3D Reconstruction of Sunglass Prototypes from 2D views

MM804 Course Project

Gauravi Chavan
dept. of Computer Science, Multimedia
University of Alberta
Edmonton, Canada
gauravi@ualberta.ca

Kumar Halder
dept. of Computer Science, Multimedia
University of Alberta
Edmonton, Canada
khalder@ualberta.ca

Ava Sehat Niaki
dept. of Computer Science, Multimedia
University of Alberta
Edmonton, Canada
asehatni@ualberta.ca

***Abstract*—There are several instances of web consumers not being satisfied with the sunglasses delivered to them owing to lack of trust or incompetence of the available websites to meet their demands. Also, as the online consumers contribute to a large section of the profit margin of such sunglasses companies; they wish to optimize their economic viability by introducing a coherent model that ensures customer satisfaction.**

Through this project, we construct a 3D custom object by extruding points on the surface of each of its two 2D faces (1 front view and 1 side view image of sunglasses respectively. It should be noted here that a single side view image is sufficient to be used at both ends of the main frame). While doing so we apply image processing techniques to get an image that meets our desired requirements in the next step. We use math and pixel information for fitting the handles.

Keywords— 3D Reconstruction, 3D Mesh Generation, Texturing

I. Motivation

From a business perspective, online e-commerce shopping websites would certainly boost if the customers are able to visualize how the pair of glasses look on them. An amalgamation of Augmented Reality and 3D Visualization to create a 3D custom object (sunglasses in our case) for integrating it in an e-commerce website would prove to be a remarkable contribution to both sectors alike. Moreover, achieving success in this use case would allow other e-commerce websites to consider this idea and reap benefit out of it. This made us inquisitive and motivated to work on this research area.

II. Introduction

Given the problem of this calibre that has relevant and relatable issues, it needs attention from researchers working in the Graphics, Animation and Visualization sector. Moreover, our use case promises a considerable contribution in the field of Visualization and Graphics domain.

To cater to the business requirements, our project work is a part of a 3 Stage Industrial Project which comprises of:

[1.] Constructing 3D models of sunglasses

[2.] Using Augmented Reality for mapping these glasses on a person's detected face

[3.] Creating a plugin embedded in the website to encapsulate the former two stages.

In this cross project, our scope is confined to the first stage only. Our objective lies in providing such prototypes of sunglasses to the users, which would make it convenient for them to select the glasses that fit their requirements.

This report is divided into twelve sections. The first three sections give a brief overview of our research work. In Section IV we briefly discuss the literature papers that we referred to and finally chalked out our own plan for our self created algorithm. In Section V, we indicate the resources that we used for developing and implementing our algorithm. In Section VI, we specify how we implement the code whereas we demonstrate the results that we have obtained from the above implementation in Section VII. In Section VIII, we discuss what we deliver through this cross project. In Section IX, we specify the challenges that we faced while we were researching in this area and implementing the code. In Section X, I have discussed how I have I contributed towards our course project and in the following Section XI we discuss the tasks which are yet to be accomplished. Finally, we indicate the literature papers that we referred in the 'References' section.

III. Problem Definition

To create 3D sunglasses that are AR ready by extruding points on the surface of each of its 2 2D faces/images.

IV. Literature Review

There has already been a lot of researches in the past that generate mesh based on the image data. Nanyang Wang et al. made a neural network-based model that can make 3D mesh just from a single RGB image [1]. There exists some similar research works that construct 3D mesh using single images [2-5]. The result is quite acceptable considering 3D mesh generation from a single image, but they are not suitable for our cause, which requires a precision high enough to be admissible by human eye. Photogrammetry is very good at getting such kinds of result [6-10], but one major drawback of photogrammetry is that for a good quality mesh, one needs to take around 40-50 images, which is quite a tedious job. They also require a lot of cleansing after generation of the required mesh. So, for commercial use in order to generate sunglasses from images, it is best to generate 3D mesh of sunglasses that require minimum user effort. In this paper, we introduce such an algorithm that takes minimum user effort with just two images of side view and front view of the sunglass and produces the 3D mesh that contains enough information to represent the resultant sunglass.

V. Development Environment

We used vtk 8.1.2, OpenCV 3.4.3.18, SciPy 1.2.1, numPy 1.15.3, libraries for creating this algorithm.

VI. Implementation Details

We first use GrabCut algorithm to eliminate the unwanted portion (the vertical frame portion of the main frame) of the side view of the sunglass. The GrabCut algorithm is interactive algorithm for Image Segmentation.

Here, we describe a rectangle around the portion we wish to eliminate, our implementation will replace these pixels with white color pixels so that we get a perfect required view of the handle.

