DIGITAL GEOMETRY (UGE MASTER 2 INFORMATIQUE "IMAGERY SCIENCES")

Laboratory Session 5:

Digital shape registration

Yukiko Kenmochi

December 14, 2020

Practical information

Submission: Upload a compressed file containing your code and report with the file name of "your last name" via eLearning.

Deadline: January 4, 2021.

Evaluation environment: Linux ubuntu (This means that your program will be compiled and ran in a Linux environment for the evaluation, without any special setting.)

Note: Please include with your code a CMakeLists.txt file.

Experiments

STEP 1: DOWNLOAD AND OBSERVE SEGMENTED IMAGES

Download the compressed file, TP3_images.zip, containing the original colored images and their segmented images. From simple visual observation of those images, describe the shapes and positions of the segmented objects in the images.

Hereafter, we treat the pre-segmented images, namely binary images whose pixel value is 255 if the pixel is in an object and 0 otherwise.

STEP 2: READ EACH PAIR OF IMAGES AND DEFINE DIGITAL SHAPE

Read each binary image and extract an object of interest in each image.

1. Use PGMReader to read each input image.
2. Convert a binary image to a "digital set" following the instruction of "DigitalSet from threholded image".
3. Construct a "digital object" from a "digital set" with a choice of a good adjacency pair.
4. Compute connected components of a "digital object" using WriteComponents just in case. If there are several components, take the largest one.

STEP 3: CALCULATE RIGID MOTIONS OF AN OBJECT FROM A SERIES OF IMAGES

There is a series of images containing an identical object with different positions. Considering the first image as the reference, calculate for each image the movement of an object of interest from the reference. Note that the camera was not moved during the image acquisition, and motions were made in the orthogonal plane to the camera direction. Thus any motion of an object is assumed to be rigid, namely a combination of rotation and translation, such that

$$\left| \begin{array}{ccccc} \mathfrak{T} & : & \mathbb{R}^2 & \to & \mathbb{R}^2 \\ & & \mathrm{x} & \mapsto & R\mathrm{x} + t \end{array} \right.$$

where R is a rotation matrix and $t \in \mathbb{R}^2$ is a translation vector.

There are various ways of estimating an object rigid motion between two images I_1 and I_2. Let us assume that a digital shape S_i is pre-segmented in an image I_i. Here we use image moments for rigid motion estimation. We first estimate a translation vector and then a rotation matrix.

For each digital shape S_i, we compute the center of mass as follows:

$$c_i = \frac{1}{|S_i|} \sum_{\mathrm{x} \in S_i} \mathrm{x}$$

so that we obtain the translation vector t between S_1 and S_2 such that

$$t = c_2 - c_1$$

Concerning the rotation matrix R, we can apply, for example, one of the following methods:

1. choosing a common feature point f_i of S_i for $i = 1, 2$, estimate R such that
$$f_1 - c_1 = R(f_2 - c_2),$$
2. calculate the covariance matrix (inertia matrix) M_i for each S_i, and compute the first eigenvector v_i (the first principal component) of M_i (see here for Principal component analysis). Then the rotation angle can be obtained as the angle between v_1 and v_2;
3. calculate the cross-covariance matrix of S_1 and S_2, and then use the Kabsch algorithm for estimating the optimal rotation matrix R that minimizes the root mean squared deviation between S_1 and S_2.

Which method gives more reasonable results? If you find other methods that would be more robust, please propose and implement them.

STEP 4: MOVE EACH OBJECT WITH THE ESTIMATED RIGID MOTION

Based on the estimated rigid motion for each image with respect to the reference image, generate a transformed binary image by applying a rigid motion to the binary image using the module of geometric transformation in DGtal. Please choose an adequate model (forward or backward?) for this calculation.

STEP 5: COMPARE THE TRANSFORMED IMAGE AND THE REFERENCE

To compare a transformed digital shape S_2' and the reference shape S_1, we can use the Hausdorff distance between them, which is defined by

$$d_H(S_1, S_2') = \max\left\{\sup_{x\in S_1}\left\{\inf_{y\in S_2'} d(x,y)\right\}, \sup_{y\in S_2'}\left\{\inf_{x\in S_1} d(x,y)\right\}\right\}$$

Here we consider that $d(x,y)$ is the Euclidean distance between x and y.

Certain authors have proposed some alternatives to the Hausdorff distance, which possess certain advantages in practice. For example, Dubuisson and Jain (1994) have proposed to replace the sup by a mean in the previous definition, to obtain a more robust dissimilarity measure. Can you give a simple example showing the difference between the Hausdorff distance and the Dubuisson-Jain dissimilarity measure? In what sense is the latter more robust than the former?

After estimating the computational complexity of a naive algorithm that computes the Hausdorff distance or the Dubuisson-Jain dissimilarity measure between two binary images, please propose another method using the distance transform for this calculation. What is its computational complexity? To calculate the distance transform for a given image, see here.

For each pair of a transformed digital shape S_2' and the reference shape S_1, calculate the Hausdorff distance and the Dubuisson-Jain dissimilarity measure between them.

Please also calculate the Hausdorff distance and the Dubuisson-Jain dissimilarity measure between the original shape S_2 before the transformation and the reference S_1, and compare the results with the previous ones. How is the transformed shape close to the reference?

Can you also visualize the difference between a transformed image and the reference for your report?

STEP 6: (OPTIONAL) IMPROVE THE MOTION ESTIMATION

Propose a method to improve the motion estimation from the above initial estimation iteratively, and implement it.

STEP 7: (OPTIONAL) IMPROVE THE IMAGE MANIPULATION

Application of rigid motion to a binary image in Step 4 would induce very noisy boundary of digital shapes. Propose an alternative method for rigid transformation of digital images and implement it.