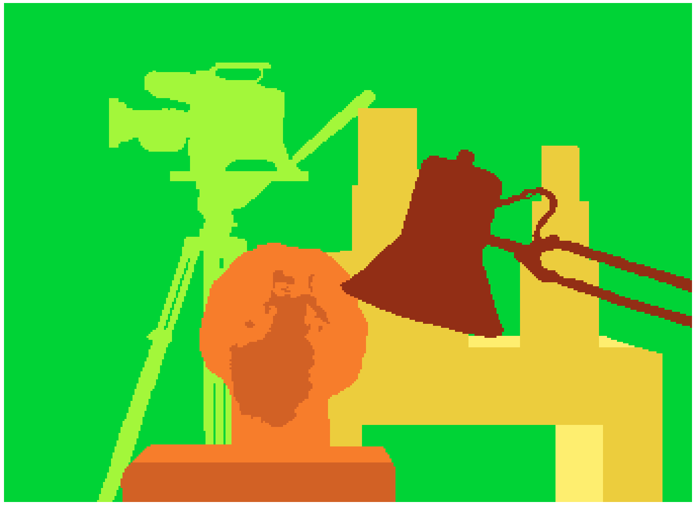
Data



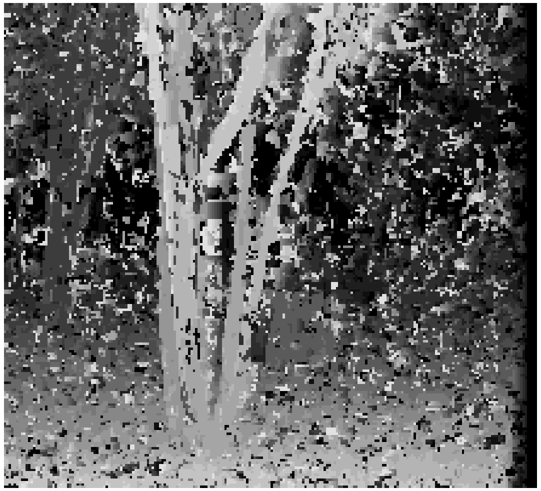
Window-based matching



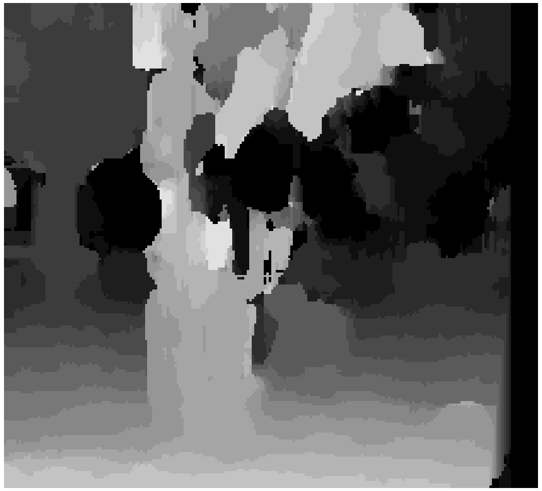
Ground truth



Effect of window size



W = 3

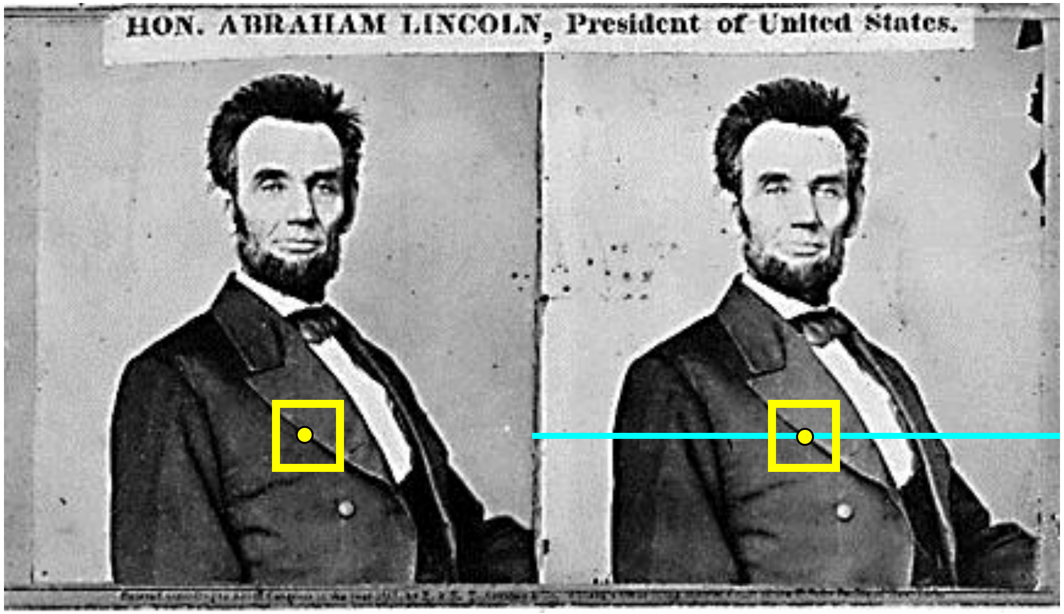


W = 20

- Smaller window
  - + More detail
  - – More noise

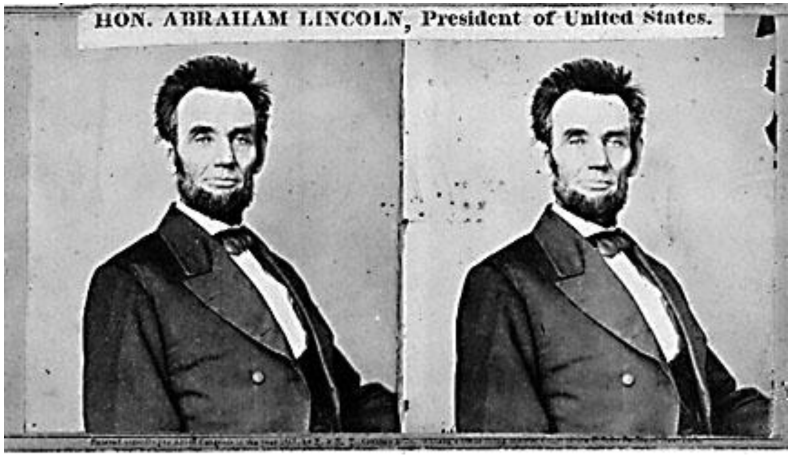
- Larger window
  - + Smoother disparity maps
  - – Less detail

The similarity constraint

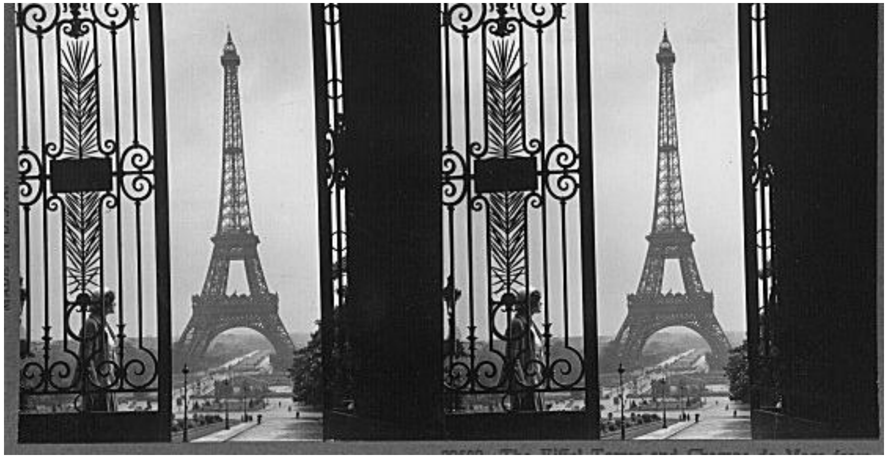


- Corresponding regions in two images should be similar in appearance
- …and non-corresponding regions should be different
- When will the similarity constraint fail?

