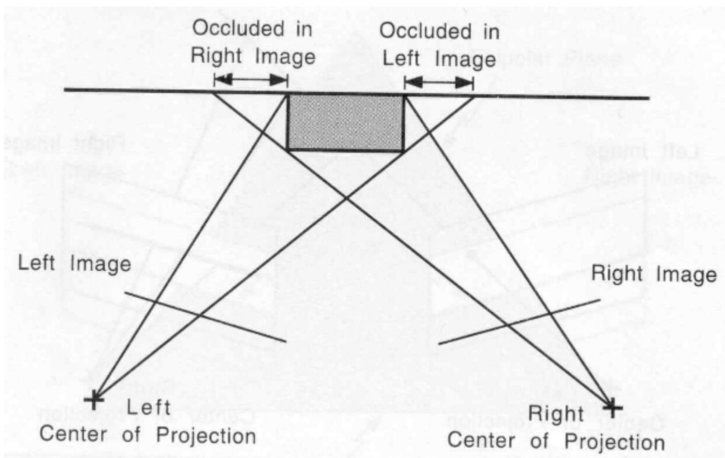
The correspondence problem

• **Why is the correspondence problem difficult?**

- Some points in each image will have no corresponding points in the other image.

(1) the cameras might have different fields of view.
(2) due to occlusion.

- A stereo system must be able to determine the image parts that should not be matched.



• **Methods for establishing correspondence**

- There are two issues to be considered:

  * how to select candidate matches?

  * how to determine the goodness of a match?

- Two main classes of algorithms:

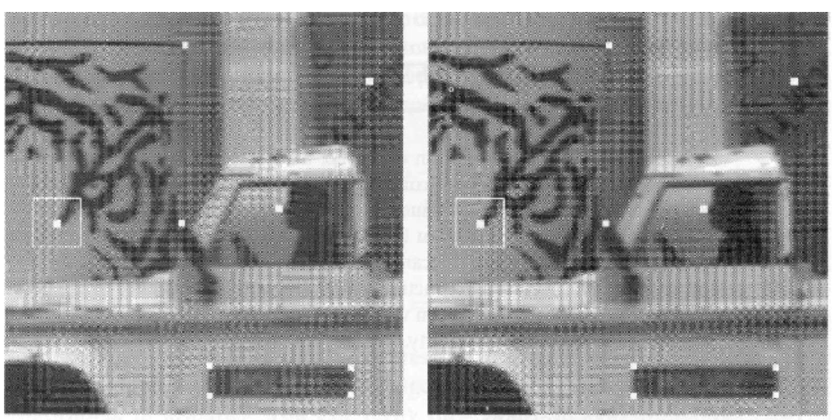
  Correlation-based: attempt to establish a correspondence by matching image intensities.

  Feature-based: attempt to establish a correspondence by matching a sparse sets of image features.

Correlation-based Methods

- Match image subwindows between the two images using *image correlation* (the oldest technique for finding the correspondence between pixels of two images).

- Scene points must have the same intensity in each image (strictly accurate for perfectly matte surfaces only).



Algorithm

Inputs:

(1) I_l and I_r

(2) the width of the subwindow $2W + 1$

(3) the search region in the right image $R(p_l)$ associated with a pixel p_l in the left image

For each pixel $p_l = (i, j)$ in the left image:

1. for each displacement $d = (d_1, d_2) \in R(p_l)$ compute

$$c(d) = \sum_{k=-W}^{W} \sum_{l=-W}^{W} I_l(i + k, j + l) I_r(i + k - d_1, j + l - d_2)$$

(cross-correlation)

2. the disparity of p_l is the vector $\bar{d} = (\bar{d}_1, \bar{d}_2)$ that maximizes $c(d)$ over $R(p_r)$

$$\bar{d} = arg \max_{d \in R}[c(d)]$$

- Usually, we normalize $c(d)$ by dividing it by the standard deviation of both I_l and I_r (*normalized cross-correlation*, e.g., $\in [0,1]$)

$$\bar{c}(d) = \frac{\sum_{k=-W}^{W} \sum_{l=-W}^{W} (I_l(i + k, j + l) - \bar{I}_l)(I_r(i + k - d_1, j + l - d_2) - \bar{I}_r)}{\sqrt{\sum_{k=-W}^{W} \sum_{l=-W}^{W} (I_l(i + k, j + l) - \bar{I}_l)^2 \sum_{k=-W}^{W} \sum_{l=-W}^{W} (I_r(i + k - d_1, j + l - d_2) - \bar{I}_r)^2}}$$

where $\bar{I}_l$ and $\bar{I}_r$ are the average pixel values in the left and right windows.

