View Interpolation

Tony Liu
College of Engineering
University of Miami
Miami, USA
yxl1837@miami.edu

***Abstract*—View interpolation, also known as view morphing, is useful in many practical applications, especially for Virtual Reality. In this report, several techniques related to view morphing are explored. The first part of the report deals with image morphing, which motivates the idea of view morphing and is included in projection-based view morphing technique. The second part of the report covers disparity-based view morphing technique, which only works for parallel view. View morphing for unparallel view is discussed in the third part of the report, projection-based view morphing.**

***Keywords*—View interpolation, image morphing, disparity, projection**

I. IMAGE MORPHING

Image morphing is a kind of special effect that changes or morphs one image into another image through transition. A simple example of image morphing is cross dissolving as shown in Fig 1.1.

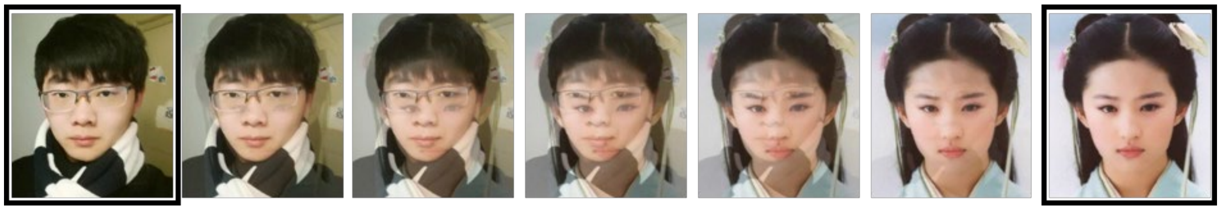


Fig 1.1: Cross dissolving effect. The leftmost image is the source image; the rightmost image is the destination image; the transition fades from source image to destination image

Cross dissolving can be achieved by linear interpolation written as:

$$I_{1.5} = w \cdot I_1 + (1-w) \cdot I_2 \ , w \in [0,1]$$

where I_1 is source image, $I_{1.5}$ is transition image, I_2 is destination image, and *w* stands for weight. When *w* is 1, the transition image is completely source image; when *w* is 0 it is completely destination image; when *w* is in-between it is the transition.

A simple method as cross dissolving will suffer from double-exposure effect. For example, when *w* is equal to 0.5, the transition image is ghost-like (as in the middle image of Fig 1.1) because the eyes of the two faces are not aligned during the transition. Thus, to make the transition more natural, the two faces need to be aligned during the transition.

Image warping is used for this purpose. Pixels are "moving" from frame to frame in the transition process. A mapping rule is used to determine the way in which the pixels in one image should be mapped to the pixels in the other image.

This report is using affine transformation for warping to make the transition more natural, which can be written as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where $[x, y, 1]^T$ is the homogeneous coordinate of a point in source image. $[x', y', 1]^T$ is the corresponding point in the transition image. The result of cross dissolving with affine warping is shown in Fig 1.2.

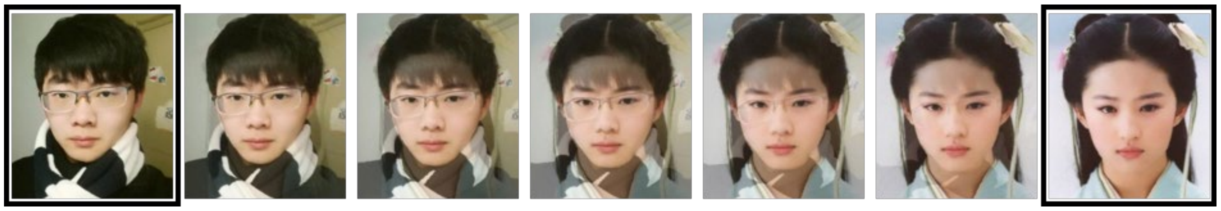


Fig 1.2: Cross dissolving with affine warping. The two faces are aligned during the transition so that transition images are not ghost-like anymore

It is obvious that the middle-way image of affine mapping rule is more natural than that of simple linear interpolation, and it is also referred as the "average face" of two faces.