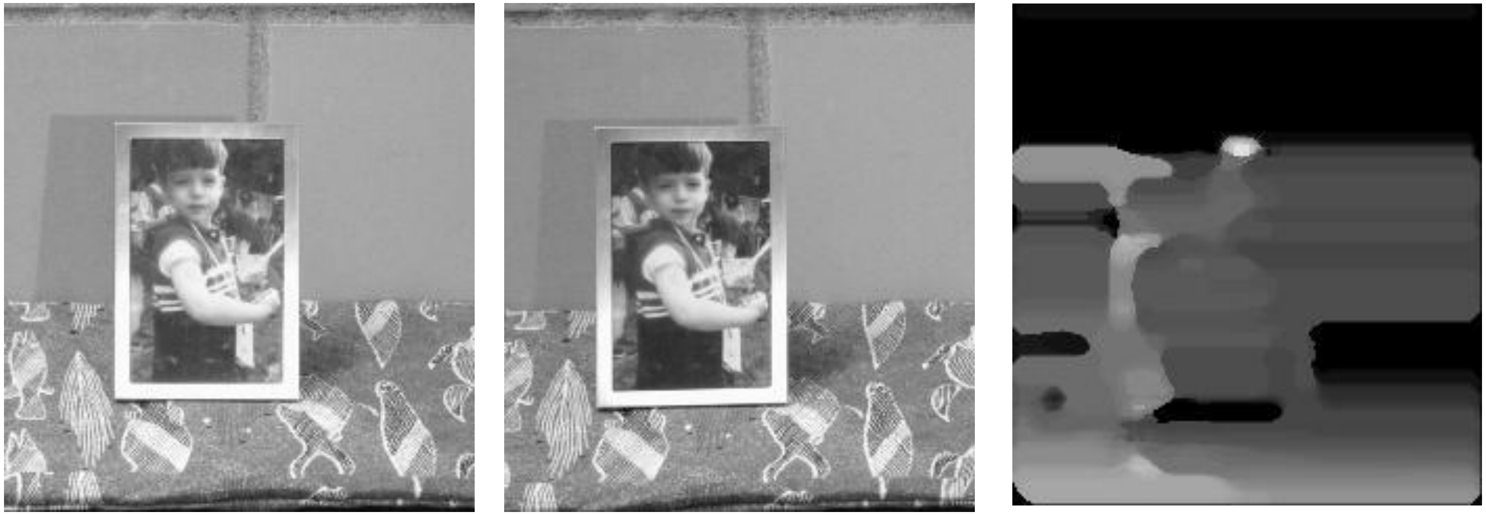
Limitations of similarity constraint



Textureless surfaces



Occlusions, repetition



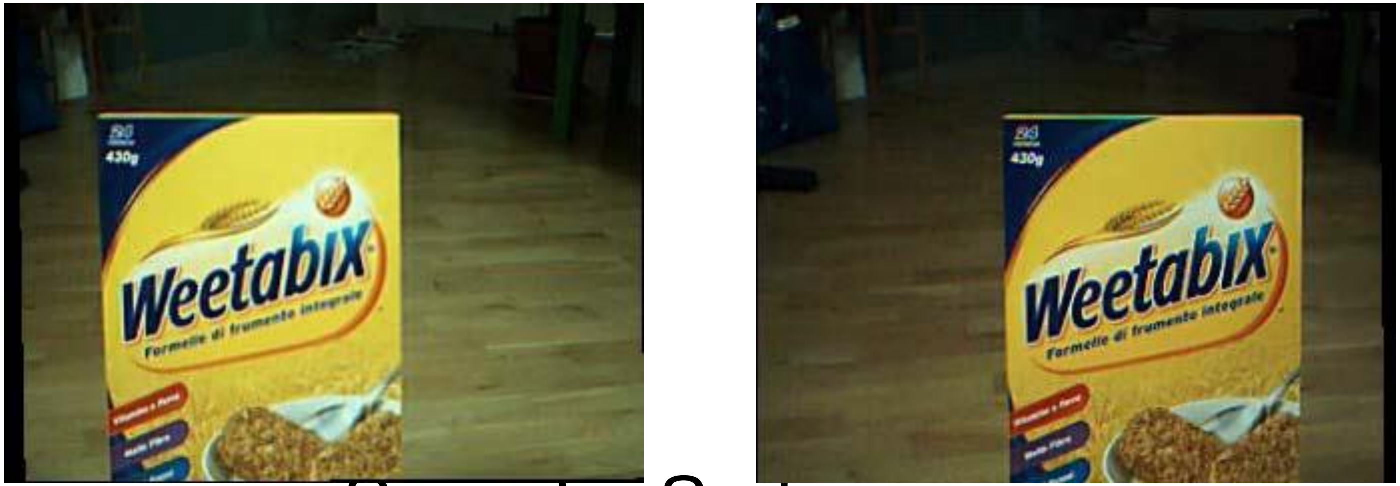
Non-Lambertian surfaces, specularities

Problems for Similarity Constraint

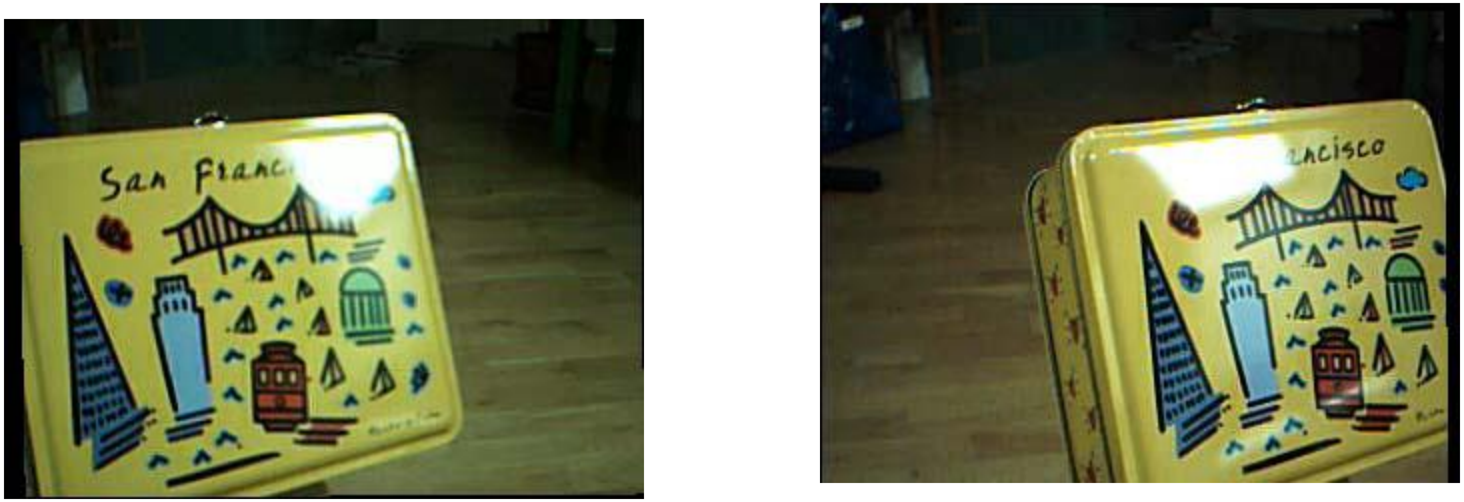
- Occlusions
  - Points with no counterpart in the other image
  - The wider the stereo baseline the more chance that there are occlusions (this should be obvious)
- Repetition
