Shape Interpolation

Team No. 17

Aayush Kumar

1. INTRODUCTION

A shape is a result of the blending of various poses. Morphing, also known as metamorphosis or shape interpolation, is a method of blending two shapes in a smooth and reasonable manner. Blending and morphing techniques include creating a smooth transition from an initial (source) object to the target object. In most cases, these morphing effects are generated using image-based techniques where the geometry of the source and target object remains similar. A method called Triangulation is widely used. A correspondence is generated between geometric features of the representation by using some affine transformations over the source body so that the target form can be generated by interpolating between the positions. In this project, we are working on techniques to interpolate any polygon or polyhedron from one representation to another.

2. LITERATURE REVIEW

Our aim is to create an in-between sequence of shapes by blending the edges of the shapes. Several ways exist to compute the relative orientation of two shapes. In Computer Graphics, it is difficult to interpolate the orientation of more than two shapes in 3D, so the following discussion will be about *two-dimensional* forms.

Initially, we can detect boundaries using the formulation of Contours using the Opencv library. From that, we will be able to point to the number of Contours in the shapes and if the number of points in the two shapes is equal (to make sure that our shape can interpolate into the required shape otherwise, we cannot do interpolation between the two shapes). After getting the required computation, we can use the OpenCV libraries to match the points of the two forms, which will get linearly interpolated.

For correspondences between the two shapes, we can use the concept of *Interpolation of intrinsic boundary representation [1][5]*.

Assume the polygons P and Q are described by their vertex positions $\mathbf{p_i}, \mathbf{q_i}$. Let $\theta_{p_i}, \theta_{Q_i}$ be the interior angles around $\mathbf{p_i}, \mathbf{q_i}$ and

$$L_{P_i} = |\mathbf{p_{i+1}} - \mathbf{p_i}|, L_{Q_i} = |\mathbf{q_{i+1}} - \mathbf{q_i}| \tag{4.6}$$

the length of the i-th edge. Additionally, let $\alpha_{P_i}, \alpha_{Q_i}$ be the angles between the i-th edge and a fixed axis. An intermediate polygon is represented by

$$L_i(t) = (1-t)L_{P_i} + L_{Q_i} \tag{4.7}$$
$$\theta_i(t) = (1-t)\theta_{P_i} + \theta_{Q_i} \tag{4.8}$$
$$\alpha_i(t) = (1-t)\alpha_{P_i} + \alpha_{Q_i} \tag{4.9}$$

However, this description will lead to an open polygon in the general case.

We know that, whether a shape is in 2D or 3D, it consists of features. We can use *Triangle to Triangle Interpolation* [4]. The affine transform between the source and target triangles was decomposed into a rotation and a stretching transformation. To achieve the least-distorted morph, the free parameters were linearly interpolated (*fig.1*). A shape can also be decomposed into pair polygons and several features using the *path interpolation approach* [3]. By blending the two polygons, intermediate polygons can be generated that help determine each feature's location. In

the end, all intermediate features must be attached without gaps so that the intermediate shape can be reconstructed. Using the intrinsic method, the corresponding features are blended. It allows the 2D shapes to blend smoothly and naturally(*fig.2*).

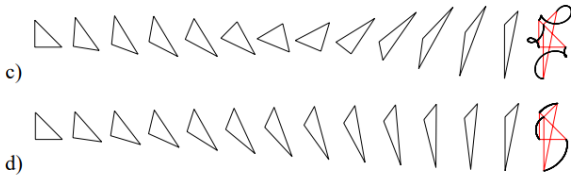


Figure 5: Transformations of a single triangle. (a) Linear vertex interpolation. (b-d) An affine map from the source to the target triangle is computed and factored into rotational and scale-shear parts. Intermediate triangles are constructed by linearly interpolating the angle(s) of rotation, the scaling factors, and the shear parameter.

fig.(1)

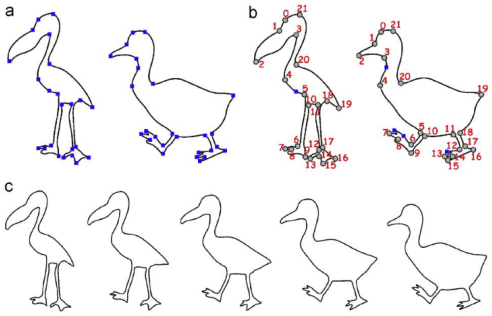
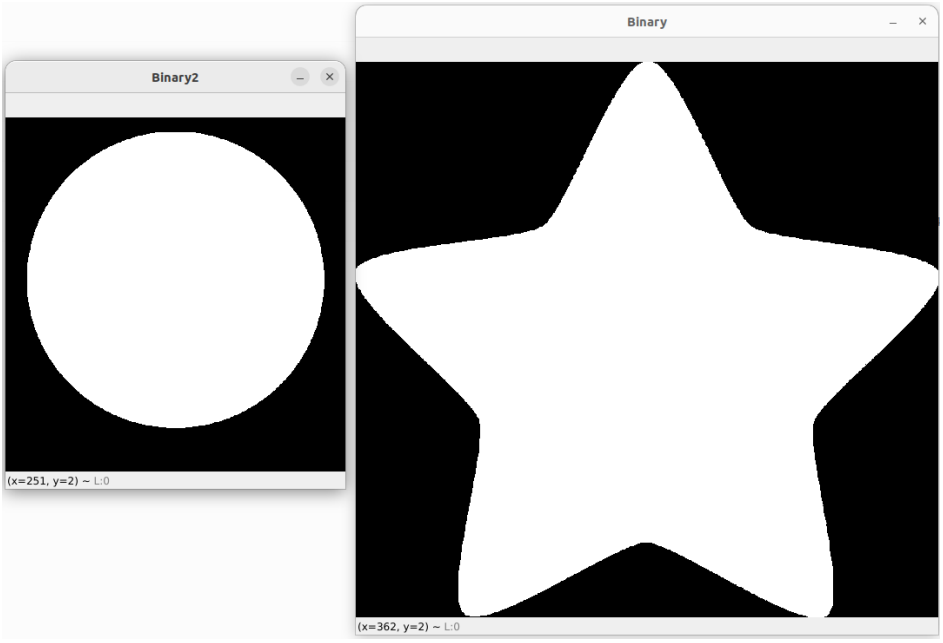


