Pasting on Deformable Surfaces with Visible Ink Using Dense SIFT Flow

Andrew Liu

***Abstract*—Pasting onto a deformable surfaces requires either a non-parametric warp or a high d.o.f global warp approximation. Previous works utilized techniques such as moving least squares or hierarchical big data to define a non-parametric mapping from an image to a known template. In this paper, I use Dense SIFT Flow to generate a dense pixel mapping which is used to warp regions from a mesh template to an input image containing a deformed mesh template. One of the latest works used infrared ink and a 1000 FPS camera to capture and isolate videos. I decided to recreate similar results using slower hardware and colored mesh grid.**

***Index Terms*—Deformable Surface, SIFT Flow, Image Warping**

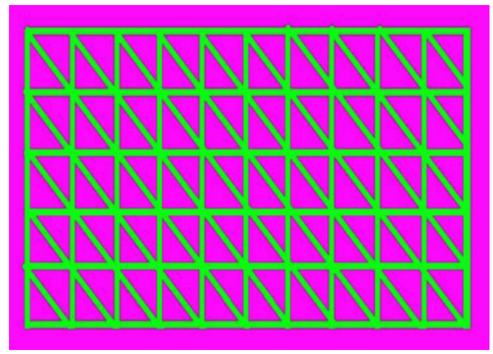

Fig. 1. Mesh grid color pattern used

I. Introduction

In this paper, we explore pasting an image onto a deformable surface such as paper or clothing. Practical applications of this include augmented reality and detection technology. In particular, we can take an image and deform it such that it fits onto the surface, matching the deformation as closely as possible. In theory, it's possible to do the reverse too - given an image and mesh grid on a deformed surface, reconstruct the image as if projecting onto a flat surface.

Previous work on deformable surface include a high speed 1000 FPS camera and infrared ink. A high speed camera can capture the changes in the mesh grid without worrying about motion blurring. Infrared ink can be detected with high fidelity using an infrared scanner. Although these equipments allow for high quality results, there aren't many high speed cameras available for use. In order to mimic the real world setting, I used an iPhone 5S as our capturing device. While iPhones can detect some levels of infrared readings, I decided to use visible light instead. This poses extra challenges such as determining bounding boxes and isolating the mesh grid.

In order to solve this we break the problem into two distinct parts:

1) Isolate and align the mesh-grid and transplant image.
2) Recover structural information about the surface using a known mesh grid. Perform a warp from a flat surface plane to the deformed surface.

Above describes my general approach for pasting onto a deformed surface. In this paper I discuss a single pair of solutions for each part which I found works well with the given constraints.

A. Previous Work

There are many approaches to deformable surfaces that I looked at. The original inspiration for this paper was a video produced by Ishikawa Watanabe Laboratory which demonstrated high speed projection on deformable surface. The algorithm they use to detect deformations wasn't available publicly (as of 12/8/2016).

Another work I looked at was Schaefer, McPhail, and Warren's Image Deformation Using Moving Least Squares. I chose not to use their algorithm because I would need to identify points of correspondences with high accuracy. With deforming and repeating mesh grid patterns, there isn't a good way to define points of correspondences without making assumptions about the structure and quality of the image. Furthermore, with occlusion the algorithm might fail to find a good set of correspondences that can warp the image well.

The last work I looked at was Tian and Narasimhan Hierarchical Data-driven Descent for Efficient Optimal Deformation Estimation. While data-driven methods are efficient and potentially provide the globally optimum solution, I don't have access to large data set nor the time to generate training pairs for the deformations.

II. Mesh Detection and Alignment

Figure 1 is the pattern used to recover the deformed surface. It consists of max-valued R and B channel background with a truss structure of max G channel pixels. I chose to have green represent the mesh grid because highly saturated greens are rare in indoor environments where I was testing this.

I choose a red blue background in order to increase the contrast with the mesh grid. This allowed me to be more aggressive with the set of energy functions I could test since the mesh grid is black in the red-blue spectrum while the background is black in the green channel.

When taking a picture of the deformed pattern, any whites in the image will also register as high green pixel value. Therefore I decided to use an energy function which exponentially penalizes pixels with high red and blue values. I created threshold values to isolate and binarize the mesh grid.

