**LAPORAN PRAKTIKUM PENGOLAHAN CITRA DIGITAL**

**5. SPATIAL TRANSFORMATIONS AND IMAGE**

**REGISTRATION**



**Disusun oleh :**

**Nama : Felix**

**NPM : 2327250059**

**Kelas : IF41**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER DAN REKAYASA**

**UNIVERSITAS MULTI DATA PALEMBANG**

**2025**

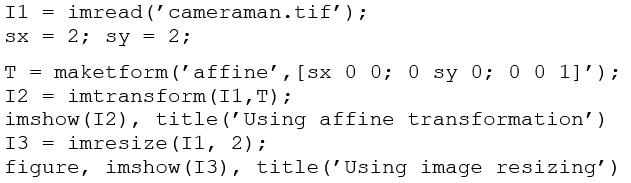
**TUTORIAL : SPATIAL TRANSFORMATIONS AND IMAGE**

**REGISTRATION**

In this tutorial, we will explore the IPT’s functionality for performing spatial transformations (using the imtransform, maketform, and other related functions). We will also show a simple example of selecting control points (using the IPT’s *Control Point Selection tool*) and using spatial transformations in the context of image registration.

In the first part of this tutorial, you will use imtransform and maketform to implement affine transformations (see Table 6.1), apply them to a test image, and inspect the results.

1. Open the cameraman image.
2. Use maketform to make an affine transformation that resizes the image by a factor [sx, sy]. The maketform function can accept transformation matrices of various sizes for N-dimensional transformations. But since imtransform only performs 2D transformations, you can only specify 3 × 3 transformation matrices. For affine transformations, the first two columns of the 3 × 3 matrices will have the values a0, a1, a2, b0, b1, b2 from Table 7.1, whereas the last column must contain 0 0 1.
3. Use imtransform to apply the affine transformation to the image.
4. Compare the resulting image with the one you had obtained using imresize.

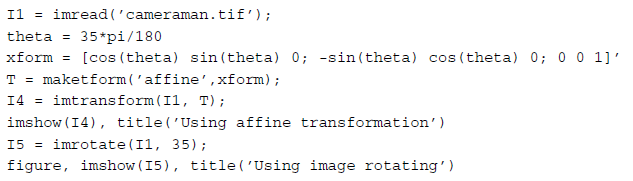


|  |  |
| --- | --- |
|  |  |
|  |

**Question 1** Compare the two resulting images (I2 and I3). Inspect size, graylevel range, and visual quality. How are they different? Why?

|  |  |
| --- | --- |
|  | Size:  Graylevel range: |
|  | Size:  Gray Lavel Range: |
| Dari kedua gambar memiliki gray lavel 7 sampai 255. Menurut saya hasil gambar yang bagus adalah yang menggunakan affine, karena gambarnya terlihat lebih besar jadi lebih jelas, dan juga warna hasil dari affine lebih bagus karena warnanya rata tidak seperti menggunakan imresize() hasilnya seperti di naikkan kecerahanya, dan keduanya tidak ada hasil gambar yang jelek semuanya bagus. Hal ini terjadi karena imtransform () dan imresize() menggunakan metode yang berbeda maka bisa terjadi hal-hal tersebut. | |

1. Use maketform to make an affine transformation that rotates an image by an angle θ.
2. Use imtransform to apply the affine transformation to the image.
3. Compare the resulting image with the one you had obtained using imrotate.

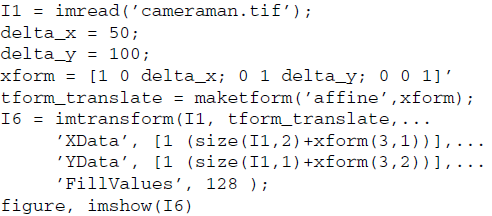


|  |  |
| --- | --- |
|  |  |
|  |

**Question 2** Compare the two resulting images (I4 and I5). Inspect size, graylevel range, and visual quality. How are they different? Why?

|  |  |
| --- | --- |
|  | Size =    Gray level: |
|  | Size :    Gray Level: |
| Dapat dilihat hasil dari kedua gambar berbeda size I1 = 259 \* 259 sedangkan size I6 = 357 \* 357, sizenya berbeda karena adanya tambahan yang di sebabkan oleh imtransform. dari segi kualitas gambar sama saja tidak ada yang berbeda. | |

1. Use maketform to make an affine transformation that translates an image by
2. Use imtransform to apply the affine transformation to the image and use a fill color (average gray in this case) to explicitly indicate the translation.
3. Display the resulting image.