  - Elements are so similar can not tell them apart locally
- Textureless Surface
  - Nothing can be matched in these regions
- Non-Lambertian Surface – Specularities
  - Makes matching much more difficult (two views not similar)
- Such problems produce spurious matches
  - False correspondences created for the above reasons

Correspondence Difficulties

Photometric Distortion and Noise

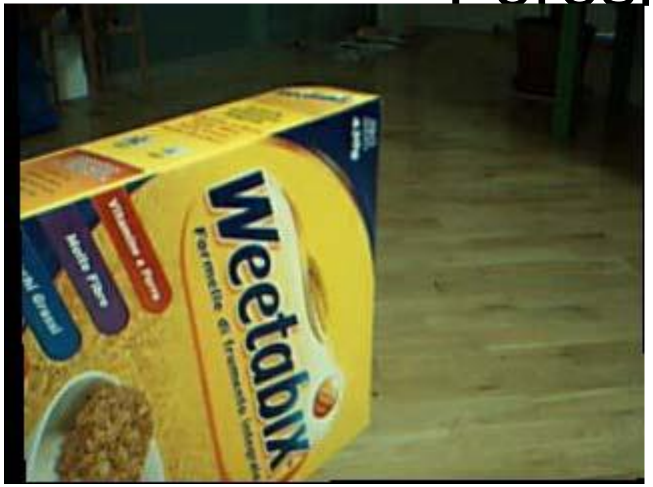

430g
Weetabix
Formelle di frumento integrale
430g
Weetabix
Formelle di frumento integrale

