3. Computer Vision 2
3.1. Corner
Corner Detection [1]

- Basic idea: Find points where two edges meet—i.e., high gradient in two directions
- “Cornerness” is undefined at a single pixel, because there’s only one gradient per point
  - Look at the gradient behavior over a small window
- Categories image windows based on gradient statistics
  - **Constant**: Little or no brightness change
  - **Edge**: Strong brightness change in single direction
  - **Flow**: Parallel stripes
  - **Corner/spot**: Strong brightness changes in orthogonal directions
Corner Detection: Analyzing Gradient Covariance

- Intuitively, in corner windows both $I_x$ and $I_y$ should be high
  - Can’t just set a threshold on them directly, because we want rotational invariance

- Analyze distribution of gradient components over a window to differentiate between types from previous slide:

  $C = \begin{pmatrix}
  \sum I_x^2 & \sum I_x I_y \\
  \sum I_x I_y & \sum I_y^2
  \end{pmatrix}$

- The two eigenvectors and eigenvalues $\lambda_1$, $\lambda_2$ of $C$ (Matlab: eig(C)) encode the predominant directions and magnitudes of the gradient, respectively, within the window

- Corners are thus where min($\lambda_1$, $\lambda_2$) is over a threshold
The Derivation of the Harris Corner Detector [2]

• The Harris corner detector is based on the local auto-correlation function of a signal; where the local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions.

• Given a shift \((\Delta x, \Delta y)\) and a point \((x, y)\), the auto-correlation function is defined as,

\[
c(x, y) = \sum_{W}[I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2
\]  

(1)

where \(I(\cdot, \cdot)\) denotes the image function and \((x_i, y_i)\) are the points in the window \(W\) (Gaussian\(^1\)) centered on \((x, y)\).

\(^1\)For clarity in exposition the Gaussian weighting factor has been omitted from the derivation.
The Derivation of the Harris Corner Detector [2]

The shifted image is approximated by a Taylor expansion truncated to the first order terms,

\[
I(x_i + \Delta x, y_i + \Delta y) \approx I(x_i, y_i) + [I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}
\]  \tag{2}

where \( I_x(\cdot, \cdot) \) and \( I_y(\cdot, \cdot) \) denote the partial derivatives in \( x \) and \( y \), respectively.
The Derivation of the Harris Corner Detector [2]

• Substituting approximation Eq. (2) into Eq. (1) yields,

\[ c(x, y) = \sum_{w} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \]

\[ = \sum_{w} \left( I(x_i, y_i) - I(x_i, y_i) - [I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \]

\[ = \sum_{w} \left( -[I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \]

\[ = \sum_{w} \left( [I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \]

\[ = [\Delta x \ \Delta y] \left[ \sum_{w} \frac{(I_x(x_i, y_i))^2}{\sum_{w} I_x(x_i, y_i)I_y(x_i, y_i)} \sum_{w} I_x(x_i, y_i)I_y(x_i, y_i) \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \]

\[ = [\Delta x \ \Delta y] C'(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \]

where matrix \( C(x, y) \) captures the intensity structure of the local neighborhood.
Harris Detector: Mathematics

$M = \text{앞 페이지 } C$

Classification of image points using eigenvalues of $M$:

- **Corner**: $\lambda_1$ and $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$; $E$ increases in all directions
- **Edge**: $\lambda_1 > \lambda_2$
- **Flat** region: $\lambda_1$ and $\lambda_2$ are small; $E$ is almost constant in all directions

$E = \text{앞 페이지 } C$
Measure of corner response:

\[ R = \det M - k \left( \text{trace } M \right)^2 \]

\[ \det M = \lambda_1 \lambda_2 \]
\[ \text{trace } M = \lambda_1 + \lambda_2 \]

\( (k - \text{empirical constant, } k = 0.04-0.06) \)

- The trace of a matrix is the sum of its eigenvalues, making it an invariant with respect to a change of basis.
Harris Detector: Mathematics

- $R$ depends only on eigenvalues of $M$
- $R$ is large for a corner
- $R$ is negative with large magnitude for an edge
- $|R|$ is small for a flat region

\[
\begin{align*}
\lambda_1 & > 0 \\
\lambda_2 & < 0
\end{align*}
\]

\[
\begin{align*}
R & > 0 \\
|R| & \text{small} \\
R & < 0
\end{align*}
\]
Harris Detector

• The Algorithm:
  – Find points with large corner response function $R$ ($R > \text{threshold}$)
  – Take the points of local maxima of $R$
Harris Detector: Workflow
Harris Detector: Workflow

Compute corner response $R$
Harris Detector: Workflow

Find points with large corner response: $R > \text{threshold}$
Harris Detector: Workflow

Take only the points of local maxima of $R$
Harris Detector: Workflow
Harris Detector Code [3]

dx = [-1 0 1; -1 0 1; -1 0 1];  % Derivative masks
dy = dx';
Ix = conv2(im, dx, 'same');  % Image derivatives
Iy = conv2(im, dy, 'same');

% Generate Gaussian filter of size 6*sigma (+/- 3sigma) and of
% minimum size 1x1.
g = fspecial('gaussian',max(1,fix(6*sigma)), sigma);
Ix2 = conv2(Ix.^2, g, 'same'); % Smoothed squared image derivatives
Iy2 = conv2(Iy.^2, g, 'same');
Ixy = conv2(Ix.*Iy, g, 'same');

cim = (Ix2.*Iy2 - Ixy.^2)./(Ix2 + Iy2 + eps); % My preferred measure.