The complete procedure for cross dissolving with affine warping is listed as below:

1. Feature matching. This is to specify which points are going to be aligned during the transition.
2. Apply Delaunay triangulation based on matched features and map corresponding triangles. Delaunay triangulation means for a given set of discrete points in a plane, connect the points to form the triangles such that no point is inside the circumcircle of any triangles.
3. Generate intermediate triangulation, $T_{1.5}$, from the Delaunay triangulation of the two faces by:

$$T_{1.5} = w \cdot T_1 + (1 - w) \cdot T_2 \ , w \in [0,1]$$

4. For each matched pair of triangles in T_1 and $T_{1.5}$:
   compute affine transformation: $\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} = P'P^{-1}$ (P and P' are 3-by-3 matrices made from the coordinates of the triangle vertices)
   Apply the affine transformation to every pixel in source image based on the triangle to which the pixels belong

5. Repeat step 4 for T_2 and $T_{1.5}$
6. Cross dissolve the two transition images with the same w as in step 3

Some intermediate results are shown in Fig 1.3

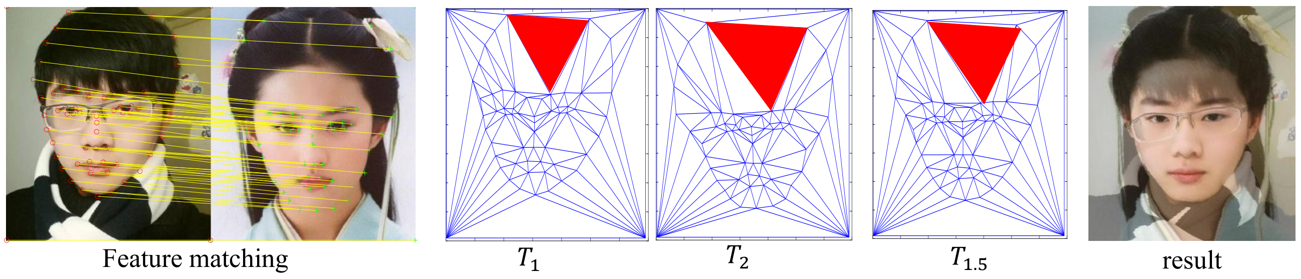


Fig 1.3 Procedure of cross dissolving with affine warping. The red triangles indicate a pair of matched triangles in T_1 and T_2 and their correspondence in $T_{1.5}$

The mapping rule is not unique. Instead of using triangles, one can also use rectangles to do the mapping, which is called mesh warping [1]. Some paper [2] is using so-called Multilevel free-form deformation (MFFD) to generate funny images.

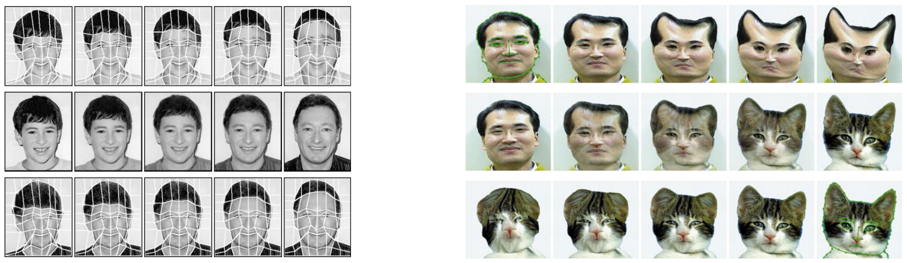


Fig 1.4. Mesh warping (left) and Multilevel free-form deformation (MFFD) based morphing (right)

There is no criterion to judge the result being good or not. The evaluation is very subjective

II. VIEW MORPHING WITH DISPARITY

View morphing is the process of creating a sequence of synthetic images that taken together, represent a smooth transition from one view of a scene to another view. The effect of view morphing with disparity is shown in Fig 2.1.

