Tampere University

Jiaqi Guo

A BRIEF SURVEY ON IMAGE MORPHING

Signal Processing
Bachelor Thesis
April 2020

ABSTRACT

Jiaqi Guo: A Brief Survey on Image Morphing
Bachelor Thesis
Tampere University
International Bachelor's Degree Programme in Science and Engineering
April 2020

Image morphing has been recognized as an interesting and powerful tool for visual effects, allowing one object smoothly transforming to another object. It is widely applied in image editing and film production applications, so there are many researches on this topic. This thesis surveys the various proposed techniques in the morphing field which try to improve the morphing quality and describes some outstanding and advanced morphing techniques. The fundamental technique shared by every algorithm is first analysed, then followed by feature-based, optimization-based and learning-based method. Each of them has their own strength and focuses on different point. An experiment is conducted to evaluate the advantages and disadvantages of those methods. The result and example from the literature reveal that there still needs further study on transformation of 2 dissimilar objects and images having large occlusions.

Keywords: image morphing, warping, image translation

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

PREFACE

This is a short survey on image morphing techniques in recent 20 years. The techniques studied are limited to 4, due to the scope of the bachelor thesis.

I would like to express my gratitude to my supervisor Joni Kämäräinen for providing me this interesting topic and encouraging me to explore more impressing research topics. I would like to thank my friend GaoMing Zheng who has been supporting me throughout the tough writing process. And I wish to thank my friends Rostislav Duda, Savanna Lujan, YuanYuan Guan and the cat Pami for allowing me to use their pictures to conduct my experiments.

I would like to show my heartful appreciation to the front-line medical staffs during the coronavirus pandemic for providing me a safe and peaceful time to finish the last part of my thesis.

Tampere, 20 April 2020

Jiaqi Guo

CONTENTS

1. INTRODUCTION 1
2. MORPHING ALGORITHMS 3
   2.1 Feature-based morphing algorithm 3
   2.2 Regenerative morphing 6
   2.3 Morphing using Structural Similarity on a Halfway Domain 8
   2.4 Geometry-Aware Unsupervised Image-to-Image Translation 10
3. EXPRERIMENTS AND RESULTS 13
   3.1 Feature-based Morphing with D-lib 13
   3.2 Liao's Method: Structural Similarity on a Halfway Domain 14
   3.3 Result 15
4. CONCLUSIONS 19
5. REFERENCES 21

LIST OF FIGURES

Figure 1. *The Morphing technique applied in the movie Star Trek VI: The Undiscovered Country (1991), where an alien transformed in to Captain James T.Kirk.* 1

Figure 2. *Image morphing. Top row: result obtained only by blending, visible ghosting effect occurs. Bottom row: images first are warped to the same position, then resulting images are blended creating a better morphing [7].* 3

Figure 3. *Flipping an edge.[5]* 4

Figure 4. *Inverse warping algorithm, (a) a pixel $g(\boldsymbol{x}')$ is resampled from its corresponding location $\boldsymbol{x} = \boldsymbol{h} - 1(\boldsymbol{x}')$ in image $f\boldsymbol{x}$. (b) Detail of the source and target pixel locations. [12]* 5

Figure 5. *The intermediate morphed frame $\boldsymbol{T}n$ is similar to the 2 source images $\boldsymbol{S}1$ and $\boldsymbol{S}2$ (Source Similarity), and the local changes from $\boldsymbol{T}n$ to 2 neighbouring frames $\boldsymbol{T}n-1$ and $\boldsymbol{T}n+1$ are smooth (Temporal Coherence) [10].* 7

Figure 6. *The Bidirectional Similarity: (a) the input source image and output image are only coherent; (b) only complete; (c) both complete and coherent [10].* 7

Figure 7. *The intermediate image function ϕ is compose by two functions. Each function is defined on the halfway domain by using a common vector field V [9].* 9

Figure 8. *The halfway parameterization. The left image is 1D vertical segments and the related correspondence vector field $v(p)$ is plotted on the right [9].* 9

Figure 9. *An extra vector $w(p)$ defines a quadratic motion path to minimize deformation during the morph process [9].* 10

Figure 10. *Architecture of the framework consists of 4 components: 2 conditional variational autoencoders in X and Y domain, and 2 transformers in appearance and geometry domain separately [15].* 11

Figure 11. *The triangulation results using 68 points generated by D-lib and manually annotated 11 points of 2 input images.* 13

Figure 12. *The interface of the prototype system for Liao's method* 14

Figure 13. *The two resource images inputs (first row) with manually selected 9 correspondences on the face used in Liao's method; Second row, the morphing sequence obtained using Liao's method; Third row, the intermediate morphing image with traditional feature-based morphing with D-lib. Both sequences have the same blending factor $\alpha = 0.2, 0.4, 0.6, 0.8$.* 15

Figure 14. *Translating a cat (left column) to a man. The morph sequence with α=0.25 0.5 0.75.* 16

Figure 15. *Morphing results when orientation changed causes large occlusion.* 16

Figure 16. *The improved fully automated method results with warped morphing sequence 0.25,0.5.0.75 [16].* 16

Figure 17. *Morphing results from the regenerative morphing. Top row: a cloud to a face. Bottom row: a boy rolling a basketball [10].* 17

Figure 18. *Linear interpolation results of geometry and appearance latent codes in unsupervised translation [15].* 18

Figure 19. *TransGaGa results on synthesis datasets (transfermation between cow and cheetah, lion and rhino) and real-world datasets (transfermation between cat human face, giraffe and horse) [15].* 18

1. INTRODUCTION

Image metamorphosis has shown to be a considerably strong and popular technique in visual special effect. There are many unforgettable morphing examples in the movies, television serious and music videos. For example, in the sixth *Star Trek* movie released in 1991, alien Martia seamless transformed into Captain James T.Kirk as shown in the Figure 1. And in *The Wolverine* (2013), the villain Yashida who attempted to steal Wolverine's healing power, his appearance changed from old to young, and back to old again.

This widely used process is known as image morphing in the field of computer graphics and has been studied and explored over 25 years in the research field. This technique is achieved by combining image warping and interpolation. It utilizes the 2D geometric transformation from source image to target image which will preserve alignment between corresponded features and creates a sequence of blended intermediate images by colour interpolation.

Figure 1. The Morphing technique applied in the movie Star Trek VI: The Undiscovered Country (1991), where an alien transformed in to Captain James T.Kirk.