% k = 0.04;
% cim = (Ix2.*Iy2 - Ixy.^2) - k*(Ix2 + Iy2).^2; % Original Harris measure.
Harris Detector Code [3]
References

3.2. Edge
Why extract edges? [1]

- Edges and lines are used in
  - object recognition
  - image matching (e.g., stereo, mosaics)
  - document analysis
  - horizon detection
  - line following robots
  - and many more apps

- More compact than pixels
Where do edges come from? [1]

Edges in images are caused by a variety of factors:

- surface normal discontinuity
- depth discontinuity
- surface color discontinuity
- illumination discontinuity
Images as functions [1]

Edges look like steep cliffs
Images as functions [1]

The gradient of an image:

\[ \nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \]

\[ \nabla f = [\frac{\partial f}{\partial x}, 0] \]

\[ \nabla f = [0, \frac{\partial f}{\partial y}] \]

The gradient points in the direction of most rapid increase in intensity

The gradient direction is given by:

\[ \theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right) \]

- how does this relate to the direction of the edge?

The edge strength is given by the gradient magnitude

\[ ||\nabla f|| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \]
The Discrete Gradient [1]

How can we differentiate a *digital* image $F[x, y]$?

- Answer: take discrete derivative ("finite difference")

$$\frac{\partial f}{\partial x}[x, y] \approx F[x + 1, y] - F[x, y]$$

How would you implement this as a cross-correlation?

$$H$$

-1  +1
The Sobel Operator [1]

Better approximations of the derivatives exist

- The Sobel operators below are very commonly used

\[
\begin{array}{ccc}
\frac{1}{8} & -1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{array}
\quad
\begin{array}{ccc}
\frac{1}{8} & 1 & 2 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{array}
\]

- The standard defn. of the Sobel operator omits the $1/8$ term
  - doesn’t make a difference for edge detection
  - the $1/8$ term is needed to get the right gradient value, however
Edge detection using the Sobel operator [1]

original image        Sobel gradient magnitude        thresholded
Effects of noise [1]

Consider a single row or column of the image
- Plotting intensity as a function of position gives a signal

\[ f(x) \]

\[ \frac{d}{dx} f(x) \]

Where is the edge?
Solution: smooth first [1]

Where is the edge? Look for peaks in $\frac{\partial}{\partial x}(h \ast f)$
Derivative theorem of convolution [1]

\[ \frac{\partial}{\partial x} (h \ast f) = \left( \frac{\partial}{\partial x} h \right) \ast f \]

This saves us one operation:

\( f \)

\( \frac{\partial}{\partial x} h \)

\( \left( \frac{\partial}{\partial x} h \right) \ast f \)

Need to find (local) maxima of a function
Laplacian of Gaussian [1]

Consider \( \frac{\partial^2}{\partial x^2} (h \ast f) \)

\[ f \]

\[ \frac{\partial^2}{\partial x^2} h \]

\[ (\frac{\partial^2}{\partial x^2} h) \ast f \]

Where is the edge? Zero-crossings of bottom graph
2D edge detection filters \[1\]

\[
h_\sigma(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2 + v^2}{2\sigma^2}}
\]

Gaussian

\[
\frac{\partial}{\partial x} h_\sigma(u, v)
\]

derivative of Gaussian (x direction)

\[
\nabla^2 h_\sigma(u, v)
\]

Laplacian of Gaussian

\[\nabla^2 \text{ is the Laplacian operator:}\]

\[
\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]
LoG filter [1]

Laplacian of Gaussian

\[ \nabla^2 h_\sigma(u, v) \]

Discrete approximation with \( \sigma = 1.4 \)
Canny edge detector [1]

1. **Smoothing**: Smooth the image with a Gaussian filter with spread $\sigma$
2. **Gradient**: Compute gradient magnitude and direction at each pixel of the smoothed image
3. **Thresholding**: Threshold the gradient magnitude image such that strong edges are kept and noise is suppressed
4. **Non-maximum suppression (thinning)**: Zero out all pixels that are not the maximum along the direction of the gradient (look at 1 pixel on each side)
5. **Tracing edges**: Trace high-magnitude contours and keep only pixels along these contours, so weak little segments go away
Canny edge detector: step 4 [1]

Check if pixel is local maximum along gradient direction
- requires checking interpolated pixels p and r
Canny edge detector [1]

\[ \| \nabla f \| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \]

norm of the gradient of smoothed image

LoG result
Canny edge detector [1]

The choice of $\sigma$ depends on desired behavior

- large $\sigma$ detects large scale edges
- small $\sigma$ detects fine features
3.3. Binocular Stereo
Stereo Vision [1]
Stereo Vision [1]

- **Basic Principle: Triangulation**
  - Gives reconstruction as intersection of two rays
  - Requires
    - calibration
    - *point correspondence*
Point Correspondence [1]

Given p in left image, where can the corresponding point p’ in right image be?
The Simplest Case: Recti-linear Configuration [1]

- Image planes of cameras are parallel.
- Focal points are at same height.
- Focal lengths same.
- Then, epipolar lines are horizontal scan lines.
The Simplest Case: Recti-linear Configuration [3]

Image plane

\[ P' = (X', Y', Z') \]

\[ P = (X, Y, Z) \]

\[ Z' = -f \quad X' = -f \frac{X}{Z} \quad Y' = -f \frac{Y}{Z} \]

\[ x = -X' \quad y = -Y' \]

\[ (X, Y, Z) \rightarrow (x, y, 1) = (f \frac{X}{Z}, f \frac{Y}{Z}, 1) \]
The Simplest Case: Rectilinear Configuration [3]

