Image Transformations

# Introduction

The intent of this article is to provide a high level understanding of Affine Spatial Transformation of planar 2-D images. These are widely used image transformations in Scanning and Printing Driver, Application and Multi-Function Printer (MFP) Firmware.

This article illustrates Raster & Vector image types along with example of commercially available formats. The key term Spatial Transformation is introduced and then its implementation through Coordinate Spaces Transformation is explained. It discusses Rendering in the context of Vector & Raster content along with rendering scenarios. It is concluded with use cases involving transformations.

# Raster & Vector Images

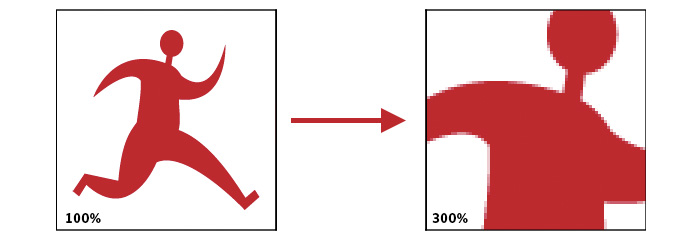
* Raster Image Formats

Raster (or bitmap) images are described by an array (or map) of bits within a rectangular grid of pixels (or dots) for each of the pixels and hence their file size is larger.

Well known Raster file types are JPEG, TIFF, BMP, PNG, GIF and portable raw formats (PPM, PGM, and PBM).

Note that bitmap font formats are Raster fonts and font image is stored in BMP, PNG, TIFF, etc format.

Raster image editor applications have their own Container formats such as CPT (Corel Photo Paint), PSD (Adobe Photoshop Document), PSP (Corel Paint Shop Pro), XCF (eXperimental Computing Facility format, native GIMP format).



When raster version is scaled, the pixels on the edge begin to show and the edges no longer look smooth.

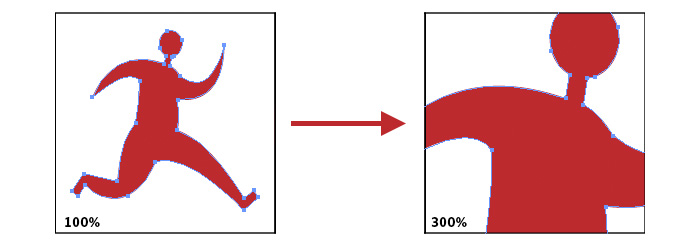
* Vector Image Formats

Vector images are described by lines, shapes, and curves which can be represented by geometric formulas for rendering the image elements at any desired display size.

Note that Outline font formats (Type1, Type3, TrueType, OpenFont, etc) are Vector fonts. Vector fonts store font outlines in WMF (Windows Meta file Format) or SVG (Scalable Vector Graphics).

Popular Vector file formats are the following: -

* Open standard - SVG, Gerber, XPS
* Application specific - AI (Adobe Illustrator Artwork), CDR (CorelDraw)
* Markup / graphics / printing languages – MathML, DrawingML, VML, WMF/EMF, HPGL
* 3D Vector formats – AMF, 3DS, etc.



When vector graphics are scaled, the edges remain crisp and sharp no matter the size.

* Compound Image Formats

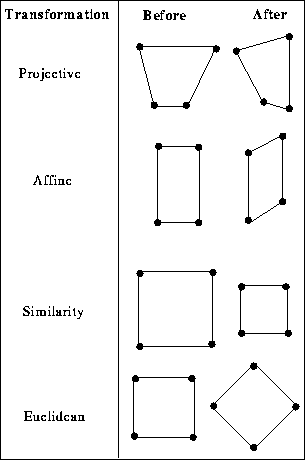
These are formats containing both pixel and vector data, possible other data, e.g. the interactive features of PDF.

* EPS (Encapsulated PostScript)
* PDF (Portable Document Format)
* PostScript, a page description language with strong graphics capabilities
* PICT (Classic Macintosh QuickDraw file)
* SWF (Shockwave Flash)
* XAML User interface language using vector graphics for images.

# Spatial Transformations

A Spatial Transformation defines a Geometric relationship between each point in the source image and transformed image.

Below is hierarchy of Spatial Transformation with increasing level of invariances (angle, length, ratio of length, parallelism, incident, cross ratio). Affine Transformations preserve Parallelism, Incident (point of intersection), and Cross Ratio (defined non-uniform scaling).