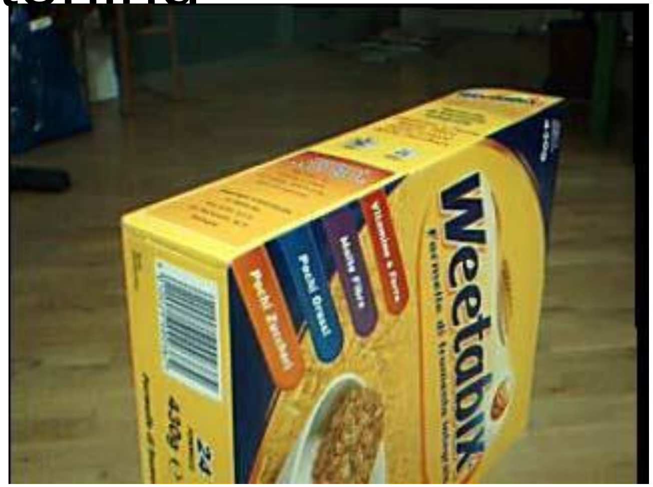
Specular Surfaces



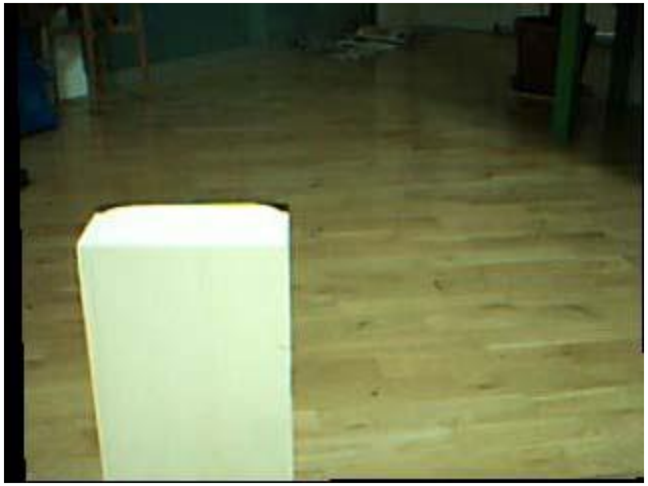
Correspondence Difficulties

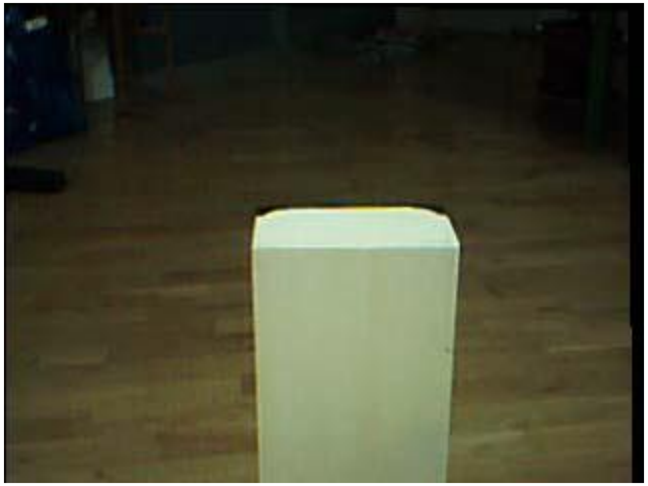
Foreshortening





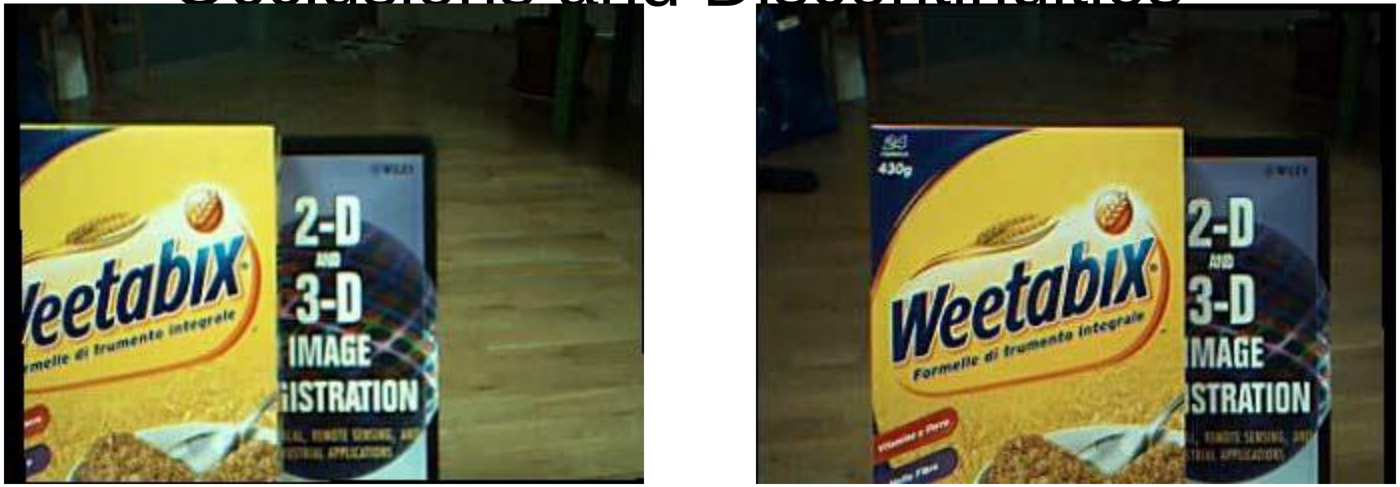
Uniform / Non-textured Surfaces



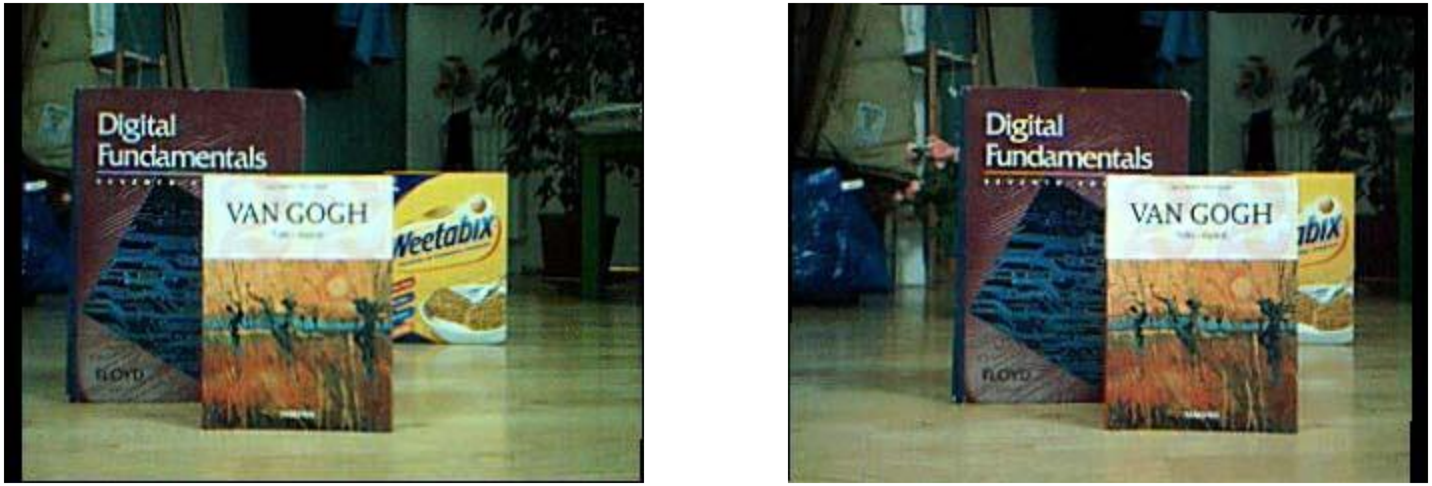


Correspondence Difficulties

Occlusions and Discontinuities


Weetabix
Formelle di frumento integrale
2-D AND 3-D IMAGE REGISTRATION
430g

Occlusions and Discontinuities