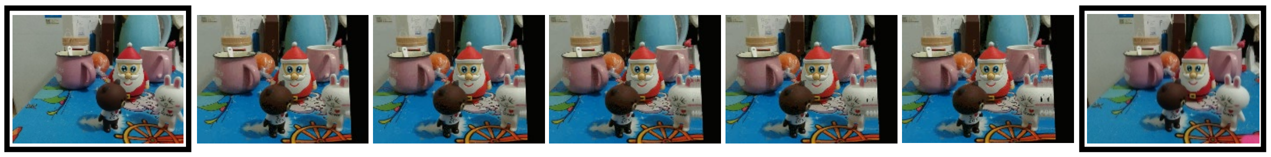


Fig 2.1 Effect of view morphing with disparity. The transition is generated from the rectified input images. That is why there is black margin appearing in transition images. The view point is changing from left to right and the shapes of the objects are preserved.

In this part, a pair of images and the ground truth disparity is used as shown in Fig 2.2.

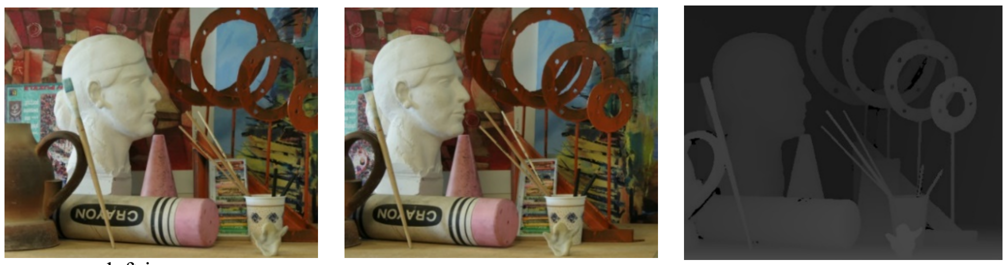


Fig 2.2 The two input images and the ground truth disparity map used in this section. Input images are already rectified. Ground truth disparity is obtained by using structed light

For any point in the left image, the disparity of the point is decreasing as the virtual camera is moving from left to right (as shown in Fig 2.3).

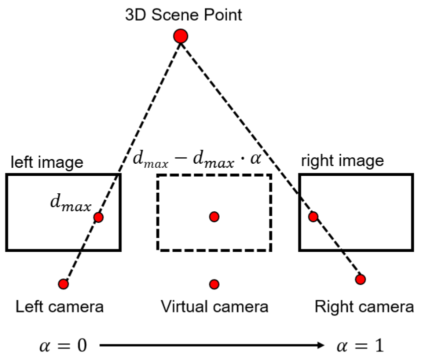


Fig 2.3 The disparity changes with the moving of the virtual camera. α is the relative position of the virtual camera w.r.t left image. For a fixed 3D scene point, the disparity of the projection of it on the left image is d_{max}. As the virtual camera is moving from left to right, the disparity of the same point is decreasing and becomes 0 when the virtual camera is overlapping with right camera.

Based on Fig 2.3, generating a virtual scene can be done by simply moving every point in the left image to the right at the distance of $d_{max} \cdot \alpha$ or every point in the right image to the left at the distance of $d_{max} \cdot (1 - \alpha)$. The result is shown in Fig 2.4.

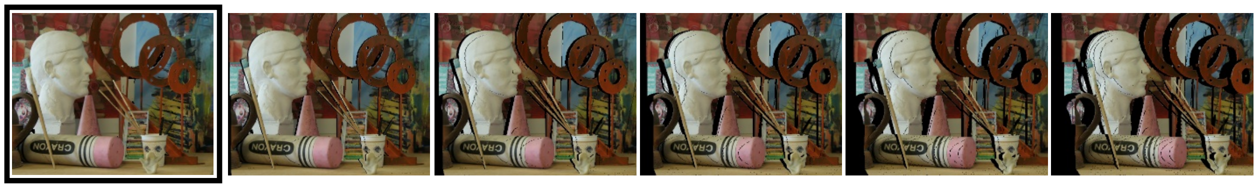


Fig 2.4 Result of view morphing with disparity. The result is generated by moving every point in right image to left

There is hole and crack in generated scenes. Hole is caused by occultation. Crack is caused by rounding error because $d_{max} \cdot \alpha$ may not always be integer.

