Solution to Homework 2

CS 543 - Hoiem

1 A feature tracker (50pt)

1.1 Keypoint Selection (15pt)

Algorithm summary:

Compute Harris keypoints

Given the input image, compute gradient I_x and I_y

Construct the second moment matrix, convoluted with Gaussian filter:

$$M = g(\sigma^2) \begin{bmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{bmatrix}$$

Compute "cornerness" heat map for either Harris or Shi-Tomasi criteria:

Harris: $M_{xx}M_{yy} - M_{xy}M_{yx}) - \alpha(M_{xx} + M_{yy}).^2$

Shi-Tomasi: $min(\lambda_1, \lambda_2)$:

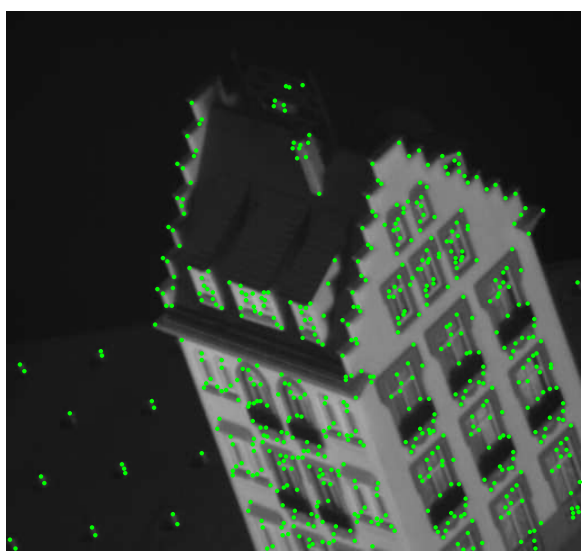
$\lambda_1 = \frac{M_{xx}}{2} + \frac{M_{yy}}{2} - \frac{\sqrt{M_{xx}^2 - 2M_{xx}M_{yy} + M_{yy}^2 + 4M_{xy}M_{yx}}}{2}$

$\lambda_2 = \frac{M_{xx}}{2} + \frac{M_{yy}}{2} + \frac{\sqrt{M_{xx}^2 - 2M_{xx}M_{yy} + M_{yy}^2 + 4M_{xy}M_{yx}}}{2}$

Threshold the heat map with τ apply non-maximum suppresion

Notes:

- If the image is not converted into double, a large threshold is needed. But that is fine.



1.2 Tracking (35pt)

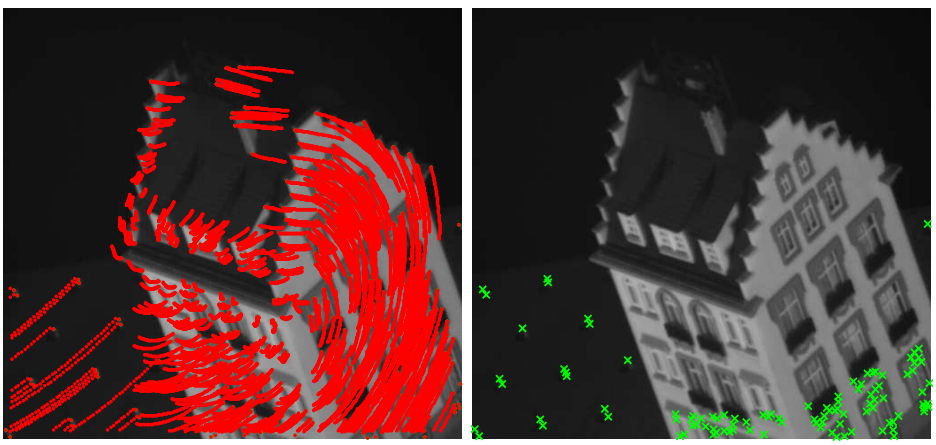
Algorithm summary:

KLT Tracking

Select keypoints from the first frame using previous part
For frame i and $i+1$
  Set x' and y' to x and y initially
  Interpolate I_x and I_y
  While not converged (or for a fixed number of iterations)
    Check if the point is out of bound, discard if it is.
    Compute I_t by taking the difference between the 15×15 window around (x, y) and (x', y'), also interpolated.
    Compute displacement (u, v) by solving
    $$\begin{bmatrix} \sum_W I_x I_x & \sum_W I_x I_y \\ \sum_W I_x I_y & \sum_W I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_W I_x I_t \\ \sum_W I_y I_t \end{bmatrix}$$
    Update the current estimate of $(x', y') = (x'_{old}, y'_{old}) + (u, v)$
  End
End

Notes:

- The interpolation is faster if you use `meshgrid` with `interp2`. You can also give `interp2` only the image values needed for interpolation instead of the whole image.



