2D Geometric Transformation Recipe

EECS 442 Computer Vision
Siyuan Chen, chsiyuan@umich.edu

We've discussed range map and domain operations in the class. The former manipulates values of pixels, and the latter changes the domain, i.e., pixel locations, while keeping the pixel values. To change the domain, we introduced geometric transformations. Say we have an original image I and a transformation T. Here we only talk about 2D case and save the 3D case to the lectures later. Then the transformed image J has the following relationship with I:

$$J(x) \ = \ I(T(x))$$

where x is a vector of coordinates of a pixel.

The transformation can be represented as the multiplication of matrices/vectors conveniently when using the homogeneous coordinates. It seems that we just need to multiply the coordinates of pixels in the original images with the transformation matrix, then we are done. However, it becomes problematic when we implement it in practice.

Basically an image is a 2D array and the coordinates of pixels are indices of that array (which are all integers). Think about these questions.

> What if the coordinates after transformation are not integers? (Actually it is what happens in the most of the cases.)
>
> If the image is rotated, or even worse, malformed, how can we put it into a regular array?
>
> Do we know how big the transformed image is?

The trick to do this is to think about the inverse process

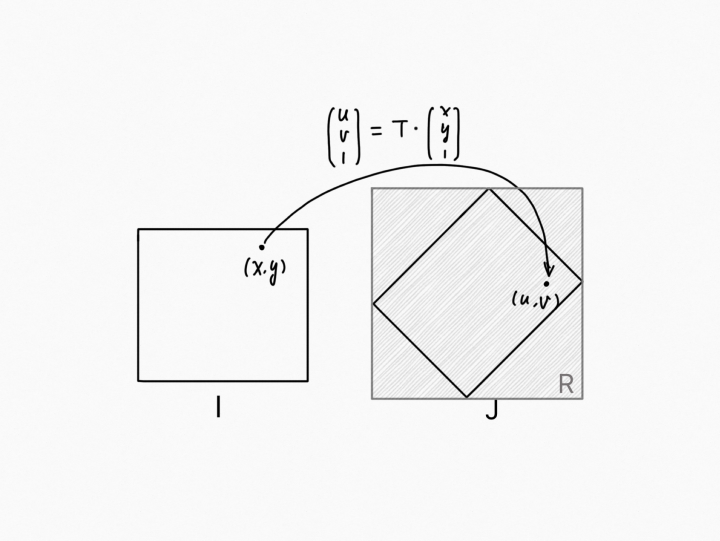
$$I(T^{-1}(y)) \ = \ J(y)$$

Since we know everything about the original image, it is manageable even though the coordinates are not integers (because we have interpolation!). Here is the recipe for implementing 2D geometric transformations.

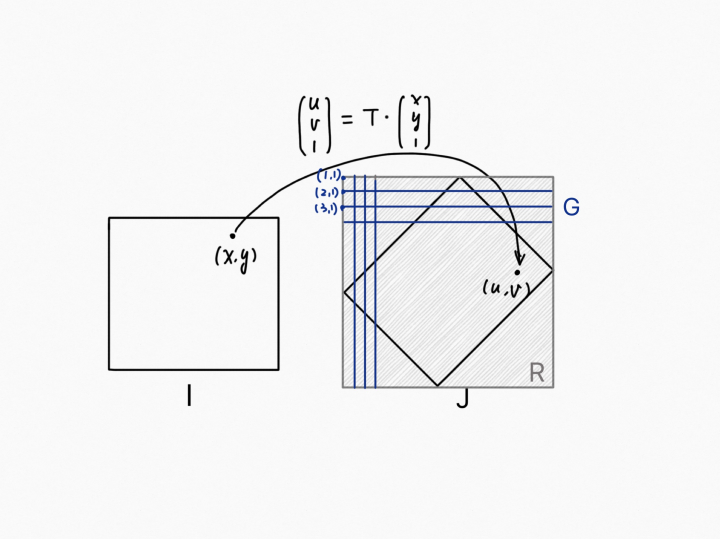
- Forward transformation. For every point (x, y) in the original image I, calculate the coordinates of point (u, v) in transformed image J by applying T to the original image I, i.e.,

  $$(u, v, 1)^T \ = \ T(x, y, 1)^T$$

  Find $min(u)$, $min(v)$, $max(u)$, $max(v)$. Then we know the range of region R that J fits into.