Alternatively, one can use two images together to fill in the hole and crack. The result is shown in Fig 2.5.

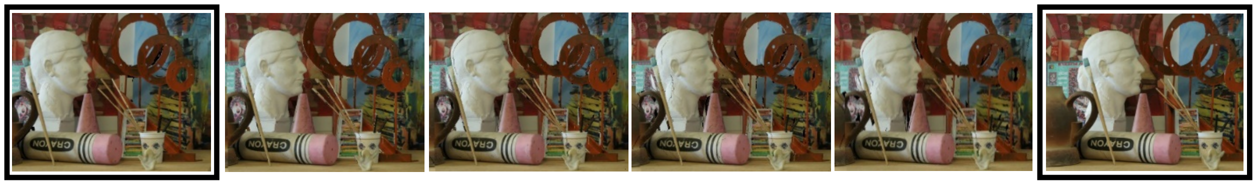


Fig 2.5 Using two input images together to generate virtual scene. Some holes and cracks disappear compared with Fig 2.4

There are two choices when assigning color to virtual scene: forward warping and inverse warping. Forward warping is to send each pixel p to its corresponding location p' while inverse warping is to get each pixel p' from its corresponding location p.

Forward warping may cause aliasing, crack, and holes due to the point landing between two pixels. Inverse warping is a better choice as we are mapping p’ back to p so there are no longer holes and it is easier to resample an image at non-integer locations.

A comparison of forward warping and inverse warping is in Fig 2.6.

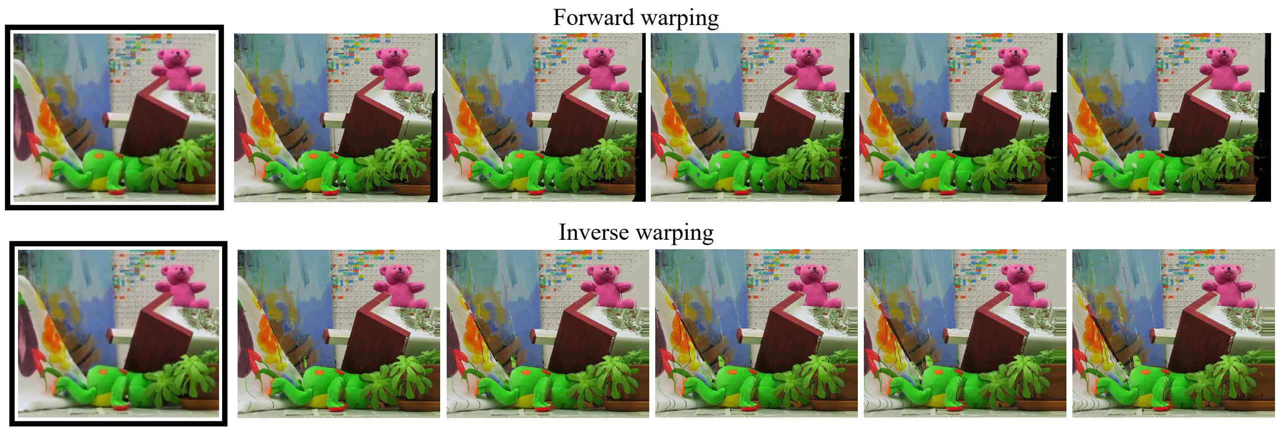


Fig 2.6 Comparison of forward warping and inverse warping. There are no holes in inverse warping result

However, there are still cracks in the result of inverse warping which are caused by the holes in disparity map. To fill in these holes. One of the solutions is to compute the integral image of the disparity map and then apply arithmetic mean filter. The result is shown in Fig 2.7. The view morphing result based on hole-filled disparity map is in Fig 2.8.

