-EE604, Sanidhya Singh, 200864

<sw>:

Software used: Adobe After effects
URL: *https://www.adobe.com/in/products/aftereffects.html*
Feature explored: Content-Aware Fill
What it does?

Many a times we capture unwanted objects in our films. This feature allows us to select those objects and remove them from the whole film.

Some results: [credits: Blue Mantle Films, YouTube**]**

Before *After*

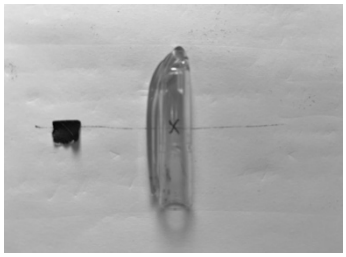
Research Paper used to explore this feature: Rosana Balanescu , Adrian Sterca, Ioan Badarınza, "Removal of Unwanted Objects from Still Photograhs," SoftCOM, 2020

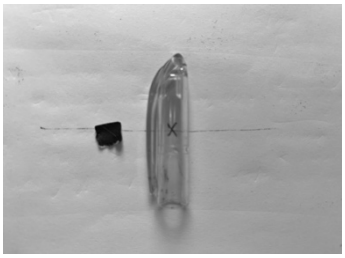
How this feature works?

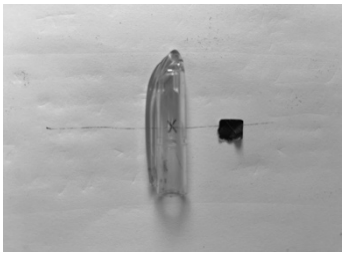
We will reconstruct a frame with the help of multiple frames extracted from the video. To fill the space occupied by the obstructing object we will use the pixel values of the other frames. The problem of finding correlation between the hidden area of the frame to the same point from the real world in the other frames is solved by following two steps:

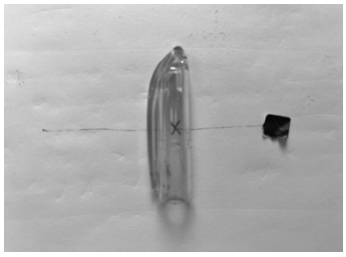
1. Matching blocks of pixels from several frames and removing blocks that contains intrusive object.
2. Blending matched block of the pixel into the final image.

I have explored this feature assuming that the camera is held static and the object to be removed is moving. Below are the images which I have captured using my cell phone (converted to B/W for simplicity) where the pen cap is still and a black paper is moving:

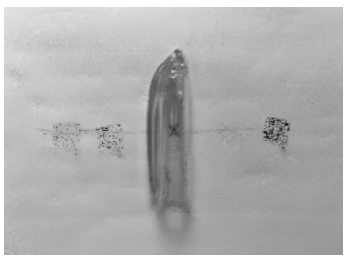








After applying the algorithm mentioned in the paper, I obtained ⟶
The obtained result contains lots noise and is far from perfect since I have just used 4 frames, increasing the number of frames will yield much accurate results.



Algorithm used to compute the result: **[Link to my jupyter notebook:** *https://drive.google.com/file/d/11b4GG2jt4COYUem6nMm56M6MYbr0tfMb/view?usp=sharing* **]**

Pseudo code for main body:

```
clean_img = empty matrix
for i in images:
        Macroblock_i = convert_to_macroblock(i)
for k in Macroblocks:
        min_mse = inf
        min_idx = 0
        for i in images - 1:
                mse_val = mse(image[i], image[i+1])
                if mse_val < min_mse:
                        min_mse = mse_val
                        min_idx = i
                end if
        end for
        Mk = blend(Macroblock_at_min_idx[k], Macroblock_at_min_idx+1[k])
        clean_img = add(Mk, clean_img)
end for
```

I have successfully implemented this algorithm by tweaking some parameters to obtain the best possible results in my jupyter notebook.

Idea behind the algorithm:

Through this algorithm we first convert the images into the form of macroblocks, which is nothing but a submatrix of the whole matrix. We then compare the corresponding macroblocks in all the frames to determine the pair of macroblocks which are most similar, as there will be high probability that these macroblocks do not contain the moving object that needs to be removed. We do so by calculating MSE of two consecutive frames. The pair of macroblocks producing the minimum MSE will be used to generate final clean image.

<q4q>:

q4q-analytical:

[Note: I have used the same image processing method as above to create this question, it is assumed that the student have read the paper mentioned]

You are given four 10 x 8 matrix representing multiple consecutive frames of an image. But, there is a problem. There is a moving obstruction in the frames. You have to use a moving object removal algorithm to generate clean image and write it in your answer sheet.