Thus, in the early study of image morphing, Wolberg in his survey considered that generating high quality morphing is straight tied to the following three problems: *feature specification*, *warp generation*, and *transition control* [14]. Feature specification means establishing feature correspondence between images. It often requires manually annotating landmarks, which is time consuming and tedious. The most challenging among the

three problems would be the warp generation, which refers to build a good correspondence map between the images. Moreover, the transition rate will directly influence the quality of the metamorphosis sequences. However, recent researches on image morphing in [9] and [10], examine it as an optimization problem with certain objective function. Later, as machine learning has become a popular topic, unsupervised learning was also applied on the image morphing. The study in [15] presented a framework to tackle this task when there are large geometry variances between a pair of images using unsupervised learning.

The scope of this brief survey is limited and will be narrowed to 2D image morphing. 3D morphing is beyond the range of this thesis. This thesis has the following structure. Chapter 2 will discuss the basic principle of 2D image morphing algorithm and briefly review several existent morphing techniques, which includes feature-based morphing, regenerative morphing, automating image morphing using structural similarity on a halfway domain which forms a 2D vector field over a domain in the halfway between 2 images, and a learning-based model using GAN. And in chapter 3, we conducted experiments by utilizing two different methods, one is halfway domain using structural similarity method, and another one is the traditional method.

2. MORPHING ALGORITHMS

Before establishing the morphing algorithm, the simple alpha-blending algorithm was widely applied to the transition between two images of different objects. This method will give a rather poor result. It causes clearly visible ghosting effect which is shown in top row in Figure 2.

The traditional and basic morphing algorithm was derived to achieve a good morphing, which preserves the feature, maintains the smoothness and avoids linearity [7]. Hence, instead of directly blending two sources, each recourse is warped towards to another source before they combine. This process is illustrated in the bottom row in Figure 2.

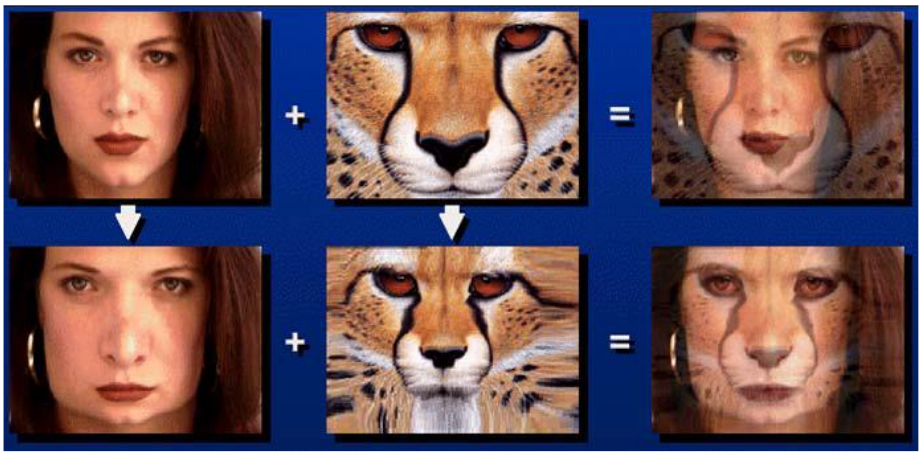

Figure 2. Image morphing. Top row: result obtained only by blending, visible ghosting effect occurs. Bottom row: images first are warped to the same position, then resulting images are blended creating a better morphing [7].

In this section, we concisely review several common morphing algorithms. The algorithms mentioned in this paper could be categorized into feature-based morphing, optimization-based morphing, and learning-based morphing.

2.1 Feature-based morphing algorithm

Feature-based morphing technique and together with all the other algorithms involve 3 steps:

- Local spatial transformation
- Local colour change
- Continuous transition

The first two steps are called image warping. While global warping only performs one transformation between source image and target image, local image warping could perform such a transformation that different regions have different motions. This will ensure that the changes over the boundaries are smooth. Here are some of the commonly used methods. The first method is to use corresponding oriented line segments to illustrate the local deformation [2]. A second approach is to use B-splines snakes [3]. This technique was used in Apple Shake, a visual effect software, and it has been widely used in medical image processing.

Yet another way is to select a sparse set of corresponding key points. A dense displacement field could be obtained by interpolating the displacement of the sparse points' set using various techniques. Among these different mathematical tools, one of the easiest and most intuitive to understand is using triangulated meshes. This approach is to apply triangulation on the set of key points, and then find affine transformation within each pair of triangles [12]. Then the colour information, could be retrieved by using inverse warping algorithm.

In the following section, we will discuss the triangular meshed warping which is used in the later experiment.

Delaunay Triangulation The triangulation divides polygon into triangles, one of which feature application is interpolation. A pair of drawing triangulations are topologically equivalent and isomorphic. They can be morphed one into another while avoiding partial crossings at the intermediate steps. The idea was already proofed in [4] and [13]. A sparse set of feature points does not have an interior, so it could be treated as a convex polygon. Hence the motion of these points could be interpolated to continuous dense motion field.

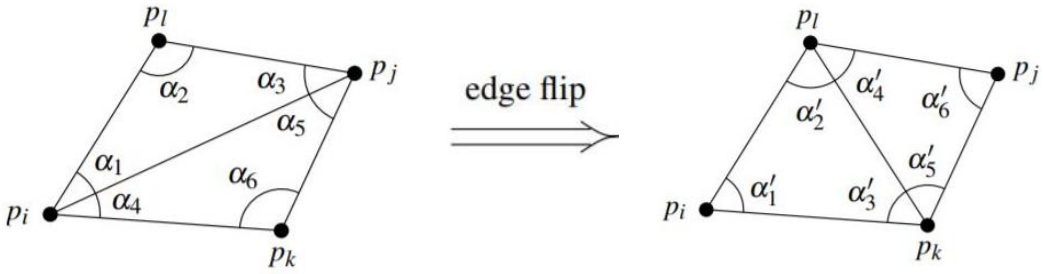

Figure 3. Flipping an edge.[5]

As shown in the Figure 3, the same data set could result in different triangulations. We could obtain a new triangulation by replacing line $p_i p_j$ with line $p_l p_k$. The triangulation on the left contains long and skinny triangles, which means that the vertices of a triangle tend to be spread far from each other, and the smallest angle of the triangle tends to be

smaller than the one in the right triangle. This edge is defined to be illegal, if flipping this edge can increase the smallest angle.

Such a situation is undesirable. We could conclude that a triangulation which contains extreme small angles is of low quality, and the triangulations are ranked by evaluating the smallest angle. This implies that if all edges in a triangulation are legal then it is a Delaunay triangulation. This follows the theorem in [5]:

> ***Theorem 1.*** *Let P be a set of points in the plane. Any angle-optimal triangulation of P is a Delaunay triangulation of P. Furthermore, any Delaunay triangulation of P maximizes the minimum angle over all triangulations of P.*

Barnhill had observed such a triangulation pairs will lead to a good interpolation [1]. In such a way we will obtain a set of triangle-to-triangle correspondences which is prepared for the further movement.