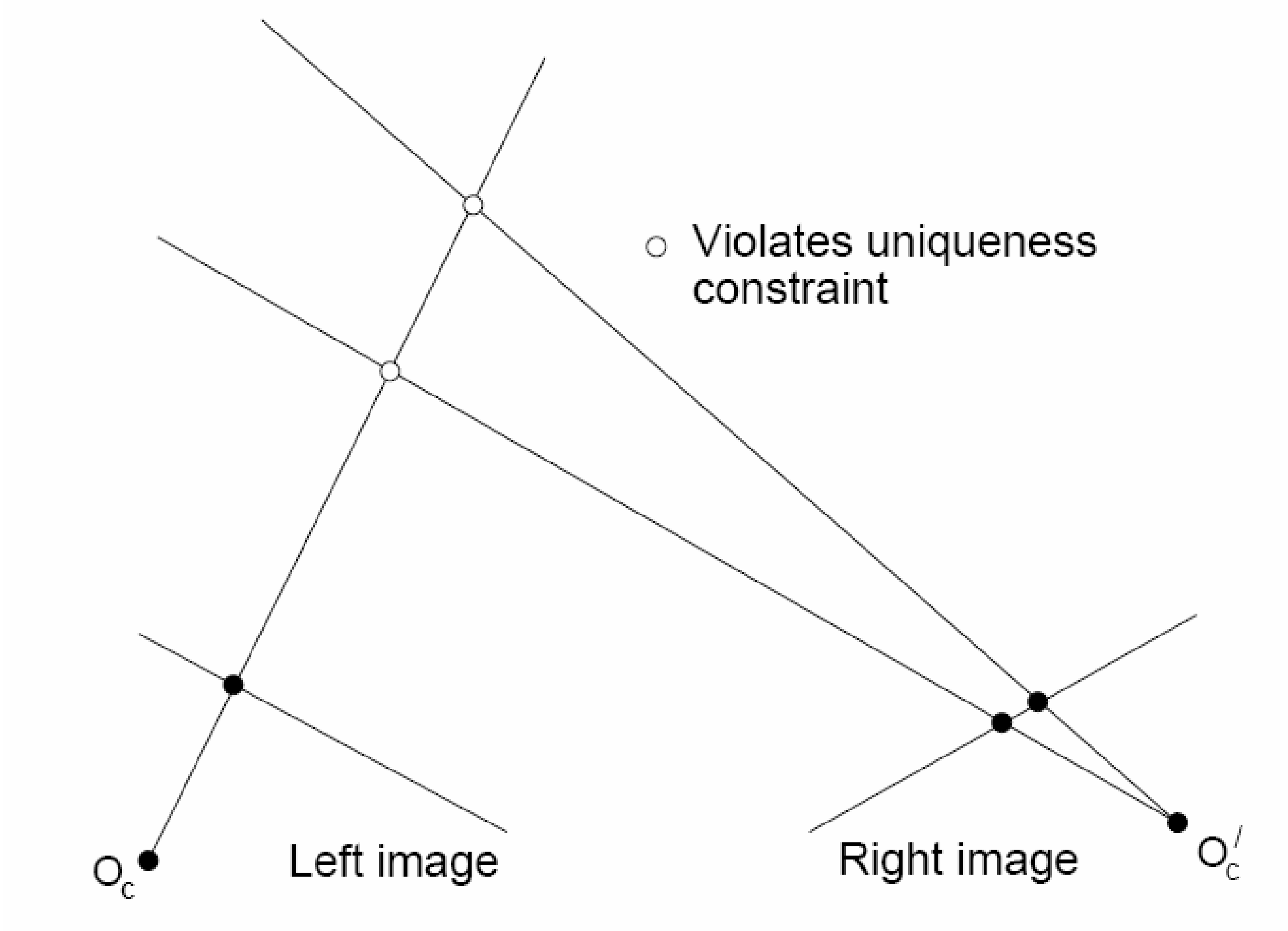


How can we improve window-based matching?

- The similarity constraint is **local** (each reference window is matched independently)
- Need to enforce **non-local** correspondence constraints
  - This means looking at the correspondences together
  - And then seeing if they make sense as a whole
- Then if correspondences are not consistent with each other we can make a decision
  - And likely discard some bad correspondences
