CS-E4850 Computer Vision Exam
CHEAT SHEET

Adam Ilyas 725819

December 18, 2018

1 Concepts

Reducing noise Use a filter kernel (Note that convolution is with filter kernel rotated 180 degrees.:
- Moving average (replace a pixel with the average of its neighborhood)
- Gaussian filter $G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2-y^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x^2}{2\sigma^2}\right)\times\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{y^2}{2\sigma^2}\right)$
Separable filter Camera projection matrix:

Handling near image border (clip filter (black), wrap around, copy edge, reflect across edge)

2 Laplacian Pyramid/ Gaussian Pyramid/

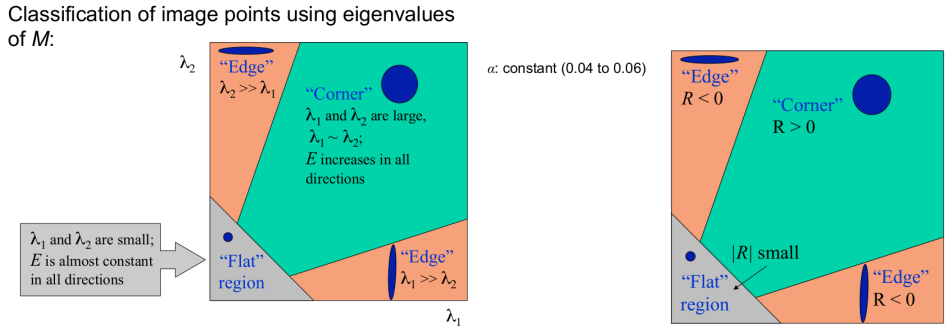
Image pyramids are multi-resolution image representations.
Gaussian Pyramid/Repeated gaussian blur and scaling down ($G_1 x_1 = x_2$) where G_1 is the gaussian blur and scaling down by half the resolution
Laplacian Pyramid/Contains the difference image between two successive Gaussian Pyramid levels ($(I - F_1 G_1)x_1$) where F_1 is scaling up to obtain the previous number of pixels.
Uses: Image compression, object detection (scale search and features), detecting stable points of interest,

Harris Corner Detector: We should easily recognize the point by looking through a small window. Shifting a window in any direction should give a large change in intensity

1) We first compute partial derivatives at each pixel. (M is a second moment matrix computed from image derivatives:)

2) Compute second moment matrix M in a Gaussian window around each pixel:

3) Compute corner response function R



$$R = det(M) - \alpha trace(M)^2$$
$$= \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

CORNER $R > 0, (\lambda_2 \approx \lambda_1)$
EDGE $R < 0, (\lambda_2 >> \lambda_1$ or $\lambda_1 >> \lambda_2)$
FLAT REGION (λ_2 and λ_1) are small, $|R|$ is small

4) Threshold R and 5) Find local maxima of response function (nonmaximum suppression)In order to pick up the optimal values to indicate corners, we find the local maxima as corners within the window

In order to pick up the optimal values to indicate corners, we find the local maxima as corners within the window

3 SIFT

Scale Invariant feature transform (SIFT):
SIFT keypoint detection is a feature detection algorithm to detect local features within images.
Detector part: Covariant: features(transform(image)) = transform(features(image))
Convolve a scale-normalized laplacian of a gaussian filter $\nabla^2_{norm} G = \sigma^2(\frac{\delta^2 G}{\delta x^2} + \frac{\delta^2 G}{\delta y^2})$ where $G(x, y) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2-y^2}{2\sigma^2})$
- Instead of using laplacian $L = \sigma(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$ we use $DoG = G(x, y, k\sigma) + G(x, y, \sigma)$
- **Scale-space extrema detection** find the maxima of response in scale-space.
- next, we eliminate edge responses. (Edges has strong laplacian responses) We can filter based on the harris response function over neighbourhoods containing blobs?

Descriptor part: Invariant: features(transform(image)) = features(image)
Scaled and rotated versions of the same neighborhood will give rise to blobs that are related by the same transformation
- **use the normalized region about the keypoint** We normalize these regions into same size circles.
- **compute gradient magnitude and orientation at each point in the region** Eliminate rotational ambiguity by assigning a unique orientation to circular image windows by creating a histogram of local gradient directions in the patch, assign orientation to peak of smoothed histogram.
- **weight them by a Gaussian window overlaid on the circle** to reduce the influence of gradients that are far from centre.
- create an orientation histogram over the 4 X 4 subregions of the window - Divide 16 X 16 sample array to 16 (4 X 4) descriptors windows, in each of these windows we compute the gradient in 8 directions, this gives a vector of 128 values.
- We then match keypoints descriptors, we threshold ratio of nearest to second nearest Descriptor. As ratio increases, less correct, more incorrect. We want our second nearest to be as far away as possible from our nearest descriptor.