Affine transformations In order to define local transformation for each triangle, the basis of the operation is an affine transformation. Once the transformation between two triangles is found, the points in one triangle can be transformed to form the new triangle. This transformation illustrates that how the faces are warped in geometric way. Affine transformation is a mapping from one affine space to another that preserves affine combinations. These include the following transformations of space: translation, rotation, uniform and nonuniform scaling, reflection and shearing.

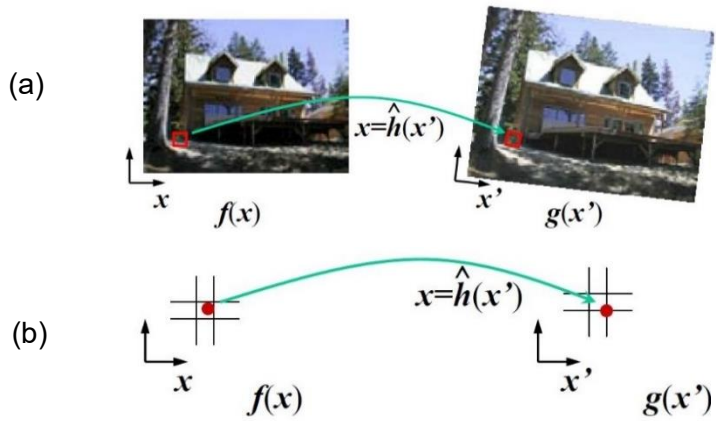

Figure 4. Inverse warping algorithm, (a) a pixel $g(\boldsymbol{x}')$ is resampled from its corresponding location $\boldsymbol{x} = \boldsymbol{h}^{-1}(\boldsymbol{x}')$ in image $f(\boldsymbol{x})$. (b) Detail of the source and target pixel locations. [12]

Inverse warping Knowing the transformation of the locations, $\boldsymbol{x}' = \boldsymbol{h}(\boldsymbol{x})$, the problem now is how to calculate the pixels in the new image $g(\boldsymbol{x}')$, given the source image $f(\boldsymbol{x})$.

One favoured solution is called inverse warping. Each pixel in the new image is sampled from the source image, and this is illustrated in the Figure 4.

Cross dissolving after obtained a sequence of interpolated frames, the pixel cross-dissolving of 2D morphing suggests a linear operation. It is a simple Alpha Blending method and works well for blending the colour information of source to target image. The cross-dissolving can be presented in the following way:

$$I_i = (1 - \alpha) \times Source + \alpha \times Target \quad (1)$$

where I_i is the intermediate image, and α is the blending control factor with the range from 0 to 1.

2.2 **Regenerative morphing**

Building upon the *bidirectional similarity*, Shechtman *et al* proposed a new image morphing approach, which does not require manual correspondences, and in which different regions of the scene move at different rates, resulting a more natural motion [10]. The author considered morphing as an example-based optimization problem with the unified objective function as Equation 2.

$$E_{Morph}(T_1, \dots, T_{N-1}) = E_{SourceSim}(T_1, \dots, T_{N-1}) + \beta E_{TempCoher}(T_1, \dots, T_{N-1}) \quad (2)$$

where the goal is to find a sequence of frames $\{T_n\}_1^{N-1}$ which is expected to have these two properties:

1. *Source Similarity* ($E_{SourceSim}$) – every local part in each frame T_n, should be similar to some part in either of the source images or both. This will limit the loss of detail and ghosting effect.
2. *Temporal Coherence* ($E_{TempCoher}$) – the similarity of all local part in the frame T_n to two near by frames T_{n-1} and T_{n+1} should be strong. Therefore, the changes across the sequence are temporally smooth in both appearance and motion.

This idea is also illustrated in Figure 5.

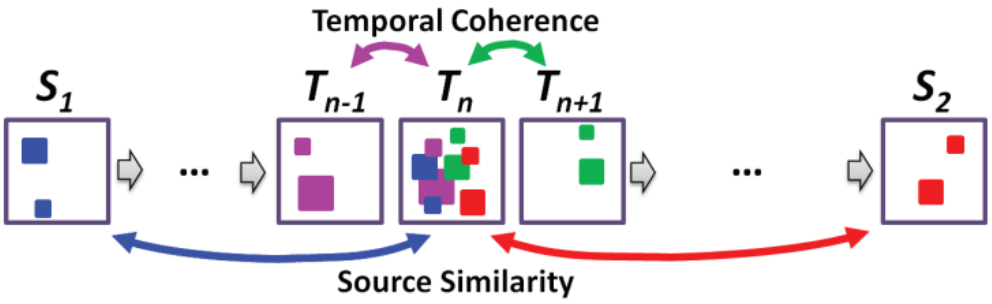

Figure 5. The intermediate morphed frame $\boldsymbol{T}_n$ is similar to the 2 source images $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ (Source Similarity), and the local changes from $\boldsymbol{T}_n$ to 2 neighbouring frames $\boldsymbol{T}_{n-1}$ and $\boldsymbol{T}_{n+1}$ are smooth (Temporal Coherence) [10].

The author defined the similarity by bidirectional similarity method. The bidirectional distance calculated distance between a source image *S* and target *T,* is composed of two terms. The first term is the *Coherence*, which penalizes the artifacts in the output and shows in the Figure 6 (a). For every pixel in the output image, there always exists a similar pixel in the source input. As for the *Completeness*, illustrated in Figure 6 (b), it means that for every pixel in the source input image, there is always a close pixel in the output image. In other words, the visual information from the input image is reserved as much as possible in the output image.

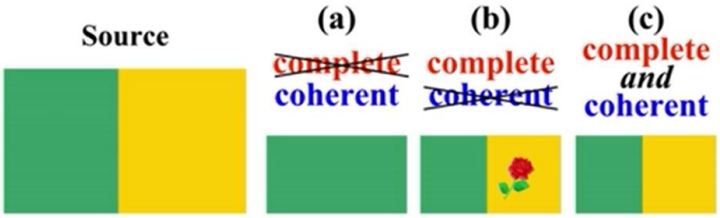

Figure 6. The Bidirectional Similarity: (a) the input source image and output image are only coherent; (b) only complete; (c) both complete and coherent [10].