- An alternative similarity measure is the *sum of squared differences* (SSD):

$$c(d) = -\sum_{k=-W}^{W} \sum_{l=-W}^{W} (I_l(i + k, j + l) - (I_l(i + k - d_1, j + l - d_2))^2$$

• Improvements

- Instead of using the image intensity values, the accuracy of correlation is improved by using *thresholded signed gradient magnitudes* at each pixel.

  - Compute the gradient magnitude at each pixel in the two images without smoothing.

  - Map the gradient magnitude values into three values: -1, 0, 1 (i.e., by thresholding the gradient magnitiude)

  - More sensitive correlations are produced this way.

• Some comments

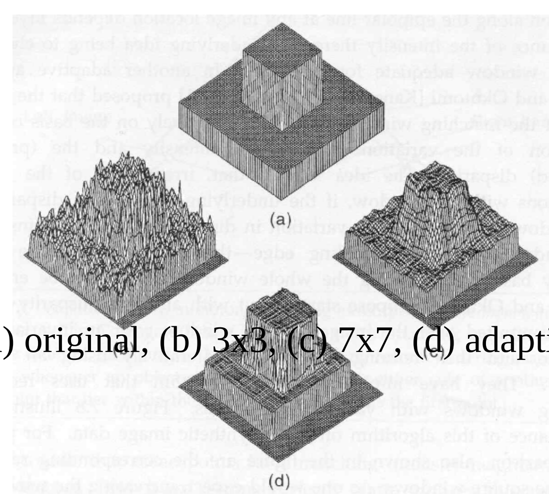
- The success of correlation-based methods depends on whether the image window in one image exhibits a distinctive structure that occurs infrequently in the search region of the other image.

- How to choose the size of the window (i.e., W)?

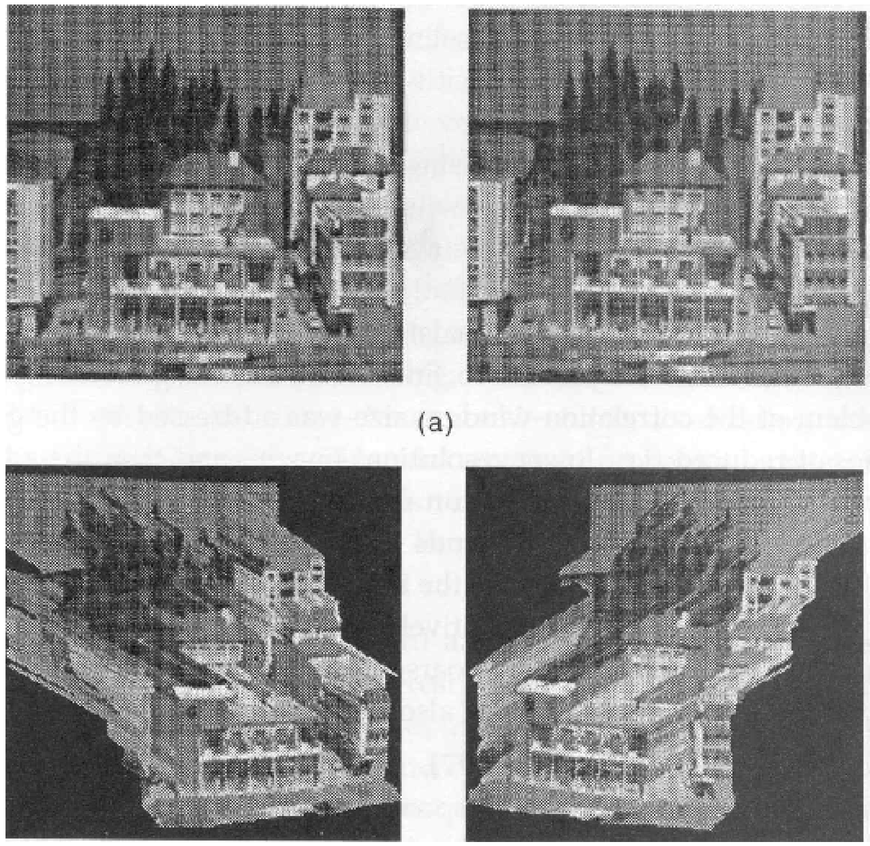
  * too small a window may not capture enough image structure, and may be too noise sensitive (i.e., many false matches).

