

2D to 3D image visualization and filter application

Alexander Raytchev

100938966

Abstract

With no more than a look, living beings can process the 3D world around them without second thought. This is not the case for computers, and is what computer vision looks to explore, exactly how can we translate real world information for a machine to accurately understand. This project will aim to see how easily a 2D image can be processed such that it would be able to receive a 3D representation. Furthermore, various filters will be added to explore if altering an image could lead a path towards an easier conversion.

Introduction

The application is a GUI that intakes any photo that's in the current directory and from there lets the user apply any of the 10+ filters that were implemented. Once the photo has been altered to the users liking the image can then be opened as a 3d graph with either matplotlib or the more sophisticated plotly. The first challenge was implementing a fully functional GUI that could have its state altered as the user goes through the program. What was challenging about this is that the state of the image had to be continuously preserved and passed along from function to function so that if the user wanted to graph the image with modifications they could do so. Secondly it was getting all dozen implemented filter algorithms (blur, edge detections, canny etc.) to be compatible with each other. The issue here was that the numbers in the array had to be consistent, it had to first always maintain the same size. Secondly the types RGB to BGR to grayscale that OpenCV naturally posses made it so that that I had to find a way of keeping all arrays consistent with each other. Lastly, and probably the biggest challenge was making the project proposal with no idea if it would work. I had never worked with or heard of computer vision before January and the proposal was based of the first two lectures. There was no guarantee that I would have been able to store an image into an array with x,y,z values that could later be plotted. The challenge in that sense was figuring out everything for myself.

Background

The background of this project comes from the very first lecture.

Digital Image

```
232 182 143 151 151 148 148 143 145 139 143 136 139 136 134 132 129 130 126 124 115 116 115 104 109 102 100 101
244 218 160 149 145 147 145 143 139 142 140 139 134 134 130 131 125 120 120 116 110 110 107 100 100 97 95 97
246 233 196 145 145 146 141 141 137 134 140 133 133 125 131 123 114 121 116 116 109 101 95 101 97 87 89 91
248 242 222 161 142 140 145 137 138 135 129 127 127 122 124 118 116 113 102 110 99 102 98 94 91 88 91 90
252 246 234 192 143 139 136 134 133 129 131 127 124 121 117 114 111 105 108 95 101 102 86 88 91 84 84 99
252 249 242 215 191 137 134 134 129 126 126 121 120 116 113 111 108 104 99 94 102 93 89 96 79 87 92 112
252 248 242 227 149 134 135 124 122 120 125 121 114 115 105 112 102 99 92 98 93 88 89 74 87 65 97 111
253 246 244 234 192 134 125 123 119 120 118 116 112 107 110 95 104 94 89 96 84 86 79 77 65 79 105 119
252 250 246 238 210 144 126 118 120 115 116 116 98 105 103 102 96 93 91 82 80 79 75 70 82 81 108 119
250 251 247 239 218 161 127 117 117 109 105 107 100 104 99 100 98 79 98 70 75 80 72 65 86 83 113 124
252 249 247 241 224 177 122 120 114 106 108 110 91 103 83 99 89 88 79 80 72 74 76 65 84 87 109 123
248 249 247 241 231 191 131 116 110 109 106 98 95 90 102 83 78 86 85 80 70 75 69 81 79 96 122 121
248 249 247 244 238 206 133 121 101 103 94 97 91 87 87 83 83 82 77 78 81 61 73 65 86 99 118 120
247 250 248 244 237 215 149 115 102 105 91 94 80 81 79 83 81 70 71 75 74 71 78 74 76 108 117 119
250 247 246 243 239 218 159 108 100 87 100 88 92 83 85 77 81 63 80 70 63 73 70 78 81 110 120 116
248 245 244 241 239 224 170 113 103 94 89 86 84 83 74 81 68 78 76 66 66 70 73 65 92 108 115 123
248 244 244 242 237 226 179 123 98 94 84 74 88 77 71 76 71 78 68 67 63 72 72 75 94 109 115 124
247 244 245 241 238 221 183 123 95 87 89 73 77 79 71 65 78 56 69 66 62 61 70 69 90 113 118 118
247 246 244 242 236 219 185 120 100 84 82 79 66 67 76 72 69 55 61 56 65 57 70 78 85 106 119 116
246 245 244 241 231 216 190 124 91 86 77 77 72 71 76 60 69 60 57 52 66 55 62 75 87 110 110 117
245 244 244 237 231 221 189 133 97 83 70 73 62 59 77 44 65 66 60 70 51 43 67 75 95 107 116 111
244 244 241 237 230 222 188 133 90 83 77 77 59 78 60 67 62 61 66 72 62 51 62 71 96 105 115 108
242 242 237 236 232 219 187 126 83 79 70 64 58 66 63 67 54 65 51 65 58 54 62 73 77 92 107 94
241 241 236 234 229 216 186 125 85 77 70 66 64 53 63 55 54 53 67 39 52 25 23 9 11 51 66 77
241 238 237 237 228 214 185 127 92 83 64 66 69 62 61 63 32 42 12 7 6 15 65 123 146 160 167 172
240 239 237 234 225 208 178 123 89 67 72 67 49 54 27 10 7 23 103 142 142 147 149 160 171 172 172 178
238 236 234 229 221 203 174 125 77 82 55 33 23 9 79 135 143 173 174 175 174 170 171 167 167 172 169 173
235 235 231 228 215 198 165 122 84 43 14 57 132 166 176 175 179 177 176 178 178 173 169 172 167 168 171 162
231 231 227 223 210 191 163 110 64 95 159 174 175 179 178 180 183 180 179 177 175 175 174 173 169 168 171 156
230 224 225 220 202 187 169 131 175 180 182 177 182 182 183 184 184 184 181 182 181 178 182 179 172 161 160 155
223 224 220 213 198 191 185 186 182 182 178 179 184 185 191 189 189 192 188 192 193 194 192 187 179 161 153 147
220 219 213 203 191 182 181 177 176 173 175 180 182 184 192 192 193 195 200 203 203 206 205 202 192 164 150 151
212 209 200 188 177 173 174 171 169 163 173 176 180 187 191 192 195 195 201 203 207 210 208 212 201 177 147 143
```

