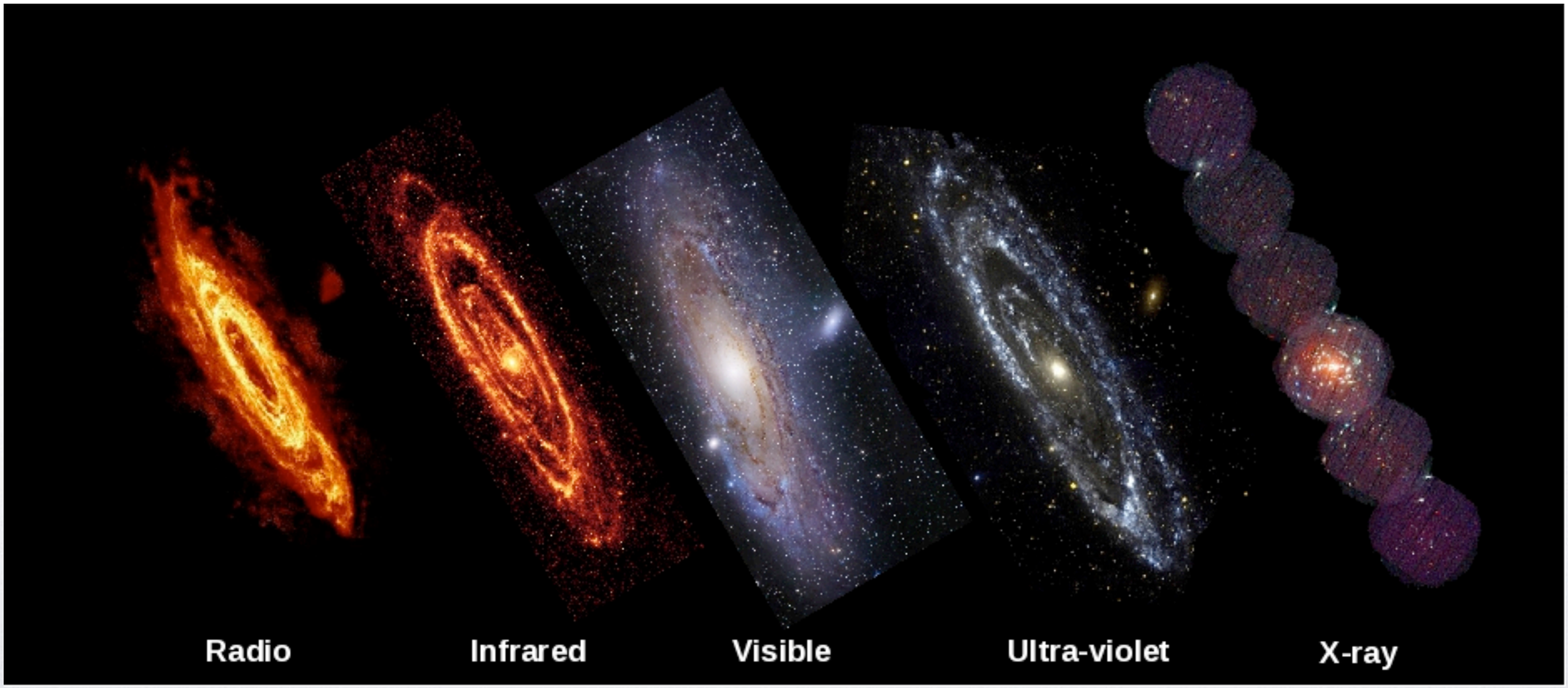
Investigating the morphology of astronomical objects using computer generated frame technique