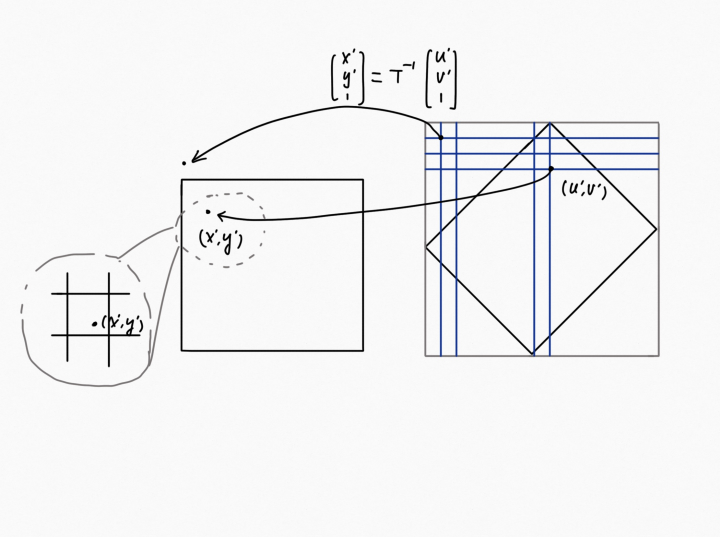
- Rasterize the region R. Create a regular, uniformed grid G that covers R. Coordinates of the point (u', v') on G should be continuous integers.



- Inverse transformation. For every point (u', v') on G, do inverse transformation, i.e.,

$$(x', y', 1)^T = T^{-1}(u', v', 1)^T$$

If (x', y') falls into the image domain, get its value by bilinear interpolation. Otherwise the value is set to zero. Assign that value to (u', v').

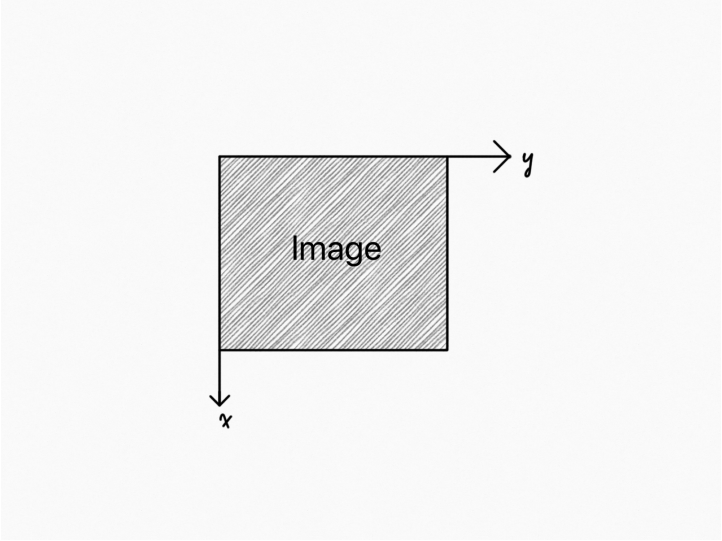


- Display G.

There are some details needed to be mentioned.

1) About the coordinates

Note that for an image, we use coordinates of which the origin is located at the top right corner of the image and x is the vertical axis with positive direction downwards and y is horizontal pointing to the right side, as the picture shows. However, when you generate a grid or apply transformations, the coordinates are assumed to be the normal coordinates we use. So you might need to change the order of x and y in some cases. (And I won't tell you what the cases are :p.)



2) Matlab built-in functions

Here is a list of Matlab functions you may feel useful.

`meshgrid`: creates a $M * N$ grid given x and y where x is $1 * N$ vector and y is $1 * M$ vector.

`reshape`: rearranges the elements of a matrix to have a shape $M * N$. The number of elements must stay the same.

`interp2`: evaluates the interpolated values given an matrix used to calculate interpolation and a list of indices indicating the query locations. You may use the function like this: `interp2(I,u,v)`, where `u,v` are column vectors.

`rgb2gray`: turns the color image to a gray-scaled image.

`imshow`: displays the image.

There are also two functions that can save all the efforts above: `affine2d` and `imwarp`. `affine2d` creates an object of 2D affine transformation given the transformation matrix. And `imwarp` takes in that object and the original image, and returns the transformed image. You won't want to use them in the submission of your assignment since you'll receive a zero. However, it is an easy way to check if your results are correct or not.