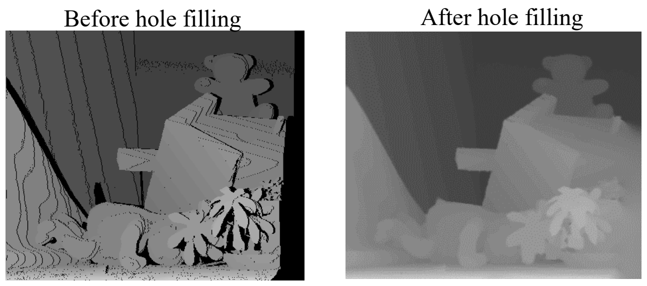


Fig 2.7 The disparity map before and after hole filling

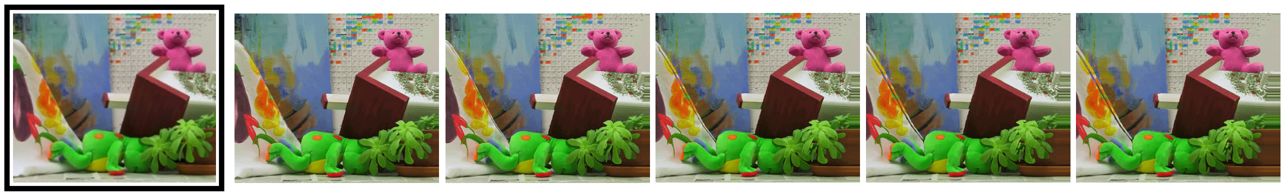


Fig 2.8 Inverse warping after disparity hole filling. Compared with Fig 2.6, cracks disappear in the virtual scene

However, generating virtual scene by disparity cannot solve the occultation problem. Because the information of the blocked object is totally lost. The problem can be solved with some state of art machine learning technique [3] which has good estimation of the blocked object. This report just briefly shows an example of view morphing result from machine learning as in Fig 2.9 without any further discussion

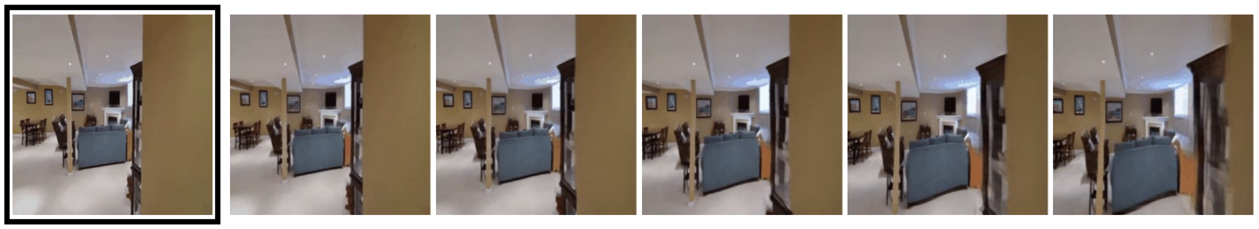


Fig 2.9 View morphing by machine learning, which requires only one single input image. The generated view has good estimation of the scene blocked by the wall

Another limitation of this method is that the left and right image are supposed to have reasonably large common region otherwise the disparity map cannot be generated. Besides, it only works for parallel view as in Fig 2.10.

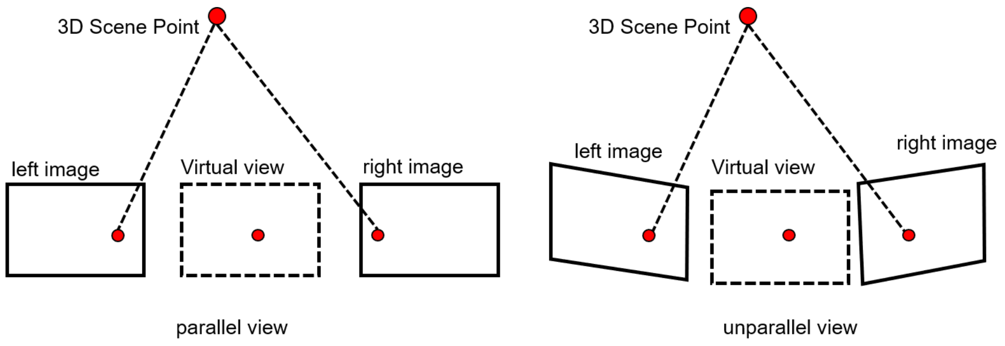


Fig 2.10 Parallel view and unparallel view

To perform view morphing for unparallel view, one may turn to projection-based view morphing technique, which is discussed in Part III.