We describe our implementation through the pseudo code as follows:

```
### Code for removing frame portion from the side view image such that only handle portion is retained; it can be noted that (255,255,255) denote the white pixels and '-1' denotes the borders of the rectangle selection; the modification is saved to image say img2
        On pressing key 'x' on the keyboard:
                cv.rectangle(img2, rect, (255,255,255), -1)
### Code for saving the modified image after cropping
        On pressing key 'w' on the keyboard:
                cv.imwrite('cropped_sideview.png', img2)
```

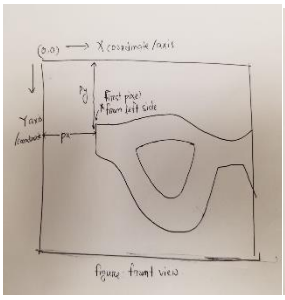
After this we proceed toward the actual implementation.

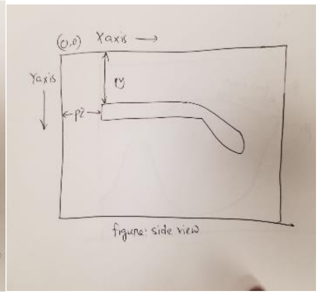
1. Creating a binary image of the front and side_view
2. Closing the gaps in the foreground and retaining only the hollow portion
3. Median filtering
4. Calculate the Laplacian
5. Calculate the sharpened image
6. Saturate the pixels in either direction
7. Apply this sharpened value on the front view

After we get a preprocessed clean thresholded images for side view and front view, we get the pixel position information of each pixel of the thresholded image, and store that in a list. Since the position values in the list is random, since it was obtained by iteration from the range of 0 to image length and image height. It is notable that even the size of the image is not square shaped, the output number x coordinate and y-coordinate will be same because there are the same number of pixels on both sides. After that, we sort the list of positions, which gives us the distance of the first pixel from left screen or Y axis as "px" and distance from top screen or X axis as "py", which represents it's x and y coordinate. This gives us a great flexibility of creating mesh, cause no wonder where the sunglass is in the image, we can always get the necessary information, and adjust their position by using this value of "px" and "py".

For the side view, we repeat the same procedure described in the front view. But this time we get two separate information, distance of the X axis of the image from the very first pixel from left side, which is actually "pz" instead of "px", and distance of the "py" from X axis. The "py" information of both the front view and side view of handle is close, if we assign the y position of handle as py, the placement is close enough in y direction. In practise, we have calculated the mean value for the pixels available alongside the horizontal line along the very first pixel, giving an approximate position of y value, and introduces some error in practise.

It is notable that there is variance of pixels in X direction of the side view image. We can take this constant value as the x coordinate of first pixel from left side of front view image. For precision in positioning, this will be a little shifted according to the value of depth of the handle. Since for this research we have manually provided the depth of the handle, this precision value is half of the depth value. After adjusting the x and z coordinate for assembling, we get the deviation of the first pixel of side view and shift the value in z direction accordingly. Thus, we connect the first handle with the left side of front view. For the joint of right side, we do the same procedure, but in an inverse way. We get the first pixel from the right side of the screen, which is actually the last pixel in X direction. The x-coordinate and y-coordinate in this case is same as the first handle. Thus, we get the assembly information of the 3 parts from the two views of sunglasses.





For creating 3D meshes from these points, it must be noted that the axis definition of 3D space and 2D space is not same. For example, in 2D image, Y axis goes downwards, while in 3D space in vtk, the Y axis is upward. So, a little adjustment of this is required to be corrected. After the correction, we plot this point clouds in 3D space. The 3D mesh can be generated in three ways, using delaunay 3D, using glyph, or creating polygons from the images and then extruding them. However, delaunay 3D does not work with 2D point dataset. We choose glyph for creating the volume of this points. We used cube source as the input of glyph, and provided the depth manually, since it is not possible to get the depth from only the top view and side view of the image. After that included the three parts in an assembly, so that we can manipulate them in the future in different software as one single object. For saving the 3D mesh, we used .obj file format, which is supported most of the external 3D software for later use.

VII. Results Evaluation

We have the following image for the side view which had a section of the main frame too. We eliminated this with the code written in

'cropping_segmenting_side_view.py' .