The general formula for this distance in Equation 3 is minimized to find the T. Later in the study, author observed that the coherence and the completeness can be generalized from multiple resources.

$$d_{BDS}(S,T) = \underbrace{\frac{1}{N_S}\sum_{s\subset S}\min_{t\subset T} D(s,t)}_{d_{Complete}(S,T)} + \underbrace{\frac{1}{N_T}\sum_{t\subset T}\min_{s\subset S} D(t,s)}_{d_{Coher}(S,T)} \quad (3)$$

Thus, the term α-Blended Completeness term is further derived as in following Equation 4:

$$d_{\alpha BlendComplete}(T, S_1S_2, \alpha) = \frac{\alpha}{N_{S_1}} \sum_{s1 \subset S_1} \min_{t \subset T} D(s_1, t) + \frac{1-\alpha}{N_{S_2}} \sum_{s2 \subset S_2} \min_{t \subset T} D(s_2, t) \quad (4)$$

where the minimization is achieved by having a disjoint region in T. For each region, it is required to be similar to S_1 or S_2 , but not strictly to both of them. And it tends to reserve the sharpness of the resource image. To avoid undesired blur caused by the colour misalignment and inherent geometric, the α-Disjoint Coherence term is used with a bias term as shown in the Equation 5.

$$d_{\alpha DisjCoher}(T, S_1S_2, \alpha) = \frac{1}{N_T} \sum_{t \subset T} \min(\min_{s1 \subset S_1} D(s_1, t), \min_{s2 \subset S_2} D(s_2, t) + D_{bias}(\alpha)) \quad (5)$$

Then combining these completeness and coherence terms together into the morphing objective function 1. The $E_{TempCoher}$ and $E_{SourceSim}$ could be rewritten as

$$E_{TempCoher}(\{T_n\}_1^{N-1}) = \sum_{n=1}^{N-1} d_{\alpha BlendBDS}(T_n, T_{n-1}, T_{n+1}, 0.5) \quad (6)$$

$$E_{SourceSim}(\{T_n\}_1^{N-1}) = \sum_{n=1}^{N-1} d_{\alpha BlendComplete}\left(T_n, S_1, S_2, \frac{n}{N}\right) + d_{\alpha Dis}(T_n, S_1, S_2, \frac{n}{N}) \quad (7)$$

where the blending factor is set to be 0.5. Finally, an iterative algorithm can be applied to optimize the objective.

This new regenerative approach replaced traditional warp and blend approach, synthesized directly from local regions in the two source images. Besides, it is able automatically create interesting morphing from visually unrelated correspondences. But it also has limitations. The way of dealing the colour is insufficient. The morphing quality will decrease in the case of view interpolation if there is change in illumination.

2.3 **Morphing using Structural Similarity on a Halfway Domain**

The third algorithm studied is proposed in [9], called *Structural Similarity on a Halfway Domain*. This algorithm utilizes fast optimization to align complicated shapes with small-

scale interactive guidance. It requires few correspondence points specified by the user, avoids ghosting effect by optimizing alignment.

The traditional method of constructing a map from one image to another image is always not satisfying. One existing challenge is that it leaves the dis-occluded region undefined in the mapping function. Because dis-occluded region is invisible in one image but not another, which will cause the discontinuity. The author solved this problem by the new approach, *Halfway Parameterization*. The idea is to define a 2D vector field *v* over half-way domain Ω between two source images and create two mapping functions ϕ_0 and ϕ_1 using the same vector field v as shown in Figure 7. The composition of these two functions is the mapping for the intermediate image. The Figure 8 provides an accurate

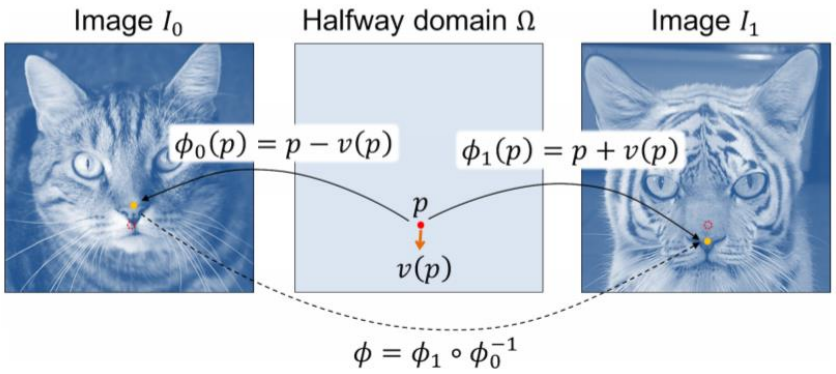

Figure 7. The intermediate image function ϕ is compose by two functions. Each function is defined on the halfway domain by using a common vector field V [9].

illustration where the discontinuities caused by simple occlusions become continuous as the vector field $v(p)$.

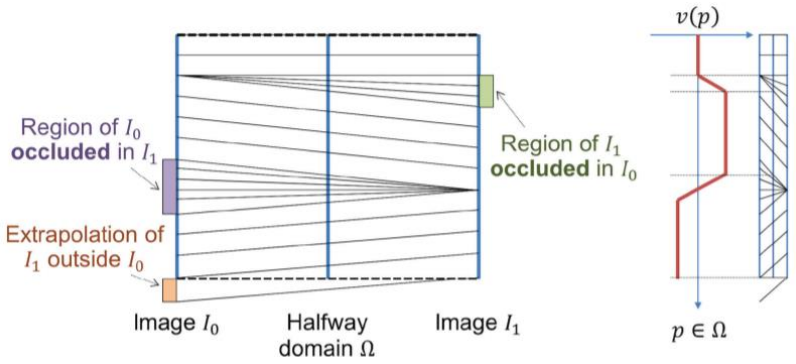

Figure 8. The halfway parameterization. The left image is 1D vertical segments and the related correspondence vector field $v(p)$ is plotted on the right [9].

In order to solve and optimize the vector field *v(p)*, the objective energy function:

$$E(p) = E_{SIM}(p) + \lambda E_{TPS}(p) + \gamma E_{UI}(p) \quad (8)$$

is then minimized. As shown in Equation 8, the energy function consists of three terms. The first term E_{SIM} is the similarity energy which is used to evaluate the similarity of edge structure of a pair of corresponding neighbourhoods. It utilizes a modified version of structural similarity index (SSIM) without a luminance term. The second term is called smoothness energy. It biases the affine transformation during optimization by minimizing the thin-plate spline (TPS) energy independently on the hallway domain grid. The third term is a soft constraint named UI energy which ensures the intermediate point of the pairs of user-specific corresponding points lies on the grid in the halfway domain.

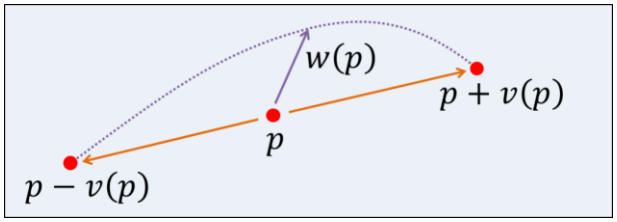

Figure 9. An extra vector $w(p)$ defines a quadratic motion path to minimize deformation during the morph process [9].