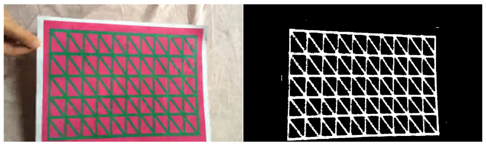

Fig. 2. Left: Pattern paper Right: Extracted mesh pattern

Next we need to align the template and the grid. Aligning the grid and template makes finding the grid's surface deformation easier. In order to align, I computed the mean x and y-value and align the grid and template. Finally I scale the template until the percentiles of both meshes align too.

One shortcoming is that the alignment is not rotationally invariant. I tried to introduce rotationally invariant feature descriptor patches before trying the alignment, unfortunately given the nature of deformable surfaces the gradient profile might look very different for similar deformations.

Another issue I encountered was the reflective property of the printer ink. Under certain deformations, certain regions would reflect the sunlight into the camera, resulting in over-exposed pixels. In addition, given the slower shutter speed of an iPhone, the mesh grid may be blurred if the deformation is too quick. This made recovering the mesh practically impossible.

III. SIFT Flow Mapping

Given the binarized deformed mesh grid and the template, we need to find a dense warp to map pixels from the template to the deformed template. I decided to use Liu, Yuen, and Torralba's SIFT Flow paper to map pixels. In their paper, they propose a minimization problem which penalizes flow-fields depending on similarity and smoothness constraints before using gradient descent to find a flow-field. The algorithm they provide has few different parameters which tuned.

Since the deformation is smooth (in this case we use a piece of paper which can't fold onto itself), we use a high smoothness parameter so that regions of pixels are preserved across the mapping. One last constraint is the distance constraint which penalizes flow-fields which have high magnitudes. Because we have already done alignment of the template and mesh grid, most of the pixel mappings should be close.

After we find the flow field, we warp our transplant image and a binary mask into the deformable surface shape. Then we combine the warped transplant image and the original grid image based on our mask.

Noticeable about this process, we did not define any points of correspondences. In the future, I will look at designing a different pattern such that points of correspondences can be identified and used. Given the repeated mesh structure of the template, it's impossible to distinguish between intersections unless we assume relative distance and spacing. Unfortunately relative spacing may not be preserved on a deformed surface.

IV. Results

More extensive results and animations can be seen online https://inst.eecs.berkeley.edu/ cs194-26/fa16/upload/files/projFinalGrad/cs194-26-abr/).

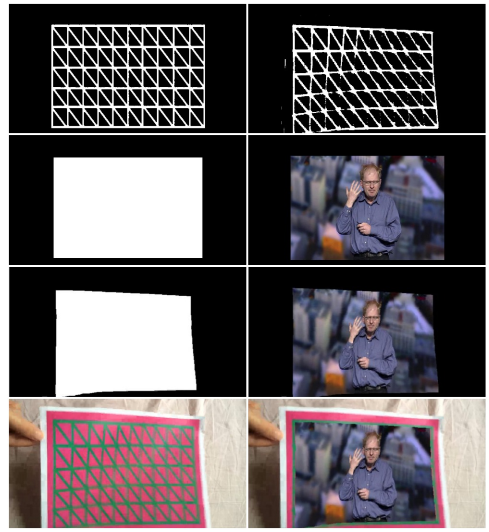

Fig. 3. Workflow for the program. (a). is the original template grid binarized. (b). is the deformed surface pattern (c). is the mask which is a bounding box around the template mesh grid. (d). is the transplant image. (e). is the warped mask. (f). is the transplant image warped. (g). the original deformed surface. (h). the combined deformed surface and transplant image.

The mesh isolation technique sometimes fails to capture enough information about the mesh structure. This is partly due to the equipment restrictions I placed on the project. When recording a moving paper, I found that deforming the paper too quickly results in motion blur. Adding bad lighting, it becomes even harder to differentiate the visible light spectrum. Both of these are solved when using a high speed camera and infrared ink.

Finally the SIFT Flow had some consistency and smoothness issues in Figure 4. This bug is localized and doesn't disrupt the overall mapping that badly. The artifact is due to the conflicting smoothness and consistency parameters required. Make things too smooth and the far corners which are the most deformed may not properly warp. Make things too consistent and the interior of the image might render with visible seams.

Figure 5 shows that it's possible to detect this mesh grid from a distance and still properly paste onto it. I used my laptop display as the mesh grid because without a light source, illumination can be a bit tricky.

Although the cursory work looks to be promising, the run time is still too high to be rendered live like in Ishikawa Watanabe Laboratory's Demo video. For a 37 frame video, it took 5 minutes to render the deformations.