Fig 2: Cropped Side View

Now we demonstrate how we extrude the pixels from 2D faces of the input images below and render a 3D sunglass out of these images. We have created an automated system for creating a 3D sunglass prototype for any provided input images. We have demonstrated the input 2D images as under for sunglass model 1:

Fig 3: Input images of Front and Side View

We get the following resultant 3D model out of the two input images above:

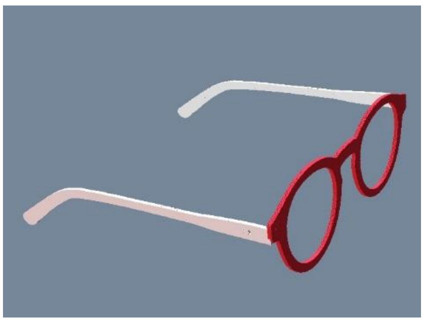


Fig 4: Output of 3D reconstruction of sunglasses of the above two images

Input images for Model 2 are as follows: (This one has an opaque coloured glass unlike the first model)

Fig 5: Input images of front and side view of the sunglass model 2

We get the following resultant 3D model out of the two input images above:

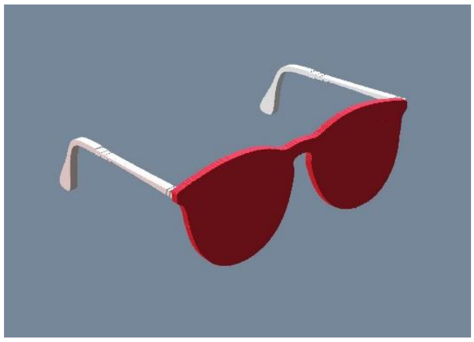


Fig 6: Output of 3D reconstruction of above two input images

VIII. Deliverables

An automated system that generates a 3D prototype of any sunglass on being provided with its side view and the front view.

IX. Challenges Faced

- There are not as many open source packages or modules for 3D mesh generation.
- One of the promising packages that we came across - pygalmesh needed linux dependencies. Therefore, we couldn't use it.
- Another package 'delunay' did not create an accurate 3D mesh, therefore, we chose not to go ahead with it.
- The math needed for automatizing the process of fixing the handles to the ends of the main frame was tricky.

X. Individual Tasks

I worked on the pre-processing part wherein it included some image processing functionalities for obtaining a perfectly edged and sharpened version of our input images. I worked on applying materials to the handles and main frame of the sunglass. I am in the process of trying to apply a texture on any prototype of sunglasses automatically. However, at the time of submitting the code we have not included it as it is difficult and needs more research.

I would like to acknowledge and be thankful to my team mates 'Ava Niaki' and 'Kumar Halder's contribution towards this course project.

XI. Future Scope

- Automating the process of integrating a transparent glass in any sunglass prototype according to its custom shape. It's challenging to deal with transparent glasses as is is difficult to separate foreground from the background owing to its intrinsic transparency. Moreover, performing background subtraction needed for a system that would automate for all prototypes of sunglasses is a tricky task. Although we tried working on this, we would need some more time to do further research.
- Automating the process of integrating a custom-made texture on the respective chosen sunglass prototype would also need some further research as it is a broad phenomenon. It should be noted that we are creating a system for mapping texture automatically for the chosen sunglass and not for a single prototype which is difficult.
- Using top view, we would augment the handles to the main frame so that it would create the exact curved representation of the handles in the 3D workspace.
- Researchers in future can consider generating a 3D prototype for asymmetric models.
- Researches in future may consider creating a model dictionary to customize the existing prototype to fit the user's preferences.
- This research idea can be utilized for including other online products like wrapping garments/clothes on a detected human body.

References

[1] "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", Nanyang Wang, Yinda Zhang , Zhuwen Li, Yanwei Fu, Wei Liu, Yu-Gang Jiang, arXiv 2018

[2] Bernardini, F., Mittleman, J., Rushmeier, H.E., Silva, C.T., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. IEEE Trans. Vis. Comput. Graph. 5(4), 349–359 (1999)

[3] Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: ECCV (2016)

[4] Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: SIGGRAPH (1987)

[5] Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: CVPR (2018)

[6] Chandler, JH, Fryer, JG, Jack, A (2005) Metric capabilities of low-cost digital cameras for close range surface measurement. The Photogrammetric Record 20(109): 12–26.

[7] James, MR, Robson, S (2012) Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application. Journal of Geophysical Research: Earth Surface 117(3): 1–17.

[8] Jester, W, Klik, A (2005) Soil surface roughness measurement - methods, applicability, and surface representation. CATENA 64(2): 174–192.

[9] Rose, JC, Paulus, S, Kuhlmann, H (2015) Accuracy analysis of a multi-view stereo approach for phenotyping of tomato plants at the organ level. Sensors (Switzerland) 15(5): 9651–9665.

[10] Westoby, MJ, Brasington, J, Glasser, NF. (2012) 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. Geomorphology 179: 300–314.