Derive expression for $Z$ as a function of $x_1, x_2, f, B$
The Simplest Case: Rectilinear Configuration [3]

\[ \begin{align*}
    x_1 &= -f \frac{X_1}{Z_1}, \\
    x_2 &= -f \frac{X_1 + B}{Z_1} = x_1 - f \frac{B}{Z_1} \\
    \Rightarrow Z_1 &= \frac{fB}{x_1 - x_2}
\end{align*} \]
The Simplest Case: Recti-linear Configuration [1]

\[ (x_l, y_l) = \left( f \frac{X}{Z}, f \frac{Y}{Z} \right) \]
\[ (x_r, y_r) = \left( f \frac{(X-T)}{Z}, f \frac{Y}{Z} \right) \]

Disparity:
\[ d = x_l - x_r = f \frac{X}{Z} - f \frac{(X-T)}{Z} \]

Then given \( Z \), we can compute \( X \) and \( Y \).

\( T \) is the stereo baseline
\( d \) measures the difference in retinal position between corresponding points
Stereo Constraints [1]

- Image plane
- Focal plane
- Epipolar Line
- Epipole

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung
Epipolar Geometry and Fundamental Matrix [1]

• The geometry of two different images of the same scene is called the *epipolar geometry*.

• The geometric information that relates two different viewpoints of the same scene is entirely contained in a mathematical construct known as *fundamental matrix*.
Baseline and Epipolar Plane [1]

- **Baseline**: Line joining camera centers $C$, $C'$
- **Epipolar plane** $\pi$: Defined by baseline and scene point $X$

from Hartley & Zisserman
Epipoles and Epipolar Lines [1]

- **Epipolar lines** $l, l'$: Intersection of epipolar plane $\pi$ with image planes
- **Epipoles** $e, e'$: Where baseline intersects image planes
  - Equivalently, the image in one view of the other camera center.

![Diagram of epipoles and epipolar lines](from Hartley & Zisserman)
Epipolar Pencil [1]

- As position of $X$ varies, epipolar planes “rotate” about the baseline (like a book with pages)
  - This set of planes is called the **epipolar pencil**
- Epipolar lines “radiate” from epipole—this is the **pencil of epipolar lines**

![Diagram of epipolar pencil](from Hartley & Zisserman)
Epipolar Constraints [1]

- Camera center \( C \) and image point \( x \) define ray in 3-D space that projects to epipolar line \( l' \) in other view (since it’s on the epipolar plane)
- 3-D point \( X \) is on this ray, so image of \( X \) in other view \( x' \) must be on \( l' \)
- In other words, the epipolar geometry defines a mapping \( x \rightarrow l' \), of points in one image to lines in the other

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung
The Fundamental Matrix [1]

- Mapping of point in one image to epipolar line in other image $\mathbf{x} \rightarrow \mathbf{l}'$ is expressed algebraically by the \textbf{fundamental matrix} $\mathbf{F}$

- Write this as $\mathbf{l}' = \mathbf{F} \mathbf{x}$

- Since $\mathbf{x}'$ is on $\mathbf{l}'$, by the point-on-line definition we know that $\mathbf{x}'^T \mathbf{l}' = 0$

- Substitute $\mathbf{l}' = \mathbf{F} \mathbf{x}$, we can thus relate corresponding points in the camera pair $(\mathbf{P}, \mathbf{P}')$ to each other with the following:

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$$
The Fundamental Matrix [2]

In computer vision, the fundamental matrix $F$ is a $3 \times 3$ matrix which relates corresponding points in stereo images. In epipolar geometry, with homogeneous image coordinates, $\mathbf{x}$ and $\mathbf{x}'$, of corresponding points in a stereo image pair, $F\mathbf{x}$ describes a line (an epipolar line) on which the corresponding point $\mathbf{x}'$ on the other image must lie. That means, for all pairs of corresponding points holds

$$\mathbf{x}'^T F \mathbf{x} = 0.$$ 

Being of rank two and determined only up to scale, the fundamental matrix can be estimated given at least seven point correspondences. Its seven parameters represent the only geometric information about cameras that can be obtained through point correspondences alone.

The above relation which defines the fundamental matrix was published in 1992 by both Faugeras and Hartley.

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung
Longuet-Higgins' essential matrix satisfies a similar relationship, the essential matrix is a metric object pertaining to calibrated cameras, while the fundamental matrix describes the correspondence in more general and fundamental terms of projective geometry. This is captured mathematically by the relationship between a fundamental matrix $\mathbf{F}$ and its corresponding essential matrix $\mathbf{E}$, which is

$$\mathbf{E} = \mathbf{K}'^T \mathbf{F} \mathbf{K},$$

$\mathbf{K}$ and $\mathbf{K}'$ being the intrinsic calibration matrices of the two images involved
Computing Fundamental Matrix [1]

Fundamental Matrix is singular with rank 2

In principal F has 7 parameters up to scale and can be estimated from 7 point correspondences

Direct Simpler Method requires 8 correspondences
Pseudo Inverse-Based [1]

\[ u^T F u' = 0 \]