許家榮 Chia-Jung Hsu
Department of Mathematics, NTU
Supervisor: Yasuhiro Hashimoto

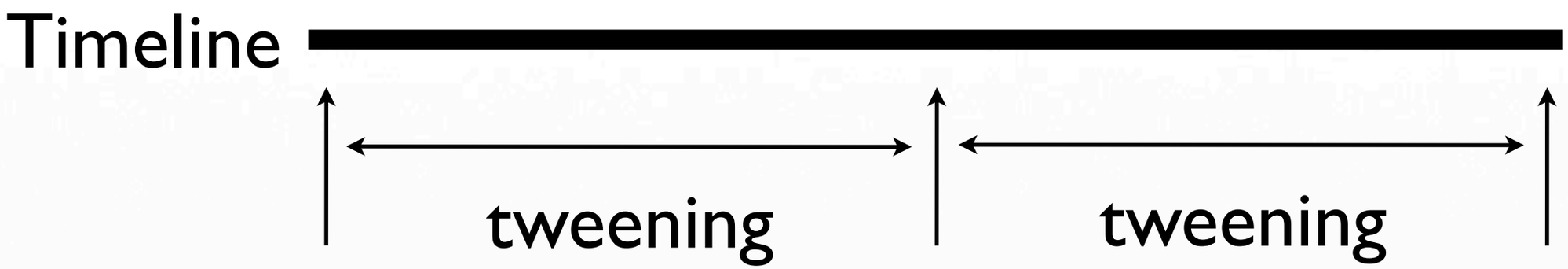
Morphology v.s. Band



Key frame

- To get full information in all wavelength, so that we can correct the redshift more generally.
- There are many intermission between all the bands.
- Only images in particular band -> key frame

Introduction to Key frame



keyframe:the images having given at some moments
tweening:the complemental transition of images

Introduction to Key frame

- tweening:a continuous transition between two images, including position, shape, color, etc.
- morphing:a transition between fully colored images

Morphing

- Morphing is composed by ‘warping’ and ‘dissolving’, so that the process looks smoothly.

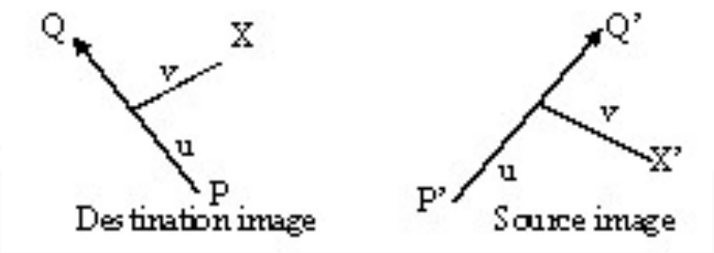
Warping

- There are many kinds of ways to warping
  - mesh-warping
  - fieldwarping

Field-warping

- Take vectors on images as reference to transform.
- Reverse mapping: from destination image, find the corresponding point on source image.

Field-warping



A pair of lines defines a mapping

$$u = \frac{(X-P)\bullet(Q-P)}{\|Q-P\|^2}$$

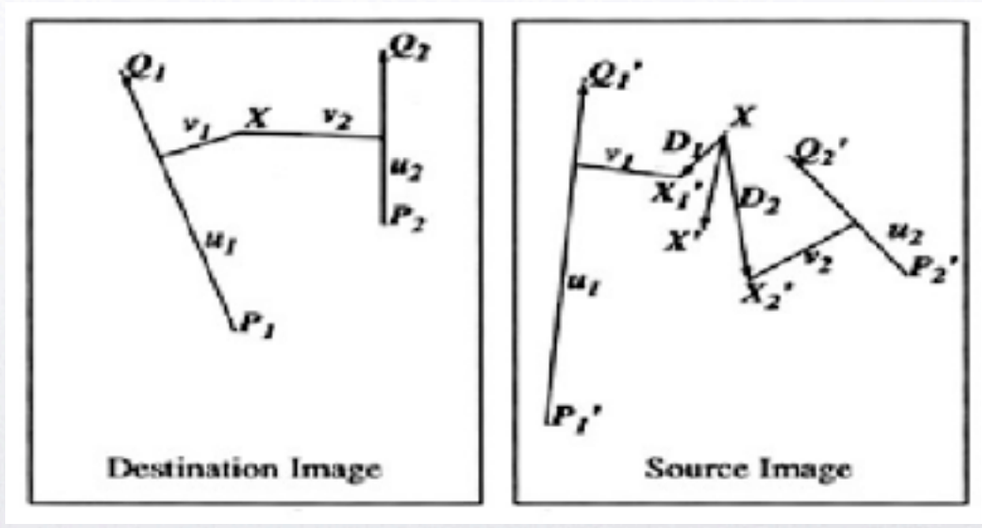
$$v = \frac{(X-P)\bullet Perpendicular(Q-P)}{\|Q-P\|}$$