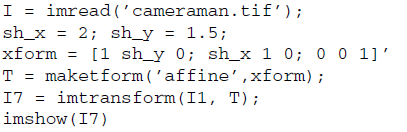


|  |  |
| --- | --- |
|  |  |

**Question 3** Compare the two images (I1 and I6). Inspect size, gray-level range, and visual quality. How are they different? Why?

|  |  |
| --- | --- |
|  | Size:    Gray level: |
|  | Size:    Gray level: |
| Dapat kita lihat size I1 dengan I6 berbeda hal itu di karenakan adanya imtransform dan maketform membuad adanya tambahan area di sebelah kanan dan hatasnya. Jika dilihat dari gray levelnya sama saja dari 7 sampai 255. | |

1. Use maketform to make an affine transformation that performs shearing by a factor [shx, shy] on an input image.
2. Use imtransform to apply the affine transformation to the image.
3. Display the resulting image.



|  |  |
| --- | --- |
|  |  |

**Image Registration**

In the last part of the tutorial, you will learn how to use spatial transformations in the context of image registration. he main steps are illustrated in a block diagram format in Figure 5.1.

1. Open the base image (Figure 5.2a) and the unregistered image (Figure 5.2b).



|  |  |
| --- | --- |
|  |  |
|  |

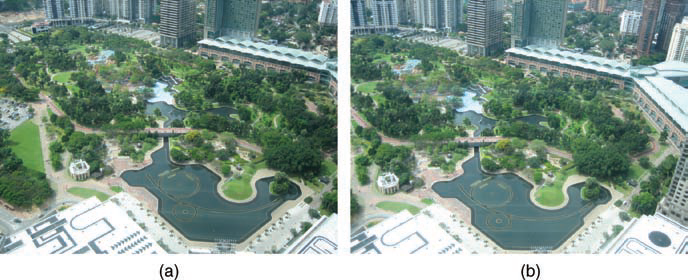
1. Specify control points in both images using cpselect (Figure 5.3). This is an interactive process that is explained in detail in the IPT online documentation. For the purpose of this tutorial, we will perform the following:
   * Open the Control Point Selection tool.
   * Choose a zoom value that is appropriate and lock the ratio.
   * Select the Control Point Selection tool in the toolbar.
   * Select a total of 10 control points per image, making sure that after we select a point in one image with click on the corresponding point in the other image, thereby establishing a match for that point. See Figure 5.4 for the points I chose.
   * Save the resulting control points using the File > Export Points to Workspace option in the menu.



|  |  |
| --- | --- |
|  |  |



**FIGURE 5.1** Image registration using MATLAB and the IPT.



**FIGURE 5.2** Interactive image registration: (a) base image; (b) unregistered image.



**FIGURE 5.3** The Control Point Selection tool.



**FIGURE 5.4** Selected points.

1. Inspect the coordinates of the selected control points.



|  |  |
| --- | --- |
|  |  |
|  |

1. Use cpcorr to fine-tune the selected control points.

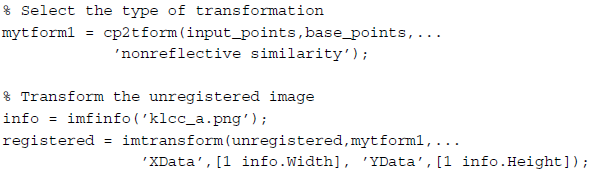


|  |  |
| --- | --- |
|  |  |

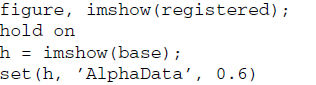
**Question 4** Compare the values for input\_points\_adj with that for input\_points. Did you notice any changes? Why (not)?

|  |
| --- |
| Hasil yang saya dapat tampak ada perubahan sedikit, karena Fungsi cpcorr hanya menyesuaikan posisi titik jika ditemukan area lokal yang lebih cocok berdasarkan nilai intensitas di sekitar titik (menggunakan pencocokan korelasi), bila tidak ada maka titik dari input\_points\_adj akan sama. |

1. This is a critical step. We need to specify the type of transformation we want to apply to the unregistered image based on the type of distortion that it contains. In this case, since the distortion appears to be a combination of translation, rotation, and scaling, we shall use the ’nonreflective similarity’ transformation type. This type requires only two pairs of control points.
2. Once we have selected the type of transformation, we can determine its parameters using cp2tform.
3. Use the resulting tform structure to align the unregistered image (using imtransform).



1. Display the registered image overlaid on top of the base image.



**Question 5** Are you happy with the results? If you had to do it again, what would you do differently?

|  |
| --- |
|  |