

HLCV Exercise 2

May 13, 2016

1 Question 4a

Homography estimation has the general purpose of finding a 3×3 -projection matrix for a projective mapping between two images in homogenous coordinates. Since the scale factor of our matrix is set to 1 we have to estimate 8 unknown parameters:

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}$$

The naive idea is to set a system of equations using point matches between $(x_i, y_i), (x'_i, y'_i)$ which would look like this

$$\begin{pmatrix} x_i & y_i & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & x_i & y_i & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & x_i & y_i & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{21} \\ h_{22} \\ h_{31} \\ h_{32} \\ h_{13} \\ h_{23} \\ 1 \end{pmatrix} = \begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix}$$

\Rightarrow 3 we get equations for our 8 unknowns from one point match.

If we extend the system with a sufficient number of points matches it becomes overspecified so we cannot solve it directly. So the idea would be using a least-squares fit.

The problem now is that wrong point matches (outliers) can totally spoil the estimation of our parameters since the least-squares fit tries to approximate all our point matches and punishes distance quadratically.

To overcome this drawback we use the RANSAC algorithm: The key idea is to select n random points and rate whether we found a set of “inliers”. In order to do this we compute the homography matrix approximating our selected points using a least-squares fit then we have a tolerance region around our fit. All the points inside that tolerance

are counted as inliers. We repeat the previous step k times and select the best point set with respect to the number of inliers it contains.

Now to provide an estimate how many times we have to sample (parameter k) to find a set of inliers given a failure probability p we can view the process as a Bernoulli process: w is the probability to get an inlier

n is the sample size = length of the Bernoulli process

k is the number of samples = number of Bernoulli processes

Given this it is easy to see that the probability that all samples fail is

$$p = (1 - w^n)^k$$

We use this formula to adapt k .

Now we can take a look how this is happening in the code: The `get_ransac_hom` function takes two lists of point coordinates `x1,y1,x2,y2` and two images `img1,img2` and returns our projection matrix `H`.

In the code segment

```
sample_size = size(x1,1); % we have sample_size samples
pFail = 0.001; % we want to be 99.9 percent sure
pInlier = 4 / sample_size; % just a very conservative guess for the
    amount of inlier (only 4);
```

we set our important parameters w to $4/\text{sample_size}$ and p to 0.1% .

We convert then our Cartesian coordinates to homogenous coordinates before we start with our RANSAC iterations.

In the present implementation we have a sample 4 random point matches and calculate their matrix `H` using

```
pos = uint32 (rand(4,1))' .* [sample_size-1, sample_size-1, sample_size
    -1, sample_size-1] + [1,1,1,1]);
testH = get_hom( x1(pos), y1(pos), x2(pos), y2(pos) );
```

We use then the matrix `H` and count the number of inlier within our given tolerance as explained above.

A slight modification is that we readjust the number of tests that we perform by

```

% adjust maximal iterations to be made
% you can reestimate the probability for picking a true pair by
% looking at the percentage of inliers of the current estimation
pCurrentInlier = inlierCount / sample_size;

if ( pCurrentInlier > pBestInlier)

    % find the amount of times k we have to draw a sample without
    % outliers
    kTests = log(pFail) / log(1 - pCurrentInlier^4);
    toDoTests = round(kTests);

    fprintf('proportion of inliers: %f, number of RANSAC iterations:
            %d\n', pCurrentInlier, toDoTests);

    % best Homography and amount of inliers
    H = testH;
    pBestInlier = pCurrentInlier;
end

if (toDoTests < j )
    break; % end here, we are pFail percent sure to have found H
end

```

In the end we collect all our inlier within the tolerance of our best chosen point match set and recalculate the H matrix using them:

```

% probability of picking an inlier from the parameter set
pInlier = pBestInlier;

% first find the inliers
hitId = zeros(sample_size,1);
for i = 1:sample_size
    p = H * [x1(i), y1(i), 1.0]';
    p = p / p(3);
    if (norm (p(1:2) - [x2(i),y2(i)]')) < distInlier)
        hitId(i) = i;
    end
end
hitId = find (hitId);

% re-estimate Homography
H = get_hom( x1(hitId), y1(hitId), x2(hitId), y2(hitId) );

```

The division by the last component $p = p / p(3)$; has to be made because we want to normalize our homogenous coordinates such that they have all scaling factor 1. Only then points can be compared among eachother.