Given the optimization result, the next step is to move each point to its target point. The common procedure is using linear interpolation, but the linear motion path could cause undesirable deformation. Thus, the author proposed a quadratic motion path by adding an extra vector $w(p)$ which defines the control point for a quadratic Bézier path as shown in Figure 9.

This model employs the halfway parameterization for building correspondences which allows us handling simple occlusions without compromising the optimization process. But processing more complex occlusions and rotations is a remaining issue.

2.4 Geometry-Aware Unsupervised Image-to-Image Translation

Early works show the effectiveness of using unsupervised in learning mapping between two different image domains. Soon, researchers learned it had advantages of transferring local textures, but had limitations on the complicated translation. For example, it could fail in mapping a pair of image domain having large geometry variation. To solve this issue, some researchers suggested to perform the translation on the higher semantic level, which means understanding the content of each component. A new approach was proposed based on these ideas [15]. In this work, instead of learning the map between the images directly, the images were decomposed into the Cartesian product of geometry latent space and appearance latent space. Furthermore, the author suggested the geometry prior loss and the conditional VAE loss, which could motivate the network to

learn independent but complementary representations, and then to perform the appearance translation and geometry translation separately.

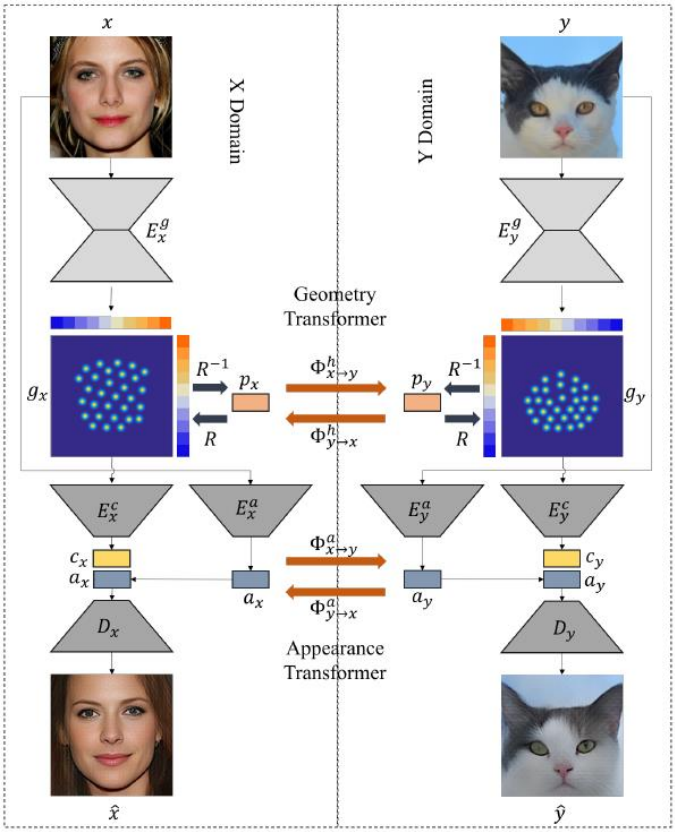

Figure 10. Architecture of the framework consists of 4 components: 2 conditional variational autoencoders in X and Y domain, and 2 transformers in appearance and geometry domain separately [15].

The Figure 10 above presents the architecture of this approach. The whole framework is composed of 2 conditional variational auto-encoders in X and Y domain, and 2 transformations in geometry and appearance. For given an input image $x \in X$, the unsupervised geometry estimator of the VAE system in X domain E_x^g will process and obtain the geometry representation g_x. It has the same resolution as the input x, and it is a 30-channel point-heatmap. Then the geometry code c_x in latent space is generated by the geometry encoder E_x^c. Meanwhile, the appearance code a_x is obtained by embedding the input image x in the appearance encoder E_x^a. Eventually, c_x and a_x will be catenated together and passed to the decoder D_x which will map the latent space back to the image space and generate the new image $\hat{x}$.

The model utilizes several loss functions so as to improve the model training. The conditional VAE loss is implemented as Equation 9 in the system:

$$\mathcal{L}_{CVAE}(\pi, \theta, \phi, \omega) = -KL\left(q_\phi(c|x, g) \parallel p(a|x)\right) + \left\|x - D(E^c(E^g(x)), E^a(x))\right\| \quad (9)$$

where calculates the KL-divergence loss and the reconstruction loss between x and $\hat{x}$. Then the prior loss is introduced to constrain the learning of the structure estimator E_x^g and E_y^g. The first term in Equation 9 is the separation loss which ensures that the heatmap generated by the E^g could efficiently cover the target of interest. The second term in the Equation 10 concentration loss which is the variance of activations g.

$$\mathcal{L}_{prior} = \sum_{i \neq j} \exp\left(-\frac{\left\|g^i - g^j\right\|^2}{2\sigma^2}\right) + Var(g) \tag{10}$$

The system considers using CycleGAN to tackle the transformation problem on the appearance latent space. In order to guarantee the visual relationship between images associated with g_x and all the translated appearance $\Phi_{X \to Y(g_x)}$, the cross-domain appearance consistency loss is introduced:

$$\mathcal{L}_{con}^a = \left\|\zeta(x) - \zeta(D_y(\Phi_{x \to y}^g \cdot E_x^g(x), \Phi_{x \to y}^a \cdot E_x^a(x)))\right\| \tag{11}$$

where ζ is the gram matrix calculated with a pretrained VGG-16 network. Furthermore, the cycle-consistency loss and adversarial loss are leveraged as well during the model training. Thus, the total loss of this approach is as follow Equation 12:

$$\mathcal{L}_{total} = \mathcal{L}_{CVAE} + \mathcal{L}_{prior} + \mathcal{L}_{con}^a + \mathcal{L}_{cyc}^a + \mathcal{L}_{cyc}^g + \mathcal{L}_{cyc}^{pix} + \mathcal{L}_{adv}^a + \mathcal{L}_{adv}^{pix} \tag{12}$$

This framework also employs the de facto and PCA embedding of coordinate during geometry transformation. The model extends the ability of CycleGAN on more complicated objects like animals and supports multi-model translation.

3. EXPRERIMENTS AND RESULTS

This chapter will present the testing process of applying two morphing algorithms mentioned in the previous chapter on real image data, as well as the direct results achieved from each approach or examples from the published paper.

3.1 Feature-based Morphing with D-lib

The implementation of feature-based morphing algorithm was done under the Anaconda-Python (3.6) environment with OpenCV installed. The OpenCV is a real-time computer vision library with many programming functions. Another supportive toolkit is D-lib. It provides help in developing modern machine learning software in C++ and Python environment. Particularly, it maintains a large number of algorithms for performing classification and feature detecton [8].

Figure 11. The triangulation results using 68 points generated by D-lib and manually annotated 11 points of 2 input images.

