CS4243 Computer Vision and Pattern Recognition

Owen Leong
owenleong@u.nus.edu

November 29, 2024

1 Images

Grayscale intensity

$$I = W_R \cdot R + W_G \cdot G + W_B \cdot B$$

$W_R = 0.299, W_G = 0.587, W_B = 0.114$
Normalized RGB: divide each component by $R + G + B$
HSV colour space

2 Processing

Adjusting brightness $x_{ij} = p_{ij} + b$
Image normalization $x_{ij} = \frac{p_{ij} - \mu}{\sigma}$
Gamma mapping $x_{ij} = 255 \cdot (\frac{p_{ij}}{255})^{\gamma}, \gamma > 0$
Histogram stretching (via min and max)
Histogram equalization $x_{ij} = Kc_{p_{ij}}$ $c_{p_{ij}}$ is the cdf value for p_{ij}
Histogram threshold for foreground/background separation
Otsu's method for automated thresholding

$$T^* = \arg\min_T (w_1(T)\sigma_1^2(T) + w_2(T)\sigma_2^2(T))$$

Convolution vs cross correlation
At boundary, do zero-padding, wrap around, copy edge, reflect across edge.
Normalized cross correlation: divide by magnitude of kernel and magnitude of input window

3 Edges

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

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

Derivative of gaussian filter
Laplace filter
Laplacian of gaussian filter
2d gaussian filter
Canny edge detector
Edge thinning using Non-maximum suppression: check in gradient direction to see if it is the maximum
Hysterisis thresholding: start with high threshold to start edge curves, and use a low threshold to continue growing them

4 Lines

Hough transform $x\cos\theta + y\sin\theta = \rho$
Hough circles $(x-a)^2 + (y-b)^2 = r^2$
Leveraging gradient information

5 Segments

5.1 Clustering using k-means

Initialize cluster centres randomly. Assign each point to the closest cluster centre. Set new cluster centre to be the mean of the points assigned to it

5.2 Superpixels

Simple Linear Iterative Clustering (SLIC) superpixels.
Use composite distance measure of colour distance and spatial distance.

- Initialize cluster centres on a grid (but move cluster centre to lowest gradient value in 3x3 neighbourhood)
- Compute distance from each cluster centre to all pixels in some neighbourhood (e.g. 2s x 2s)
- Set new cluster centre to mean of the points assigned to it

5.3 Mean shift algorithm

For each data point, compute a window around it, compute centroid. Shift centre of window to centroid and repeat.
Window size corresponds to bandwidth

6 Visual Textures

6.1 1D Gabor filter

$$e^{-\frac{x^2}{2\sigma^2}}\sin(2\pi f x)$$

Parameters σ determines rate of decay of exponential envelope and f determines frequency of sinusoid

6.2 2D Gabor filter

$$f = \frac{1}{2\pi\sigma^2} e^{-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}} \sin(\frac{2\pi x'}{\lambda} + \phi)$$

$x' = x\cos\omega + y\sin\omega, y' = -x\sin\omega + y\cos\omega$

$\lambda = \frac{1}{f}$ is the wavelength.
Can use sin or cos for even or odd functions.
γ determines whether gabor function is circular ($\gamma = 1$) or elliptical.
σ is stdev of gaussian envelope.
ϕ is phase of sinusoidal function.
ω is orientation of the stripes of Gabor function. $\omega = 0$ gives vertical stripes, $\omega = 90°$ gives horizontal stripes.

6.3 Textons

Filter bank responses used as representations. Cluster features from training images to form texton dictionary. For each region, describe it as histogram of textons. Use textures to identify boundaries using different in histogram of textons. Gabor filter is a gaussian multiplied by sinusoid.

7 Keypoints

Descriptor should change when movement is made in any direction, hence should detect corners.

7.1 Harris Corner Detector

Second moment matrix

$$H = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

$$A = \sum_W I_x^2, \quad B = \sum_W I_x I_y, \quad C = \sum_W I_y^2$$

$$E(u,v) = \sum_W [I(x+u, y+v) - I(x,y)]^2$$

$$\approx \begin{bmatrix} u & v \end{bmatrix} H \begin{bmatrix} u \\ v \end{bmatrix}$$

Good corner has $R = \min(\lambda_1, \lambda_2)$ large
Efficient approximation

$$R = \lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2 = \det H - \kappa \mathrm{tr}(H)^2$$

- Non-maximum suppression searches for max values and zeroes-out surrounding window.
- Adaptive non-maximum suppression picks corners which are local maxima and whose response is significantly greater than all neighbouring local maxima within some radius r.

Weighted window

$$H = \sum_W w_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

7.1.1 Equivariance and Invariance

Equivariant: transforming the image transforms the detection as well
Invariant: transforming the image does not change the detection
Harris corner detected location is equivariant to translation and response is invariant to translation.
Harris corner location is equivariant to rotation and response is invariant to rotation.
Invariant to additive change in intensity but not scaling of intensity.
Not equivariant to scaling.
Handle multi-scale by using a fixed window size but using a Gaussian pyramid to scale the original image.

8 Descriptors

Vector representations that characterize a region of the image.
Should be invariant and discriminative.
After applying Harris corners, find dominant gradient direction for the image patch and rotate patch according to this angle.

8.1 MOPS

Multi-scale oriented patches takes 40×40 windows around keypoint, subsamples every 5th pixel, rotates to horizontal by computing dominant edge direction, normal window by subtracting mean and dividing by stdev, then take wavelet transform to get 64-dimensional vector.

8.2 GIST

GIST descriptor divides image patch into 4×4 cells and applies Gabor filters. Divide into 4×4 cells (spatial partitioning) to retain spatial information. Result of descriptor is $4 \times 4 \times N$, where N is size of filterbank.