Each point correspondence can be expressed as a linear equation

\[
\begin{bmatrix}
F_{11} & F_{12} & F_{13} \\
F_{21} & F_{22} & F_{23} \\
F_{31} & F_{32} & F_{33}
\end{bmatrix}
\begin{bmatrix}
u' \\
v'
\end{bmatrix}
= 0
\]

\[
\begin{bmatrix}
u u' & u v' & u' v & v v' & v' & u' & v' & 1
\end{bmatrix}
\begin{bmatrix}
F_{11} \\
F_{12} \\
F_{13} \\
F_{21} \\
F_{22} \\
F_{23} \\
F_{31} \\
F_{32} \\
F_{33}
\end{bmatrix} = 0
\]
Pseudo Inverse-Based [1]

8 corresponding points, 8 equations.

\[
\begin{pmatrix}
u_1 u'_1 & u_1 v'_1 & u_1 & v_1 u'_1 & v_1 v'_1 & v_1 & u'_1 & v'_1 \\
u_2 u'_2 & u_2 v'_2 & u_2 & v_2 u'_2 & v_2 v'_2 & v_2 & u'_2 & v'_2 \\
u_3 u'_3 & u_3 v'_3 & u_3 & v_3 u'_3 & v_3 v'_3 & v_3 & u'_3 & v'_3 \\
u_4 u'_4 & u_4 v'_4 & u_4 & v_4 u'_4 & v_4 v'_4 & v_4 & u'_4 & v'_4 \\
u_5 u'_5 & u_5 v'_5 & u_5 & v_5 u'_5 & v_5 v'_5 & v_5 & u'_5 & v'_5 \\
u_6 u'_6 & u_6 v'_6 & u_6 & v_6 u'_6 & v_6 v'_6 & v_6 & u'_6 & v'_6 \\
u_7 u'_7 & u_7 v'_7 & u_7 & v_7 u'_7 & v_7 v'_7 & v_7 & u'_7 & v'_7 \\
u_8 u'_8 & u_8 v'_8 & u_8 & v_8 u'_8 & v_8 v'_8 & v_8 & u'_8 & v'_8 \\
\end{pmatrix}
\begin{pmatrix}
F_{11} \\
F_{12} \\
F_{13} \\
F_{21} \\
F_{22} \\
F_{23} \\
F_{31} \\
F_{32}
\end{pmatrix}
= 
\begin{pmatrix}
1 \\
1 \\
1 \\
1 \\
1 \\
1 \\
1 \\
1
\end{pmatrix}
\]

Invert and solve for \( \mathcal{F} \).

(Use more points if available; find least-squares solution to minimize
\[
\sum_{i=1}^{n} (p_i^T \mathcal{F} p'_i)^2
\] )
The Eight-Point Algorithm [1]

• Input: n point correspondences (n >= 8)
  - Construct homogeneous system $A\mathbf{x} = 0$ from:
    - $\mathbf{x} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})$: entries in $F$
    - Each correspondence gives one equation
    - $A$ is a nx9 matrix (in homogenous format)
  - Obtain estimate $\hat{F}$ by SVD of $A$
    - $x$ (up to a scale) is column of $V$ corresponding to the least singular value
  - Enforce singularity constraint: since Rank ($F$) = 2
    - Compute SVD of $\hat{F}$
    - Set the smallest singular value to 0: $D \rightarrow D'$
    - Correct estimate of $F$: $F' = UD'V^T$
• Output: the estimate of the fundamental matrix, $F'$
• Similarly we can compute $E$ given intrinsic parameters
Locating the Epipoles from F [1]

- Input: Fundamental Matrix F
  - Find the SVD of F
  - The epipole $e_l$ is the column of V corresponding to the null singular value (as shown above)
  - The epipole $e_r$ is the column of U corresponding to the null singular value
- Output: Epipole $e_l$ and $e_r$

\[ \bar{p}_r^T F \bar{p}_l = 0 \]
\[ \bar{p}_r^T F \bar{e}_l = 0 \]
\[ F \bar{e}_l = 0 \]

$e_l$ lies on all the epipolar lines of the left image

True For every $p_r$

F is not identically zero

\[ F = UDV^T \]
Corner Detection [5]

```matlab
thresh = 500;  % Harris corner threshold

% Find Harris corners in image1 and image2
[cim1, r1, c1] = harris(im1, 1, thresh, 3);
[cim2, r2, c2] = harris(im2, 1, thresh, 3);
```
Correlation-Based Matching [5]

dmax = 50;  % Maximum search distance for matching
w = 11;    % Window size for correlation matching

% Use normalised correlation matching
[m1,m2] = matchbycorrelation(im1, [r1';c1'], im2, [r2';c2'], w, dmax);

% Display putative matches
show(im1,3), set(3,'name','Putative matches')

for n = 1:length(m1);
    line([m1(2,n) m2(2,n)], [m1(1,n) m2(1,n)])
end

Putative matches
RANSAC-Based Fundamental Matrix Estimation [5]

% Assemble homogeneous feature coordinates for fitting of the fundamental matrix, note that [x,y] corresponds to [col, row]
x1 = [m1(2,:); m1(1,:); ones(1,length(m1))];
x2 = [m2(2,:); m2(1,:); ones(1,length(m1))];

% Distance threshold for deciding outliers

Inlying matches
Epipolar Lines [5]

% Step through each matched pair of points and display the % corresponding epipolar lines on the two images.

\[ l_2 = F \cdot x_1; \quad \text{Epipolar lines in image2} \]
\[ l_1 = F' \cdot x_2; \quad \text{Epipolar lines in image1} \]

% Solve for epipoles
\[
[U, D, V] = \text{svd}(F);
\]
e1 = hnormalise(V(:,3));
e2 = hnormalise(U(:,3));

for n = inliers
    figure(1), clf, imshow(im1), hold on, plot(x1(1,n),x1(2,n),'r+');
    hline(l1(:,n)); plot(e1(1), e1(2), 'g*');

    figure(2), clf, imshow(im2), hold on, plot(x2(1,n),x2(2,n),'r+');
    hline(l2(:,n)); plot(e2(1), e2(2), 'g*');
end
Epipolar Lines [5]
Estimation of Camera Matrix [6]

• Once the essential matrix is known, the camera matrices may be retrieved from $E$. In contrast with the fundamental matrix case, where there is a projective ambiguity, the camera matrices may be retrieved from the essential matrix up to scale and a four-fold ambiguity.

