Rectangling Panoramic Images via Warping Using SVD

Chih Chiang Chang(張致强B05902064), Chi Ming Chung(鍾起鳴B05902026)

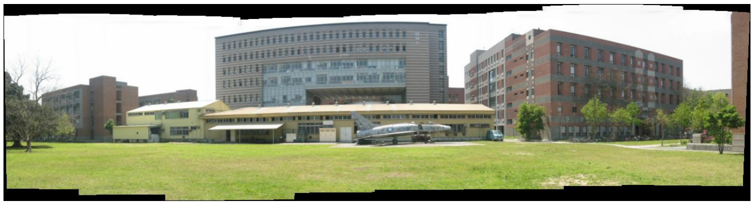


Figure 1: *from YY Chuang VFX sample*

1 Pipe Line

input image → local warping → unwarp → global warping

2 Local Warping

2.1 Find border of the image

One of the hardest parts in our implementation is to find the border of an image, since the input image is usually not supplied with a border map. Therefore, we design a way to find the border using breath-first search. First, we en-queue pixels which are on the boundary segments(4 sides of the target rectangle). For each pixel in the queue, we declare it a border pixel if its intensity is lower than a threshold, and put its 4-connected neighbors into the queue. After obtaining our border, we apply a 3x3 dilation and gaussian blur to smooth the edge between border and non-border region.

the border of Figure 1 (inverted):

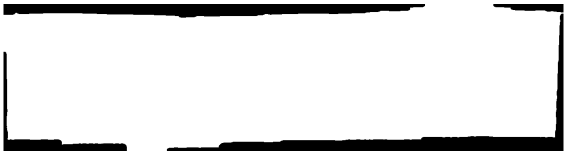


Figure 2

2.2 Seam carving

We follow the way in[AVIDAN, S., AND SHAMIR, A. 2007. Seam carving for content- aware image resizing]. As the paper[Kaiming He, Huiwen Changand Jian Sun, Rectangling Panoramic Images via Warping] imply, they define the "boundary segment" as a connected sequence of missing pixels on one of the four sides, which we find from the border map we constructed above. Each time select the longest boundary segment and insert one seam. If a seam is block by a pixel, we run the seam carving algorithm in a sub-image.

2.3 Unwarp

We record the displacement of each pixel during seam carving process for unwarping. A squared grid is created on the local-warped image and is unwarped back to the input image shape(the paranoma image). The distorted quad mesh obtained here is used in the next step, global warping. After global warping, we need a way to preserve features, namely, shapes and boundaries.

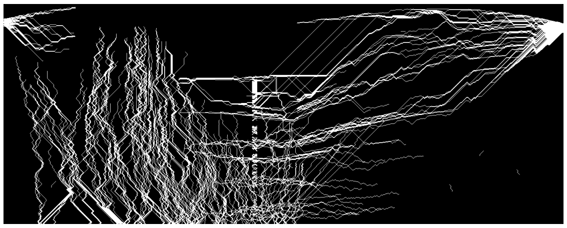


Figure 3: *visualize seams*

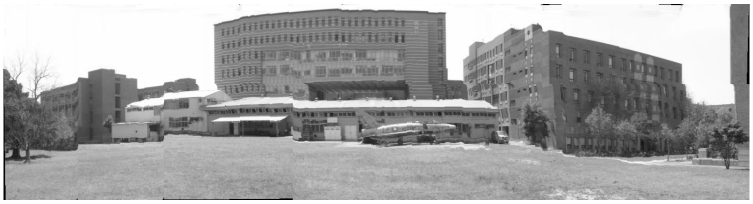


Figure 4

2.4 Energy function

We follow the way in [Kaiming He, Huiwen Changand Jian Sun, Rectangling Panoramic Images via Warping] to preserve energy.
Shape Preservation. [Zhang et al. 2009; Chang and Chuang 2012]

$E_s(V) = \frac{1}{N}\sum_q \parallel A_q(A_q^T A_q)^{-1}A_q^T - I)V_q \parallel^2$
N is the number of quads in the mesh, q is a quad index, I is a unit matrix, and A_q is a 8×4 matrix and V_q is a 8×1 vector on the quad:

$$A_q = \begin{bmatrix} \hat{x_0} & -\hat{y_0} & 1 & 0 \\ \hat{y_0} & \hat{x_0} & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x_3} & -\hat{y_3} & 1 & 0 \\ \hat{y_3} & \hat{x_3} & 0 & 1 \end{bmatrix}, V_q = \begin{bmatrix} x_0 \\ y_0 \\ \vdots \\ x_3 \\ y_3 \end{bmatrix}$$

Here we use $(\hat{x_0}, \hat{y_0}), ..., (\hat{x_3}, \hat{y_3})$ to denote the four pairs of coordinates of the input quad, and $(x_0, y_0), ..., (x_3, y_3)$ those of the output quad.
We denote number 0 as top-left vertex, 1 as top-right vertex, 2 as bottom-left vertex, 3 as bottom-right vertex.
Boundary Preservation. [Kaiming He, Huiwen Changand Jian Sun, Rectangling Panoramic Images via Warping]

$$E_B(V) = \sum_{v_i \in L} x_i^2 + \sum_{v_i \in R} (x_i - w)^2 + \sum_{v_i \in T} y_i^2 + \sum_{v_i \in B} (y_i - h)^2$$