4 Model Estimation

RANSAC Random Sample Consensus, for model fitting in the presence of outliers
Repeat N times:
- Choose a small subset of points (s points) uniformly at Random
- fit a model to that subset of s points (least squares fitting) and compute error function
- find all remaining points (inliers) that are close enough (t distance) to the model and reject the rest as outliers
- if there are **d** or more inliers, refit with all inliers

Choosing of parameters (**s** initial min no. of points, distance threshold **t**, number of samples **N**, Consensus set size **d**)
Pros: Simple general applicable to many problems **Cons:** Many parameters to tune, doesnt work well with low inlier ratio, cant get good initialization

5 Hough transform

for each feature keypoint: fit points to obtain parameters.
model can be linear ($y = mx + b$) or polar ($x \cos\theta + y \sin\theta = \rho$)
$(x, y) \rightarrow (m, b)$ points $\rightarrow$ parameter space, voting scheme, discretize parameter space to bins.
for each feature point in image, put a vote in every bin in parameter space that could've generated this points. Find the bins with most points.

Assumption that a noisy model will not vote consistently for any single model.
missing data doesnt matter as long as there are enough features to agree on a good model

6 Large-scale object instance recognition/ retrieval

inverted index: pre-compute index to enable faster search at query time by storing a mapping from content, such as words or numbers, to its locations.
for text document, indexing is an efficient way to find all pages which contains a given word
We want find all images with a particular feature by mapping our high-dimensional descriptors features with visual codewords by clustering.
Determine which word to assign a new image feature by finding the closest cluster center

Bag of visual words Summarize entire image based on its distribution (histogram) of codeword occurrences. imagine each image is a document with words. $sim(d_j, q) = \frac{\sum d(i) \cdot q(i)}{||d|| \cdot ||q||}$

term frequency - inverse document frequency
term frequency $= \frac{n_{id}}{n_d} =$ # word i in doc d/# of words in doc d
inverse document frequency $= log\frac{N}{n_i} = \log \frac{\text{total docs}}{\text{\# of doc containing word i}}$
Effective when we are able to find reliable features within clutter (more important)
Precision-Recall Curve y vs x: Precision vs Recall, to observe the trade off.
Precision: correct / returned
Recall: correct / actual correct that shd be returned
Pros:
flexible to geometry, viewpoint, deformations
a compact summary of the pages
provides a fixed dimensional vector representations for sets
Cons:
background and foreground information mixed when the bag cover the whole image (importance of features not taken into account)
basic model ignores geometry, must verify or encode via features

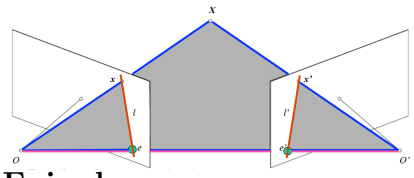
7 optical flow and keypoint tracking

Uses of motion: 3d shape reconstruction, event and activity recognition,
Given 2 subsequent frames of a video, the optical flow field indicates the apparent motion of a pixel. If we have 2 or more frames we can track features from one frame to the other by following the optical flow.
Structure from motion given a set of corresponding points in 2 or more image, we can compute camera parameters and 3D point coordinates.
Motion field projection of 3D scene motion to image (2d)
Lucas-Kanade Optical Flow Optical flow is the apparent motion of brightness patterns in the image. ideally, optical flow is the same as motion field (3d scene motion to 2d)
key assumptions: (small motion) points doesnt move far, (spatial coherence) points move like their neighbours, (Brightness constancy) project of points looks the same in each frame
estimating optical flow: given 2 subseq. frames, estimate flow from
Template matching (using 2 image and keypoint detection)
slow (need to check for more location), may be useful if large movement

Camera calibration process of determining internal camera parameters, which define the mapping between light rays and image pixel

8 Two view geometry and stereo vision

Two view geom./ epipolar geom. : geometric constraints between 2 views
stereo vision: using 2 views to measure depths of scene points.
structure: given projections of objects in 2 or more image, compute 3d coord.
Stereo correspondence: Given a point in one of the images, where could its corresponding points be in the other images
Motion: Given a set of corresponding points in two or more images, compute the camera parameters



Baseline: Line connecting 2 cameras O, O'
Epipolar plane: Plane containing baseline, and object x
Epipoles:
intersection of baseline with image planes, projections of the other camera centres, vanishing point in the motion direction
Epipolar line: intersection of epipolar planes with image plane.
Essential matrix

Object detection by sliding window

Viola-Jones face detector
Convolutional neural network