

Introduction to Digital Image Processing

Exercise No. 4.: Reminder - Finding the motion between images

Due: 2-02-2006

1 Lucas-Kanade Image Alignment with Pyramids

Here we describe how to find the motion between two gray-scale images I_1, I_2 using a simple motion model of 2D Translation:

$$p_2 = p_1 + \begin{pmatrix} u \\ v \end{pmatrix}$$

The algorithm outline is as follows, representing the motion by $\vec{s} = (u, v)$

1. Compute a Gaussian pyramid for both images of k levels, $pyr1$ and $pyr2$. It is recommended to make sure that the smallest level is not smaller than 30×30 .
2. Start with some initial guess (e.g. no motion, $\vec{s} = (0, 0)$)
3. For every level of the pyramids j from k to 0:
 - Update \vec{s} according to this level.
 - Let $Im_1 = pyr1\{j\}, Im_2 = pyr2\{j\}$.
 - Compute the derivative images I_x, I_y on Im_2 .
 - Compute the equation matrix A (see section 2).
 - iterate until convergence:
 - Warp Im_1 according to the current motion estimate to get Im_3 .
 - Let $I_t = Im_2 - Im_3$.
 - compute \vec{b} (see section 2).
 - Solve for the motion equations and get $\vec{\hat{s}}$ (see section 2).
 - update \vec{s} by $\vec{\hat{s}}$

2 The motion equations

For every pixel (x, y) , we assume a motion of $(u(x, y), v(x, y))$. The equations are derived from the constant brightness constraint:

$$I_2(x + u(x, y), y + v(x, y)) = I_1(x, y)$$

The Taylor Expansion of I_2 around (x, y) gives:

$$I_2(x + u(x, y), y + v(x, y)) \sim I_2(x, y) + I_x(x, y)u(x, y) + I_y(x, y)v(x, y)$$

Combining the two equations yields:

$$I_x(x, y)u(x, y) + I_y(x, y)v(x, y) + I_t(x, y) \sim 0$$

Where $I_t(x, y) = I_2(x, y) - I_1(x, y)$
 (Note that I_x and I_y are estimated from I_2 .)
 We wish to minimize the error in the l_2 norm:

$$\sum_{x,y} (I_x(x, y)u(x, y) + I_y(x, y)v(x, y) + I_t(x, y))^2$$

When the motion is translation, we get constant motion for all pixels:

$$u(x, y) = u$$

$$v(x, y) = v$$

To make things short, we omit the indexes x, y , to get the following error function to minimize:

$$\sum_{x,y} (I_x u + I_y v + I_t)^2$$

To minimize it, we compute the derivatives of the above expression with respect to u and v and set them to 0, receiving the final equation set:

$$\begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{pmatrix}$$

We put the left parts of the equations in a 2x2 matrix A and parameters vector $\vec{s} = (u, v)^T$, and the right parts of the equations in a 2x1 vector b . So $A\vec{s} = \vec{b}$, a nice and simple linear equation set.

2.1 Notes on the Motion Computation

- A is common to all iterations of the same pyramid level.
- Test your exercise first on small translations and one pyramid level to find more obvious bugs.
- Test your exercise on small real images, and not on caricatures.
- To increase speed, you may compute the motion without using the biggest pyramid level. In this case, you should adjust the motion to match the input image coordinates.
- When computing derivatives, first smooth the image. Use the same smoothing for the computation of I_t
- You should not use pixels which are close to the image borders. These pixels may not overlap informative pixels in the other image, and the derivatives estimates near the image borders are not correct.

2.2 The Motion Equations for Rotation and Translation

Again, we minimize the l_2 error function:

$$\sum_{x,y} (I_x(x, y)u(x, y) + I_y(x, y)v(x, y) + I_t(x, y))^2$$

When the motion is given by rotation and translation, we get:

$$p_2 = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} p_1 + \begin{pmatrix} dx \\ dy \end{pmatrix}$$

so

$$u(x, y) = \cos(\theta)x - \sin(\theta)y + dx - x$$

$$v(x, y) = \sin(\theta)x + \cos(\theta)y + dy - y$$

Assuming a small rotation angle, we substitute: $\cos(\theta)$ by 1 and $\sin(\theta)$ by θ , and we get

$$u(x, y) = -\theta y + dx$$

$$v(x, y) = \theta x + dy$$

To make things short, we omit the indexes x, y , to get the following expression:

$$\sum_{x,y} (I_x(-\theta y + dx) + I_y(\theta x + dy) + I_t)^2$$

Computing the derivatives with respect to dx , dy and θ and setting them to 0, yields a linear equation set with 3 unknowns, as shown in class. We put the left parts of the equations in a 3x3 matrix A and parameters vector $\vec{s} = (dx, dy, \theta)^T$, and the right parts of the equations in a 3x1 vector b .

2.3 Notes on the Motion Computation for rotation

- For the small motion assumption to be valid, it is recommended to solve the equations by posing the origin of the image (0,0) on the center of the image. This means that the x, y coordinates in the equations should be given in center-origin coordinate system, and the transformation should be adapted to the warping accordingly.
- Use the approximation $\cos(\theta) = 1$ and $\sin(\theta) = \theta$ only for solving the equation set. To actually describe the motion, use the original representation.
- To make the concatenation of motions simpler, you are advised to use the matrix representation of the motion:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & dx \\ \sin(\theta) & \cos(\theta) & dy \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$