2.5 Solve the energy function

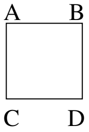
According to above equations, a quad can be represent by 4 vertices. Therefore we can solve the equations with SVD, since the energy functions are all mean square error problems.
We show our matrix below:
denote T as vertices number, N as quad numbers.

$$A = \begin{bmatrix} \underset{8\times T}{A_0} \\ \vdots \\ \underset{8\times T}{A_N} \\ \underset{Nums(v_i \in L)\times T}{v_i \in L} \\ \underset{Nums(v_i \in R)\times T}{v_i \in R} \\ \underset{Nums(v_i \in T)\times T}{v_i \in T} \\ \underset{Nums(v_i \in B)\times T}{v_i \in B} \end{bmatrix}, x = \begin{bmatrix} x_0 \\ y_0 \\ x_1 \\ y_1 \\ \vdots \\ x_T \\ y_T \end{bmatrix}, b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 10^8 \\ \vdots \\ 10^8 \end{bmatrix}$$

(According to paper, we use 10^8 approximately to infinity)

2.6 Interpolation

After finishing the global warping, we only have the correct warping displacement of the mesh vertices. We need to warp all the pixels in the panoramic image according to the displacement of the mesh. Therefore, bilinear interpolation is used on each pixel interpolated from the 4 vertices. Our quad denotes ABCD as vertices:



$$\begin{bmatrix} x_A & x_B & x_C & x_D \\ y_A & y_B & y_C & y_D \\ 10^8 & 10^8 & 10^8 & 10^8 \\ \lambda & 0.0 & 0.0 & 0.0 \\ 0.0 & \lambda & 0.0 & 0.0 \\ 0.0 & 0.0 & \lambda & 0.0 \\ 0.0 & 0.0 & 0.0 & \lambda \end{bmatrix} \begin{bmatrix} interp_i \\ \vdots \end{bmatrix} = \begin{bmatrix} x_{pixel} \\ y_{pixel} \\ 10^8 \\ \lambda \\ \lambda \\ \lambda \\ \lambda \end{bmatrix}$$

The third row of the leftmost matrix constraints the variables summing to 1.
the bottom rows are solving: min $\lambda \parallel interp_i - 1 \parallel$
to make sure the $interp_i$ are close to 1.
(We set λ = 0.2 here)

3 Improvement

According to the paper we read, there is an addition energy function which is line preservation that we did not apply yet.
The function we use as interpolation may have some logical flaws. If the missing pixels are getting through the image and cause a hole inside it, we did not do well on it.
We assume the quads are all convex, therefore in some extreme cases we might fail.

Figure 5

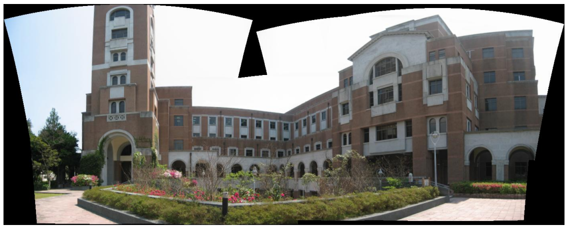


Figure 6: *NTU library for input paranoma, from YY Chuang VFX sample*

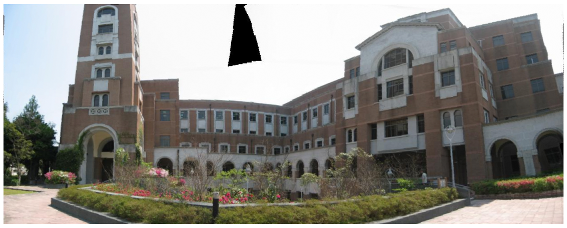


Figure 7: *the result of NTU library*

4 Artifact

5 Citations and References

5.1 Citations

Conrad Sanderson and Ryan Curtin. Armadillo: a template-based C++ library for linear algebra. Journal of Open Source Software, Vol. 1, pp. 26, 2016.

Conrad Sanderson and Ryan Curtin. A User-Friendly Hybrid Sparse Matrix Class in C++. Lecture Notes in Computer Science (LNCS), Vol. 10931, pp. 422-430, 2018.

5.2 Third-Party Material

Armadillo C++ library for linear algebra and scientific computing
OpenCV Library

5.3 References

AVIDAN, S., AND SHAMIR, A. 2007. Seam carving for content-aware image resizing. In SIGGRAPH 2007.

KAIMING HE, HUIWEN CHANG, AND JIAN SUN. Rectangling Panoramic Images via Warping. In SIGGRAPH 2013

6 Test Program

The code of our implementation can be found here: https://github.com/George0828Zhang/Rectangling-Warp