Types of Affine Transformations: -

* Translation
* Rotation
* Uniform Scale
* Non-uniform Scale
* Shear

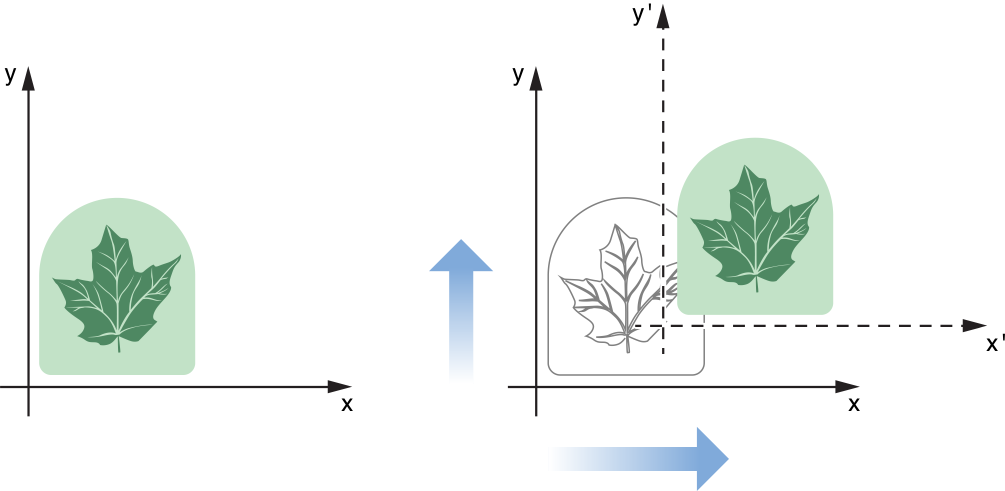
# Coordinate Spaces, Transformations and Raster & Vector Images Rendering

Computer vision software systems use Coordinate Spaces and Transformations to scale, rotate, translate, shear, and reflect graphics output.

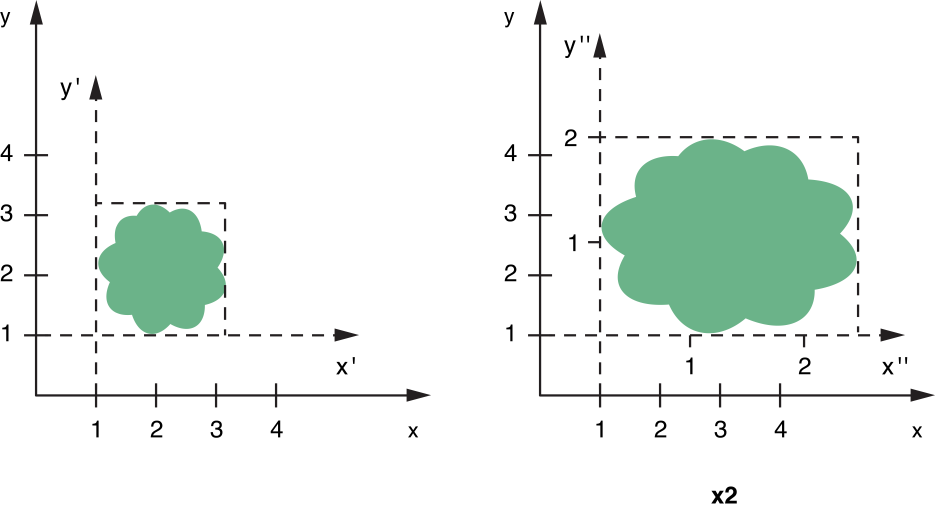
Though different Operating Systems use slightly different set of Coordinate Spaces (Windows OS has World, Page, Device and Physical Device spaces and Mac OS X has User and Device spaces), the system maps drawing content (image) from one Coordinate Space into the next using a Spatial Transformation until the output appears in its entirety on the physical device (monitor or printer). And similar transformations happen during document scanning. Even all functions (Print, Copy, Scan and Fax) in Print Imaging Pipeline in MFP firmware require these transformations.

These image transformations effectively means transferring Color values of source pixel (x, y) to its new location (x’, y’) in destination Coordinate Space. Below listed are details of different Affine Transformations along with operator matrix.