• A 3x3 matrix is an essential matrix if and only if two of its singular values are equal, and the third is zero.

• We may assume that the first camera matrix is $P=[[I|0]]$. In order to compute the second camera matrix, $P'$, it is necessary to factor $E$ into the product $SR$ of a skew-symmetric matrix and a rotation matrix.

• Suppose that the SVD of $E$ is $U\text{diag}(1,1,0)V^T$. Using the notation of $W$ and $Z$, there are (ignoring signs) two possible factorizations $E=SR$ as follows:

$$S=UZU^T, R=UWV^T \text{ or } UW^TV^T$$

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$ZW = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$E=SR=(UZU^T)(UWV^T) = UZ(U^TU)WV^T = UZWV^T = U \text{ diag}(1,1,0) V^T$$
Estimation of Camera Matrix [7]

\[ S = UZU^T, \ R = UWV^T \text{ or } UW^TV^T \]

- For a given essential matrix \( E = U \text{diag}(1,1,0)V^T \), and first camera matrix \( P = [I|0] \), there are four possible choices for the second camera matrix \( P' \), namely

\[ P' = [UWV^T|u_3] \text{ or } [UWV^T|-u_3] \text{ or } [UW^TV^T|u_3] \text{ or } [UW^TV^T|-u_3] \]

Where, \( u_3 \) is the last column of \( U \).

The four possible solutions for calibrated reconstruction from \( E \).
Linear triangulation methods [8]

• In each image we have a measurement $x = PX$, $x' = P'X$, and these equations can be combined into a form $AX = 0$, which is an equation linear in $X$.

• The homogeneous scale factor is eliminated by a cross product to give three equations for each image point, of which two are linearly independent.

• For the first image, $x \times (PX) = 0$ and writing this out gives

\[
\begin{bmatrix}
0 & -1 & y \\
1 & 0 & -x \\
-y & x & 0
\end{bmatrix}
\begin{bmatrix}
p_{1T}X \\
p_{2T}X \\
p_{3T}X
\end{bmatrix} = 0
\]

Where, $p^iT$ are the rows of $P$.

\[
A = \begin{bmatrix}
xp_{3T} - p_{1T} \\
yp_{3T} - p_{2T} \\
x'p_{3T} - p_{1T} \\
y'p_{3T} - p_{2T}
\end{bmatrix}
\]

\[x(p_{3T}X) - (p_{1T}X) = 0\]
\[y(p_{3T}X) - (p_{2T}X) = 0\]
\[x(p_{2T}X) - y(p_{1T}X) = 0\]
Linear triangulation methods [8]

• An equation of the form $AX=0$ can then be composed, with

$$A = \begin{bmatrix}
    x p^{3T} - p^{1T} \\
    y p^{3T} - p^{2T} \\
    x' p^{i3T} - p^{i1T} \\
    y' p^{i3T} - p^{i2T}
\end{bmatrix}$$

Where two equations have been included from each image, giving a total of four equations in four homogeneous unknowns. This is a redundant set of equations, since the solution is determined only up to scale.

$X$ can be calculated by SVD of $A$. 
Fundamental matrix estimation methods

\[ x_{cam} = PX \]
\[ x_{cam} \times (PX) = 0 \]
\[ x'_{cam} = P'X \]
\[ x'_{cam} \times (P'X) = 0 \]

\[
\begin{align*}
x p^{3T} X - p^{1T} X &= 0 \\
y p^{3T} X - p^{2T} X &= 0 \\
x p^{2T} X - y p^{1T} X &= 0 \\
x' p^{3T} X - p'^{1T} X &= 0 \\
y' p^{3T} X - p'^{2T} X &= 0 \\
x' p^{2T} X - y' p'^{1T} X &= 0
\end{align*}
\]

\[ AX = 0 \quad \text{where} \quad A = \begin{bmatrix} x p^{3T} & - p^{1T} \\ y p^{3T} & - p^{2T} \\ x' p^{3T} & - p'^{1T} \\ y' p^{3T} & - p'^{2T} \end{bmatrix} \]
Stereo Rectification [1]

How can we make images as in recti-linear configuration?

→ Stereo Rectification

• Image Reprojection
  - reproject image planes onto common plane parallel to line between optical centers

• Notice, only focal point of camera really matters
References

3.4. Stereo Matching
Stereo Vision [1]

Triangulate on two images of the same point to recover depth.
- Feature matching across views
- Calibrated cameras

Matching correlation windows across scan lines
Reduction of Searching by Epipolar Constraint [1]

- Epipolar Constraint
  - Matching points lie along corresponding epipolar lines
  - Reduces correspondence problem to 1D search along conjugate epipolar lines
  - Greatly reduces cost and ambiguity of matching
Photometric Constraint [1]

Same world point has same intensity in both images.