- Another option is to use active sensors!
  - Project your own correspondences on the scene!
  - For example, shine a single laser beam around the scence

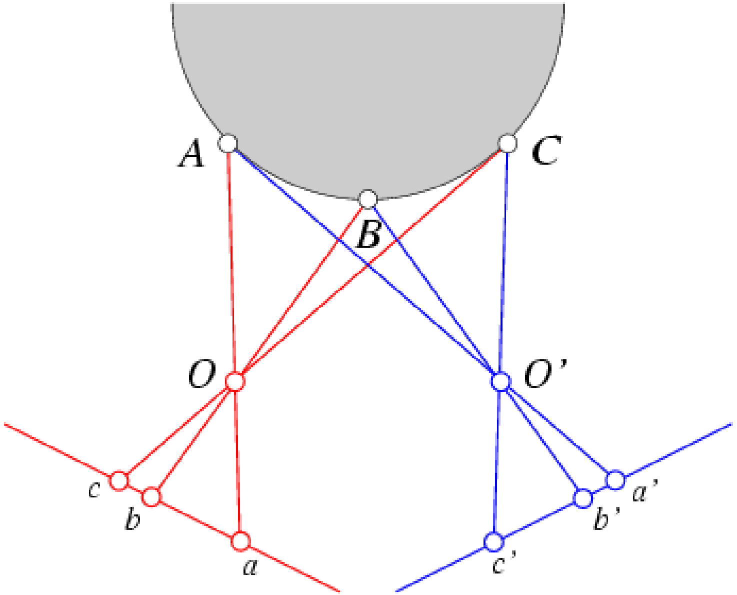
Non-local constraints

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image



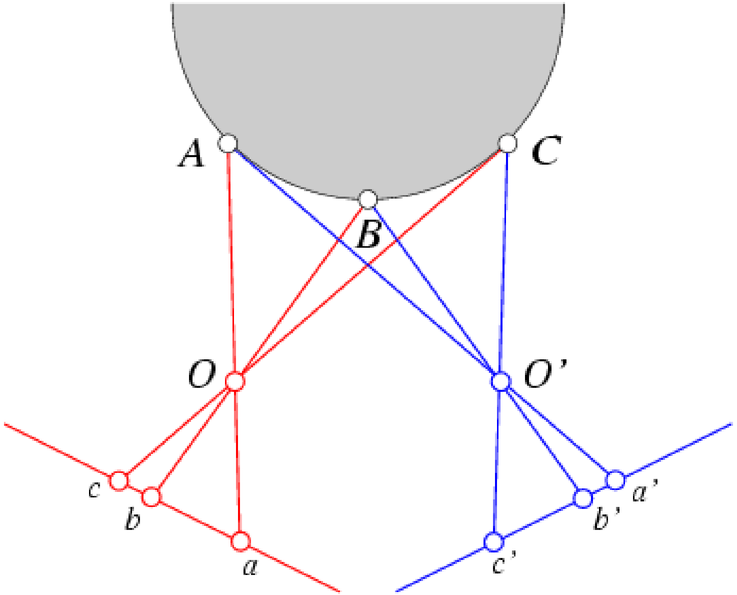
Non-local constraints

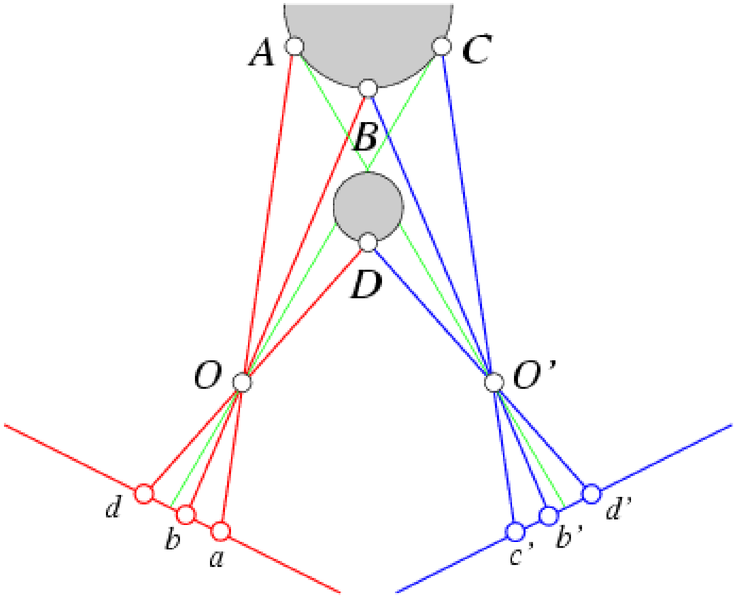
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views