To obtaining corresponding points, we used the pre-trained facial landmark detector inside the D-lib, to obtain the location of 68 (x, y)-coordinates that map to facial features on the face. In addition, 3 points (one on the neck, two on each side of the shoulder) are manually added to the feature point set. 7 corner points of the image were added as well in order to have a more precise result. Hence, we collected a sparse set of 79 key points for each image.

On the set of sparse feature points we performed Delaunay triangulation. The output of Delaunay triangulation is a list of triangles represented by the indexes of points in the array of 79 points. In this particular task, the triangulation produces 148 triangles connecting the 79 key points. The triangulation result is stored as an array of three columns.

The visualization of the triangulation result generated from two source images is shown in Figure 11.

3.2 Liao's Method: Structural Similarity on a Halfway Domain

The full code of the project is compiled in the x64 Windows with the compiler Visual Studio 2010 and the libraries CUDA 6.5, CUSP 0.3.0, QT 5.3.2, OpenCV 3.0, and Intel MKL 2015. We attempted to build up this program but failed. But luckily, the author provided an executable application.

As shown in the Figure 12, it is the user interface of the prototype application with a test example. The two images lay on the top row are the input source images, which are required to be in the same size. Mouse clicking operation allows user to add, delete or correct user-specific corresponding points on the images. The halfway domain image of the current task is in the bottom left part. And finally, the bottom right part is the output morphing animation. The bottom parts are updated interactively while the optimization algorithm is running in the background. In addition, the optimization process will restart instantly if the user modifies the corresponding points during the process.

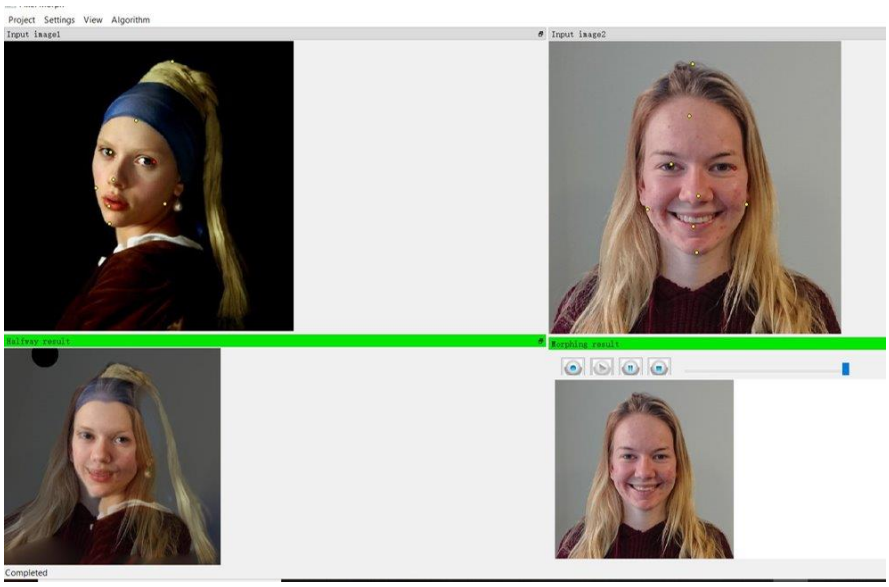

Figure 12. The interface of the prototype system for Liao's method

3.3 **Result**

The Figure 13 below presents the results from the task where one face object morphs to another face object. Two images listed in the top row are the source images. There are 9 points which are manually selected used as the user specific correspondence points in Liao's prototype system. We generated 4 intermediate images where the blending factor $\alpha = 0.2, 0.4, 0.6, 0.8$ respectively, to illustrate the changing morphing sequence for both of the methods. It is quite obvious that some undesirable features occurred in the intermediate images, for example the faded hair on the shoulder and on the forehead. This is due to the fact that there are no correspondence points for the hair in the other image. The Liao's method on the second row in Figure 13 handled the forehead hair part a little better than the first method. But the facial transformation is processed smoothly in general for both the cases.

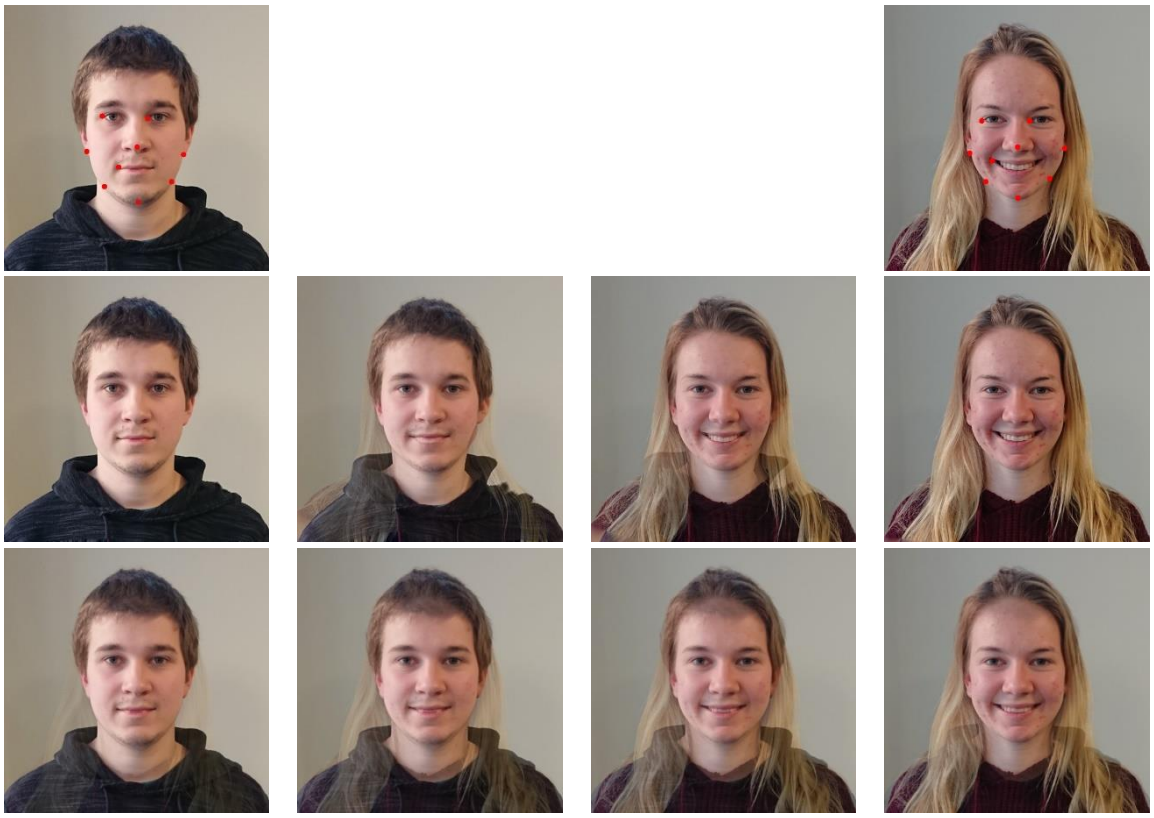

Figure 13. The two resource images inputs (first row) with manually selected 9 correspondences on the face used in Liao's method; Second row, the morphing sequence obtained using Liao's method; Third row, the intermediate morphing image with traditional feature-based morphing with D-lib. Both sequences have the same blending factor $\alpha = 0.2, 0.4, 0.6, 0.8$.