– True for Lambertian surfaces
  • A Lambertian surface has a brightness that is independent of viewing angle
– Violations:
  • Noise
  • Specularity
  • Non-Lambertian materials
  • Pixels that contain multiple surfaces
Photometric Constraint [1]

For each epipolar line
  For each pixel in the left image
  • compare with every pixel on same epipolar line in right image
  • pick pixel with minimum match cost

This leaves too much ambiguity, so:
  Improvement: match windows
Correspondence Using Correlation [1]

Left

Right

scanline

SSD error

disparity
Sum of Squared Difference (SSD) [1]

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

We define the window function:

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

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

$$C_r(x, y, d) = \sum_{(u, v) \in W_m(x, y)} [I_L(u, v) - I_R(u - d, v)]^2$$
Image Normalization [1]

- Even when the cameras are identical models, there can be differences in gain and sensitivity.
- For these reason and more, it is a good idea to normalize the pixels in each window:

\[
\hat{I}(x, y) = \frac{I(x, y) - \bar{I}}{\sqrt{\sum_{(u, v) \in W_m(x, y)} [I(u, v)]^2}}
\]

**Average pixel**

\[
\bar{I} = \frac{1}{|W_m(x, y)|} \sum_{(u, v) \in W_m(x, y)} I(u, v)
\]

**Window magnitude**

\[
\|I\|_{W_m(x, y)} = \sqrt{\sum_{(u, v) \in W_m(x, y)} [I(u, v)]^2}
\]

**Normalized pixel**
Images as Vectors [1]

Each window is a vector in an $m^2$ dimensional vector space. Normalization makes them unit length.

“Unwrap” image to form vector, using raster scan order.
Image Metrics [1]

(Normalized) Sum of Squared Differences
\[ C_{SSD}(d) = \sum_{(u,v) \in W_m(x,y)} [(\hat{I}_L(u,v) - \hat{I}_R(u-d,v))]^2 \]
\[ = \| w_L - w_R(d) \|^2 \]

Normalized Correlation
\[ C_{NC}(d) = \sum_{(u,v) \in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v) \]
\[ = w_L \cdot w_R(d) = \cos \theta \]

\[ d^* = \arg \min_d \| w_L - w_R(d) \|^2 = \arg \max_d w_L \cdot w_R(d) \]
Stereo Result [1]

Left

Disparity Map

Images courtesy of Point Grey Research

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung
Window Size [1]

- Effect of window size
- Some approaches have been developed to use an adaptive window size (try multiple sizes and select best match)

Better results with *adaptive window*

Ordering Constraint [3]

- If an object a is left on an object b in the left image then object a will also appear to the left of object b in the right image.

Ordering constraint... ...and its failure

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung
Smooth Surface Problem [3]

- Correspondence fail for smooth surfaces

- There is currently no good solution to the correspondence problem
Occlusion [1]

Diagram showing the concepts of left and right occlusion with scanlines and matching points.
Search over Correspondence [1]

Three cases:
- Sequential – add cost of match (small if intensities agree)
- Occluded – add cost of no match (large cost)
- Disoccluded – add cost of no match (large cost)
Dynamic programming yields the optimal path through grid. This is the best set of matches that satisfy the ordering constraint.
Dynamic Programming [3]

Local errors may be propagated along a scan-line and no inter scan-line consistency is enforced.
Search in the right image... the disparity \((dx, dy)\) is the displacement when the similarity measure is maximum.
**Segment-Based Stereo Matching [3]**

**Assumption**

- Depth discontinuity tend to correlate well with color edges
- Disparity variation within a segment is small
- Approximating the scene with piece-wise planar surfaces
Segment-Based Stereo Matching [3]

• Plane equation is fitted in each segment based on initial disparity estimation obtained SSD or Correlation

• Global matching criteria: if a depth map is good, warping the reference image to the other view according to this depth will render an image that matches the real view

• Optimization by iterative neighborhood depth hypothesizing
Segment-Based Stereo Matching [3]

Hai Tao and Harpreet W. Sawhney

E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung
Segment-Based Stereo Matching [3]
Smoothing by MRF [2]
Smoothing by MRF [4]

(a) Raw low-res depth map  (b) Raw low-res 3D model  (c) Image mapped onto 3D model
Smoothing by MRF [4]

(d) MRF high-res depth map
(e) MRF high-res 3D model
(f) Image mapped onto 3D model
Stereo Testing and Comparison [1]
Stereo Testing and Comparison [1]

Window-based matching (best window size)  Ground truth
Stereo Testing and Comparison [1]

State of the art method

Ground truth


E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung
Intermediate View Reconstruction [1]
Intermediate View Reconstruction [1]
References

3.5. Optical Flow
Motion in Computer Vision

Motion

• Structure from motion
• Detection/segmentation with direction

[1]
Motion Field v.s. Optical Flow [2], [3]

**Motion Field:** an ideal representation of 3D motion as it is projected onto a camera image.

**Optical Flow:** the approximation (or estimate) of the motion field which can be computed from time-varying image sequences. Under the simplifying assumptions of 1) Lambertian surface, 2) pointwise light source at infinity, and 3) no photometric distortion.
Motion Field [2]

• An ideal representation of 3D motion as it is projected onto a camera image.
• The time derivative of the image position of all image points given that they correspond to fixed 3D points. “field : position → vector”

• The motion field $\mathbf{v}$ is defined as

$$
\mathbf{v} = f \frac{Z \mathbf{V} - V_z \mathbf{P}}{Z^2}
$$

where $\mathbf{V} = -\mathbf{T} - \omega \times \mathbf{P}$

$\mathbf{P}$ is a point in the scene where $Z$ is the distance to that scene point.
$\mathbf{V}$ is the relative motion between the camera and the scene,
$\mathbf{T}$ is the translational component of the motion,
and $\omega$ is the angular velocity of the motion.
Motion Field [5]