$$x' = P' + u\bullet(Q'-P') + \frac{v\bullet Perpendicular(Q'-P')}{\|Q'-P'\|}$$

u is the ratio along the line, and v is the distance from the line

Field-warping

- For multiple pairs of lines, every pair defines a mapping.
- From one point on destination image to the corresponding point on source image, we can get a vector.



Field-warping

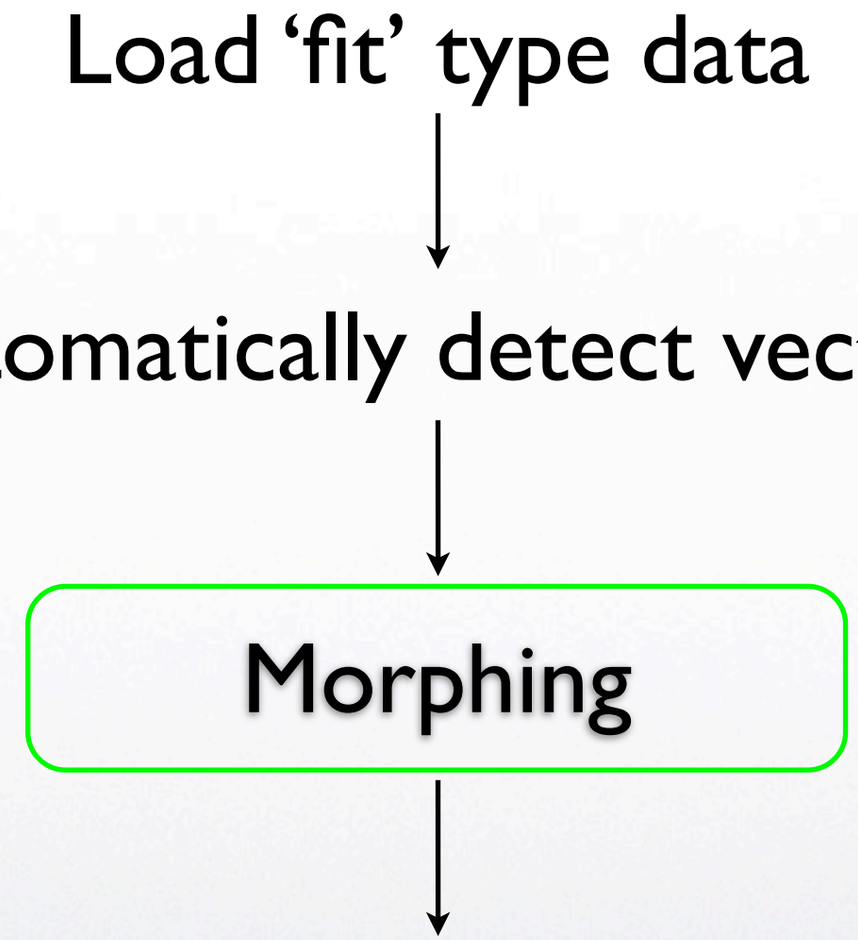
$$weight = \left(\frac{length^{p}}{(a + dist)} \right)^{b}$$

- Summation of all vectors got from all pairs of lines gives a reasonable position.