Another experiment was also conducted for comparing the performances of these 2 methods on rather huge orientation changes occurring on the same object. The result in Figure15 shows results from both methods suffered significant blending artifacts especially in the neck region when the ideal correspondence is discontinuous because of the occlusion. The Liao's method had slightly better results shown in the second row in the aspect of ghosting effects. It was expected to be a smooth head rotation process during

the morphing. The picture in Figure 14 also presents a failed morphing between a man and a cat.

Later, an improved fully automated image morphing example proposed by Liao [16]. The two original images shown in Figure 16 are warped into a morphing sequence based on sparse Neural Best-Buddies, which replace the user guided correspondences in the energy function of in the previous work (Equation 8).

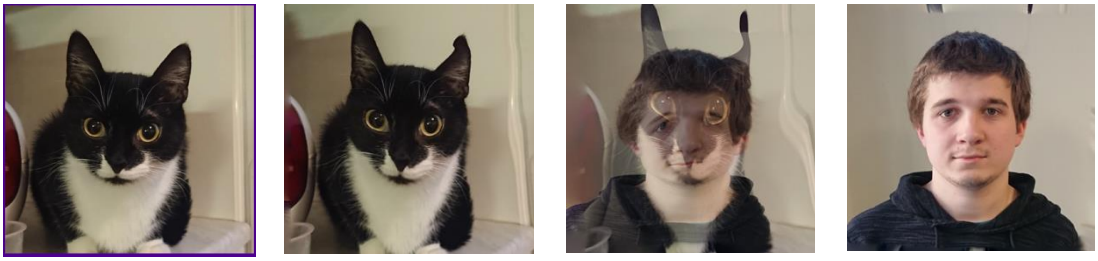

Figure 14. Translating a cat (left column) to a man. The morph sequence with α=0.25 0.5 0.75.

Figure 15. Morphing results when orientation changed causes large occlusion.

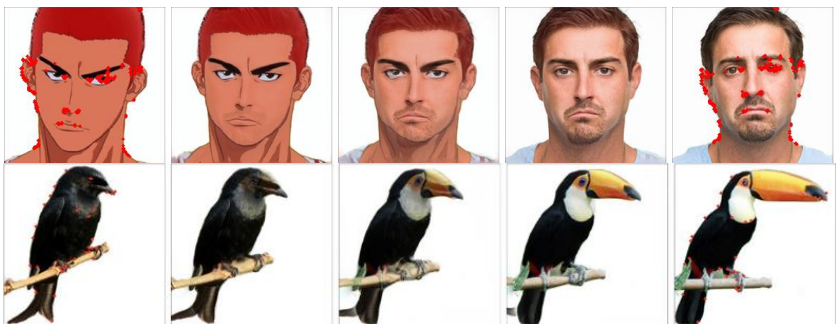

Figure 16. The improved fully automated method results with warped morphing sequence 0.25,0.5.0.75 [16].

Unfortunately, the regenerate morphing and unsupervised translation are not open source project. It is unlikely to test the model with own data. Here are the demo and examples presented by the original paper and project.

Figure 17 presents the results from the *Regenerative morphing*. Top row shows the cloud into a face. It is hard to even manually define the correspondences between two images, which is a requirement when using traditional methods. The result morphs in the middle transform smoothly from one to another. And the facial features merge into closed similar cloud pattern. The bottom row shows a boy rolling a basketball. Here it indicates the performance of this model when handling larger non-rigid motions between the two images. In this example, automatic SIFT and 17 manual correspondences are used to improve the result morphs. Note the non-rigid motion of the face and arm.

Figure 17. Morphing results from the regenerative morphing. Top row: a cloud to a face. Bottom row: a boy rolling a basketball [10].

The Figure 18 shows the results after performing a linear interpolation to geometry code and appearance code respectively from [15]. It is aiming to assess if the disentangled latent space is densely added. It is obvious that both the geometry and appearance of images can be translated smoothly from source to target image.

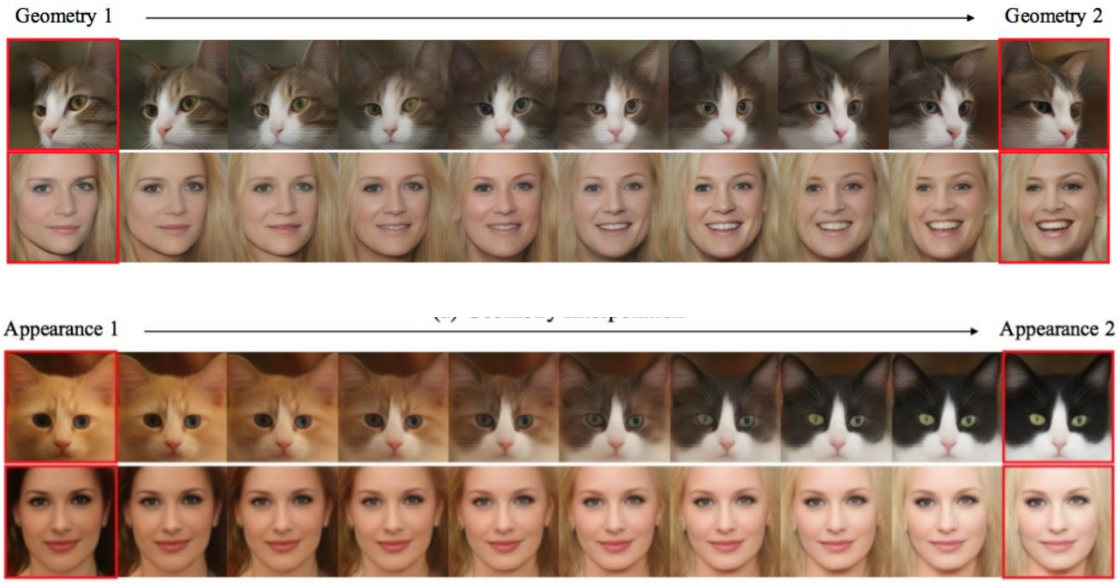

Figure 18. Linear interpolation results of geometry and appearance latent codes in unsupervised translation [15].