3D point \( P (X,Y,Z) \) and 2D point \( p (x,y) \), focal length \( f \)

\[
p = f \frac{P}{Z} \quad \begin{cases} 
  x = f \frac{X}{Z} \\
  y = f \frac{Y}{Z} 
\end{cases} \tag{1}
\]

Motion field \( v \) can be obtained by taking the time derivative of (1)

\[
\begin{align*}
  v_x &= f \left( \frac{V_x}{Z} - X \frac{V_z}{Z^2} \right) \\
  v_y &= f \left( \frac{V_y}{Z} - Y \frac{V_z}{Z^2} \right) 
\end{align*} \tag{2}
\]
The motion of 3D point $P, V$ is defined as flow

$$ V = -T - \omega \times P $$

$$
\begin{align*}
V_x &= -T_x - \omega_y Z + \omega_z Y \\
V_y &= -T_y - \omega_z X + \omega_x Z \\
V_z &= -T_z - \omega_x Y + \omega_y X
\end{align*}
$$

By substituting (3) into (2), the basic equations of the motion field is acquired

$$
\begin{align*}
v_x &= \frac{T_x x - T_x f}{Z} - \omega_y f + \omega_z y + \frac{\omega_x x y}{f} - \frac{\omega_y x^2}{f} \\
v_y &= \frac{T_y y - T_y f}{Z} + \omega_x f - \omega_z x - \frac{\omega_y x y}{f} + \frac{\omega_x y^2}{f}
\end{align*}
$$
Motion Field [5]

The motion field is the sum of two components, one of which depends on translation only, the other on rotation only.

\[
\begin{align*}
    v_x &= \frac{T_Z x - T_X f}{Z} - \omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f} \\
    v_y &= \frac{T_Z y - T_Y f}{Z} + \omega_x f - \omega_z x - \frac{\omega_y xy}{f} + \frac{\omega_x y^2}{f}
\end{align*}
\]  

(4)

Translational components

Rotational components

\[
\begin{align*}
    v_x &= -\omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f} \\
    v_y &= +\omega_x f - \omega_z x - \frac{\omega_y xy}{f} + \frac{\omega_x y^2}{f}
\end{align*}
\]  

(5) (6)
Motion Field: Pure Translation [2]

If there is no rotational motion, the resulting motion field has a peculiar spatial structure.

If (5) is regarded as a function of 2D point position,

\[ \begin{align*}
    v_x &= \frac{T_Z x - T_x f}{Z} \\
    v_y &= \frac{T_Z y - T_y f}{Z}
\end{align*} \]  \quad (5)

If \((x_0, y_0)\) is defined as in (6)

\[ \begin{align*}
    x_0 &= f \frac{T_X}{T_Z} \\
    y_0 &= f \frac{T_Y}{T_Z}
\end{align*} \]  \quad (6)

\[ \begin{align*}
    v_x &= \frac{T_Z}{Z} \left( x - f \frac{T_X}{T_Z} \right) \\
    v_y &= \frac{T_Z}{Z} \left( y - f \frac{T_Y}{T_Z} \right)
\end{align*} \]  \quad (7)
Motion Field: Pure Translation [2]

Equation (7) say that the motion field of a pure translation is radial. In particular, if $T_Z<0$, the vectors point away from $p_0 (x_0, y_0)$, which is called the focus of expansion (FOE). If $T_Z>0$, the motion field vectors point towards $p_0$, which is called the focus of contraction. If $T_Z=0$, from (5), all the motion field vectors are parallel.

$$
\begin{align*}
    v_x &= \frac{T_Z x - T_X f}{Z} \\
    v_y &= \frac{T_Z y - T_Y f}{Z}
\end{align*}
$$

(5)

$$
\begin{align*}
    v_x &= -f \frac{T_X}{Z} \\
    v_y &= -f \frac{T_Y}{Z}
\end{align*}
$$

(8)
Motion Field: Motion Parallax [6]

Equation (8) say that their lengths are inversely proportional to the depth of the corresponding 3D points.

\[
\begin{align*}
    v_x &= -f \frac{T_x}{Z} \\
    v_y &= -f \frac{T_y}{Z}
\end{align*}
\] (8)

This animation is an example of parallax. As the viewpoint moves side to side, the objects in the distance appear to move more slowly than the objects close to the camera [6].
Motion Field: Motion Parallax [2]

If two 3D points are projected into one image point, that is coincident, rotational component will be the same. Notice that the motion vector $V$ is about camera motion.

\[
\begin{align*}
    v_x &= \frac{T_Z x - T_x f}{Z} - \frac{\omega_y y f + \omega_z y + \frac{\omega_x y f}{f}}{f} - \frac{\omega_x x^2}{f} \\
    v_y &= \frac{T_Z y - T_y f}{Z} + \frac{\omega_x x f - \omega_z x - \frac{\omega_y y f}{f}}{f} + \frac{\omega_x y^2}{f}
\end{align*}
\] (4)
The difference of two points’ motion field will be related with translation components. And, they will be radial w.r.t FOE or FOC.

