

CS 223B: Introduction to Computer Vision

Midterm Guidelines and Sample Questions

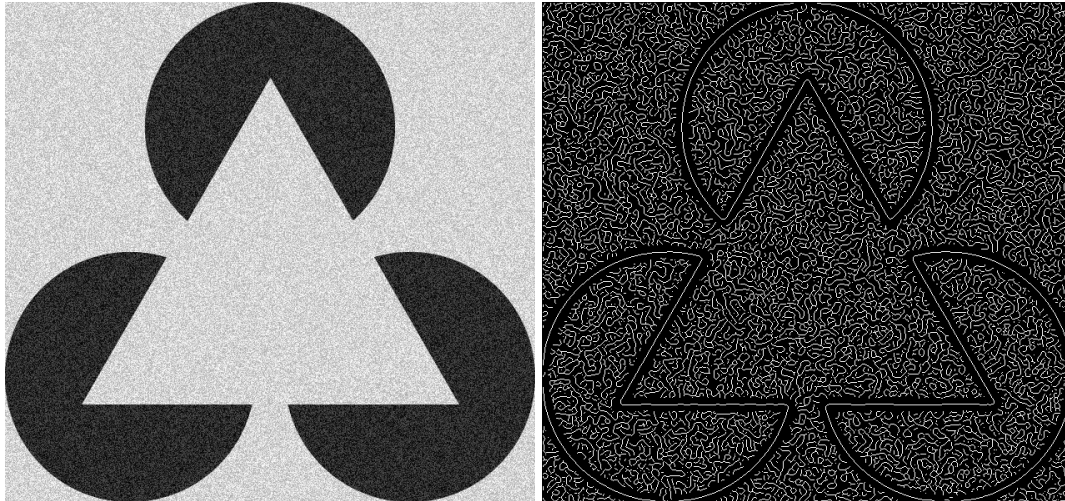
Midterm guidelines

1. A one-page, double-sided cheat sheet is allowed in the exam
2. The exam questions will be based (mostly) on the topics below
 - Projection models
 - Perspective models
 - Rigid transformation
 - Intrinsic and extrinsic parameters
 - Projection matrix
 - Projection matrix theorem (Faugeras, 1993)
 - Camera calibration
 - Epipolar geometry
 - Definitions
 - Rectification
 - Essential matrix
 - Fundamental matrix
 - 8-point algorithm
 - Stereo vision
 - Human stereopsis
 - Binocular fusion
 - Cooperative model
 - Fusion
 - Segmentation
 - Gestalt rules
 - Unsupervised clustering
 - Graph cut based algorithms
 - Hough transform
 - RANSAC
 - Linear filters (2D)
 - Convolution
 - Linear shift invariant system
 - Sampling
 - Aliasing
 - Detectors & Descriptors
 - Optical flow
 - Human motion classification

Sample Questions

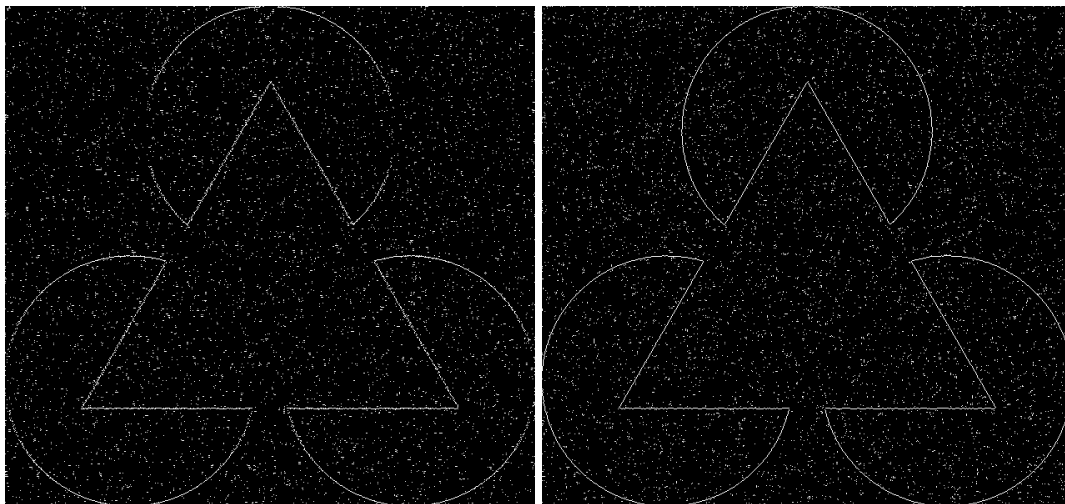
1 Image Filters

For the image (a) below, which of the images (b), (c), or (d) is the result of applying the Canny edge detector? Explain your answer.



(a) Original Image

(b)



(c)

(d)

2 RANSAC Algorithm

RANSAC (RANDOM SAMPLE CONSENSUS) is an algorithm for fitting a model to very noisy data. Its basic idea is to fit models to small samples of data, and then look for “consensus” from the entire dataset.

(a) Complete the following pseudo-code for RANSAC

INPUT:

data - a set of observations

model - a model that can be fitted to data

n - the minimum number of data required to fit the model (fixed for a given model)

t - a threshold value for determining when a datum fits a model

d - the number of “consensus” data points required for a “good” model

OUTPUT: *BestModel*

FOR (a number of iterations)

CurrentSet := sample a data subset of size n

CurrentModel := model fitted to *CurrentSet*

ConsensusSet := *CurrentSet*

FOR every data point not in *CurrentSet*

END FOR

IF $\#(\textit{ConsensusSet}) > d$

END IF

END FOR

RETURN *BestModel*

(b) Suppose you run the above algorithm on a dataset, and you find that the algorithm is able to return a pretty good model, but runs unbearably slow. How can you change the parameters to tackle this?