III. PROJECTION-BASED VIEW MORPHING

As mentioned, image morphing motivates the idea of view morphing. For example, for input images as in Fig 3.1. The shape of the object in transition images is not preserved using the method discussed in Part I. One would expect natural transition shown on the bottom.

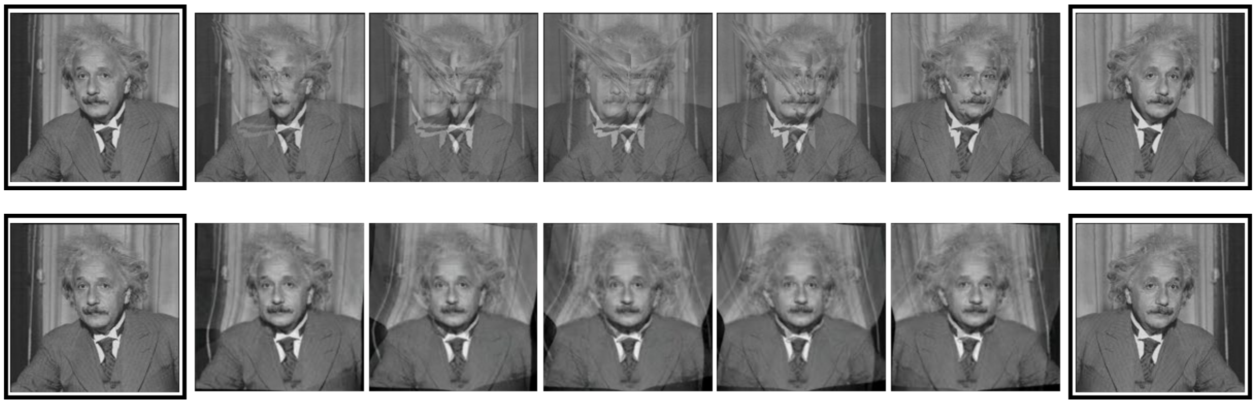


Fig 3.1. image morphing (top) and expectation (bottom)

The basic idea of this technique is to prewarp two images (usually unparallel view) prior to computing a morph and then postwarp the morphed image back to a desired plane. The purpose of prewarping is to convert the unparallel view to parallel view so that the image morphing technique introduced in Part I can be directly applied on the parallel view.

If the fundamental matrix of the two images has the form as:

$$\hat{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

then the two image planes are parallel. Based on this, one can choose two projective transforms $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ such that $\left(\boldsymbol{H}_2^{-1}\right)^T F \boldsymbol{H}_1^{-1} = \hat{F}$, where F is the original fundamental matrix of the two images. An 8-point algorithm [4] is used for finding F. To obtain $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$, the basic idea is to apply a rotation in depth to make the images planes parallel and then align the corresponding scanlines by an affine transformation. The detail is in [5]

The complete procedure for this shape-preserved view morphing is as follows:

1. **Prewarp**: apply projective transforms $\boldsymbol{H}_1^{-1}$ to I_1 and $\boldsymbol{H}_2^{-1}$ to I_2, producing prewarped images $\hat{I}_1$ and $\hat{I}_2$
2. **Morph**: form $\hat{I}_{1.5}$ by linearly interpolating positions and colors of corresponding points in $\hat{I}_1$ and $\hat{I}_2$, using the image morphing technique as in Part I
3. **Postwarp**: apply $\boldsymbol{H}_s$ to $\hat{I}_{1.5}$, yielding image $I_{1.5}$, where $\boldsymbol{H}_s$ is the estimated homography that maps $\hat{I}_{1.5}$ to $I_{1.5}$