The Figure 19 below shows the results of performing the TransGaGa model on both near-rigid (faces) and non-rigid (animals) objects. The model performs robustly when handling large shape variations and unconstrained appearance in both rigid and non-rigid scenarios. It has a rather high capacity for translation between more complicated objects.

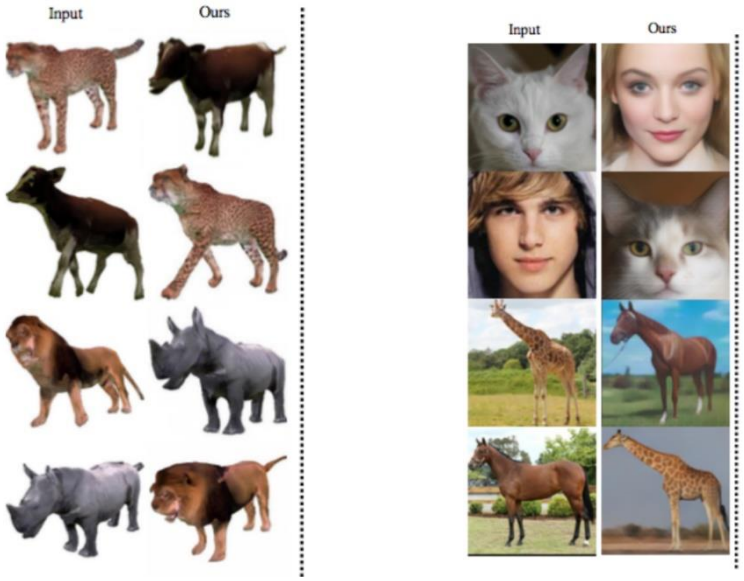

Figure 19. TransGaGa results on synthesis datasets (transfermation between cow and cheetah, lion and rhino) and real-world datasets (transfermation between cat human face, giraffe and horse) [15].

4. CONCLUSIONS

Image morphing is an essential and interesting technique especially for visualization related industry. In this work, we provided a comprehensive overview of various available literature in the field of morphing techniques and surveyed some of those which have significant contributions. Those methods are varied from traditional feature-based methods to the state of art learning-based methods. The algorithms in each method were discussed with detail, as well as the strengths and weakness.

We successfully implemented one of the traditional feature-based mesh morphing methods and provided a qualitative comparison of the results obtained from this method and from the structural similarity in halfway domain method using facial images. None of these two methods produced the plausible morphing result. Thus, we illustrated the example results from the published papers of other methods as well. This delivered a better understanding of those methods which do not have open sources and are difficult to implement and train the model.

The Feature-based triangular meshed morphing is easy to understand and implement. It is motivated by a feasible straightforward combination requiring triangular meshes as features and inverse interpolation to formulate the warp function. But this requires a good knowledge of the manual corresponding features. The process is not an efficient way to find a morph when we did not depend on the powerful modern toolboxes. The regenerative morphing replaced traditional warp and blend method with a regenerative approach by formulating an optimization task. It attempted to eliminate requirement of the manual correspondence and obtained the morph even when the objects in target and source image are very dissimilar. By applying bidirectional similarity of each frame to its neighbours and source images, this approach minimized the ghosting effects and created an interesting and artistic morph. Morphing using structural similarity on halfway domain method is another optimization-based method. Using a halfway parameterization for building correspondences ensured the model easily handling simple occlusions without making compromises in the optimization process. The model matched neighbourhoods with the same structure properly, even for those with different colour distributions.

The newest approach based on GAN system has further improved the quality of the mapping process. The Geometry-Aware Unsupervised Image-to-Image Translation used auto-encoder to separate the geometry and appearance information, then built the translation on each latent space. Thereby, an effective translation between objects with complex structures could be realized.

For the cases where the input images are sufficient similar in the structure, all the methods could produce a high-quality result automatically, or with a few user-guided points to improve the result. However, more complicated cases, where feature information has

huge occlusions and images are dissimilar objects, will still require a large amount of future research to produce a visually pleasing morph.

5. REFERENCES

[1] R. E. Barnhill, "Representation and Approximation of Surfaces," Mathematical Software, pp. 69-120, 1977. DOI: //doi.org/10.1016/B978-0-12-587260-7.50008-X.

[2] T. Beier, S. Neely and T. Beier, "Feature-based image metamorphosis," Comput. Graphics ACM, vol. 26, (2), pp. 35-42, 1992. DOI: 10.1145/142920.134003.

[3] P. Brigger, J. Hoeg and M. Unser, "B-spline snakes: a flexible tool for parametric contour detection," IEEE Trans. Image Process., vol. 9, (9), pp. 1484-1496, 2000.

[4] S. S. Cairns, "Deformations of Plane Rectilinear Complexes," The American Mathematical Monthly, vol. 51, (5), pp. 247-252, 1944.

[5] M. de Berg et al, Computational Geometry Algorithms and Applications. 2008. DOI: 10.1007/978-3-540-77974-2.

[6] M. S. Floater and C. Gotsman, "How to morph tilings injectively," J. Comput. Appl. Math., vol. 101, (1), pp. 117-129, 1999.

[7] J. Gomes et al, Warping & Morphing of Graphical Objects. 1999.

[8] D. E. King, "Dlib-ml: A machine learning toolkit," Journal of Machine Learning Research, vol. 10, pp. 1755-1758, 2009.

[9] J. Liao et al, "Automating Image Morphing Using Structural Similarity on a Halfway Domain," ACM Transactions on Graphics (TOG), vol. 33, (5), pp. 1-12, 2014.

[10] E. Shechtman et al, "Regenerative morphing," in 2010. DOI: 10.1109/CVPR.2010.5540159.

[11] V. Surazhsky and C. Gotsman, "Intrinsic morphing of compatible triangulations," Int. J. Shape Model., vol. 9, (2), pp. 191-201, 2003. . DOI: 10.1142/S0218654303000115.

[12] R. Szeliski, Computer Vision Algorithms and Applications. (1st ed.) 2011. DOI: 10.1007/978-1-84882-935-0.

[13] C. Thomassen, "Deformations of plane graphs," Journal of Combinatorial Theory, Series B, vol. 34, (3), pp. 244-257, 1983.

[14] G. Wolberg, "Image morphing: a survey," The Visual Computer, vol. 14, (8), pp. 360-372, 1998.

[15] W. Wu et al, "TransGaGa: Geometry-Aware Unsupervised Image-to-Image Translation," 2019.

[16] K. Aberman et al, "Neural best-buddies: sparse cross-domain correspondence," ACM Transactions on Graphics (TOG), vol. 37, (4), pp. 1-14, 2018. DOI: 10.1145/3197517.3201332.