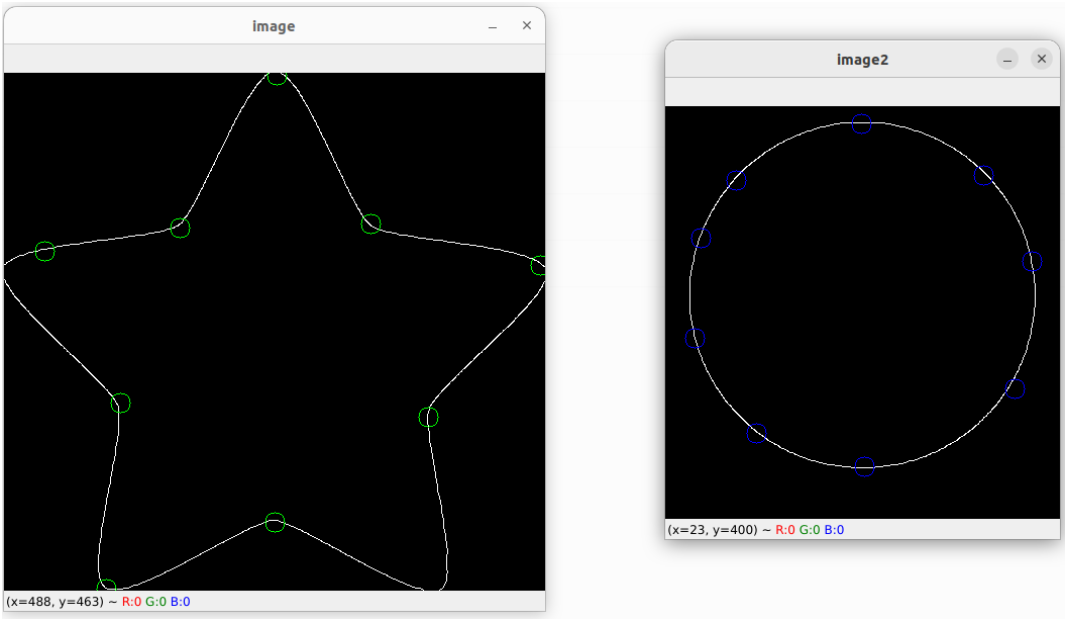
fig.(2)

3. MILESTONES

S. No.	*Milestone*	*Member*	*Completed*
1	RGB to Binary Image Formation using OpenCV (for both source and target images)	Aayush	✅
2	Contours Detection	Aayush	✅
3	Feature Points + Edge Detection	Aayush	✅
4	Blending of Shapes to show Interpolation	Aayush	✅

Update till Project Evaluation - III





We have made the contours detection in the source and target image respectively. After that we have accomplished the milestones of Feature Points Detection & Edge Detection in the source and target images using OpenCV libraries and tutorials. Milestones accomplishments are also mentioned in the above table.

REFERENCES

1. https://dl.acm.org/doi/pdf/10.1145/133994.134001
   Sederberg, T.W., Gao, P., Wang, G., Mu, H.: 2D shape blending: An intrinsic solution to the vertex path problem. In: SIGGRAPH 1993 Proceedings, vol. 27, pp. 15–18 (1993)
2. http://www.cs.technion.ac.il/~mirela/publications/psibd.pdf
3. https://shorturl.at/quGKS
4. https://link.springer.com/chapter/10.1007/11424857_113
5. https://tuprints.ulb.tu-darmstadt.de/213/5/diss4a.pdf
6. https://learnopencv.com/opencv-threshold-python-cpp/
7. https://scialert.net/fulltext/?doi=itj.2014.2425.2430
   Chao Song , 2014. Improvement of As-Rigid-As-Possible Interpolation in Shape Correspondence. *Information Technology Journal, 13: 2425-2430.*
8. http://www.cs.tau.ac.il/~dcor/online_papers/papers/arap.pdf
   Alexa, M., Cohen-Or, D., Levin, D.: As-rigid-as-possible shape interpolation. In: SIGGRAPH 2000 Proceedings, pp. 157–164 (July 2000)