Targets of program

- 1.multiple images input
- 2.interpolation -> extrapolation
- 3.interpolation algorithm -> fitting algorithm
- MOST IMPORTANT:AUTOMATICALLY

Structure of total program



Structure of morphing

Load images and vectors for each images

↓

Use Lagrange interpolation method to compute a series of reasonable vectors at particular time

↓

Take each series of vectors and the “series of reasonable vectors” to execute field-warping for each images

↓

Combine all the images in a ratio related to time

Header of real code

```
#include<opencv/highgui.h>

typedef struct{
    CvPoint startpoint;
    CvPoint endpoint;
    CvMat *vec;
}Vec;
//This structure define vector type so that I can use this type to claim 2D array.
//( 1D for multiple lines, 1D for mutliple images)

void assignvec( Vec *imgvec, int linenum, int serialnum);
//assign the vector in each image
void lagrangeinter( Vec **vec, Vec *warpvec, double t, int imagenum, int linenum);
//compute the function of vector with time
double weightfactor( Vec line, int i, int j);
//compute the weight of each vector, used in fieldwarping function
void fieldwarping( IplImage *img, IplImage *warpimg, Vec *imgvec, Vec *warpvec, int linenum);
//mainly execute, adjust source image(img) to destination image(warpimg)
void crossdissolve( IplImage **img, IplImage *morphimg, int imagenum, double t);
//combine all the warpimg
```