Seeing this image instantly gave me the idea that this could have potential to be something more. Something that other students and other people in general would see and be able to recognize that it is indeed the building blocks of an image. Another inspiration that came to me was that the application could be used as a teaching tool for computer vision. Because I struggled in the beginning of this course having a tool that would show you in real time what each filter does seemed like a good idea for myself and for potential future students who take the course. For example, I have an image called saltandpeper.jpg, if you select the image then press on the median filter the noise vanishes, this could be very beneficial to other visual learners. That is how my project had two focuses, create a 3d representation and get a lot of filters working. Then later investigate if the filters help make better 3d images.

Approach

The first step that needed to be done in order to get the approach to work was to resize every image with `cv2.resize()` and get them all into the same grayscale format. Otherwise there were issues with the rgb bgr gray defaults that are present. For example, an image that was not in grayscale format could not receive a lot of the edge detection filters.

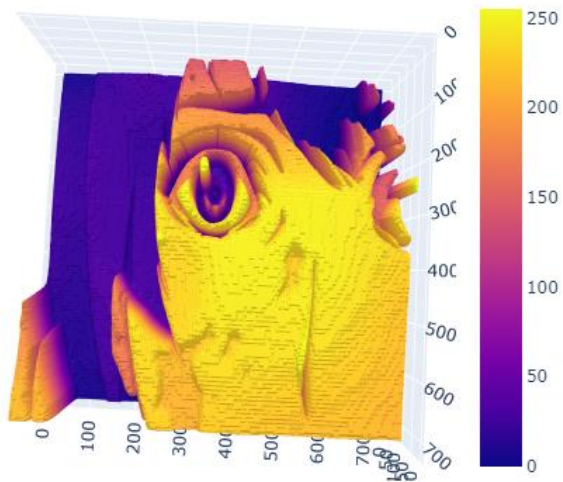
Next up were all the filters. The data of the image had to be passed into the functions which then called OpenCV prebuilt functions. Besides, Gray scaling, resizing, reading and passing the matrices no other algorithms were used. The application is more of a fun learning GUI that tends to borrow already prebuild filters from the OpenCV library. It uses average blur, Gaussian Blur, median blur, bilateral blur, laplacian, canny, sobel horizontal and sobel vertical edge detection, adaptive mean threshold and finally adaptive Gaussian threshold.

Results

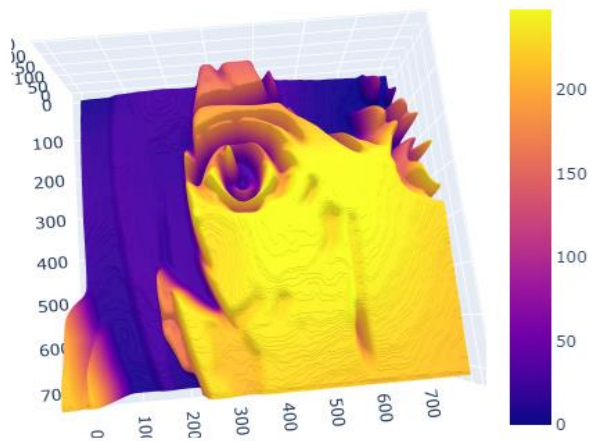
Original image



3D plot with plotly



3D plot with plotly after smoothing filters are applied



Here we can see in the results the difference in the intensity of the pixels when a smoothing filter is applied. In a normal view the image looks blurry but with the software you can clearly see the difference and how the computer would interpret the two images. And this point is key, computer vision is about what the computer can see. The only limitation was that I was not able to properly export the data the way I wanted to. Originally, I wanted to first get the data into excel, however it became apparent that each value in image was a very large array. This hindered plans to potentially port the image vectors anywhere else. If I was to recreate this project, I would have prioritized this feature and implement my work from the start to have a chance at exporting the values.

List of Work:

All Work was done by me as I was alone and did not have a group.

GitHub repository: <https://github.com/raytchev3004/comp4102>

References

OpenCV official documentation for basic algorithms

https://docs.opencv.org/master/d6/d00/tutorial_py_root.html

Programming manual provided on cuLearn for various functions with PIL, array, and cv2

Programming with python, Jan Erik Solem

Mathplotlib documentation for graph

<https://matplotlib.org/3.2.1/contents.html>

Plotly documentation for graph

<https://plotly.com/python/3d-axes/>

Tkinter documentation for GUI help

<https://docs.python.org/3/library/tk.html?>

stackoverflow for fixing code that would not compile