The visualization of the algorithm is shown in Fig 3.2

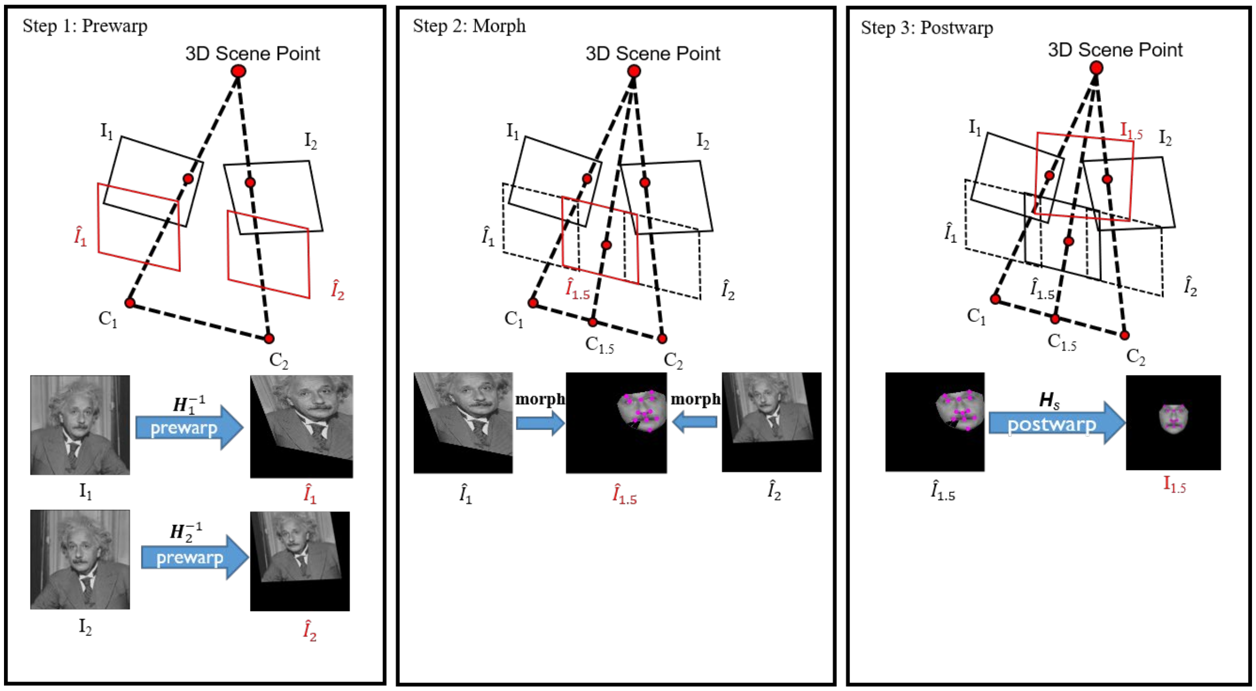
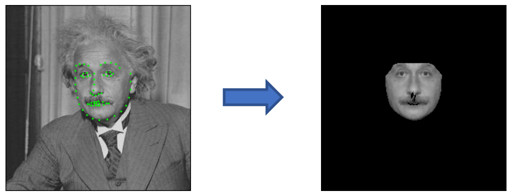


Fig 3.2 Visualization of the 3-step view morphing. In the first step, the two input images are prewarped by applying $\boldsymbol{H}_1^{-1}$ and $\boldsymbol{H}_2^{-1}$, producing $\hat{I}_1$ and $\hat{I}_2$. The image plane of $\hat{I}_1$ and $\hat{I}_2$ are now parallel. In the second step, the two images are morphed by the same method introduced in Part I. In the third step, the morphed face is postwarped to a desired plane by $\boldsymbol{H}_s$, which is the homography that map $\hat{I}_{1.5}$ to $I_{1.5}$ and can be obtained by the specified key points such as the magenta points in the figure

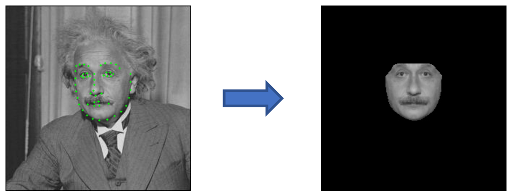
The 8-point algorithm for estimating F is easy to implement and camera information is not required. But the initial version of it is not robust to noise and inaccurate key points. However, if an extra normalization step is added into the algorithm the performance is said to be comparable with some of the best iterative algorithms.

