Problem Set 10

Computer Vision
University of Bern
Fall 2021

1 Outlier Rejection

The RANSAC algorithm is used for outlier rejection, and it repeats the following four steps.

- Select a small set of data points randomly.
- Compute model parameters for those points.
- Find inliers to this model.
- If the number of inliers are large enough, recompute model parameters.

1. Let p denote the probability of a data-point being an outlier. Assume we choose the seed point randomly from all points in an i.i.d. fashion. Compute the probability of finding the correct model parameters if we try N samples and we need at least m points for parameter estimation.

2. Let's register two images to create a panorama image. For this you have to use a homographic transformation model between images. Assume that 10 percent of the matching points are incorrect. How many times do we need to try sampling the points to find the correct model with 99 percent probability?

3. Which is more difficult? Stereo matching or image stitching for panorama pictures? Assume that the quality of feature matching is the same in both

cases and you use RANSAC.

4. Give a scenario, where RANSAC would fail. How would you change RANSAC to cope with the problem?

2 Sliding windows

The pipeline of the sliding window approach is simple. First we compute a feature representation of the image. It is a 3 dimensional array. The first two dimensions correspond to the image coordinates and the third are the channels of features. An important parameter is the stride. When the features are computed n pixels apart, the stride is n. In the case of raw RGB "features" the stride is 1 and the number of channels are 3. In the case of HOG features, the stride is usually 8 and there are 31 channels.

1. The size of the image is 400×400 pixels. We use HOG features with stride 8 and the number of channels is $c = 31$. The template size is 5×5 pixels. How long does it take to compute the feature responses, if we have a processor with 10 GFlops?

2. Consider different scales. The image size is 400×400 pixels for the largest magnification. We use 4 octaves and an octave resolution $r = 10$ to render the images at different scales. All the other parameters are the same as above. What is the running time?

3 Pictorial Structures

The score of part placements can be defined by the function J below, where l_i are the part locations ($i = 1, \ldots, k$), $m_i(l_i)$ are the unary terms and $d_{ij}(l_i, l_j)$ are the binary terms between part placements. The part placements are on a discrete grid, l_i can only be placed at N locations.

$$J(L) = \sum_i m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \tag{1}$$

We also know that the the structure of the graph (V, E) is a tree. We need to minimize J with respect to l_i to get the optimal part placements.

1. What happens when $d_{ij}(l_i, l_j) = 0$ for all $(v_i, v_j) \in E$? What is the running time of the optimization?

2. What happens when $d_{ij}(l_i, l_j) = 0$ when $l_i = l_j$ and $d_{ij}(l_i, l_j) = +\infty$ otherwise? What is the running time of the optimization?

3. When $d_{ij}(l_i, l_j) = a_{ij}\|l_i - l_j\|^2$, what is the running time of the most naive optimization?

4. Give a faster optimization algorithm by eliminating the variables leaf by leaf.

5. The generalized distance transform solves optimization $f(p) = \min_{q\in[1,N]} m(q) + d(p, q)$. It takes $\mathcal{O}(N)$ time to calculate. What is the running time of optimizing J using distance transform?