Matrices:

1.)

```
[[220 222 212 213 211 198 190 188]
 [199 191 195 189 199 188 183 189]
 [203 186 189 182 174 168 177 176]
 [198 193 185 102 171 162 162 159]
 [199 196 188 115 166 164 159 164]
 [193  18 179 116 161 161 156 159]
 [186 186 177 105 154 158 156 156]
 [187 181 176  94 152 161 160 158]
 [177 172 173 169 150 166 161 158]
 [176 177 179 173 156 162 158 155]]
```

2.)

```
[[217 217 218 216 206 199 183 190]
 [200 189 194 195 190 182 179 177]
 [200 194 183 184 169 172 179 173]
 [197 191 187  91 170 162 162 158]
 [194 189 188 107 166 163 157 158]
 [185 192  17 112 167 161 159 161]
 [182 178 175 106 163 154 155 154]
 [180 180 176  97 162 159 157 155]
 [172 177 182 168 164 162 157 156]
 [171 175 170 173 163 157 154 151]]
```

3.)

```
[[220 221 212 207 201 190 173 184]
 [205 197 201 197 185 181 174 175]
 [203 193 183 183 171 175 171 168]
 [200 188 179  92 174 164 158 157]
 [193 187 182 128 178 170 154 163]
 [180 189 174 138 177  78 157 150]
 [184 178 168 112 167 152 151 155]
 [179 176 170 115 161 156 156 150]
 [170 172 175 171 157 159 153 152]
 [168 167 170 172 157 158 156 151]]
```

4.)

```
[[220 207 214 208 203 189 188 186]
 [213 204 207 201 191 188 179 185]
 [209 197 191 186 171 181 179 177]
 [203 195 184  97 174 165 170 165]
 [196 193 182 118 166 168  97 166]
 [193 196 179 111 164 168  75 165]
 [188 193 176  97 156 160 156 159]
 [186 183 177 135 155 162 161 162]
 [175 175 173 173 147 166 159 156]
 [180 175 177 173 157 163 158 154]]
```

Ans:

There are multiple answers possible depending upon the choice of the size of macroblock. But to obtain optimal results one should use 2 x 2 size of macroblocks.

In total we will have $80/4 = 20$ macroblocks.

For each macroblock, we have to do 3 calculations as 4 images are given, so we end up doing 80 calculations for this problem. Alternatively, one can ignore the last row and choose a macroblock of 3 * 2, to reduce the calculations to 36.

To calculate MSE, we will employ the definition of "new quality index - Q," which is defined as:

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{\left(\sigma_x^2 + \sigma_y^2\right)\left[(\bar{x})^2 + (\bar{y})^2\right]}$$

Where,

x = k'th macroblock of i'th image
y = k+1'th macroblock of i'th image

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

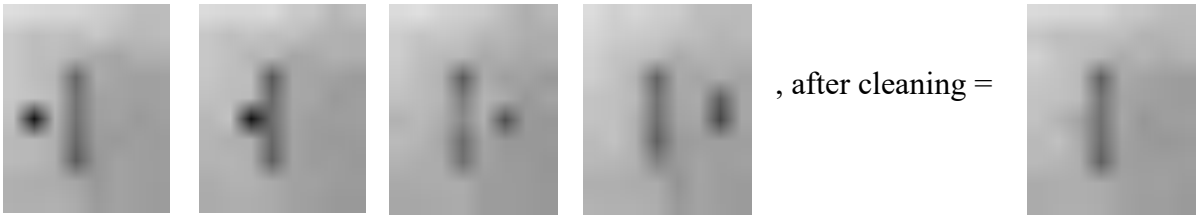
$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}).$$

Once the macroblock having minimum index is found, blend it with its neighbouring (min_idx + 1' th) macroblock by averaging their values.
Repeating this for all the macroblocks we obtain the final clean matrix as (decimals are rounded off):

```
[[217 217 212 213 205 198 188 186]
 [200 189 194 189 189 181 179 185]
 [208 195 190 185 171 179 177 175]
 [202 194 184  97 173 164 161 158]
 [196 193 188 114 167 163 156 158]
 [193 179 164 115 167 153 158 160]
 [187 192 174 106 163 153 155 155]
 [186 182 175  98 161 158 159 157]
 [171 176 173 172 163 161 160 157]
 [170 174 177 173 162 157 157 154]]
```

Pictorially (obtained using the aforementioned jupyter notebook):


, after cleaning =

q4q–mcq:

Q1. Which of the following is/are true regarding an algorithm to remove moving objects from a movie:

a) Use of submatrix instead of single pixel to employ mean-squared error like methods to find correlation between multiple frames
b) There is a low probability that the corresponding macroblock in the neighbouring image will not contain moving object that needs to be removed
c) Adding noise to the image will fail the algorithm
d) To remove fast moving object we will need more number of frames per second

Ans: a,d

b is incorrect, it should be high instead of low.

c is incorrect, as any static noise won't create much of a difference between MSE of two neighbouring frames.

Q2. Which of the following option/s can improve the obstructive object removal techniques:

a) Increasing the size of Macroblock can improve the algorithm
b) Decreasing the size of Macroblock can improve the algorithm
c) Increasing the number of frames can improve the algorithm
d) Decreasing the number of frames can improve the algorithm

Ans: c

There is no direct correlation between algorithms result with the size of the Macroblock. One should test several size of the Macroblock depending upon the object to be removed, how big it is and how fast it's moving.

Increasing the number of frames will definitely improve the algorithm.

-EE604, Sanidhya Singh, 200864