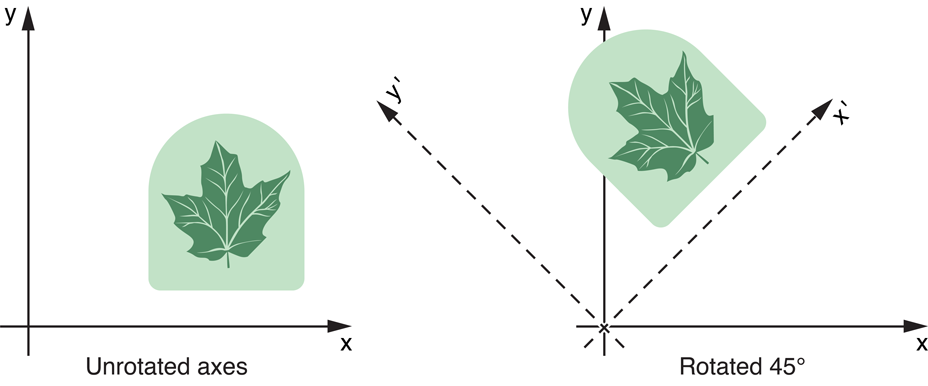
* Translation - Translation involves shifting the origin of the current coordinate system horizontally and vertically by a specific amount. Translation is probably used the most because it can be used to position graphic elements in destination Coordinate Space.



* Scaling - It lets you stretch or shrink the units along the x and y axes independently.



* Rotation – It changes the orientation of the coordinate axes by rotating them around the current origin.



* Combined Ordered Transformations – It can be observed that Transformation operation can be achieved through matrix operations. So then, multiple Transformation operations can be applied together through series of matrix multiplications but with only caution that they have to be applied in given order because matrix multiplication are not commutative. This caution can be easily observed by seeing the difference between applying a sample translation and then sample rotation and re-try these operations in reverse order – the resulting transformed images would appear different.

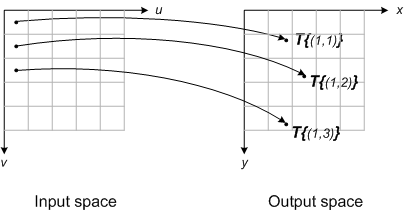
For obvious performance reasons, imaging systems accumulates desired Affine Transformations across logical Coordinate Spaces. It then applies this accumulated transform to the given image for its rendering on desired Physical Coordinate Space (Monitor, Printer, etc). This approach is able to naturally handle resolution independency while rendering Vector images whereas rendering of Raster image involving resolution change requires proportionate interpolation or extrapolation.

# Imaging Libraries & Frameworks

Different OS have their own native imaging libraries & frameworks and there are open source alternatives too. Some well-known examples are the following: -

* Windows – GDI+, Direct2D
* Mac OS X – Core Image
* Open Source – OpenGL, OpenCV, GIMP, ImageMagick, etc.
* Browsers – CSS Script

Note that any imaging library will face inherent challenge of mapping each of the source Coordinate Space point, i.e., source pixel to exact single destination Coordinate Space point, i.e., destination pixel.



Hence, all libraries & frameworks use some kind of scheme to identify nearest neighbors and then apply some averaging/interpolation algorithm to establish appropriate mapping between Coordinate Spaces.

# Commercial Rendering Engines capability of handling Raster & Vector Content

Here we present a list of various imaging solutions along with its Rendering Engine and if they handle Vector content.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Imaging Solution Type** | **OS** | **Solution Platform** | **Rendering Engine** | **Supports Vector** | **Remarks** |
| User Interfaces | Windows | WPF | DirectX | Y | WPF applications UI are vector-based and scale lossless based on Windows DPI settings. |
|  | Windows | Windows Forms (using GDI / GDI+) | DirectShow | N | GDI / GDI+ is based on the concept of a device context, where applications obtain handles to the device context, and use the handles in order to interact with the device.  GDI / GDI+ rendering mode is ‘immediate’ - application repaints the area that becomes invalidated. |
|  | Mac OS X | Core Graphics | Quartz | Y |  |
| Browsers | All | Chrome & Opera | Blink | Y |  |
|  |  | Tizen, Safari, Android Browser | WebKit | Y |  |
|  |  | IE & Edge | Trident & EdgeHTML | Y |  |
| Applications | All | MS Office & Outlook | MS Word Engine | Y |  |
|  | Windows | XPS | DirectX | Y |  |
|  | All | Adobe Apps (AIR, CS, Illustrator) | WebKit | Y |  |

# References: -

[1] <https://en.wikipedia.org/wiki/Image_file_formats>