8.3 SIFT features

8.3.1 Multi-scale extrema (blob) detection

Estimate laplacian of gaussian response, using difference of gaussians, over different scales (different sigmas). Also done at different octaves (scaling of image). Local maxima if response is greater than spatial neighbours and neighbouring sigmas. Scale of the keypoint is based on sigma and octave.

8.3.2 Key-point localization

Refine locations of keypoint to sub-pixel accuracy. Reject maxima by threshold. Reject maxima where curvature is insufficient, by Hessian of different of gaussians response, since straight edges are not good for localization compared to corners. Ratio of eigenvalues.

8.3.3 Orientation assignment

Dominant edge gradient.

8.3.4 Descriptor

Take 16×16 window around keypoint, then partition into 4×4 grid of cells. Compute gradient orientations and magnitudes for each pixel, reweight magnitudes according to gaussian centred on keypoint and discard pixels with low magnitude. Of remaining edge orientations, create histogram with 8 bins for each cell. For rotation invariance, shift histogram binning by its dominant orientation (e.g. largest histogram value shifted to 0). Finally, collapse into a vector ($16 \times 8 = 128$ dimensions). Normalize vector to unit length, clamp values based on threshold and re-normalize.
To match two features, threshold ratio between best descriptor and second best descriptor. Best should be approx less than 0.7 times next best.

8.4 Precision / Recall

TP: correct match
FP: wrong match between two features
FN: pair of matching features not matched
TN: detected features not part of any feature pair
Precision: $\frac{TP}{TP+FP}$ how accurate are the features pairs declared as matches?
Recall: $\frac{TP}{TP+FN}$ was the algorithm able to find all the actual pairs of features?
Specificity: $\frac{TN}{TN+FP}$ can the algorithm correctly disregard the features which are not part of any pair?
Measure performance by Area under ROC (Receiver Operator Characteristic) curve, which plots true positive rate against false positive rate.

9 Homography

$x' = Hx$ and solve using DLT
Data normalization:

$$T = \begin{bmatrix} s & 0 & -sc_x \\ 0 & s & -sc_y \\ 0 & 0 & 1 \end{bmatrix}, s = \frac{\sqrt{2}}{d}$$

where c is centroid, d is mean distance of all points from centroid
Apply DLT to correspondences $\tilde{x}_i \leftrightarrow \tilde{x}'_i$ and set $H = T'^{-1}\tilde{H}T$
Apply RANSAC
Warping to resample

9.1 RANSAC

$$N = \frac{\log(1-p)}{\log(1-(1-e)^s)}$$

N is number of iterations
e is probability that point is outlier
s is number of points needed for the sample for one fitting
p is probability that at least one set of points does not contain any outliers

10 Optical Flow

Brightness constancy assumption:

$$I(x(t), y(t), t) = C$$

Small motion assumption:

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

$$I_x u + I_y v + I_t = 0$$

Approximate I_x, I_y using sobel filter or any spatial derivative.
Approximate I_t using frame differencing.

10.1 Lucas-kanade flow

Assume that an image patch has constant flow, e.g. 5x5 image patch

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ \dots & \dots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ \dots \\ I_t(p_{25}) \end{bmatrix}$$

$$A^T A = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

λ_1/λ_2 should not be too large.
Aperture problem, where within the small window the movement is ambiguous. Does not contain corner points. Least squares problem is not well conditioned.
If actual motion is not small enough, small motion assumption is violated causing aliasing. Solution is to run coarse-to-fine optical flow. Compute optical flow at lower resolution first, then warp image, then compute at next resolution, etc.

10.2 Horn-Schunck Optical flow

$$\min_{u,v} \sum_{i,j} [E_s(i,j) + \lambda E_d(i,j)]$$

$$\hat{u}_{kl} = \bar{u}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_x$$

$$\hat{v}_{kl} = \bar{v}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_y$$

11 Tracking

11.1 Lucas-Kanade-Tomasi Tracker

Assume Δp is small and linear around p_0

$$\min_{\Delta p} \sum_x \left[I(W(x;p)) + \Delta I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2$$

$T(x)$ is the template, $I(x)$ is the image

$$\Delta p = H^{-1} \sum_x \left[\Delta I \frac{\partial W}{\partial p} \right]^T [T(x) - I(W(x;p))]$$

$$H = \sum_x \left[\Delta I \frac{\partial W}{\partial p} \right]^T \left[\Delta I \frac{\partial W}{\partial p} \right]$$

11.2 Template Matching

11.2.1 MOSSE Filter

Minimum output sum of squared error
Goal is to compute a filter g such that when the input image is convolved with g, then the output y will have a well-localized response.

$$g = \arg\min_g \frac{1}{N} \sum (g \otimes x_i - y_i)^2 + \lambda |g|^2$$

$$g = (X^T X + \lambda I)^{-1} X^T y$$

Let $\hat{g} = F(g) = \frac{\sum \hat{x}_i^* \odot \hat{y}_i}{\sum \hat{x}_i^* \odot \hat{x}_i + \lambda}$, $\hat{g}_N = \frac{\hat{a}_N}{\hat{b}_N + \lambda}$

$$\hat{a}_N = \sum_{i=1}^N \hat{x}_i^* \odot \hat{y}_i, \quad \hat{b}_N = \sum_{i=1}^N \hat{x}_i^* \odot \hat{x}_i$$

$$\hat{a}_{N+1} = (1-\eta)\hat{a}_N + \eta(\hat{x}_{N+1}^* \odot \hat{y}_{N+1})$$

$$\hat{b}_{N+1} = (1-\eta)\hat{b}_N + \eta(\hat{x}_{N+1}^* \odot \hat{x}_{N+1})$$