  * too large a window makes matching less sensitive to noise (desired) but also to actual variations of image intensity (undesired -- it causes discontinuities in the disparity map).

  * an *adaptive searching window* has been proposed in the literature.



((a) original, (b) 3x3, (c) 7x7, (d) adaptive)



(3D info recovered using adaptive window)

- How to choose the size and location of $R(p_l)$?

  * if the distance of the fixating point from the cameras is much larger than the baseline, the location of $R(p_l)$ can be chosen to be the same as the location of p_l.

  * the size of $R(p_l)$ can be estimated from the maximum range of distances we expect to find in the scene.

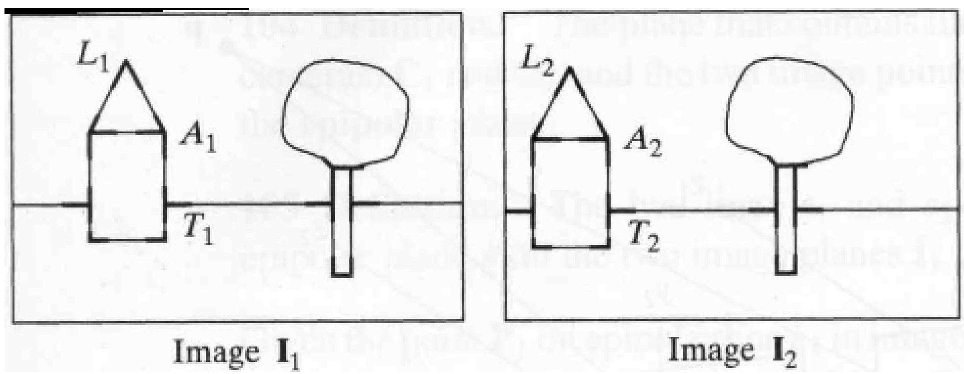
  * we will see that the search region **can always be reduced to a line !!**

Feature-based Methods

- Look for a feature in an image that matches a feature in the other.

- Typical features used are:

    * edge points
    * line segments
    * corners



- A set of *features* is used for matching; a line feature descriptor, for example, could contain:

    * the length, l
    * the orientation, θ
    * the coordinates of the midpoint, m
    * the average intensity along the line, i

- Similarity measures are based on matching feature descriptors:

$$S = \frac{1}{w_0(l_l - l_r)^2 + w_1(\theta_l - \theta_r)^2 + w_2(m_l - m_r)^2 + w_3(i_l - i_r)^2}$$

where w_0, ..., w_3 are weights (determining the weights that yield the best matches is a nontrivial task).

Algorithm

Inputs:

(1) I_l and I_r
(2) features and their descriptors in both images
(3) the search region in the right image $R(f_l)$ associated with a feature f_l in the left image

For each feature f_l in the left image:

1. Compute the similarity between f_l and each image feature in $R(f_l)$
2. Select the right-image feature f_r, that maximizes the similarity measure.
3. Save the correspondence and disparity $d(f_l, f_r)$

• Correlation-based vs feature-based approaches

Correlation-based methods

- Easier to implement than feature-based methods.

- Provide a dense disparity map (useful for reconstructing surfaces).

- Need textured images to work well (many false matches otherwise).

- Don't work well when viewpoints are very different (due to forshortening and change in illumination direction).

Feature-based methods:

- Suitable when good features can be extracted from the scene.

- Faster than correlation-based methods.

- Provide sparse disparity maps (OK for applications like visual navigation)

- Relatively insensitive to illumination changes.

• Structured lighting

- Feature-based methods are not applicable when the objects have smooth surfaces (i.e., sparse disparity maps make surface reconstruction difficult).

- Patterns of light are projected onto the surface of objects, creating interesting points even in regions which would be otherwise smooth.

- Finding and matching such points is simplified by knowing the geometry of the projected patterns.