Example 1

Time = 0

source image

Example 1

Time = 1

source image

Example 1

Time = 0.5

image 1 after warping

Example 1

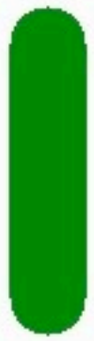
Time = 0.5

image 2 after warping

Example 1

Time = 0.5

image after morphing



Example 1

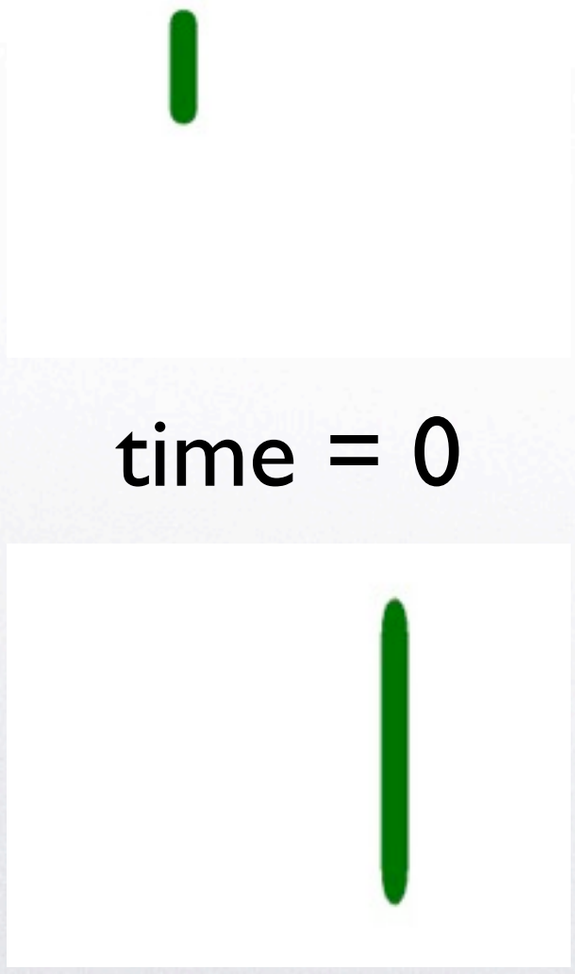
time = 0

time = 1.0

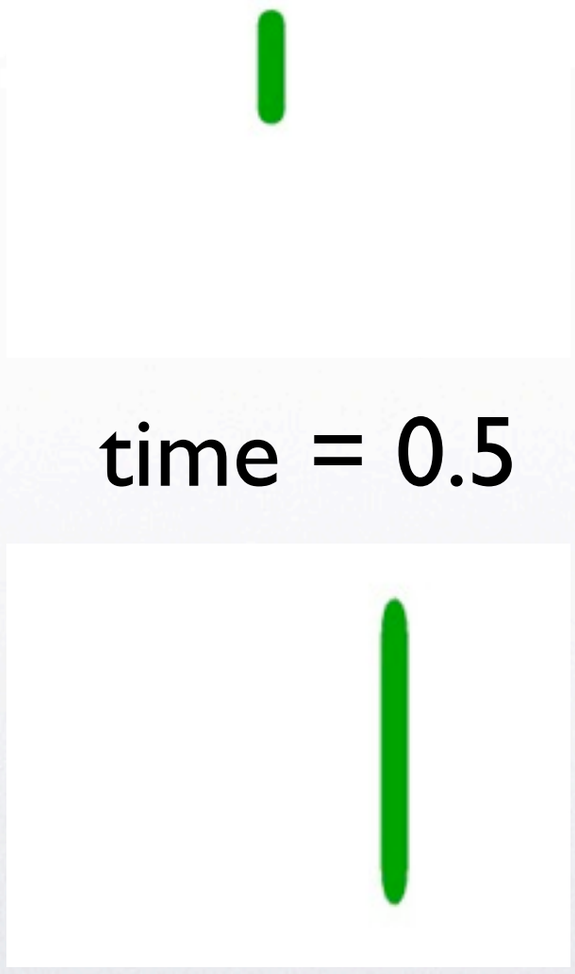
image after morphing

time = 0.5

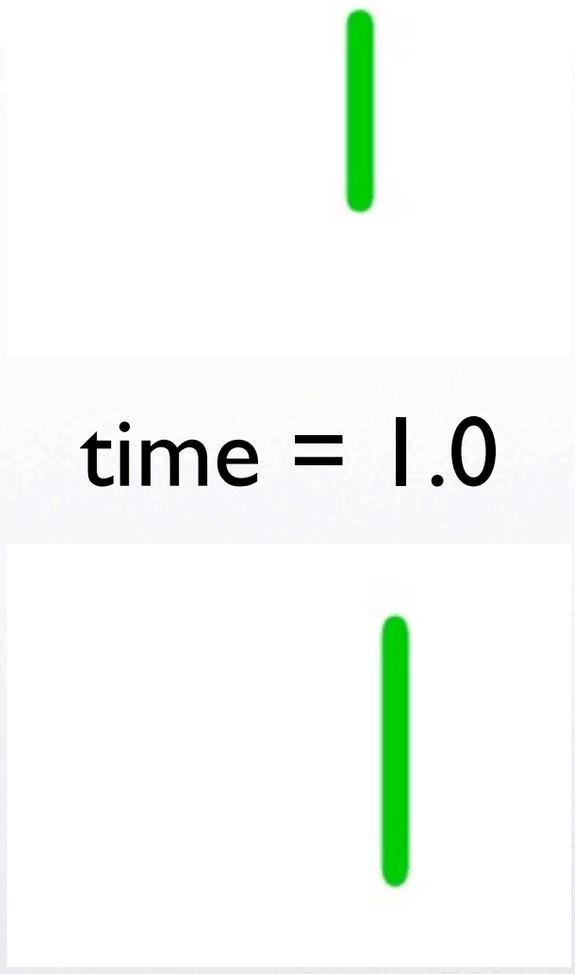
example-line



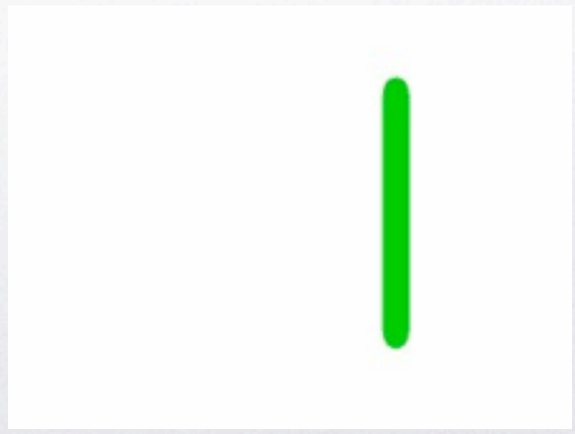
time = 0



time = 0.5



time = 1.0

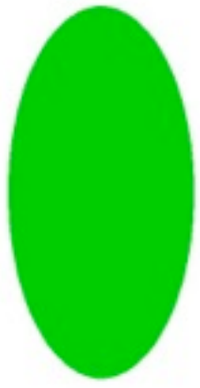


time = 1.2

Example 2

image 1