V. Conclusion

Although the algorithm is still highly unoptimized, the proof of concept approach shows promising results. In terms of mesh grids, infrared ink is the best way to go because visible light is too dependent on lighting conditions.

For slowly deforming surfaces, having a slow shutter speed is acceptable. However quick deformations result in blurred motion on an iPhone which makes mesh recovery almost impossible.

While the truss structure does a good job of modeling the deformation surfaces, we could design a different pattern

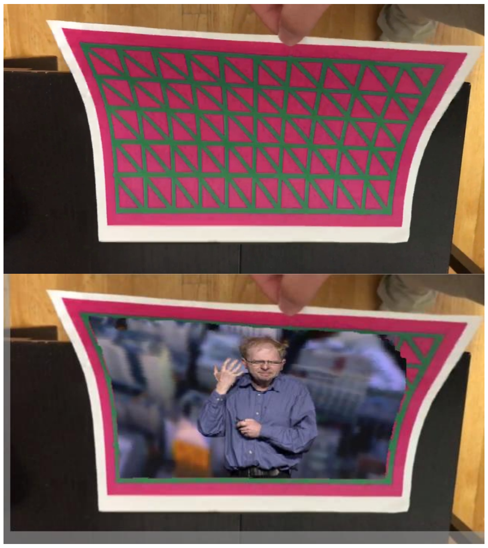

Fig. 4. Extreme deformation causes issues with SIFT Flow parametrization.

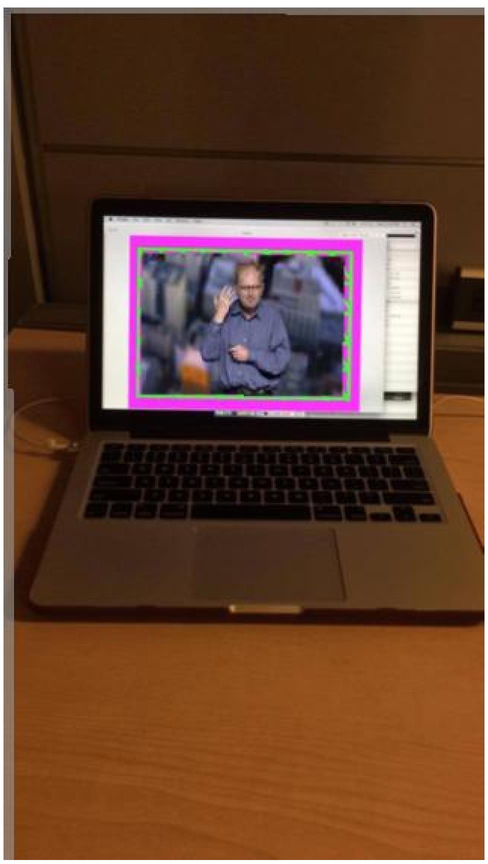

Fig. 5. Detecting mesh grid at distance.

which has the similar truss structure but also introduce distinguishable intersections for points of correspondences.

If we were able to accurately identify pairs of correspondences, then there are other approaches we could try such as the Moving Least Squares warp or gradient descent for non-parametric warp. Since we can define a loss function as the sum of pixel differences between the original grid and template, we can potentially find a local parametric warp for each region iteratively until we reach a local minimum.

In addition, with points of correspondences, we can introduce rotationally invariant patch descriptors and take the average angle to get the orientation of the entire page. Then we correct it and perform the alignment from the previous section.

Future steps:

1) Reverse template imaging. If we have an unknown image painted onto an unknown deformed surface, we can use a projector to project a visible mesh grid template onto the surface, use the above algorithm to reverse the deformed surface and recover the image.
2) Handle partial occlusions. Since we have a color background, we can extract the red-blue ratio instead and create a mask where the occluded parts of the mesh grid are black. This may have unknown consequences for SIFTFlow.
3) Design a mesh grid so that points of correspondences can be recovered. Used for rotationally invariance and other warping algorithms.

VI. References

1) *Dynamic projection mapping onto deforming non-rigid surface using a high-speed projector* by Ishikawa Watanabe Laboratory
2) *Image Deformation Using Moving Least Squares* by Schaefer, McPhail, Warren
3) *Hierarchical Data-driven Descent for Efficient Optimal Deformation Estimation* by Tian, Narasimhan