Non-local constraints

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views





Ordering constraint doesn't always hold

Correlation Approach

PROS

- Easy to implement
- Produces dense disparity map
- Can be slow if implemented poorly

CONS

- Needs textured images to work well
- Inadequate for matching image pairs from very different viewpoints due to failure of similarity constraint
  - Poor for wide baseline matching
- Window may cover points with quite different disparities
- Inaccurate disparities on the occluding boundaries

Feature-based Approach

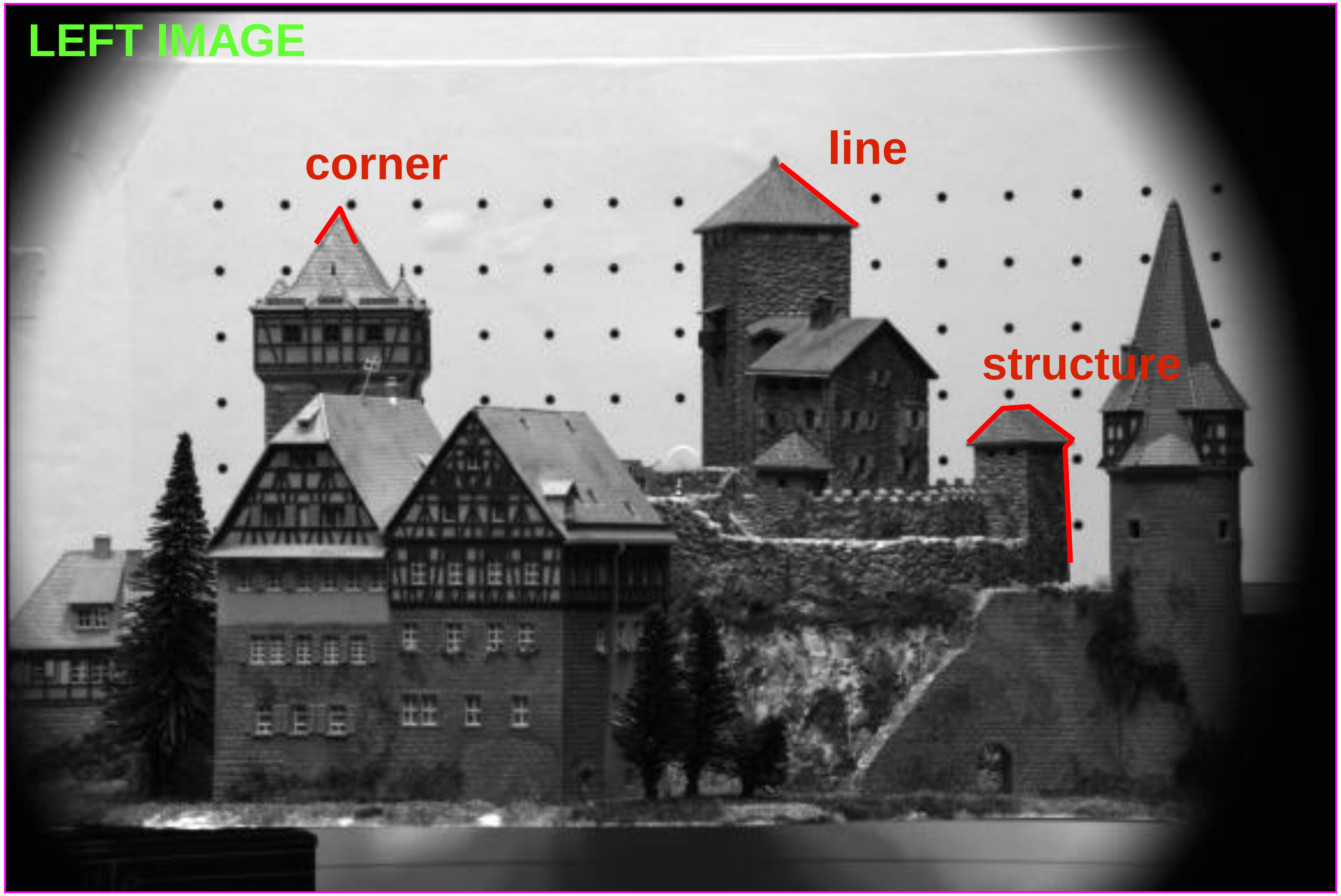
Features

- Edge points
- Lines (length, orientation, average contrast)
- Corners (including Harris, SURF, SIFT, etc.)