image 2



Example 2

image 1 after warping

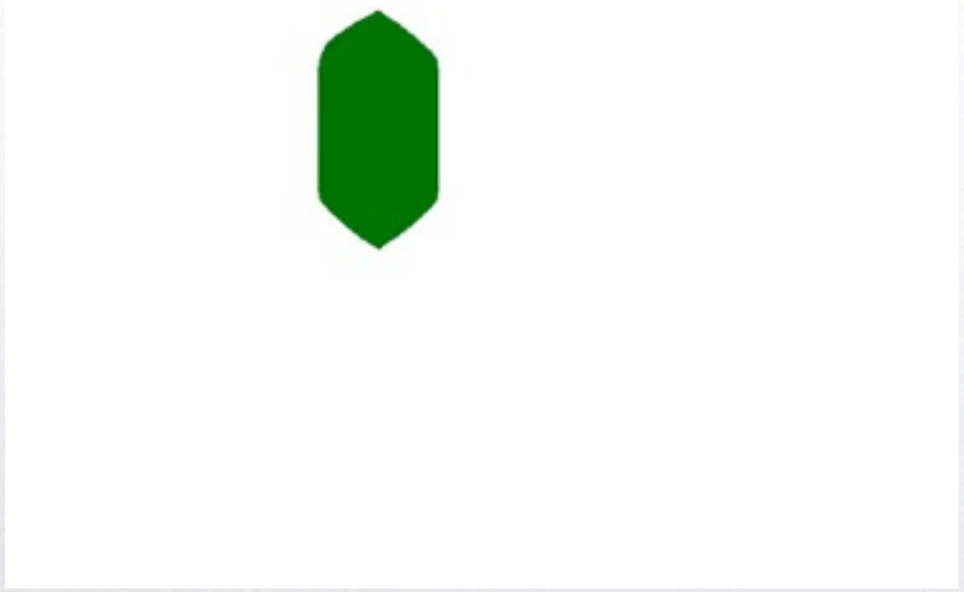
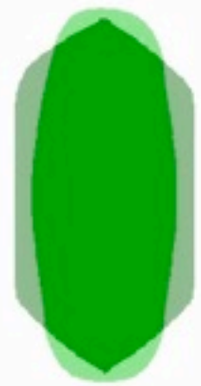


image 2 after warping

Example 3

Time = 0.5

image after morphing



Example 3

image 1

image 2

Example 3

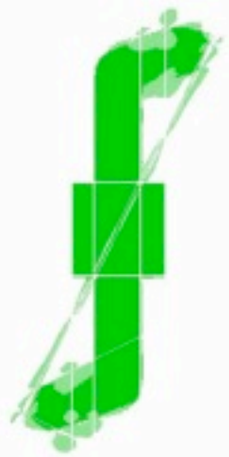
image 1 after warping

image 2 after warping

Example 3

Time = 0.5

image after morphing



To do in the future

- Fitting algorithm
- Automatically detect vector

Reference

- http://planck.cf.ac.uk/science/mm-wave-astronomy
- http://davis.wpi.edu/~matt/courses/morph/2d.htm
- http://en.wikipedia.org/wiki/Key_frame

END

Thank you