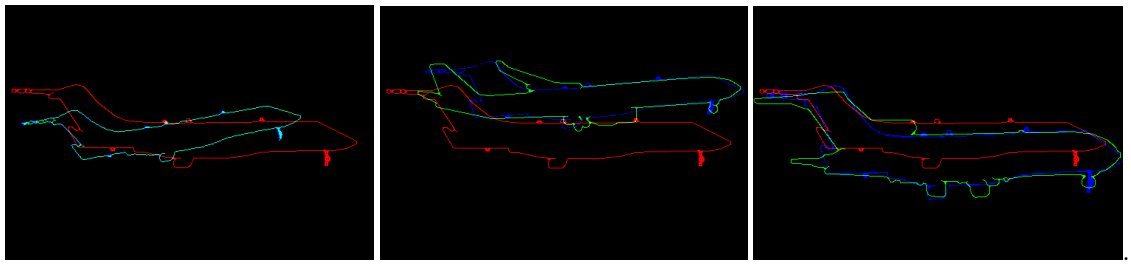
2 Shape alignment (30pt)

Algorithm summary:

KLT Tracking

Align the center of mass of Image 1 with that of Image 2.
Compute the distance transformation of Image 2
While not converged
  For each point in Image 1, find its nearest point in Image 2 from distance transformation.
  Compute the parameters of affine transformation (or any transformation of you choose)
  Compute the error of the transformed Image 1 to Image 2
  Check if the error has converged
End

This shows the results on aligning `object2` to `object2t`, `object2` to `object1` and `object2` to `object3`. Red is the original image, green is the target and blue is the best fit.



The run time is:

Method	Running time	Error
Object2t	0.236	0.346
Object1	0.212	5.445
Object3	0.241	4.499

3 Object instance recognition (20pt)

3.1 Keypoint matching (5 pt)

We first calculate the distance between g and each of the descriptors from template image $f_1, \dots f_n$. g matches its closest neighbour if the ratio of distances between g and its closest neighbour to g and second closest neighbour is below a certain threshold θ.

Match keypoint

Calculate the L2 distance of g and $f_1, \dots f_n$:

$Dist(g, f_i) = ||g - f_i||$.

Suppose the closest point to g is f_1 and second closest is f_2:

g matches with f_1 if $Dist(g, f_1) < \rho Dist(g, f_2)$

Where threshold $\rho = 0.7$ in our case.

3.2 Keypoint matching (15 pt)

Given that the first image has an object bounding box of (x_1, y_1, w_1, h_1) and the matched descriptors are $(u_1, v_1, s_1, \theta_1)$ and $(u_2, v_2, s_2, \theta_2)$. Width, height and orientation are: $w_2 = \frac{s_2}{s_1} w_1$, $h_2 = \frac{s_2}{s_1} h_1$, $o_2 = \theta_2 - \theta_1$.

x_2 and y_2 are calculated from the displacement vector of $[u_2, v_2]^T$ to the center of the object in image2. The displacement vector from the keypoint1 to the center of the object in image1 is $[\mathrm{x}_1 - \mathrm{u}_1, \mathrm{y}_1 - \mathrm{v}_1]^T$. Considering the scale difference in image2 it becomes $\mathrm{s}_2/\mathrm{s}_1\ [\mathrm{x}_1 - \mathrm{u}_1, \mathrm{y}_1 - \mathrm{v}_1]^T$

Applying the rotation matrix: $\mathrm{s}_2/\mathrm{s}_1 \begin{bmatrix} cos(\theta_2 - \theta_1) & -sin(\theta_2 - \theta_1) \\ sin(\theta_2 - \theta_1) & cos(\theta_2 - \theta_1) \end{bmatrix} [\mathrm{x}_1 - \mathrm{u}_1, \mathrm{y}_1 - \mathrm{v}_1]^T$. Therefore $[x_2, y_2]^T = [u_2, v_2]^T + \frac{s_2}{s_1} \begin{bmatrix} cos(\theta_2 - \theta_1) & -sin(\theta_2 - \theta_1) \\ sin(\theta_2 - \theta_1) & cos(\theta_2 - \theta_1) \end{bmatrix} [\mathrm{x}_1 - \mathrm{u}_1, \mathrm{y}_1 - \mathrm{v}_1]^T$.

And this gives all parameters needed to specify the location of the object $(x_2, y_2, w_2, h_2, o_2)$.