\[
\begin{align*}
\Delta v_x &= (T_Z x - T_X f) \left( \frac{1}{Z_1} - \frac{1}{Z_2} \right) = (x - x_0) T_Z \left( \frac{1}{Z_1} - \frac{1}{Z_2} \right) \\
\Delta v_y &= (T_Z y - T_Y f) \left( \frac{1}{Z_1} - \frac{1}{Z_2} \right) = (y - y_0) T_Z \left( \frac{1}{Z_1} - \frac{1}{Z_2} \right)
\end{align*}
\]
Motion Parallax

The relative motion field of two instantaneously coincident points:

1. Does not depend on the rotational component of motion
2. Points towards (away from) the point $p_0$, the vanishing point of the translation direction.
Motion Field: Pure Rotation w.r.t Y-axis [7]

If there is no translation motion and rotation w.r.t x- and z- axis, from (4)

\[
\begin{align*}
  v_x &= -\omega_y f - \frac{\omega_y x^2}{f} \\
  v_y &= \frac{\omega_y xy}{f}
\end{align*}
\]

Fig. 2. (a) Motion field of a pure translation ($T_x$ and $T_z$). (b) Motion field of a pure rotation ($R_y$). The crosses and circles represent the feature points in the first and the second images, respectively, and the lines join the matches.
Motion Field: Pure Rotation w.r.t Y-axis [7]

Translational Motion

Distance to the point, Z, is constant.

Rotational Motion

Distance to the point, Z, is changing. According to Z, y is changing, too.
Motion Field: Optical Flow [4]

For a 2D+1 dimensional case (3D or n-D cases are similar) a voxel at location \((x,y,t)\) with intensity \(I(x,y,t)\) will have moved by \(\delta x\), \(\delta y\), and \(\delta t\) between the two image frames, and the following image constraint equation can be given:

\[
\hat{I}(x,y,t) = \hat{I}(x + \delta x, y + \delta y, t + \delta t)
\]

Assuming the movement to be small, the image constraint at \(I(x,y,t)\) with Taylor series can be developed to get:

\[
I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + H.O.T.
\]

where H.O.T. means higher order terms, which are small enough to be ignored. From these equations it follows that:

\[
\frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t = 0
\]

or

\[
\frac{\partial I}{\partial x} \frac{\delta x}{\delta t} + \frac{\partial I}{\partial y} \frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} \frac{\delta t}{\delta t} = 0
\]

which results in

\[
\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0
\]

where \(V_x\), \(V_y\) are the \(x\) and \(y\) components of the velocity or optical flow of \(I(x,y,t)\) and \(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}\) are the derivatives of the image at \((x,y,t)\) in the corresponding directions. \(I_x, I_y, I_t\) can be written for the derivatives in the following.

Thus:

\[
I_x V_x + I_y V_y = -I_t
\]

or

\[
\nabla I^T \cdot \vec{V} = -I_t
\]

The Image Brightness Constancy Equation [2]
\[ \nabla I^T \cdot V = -I_t \]

**Assumption**

The image brightness is continuous and differentiable as many times as needed in both the spatial and temporal domain.

The image brightness can be regarded as a plane in a small area.
Motion Field: Lucas–Kanade Method [8]

Assuming that the optical flow \((V_x, V_y)\) is constant in a small window of size \(m \times m\) with \(m > 1\), which is centered at \((x, y)\) and numbering the pixels within as 1...\(n\), \(n = m^2\), a set of equations can be found:

\[
\nabla I^T \cdot \mathbf{V} = -I_t
\]

\[
\begin{pmatrix}
I_{x_1} V_x + I_{y_1} V_y = -I_{t_1} \\
I_{x_2} V_x + I_{y_2} V_y = -I_{t_2} \\
\vdots \\
I_{x_n} V_x + I_{y_n} V_y = -I_{t_n}
\end{pmatrix}
\]

With this there are more than two equations for the two unknowns and thus the system is over-determined. Hence:

\[
\begin{pmatrix}
I_{x_1} & I_{y_1} \\
I_{x_2} & I_{y_2} \\
\vdots & \vdots \\
I_{x_n} & I_{y_n}
\end{pmatrix}
\begin{pmatrix}
V_x \\
V_y
\end{pmatrix}
= 
\begin{pmatrix}
-I_{t_1} \\
-I_{t_2} \\
\vdots \\
-I_{t_n}
\end{pmatrix}
\]

or

\[
A \tilde{\mathbf{v}} = -\mathbf{b}
\]

To solve the over-determined system of equations, besides other methods, the least squares method can also be used:

\[
A^T A \tilde{\mathbf{v}} = A^T (-\mathbf{b}) \text{ or } \tilde{\mathbf{v}} = (A^T A)^{-1} A^T (-\mathbf{b})
\]

or

\[
\begin{pmatrix}
V_x \\
V_y
\end{pmatrix}
= \left[ \sum I_{x_i}^2 \quad \sum I_{x_i} I_{y_i} \quad \sum I_{y_i}^2 \right]^{-1} \left[ -\sum I_{x_i} I_{t_i} \quad -\sum I_{y_i} I_{t_i} \right]
\]

with the sums running from \(i = 1\) to \(n\).
Motion Field: Aperture Problem [2], [9]

The component of the motion field in the direction orthogonal to the spatial image gradient is not constrained by the image brightness constancy equation. ➔ Given local information can determine component of optical flow vector only in direction of brightness gradient.

The aperture problem. The grating appears to be moving down and to the right, perpendicular to the orientation of the bars. But it could be moving in many other directions, such as only down, or only to the right. It is impossible to determine unless the ends of the bars become visible in the aperture.
References


E-mail: hogijung@hanyang.ac.kr
http://web.yonsei.ac.kr/hgjung