Matching algorithm

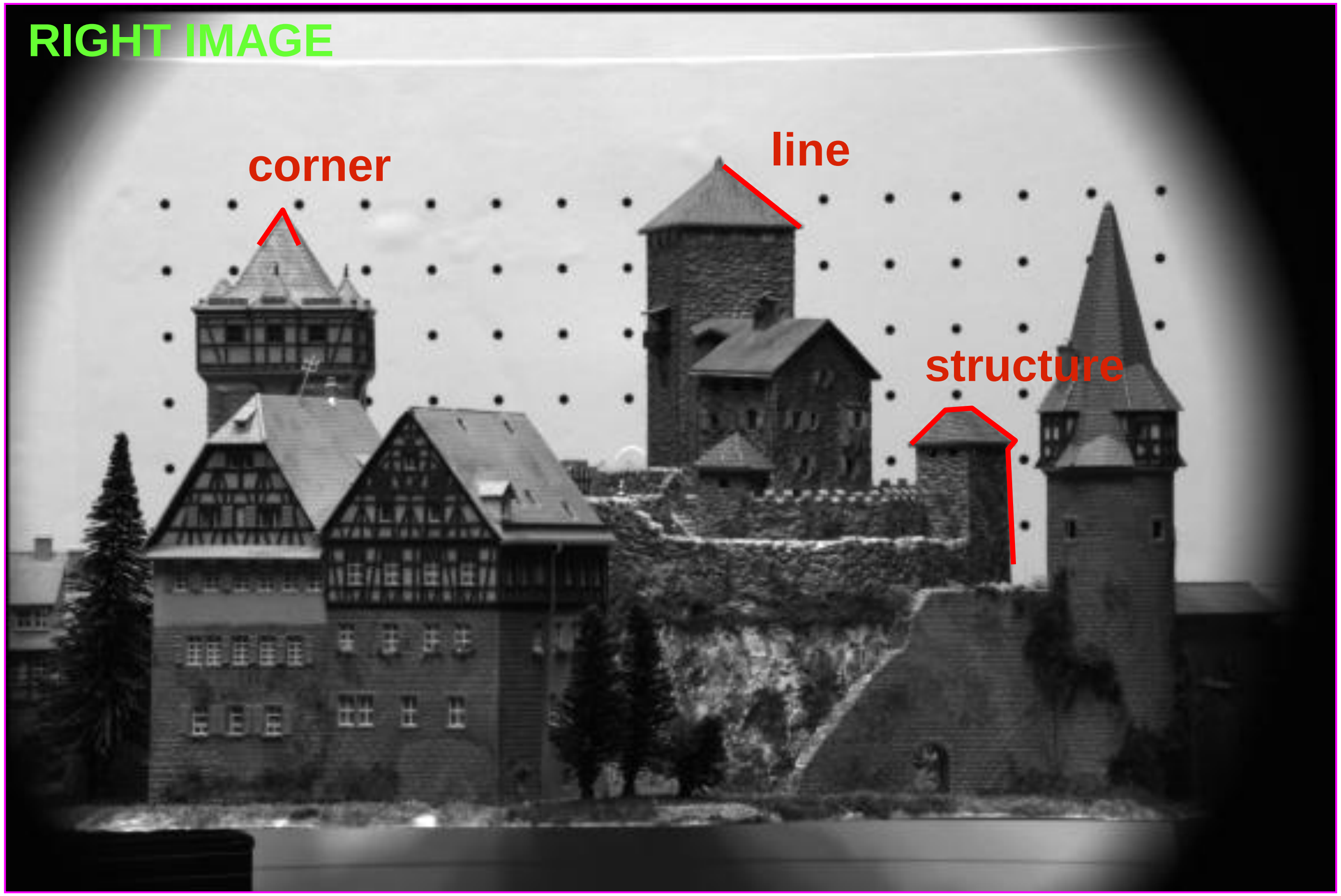
- Extract features in the stereo pair
- Define a suitable similarity measure for these features
- Use constraints to reduce number of matches
- Geometric constraints
  - Need only match features on same horizontal line
- Analytic constraints
  - Uniqueness – each feature has at most one match
  - Continuity – disparity varies continuously almost everywhere across this image

Feature-based Approach



For each feature in the left image…

Feature-based Approach



Search in the right image… the disparity (dx, dy) is the displacement when the similarity measure is maximum

Matching corner features

- For Harris corners usually just match small windows around the corner (using SSD)
- But SURF/SIFT corners have much more powerful and complex descriptor
- SURF descriptor is a 64 element float
  - Choosing best match between a set of SURF descriptors is a closest point problem (find_obj.cpp in OpenCV examples)
  - For simple stereo means finds closest (in terms of 64 elements) match in left and right image
- SURF descriptors have good invariance but are slow to match (can only handle hundreds)
  - Invariance means they work well for wide baseline images

Feature-based Approach

PROS

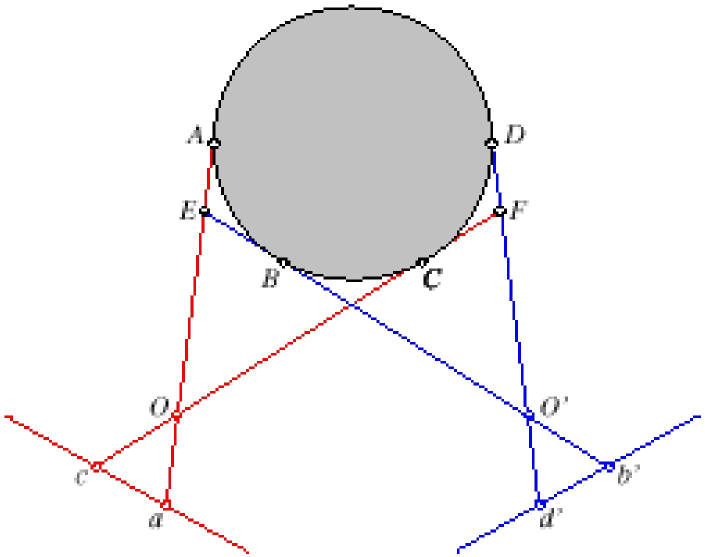
- Relatively insensitive to illumination changes
- Good for man-made scenes with strong lines but weak texture or textureless surfaces
- Work well on the occluding boundaries (edges)
- Could be faster than the correlation approach
- With SURF/SIFT features can work well for wider baseline

CONS

- Only produces a sparse depth map (usually hundreds of pts)
- Feature extraction is slow and matching is sometimes slower
  - Finding good feature descriptors is difficult
  - Lot of work done for matching SURF/SIFT features
  - But problem still difficult; i.e. what is similarity between two lines?

A Last Word on Correspondences

Correspondence fail for smooth surfaces



There is currently no good solution to the correspondence problem