There is a systematic method to obtain $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$, but not the case for $\boldsymbol{H}_s$, which is used for postwarping the morphed face, $\hat{I}_{1.5}$, to a desired plane, which is $I_{1.5}$. The author suggests finding $\boldsymbol{H}_s$ "interactively", which means "trial and error" or, in other words, "try different parameters until it yields satisfactory result". However, the plane of the final morphed image is unknown so that the process of finding $\boldsymbol{H}_s$ could be magic. However, the biggest disadvantage is that it makes it very difficult to automate the whole view morphing process since the final parameters are manually set.

In the image morphing step, if the corresponding triangles were too small or the points were not positioned in a way that allowed us to draw corresponding triangles, this would lead to information loss and holes in the morphed image as in Fig 3.3.



Morph with 68 points



Morph with 52 points

Fig 3.3 Morph the face with different number of features. There are many facial landmark around the mouth, some of them would give very small triangles that fail to map the other images and thus result in information loss. However, if we reduce the number of points around mouth, the problem is solved

Finally, this projection-based view morphing technique only works for statics scene, that is, the relative location of object remains unchanged when generating virtual view. Some paper that improve this method has the so-called dynamic view morphing [6] which generates a virtual scene with the object changing its location. This report just shows the result achieved by the technique as in Fig 3.4 without discussion.

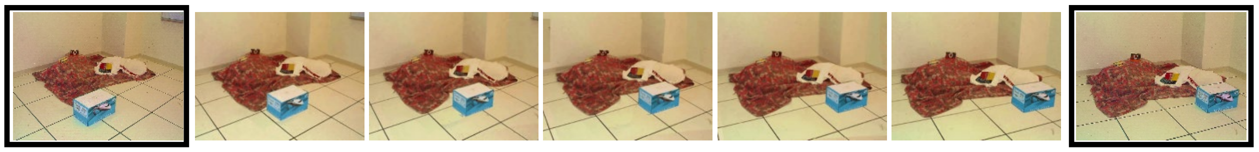


Fig 3.4 Dynamic view morphing

IV. Conclusion

In this report, I explored three kinds of techniques: image morphing, view morphing with disparity, and projection-based view morphing. Image morphing can create transitions from source image to destination image, but the shape of the object may not be preserved. View morphing with disparity can generate the transitions with the shape of the object being preserved but it only works for parallel view. Finally, projection-based view morphing incorporates image morphing technique and works for unparallel view.

REFERENCE

[1] Smythe, D.B., 1990. A two-pass mesh warping algorithm for object transformation and image interpolation. Rapport technique, 1030, p.31.

[2] Lee, S., Wolberg, G., Chwa, K.Y. and Shin, S.Y., 1996. Image metamorphosis with scattered feature constraints. IEEE transactions on visualization and computer graphics, 2(4), pp.337-354.

[3] Wiles, O., Gkioxari, G., Szeliski, R. and Johnson, J., 2019. SynSin: End-to-end View Synthesis from a Single Image. arXiv preprint arXiv:1912.08804.

[4] Hartley, R.I., 1997. In defense of the eight-point algorithm. IEEE Transactions on pattern analysis and machine intelligence, 19(6), pp.580-593.

[5] Seitz, S.M. and Dyer, C.R., 1996, August. View morphing. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (pp. 21-30).

[6] Manning, R.A. and Dyer, C.R., 1999, June. Interpolating view and scene motion by dynamic view morphing. In Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149) (Vol. 1, pp. 388-394). IEEE.