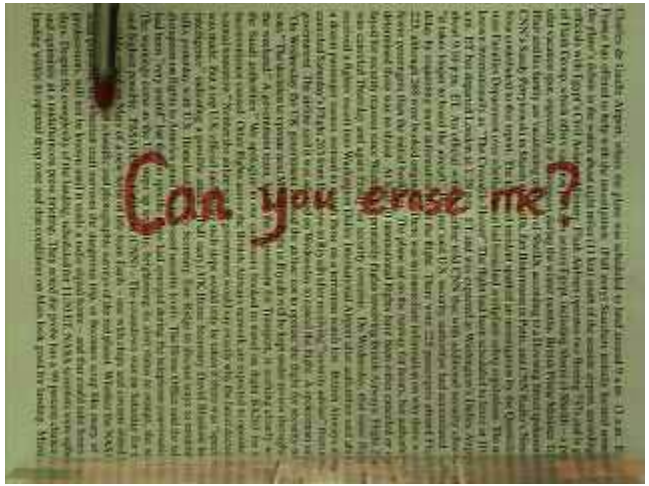
“Magic” Eraser

- Presented by Lauren Hutson and William Spies
- Demonstrated on 11/28/2017 to the class of EECS432, Northwestern University

Goal



- Erase the words written on the newspaper in the video.
- Replace the erased text with a similar newspaper texture.

Notes

- Need to be able to access video feed and convert each frame into “Image.”
  - OpenCV has good tools for live video access and accessing video files.
- Source video is fairly noisy, might create problems with masks.
  - Good candidate for using morphological tools.
- Marker and text are both **red**.
  - Can use OpenCV “InRange” function to define color space boundaries for the final mask.
- Texture synthesis requires some filtering to only grab “newspaper” textures.

- Ideally, all code runs in less than 33.3 milliseconds (can resolve for video recorded at 30 frames per second).

Strategy

1. Gain access to the frame data from the provided video file.
2. Develop a “mask” for the desired content to erase; in this case, the red ink.
3. Detect the current position of the red pen.
4. Erase the portion of the red text to the right of the pen’s current position.
5. Synthesize newspaper texture through use of a non-parametric sampling algorithm to fill the gap left from the erasure.
6. Combine synthesized newspaper texture and original image.
7. Display the final results!

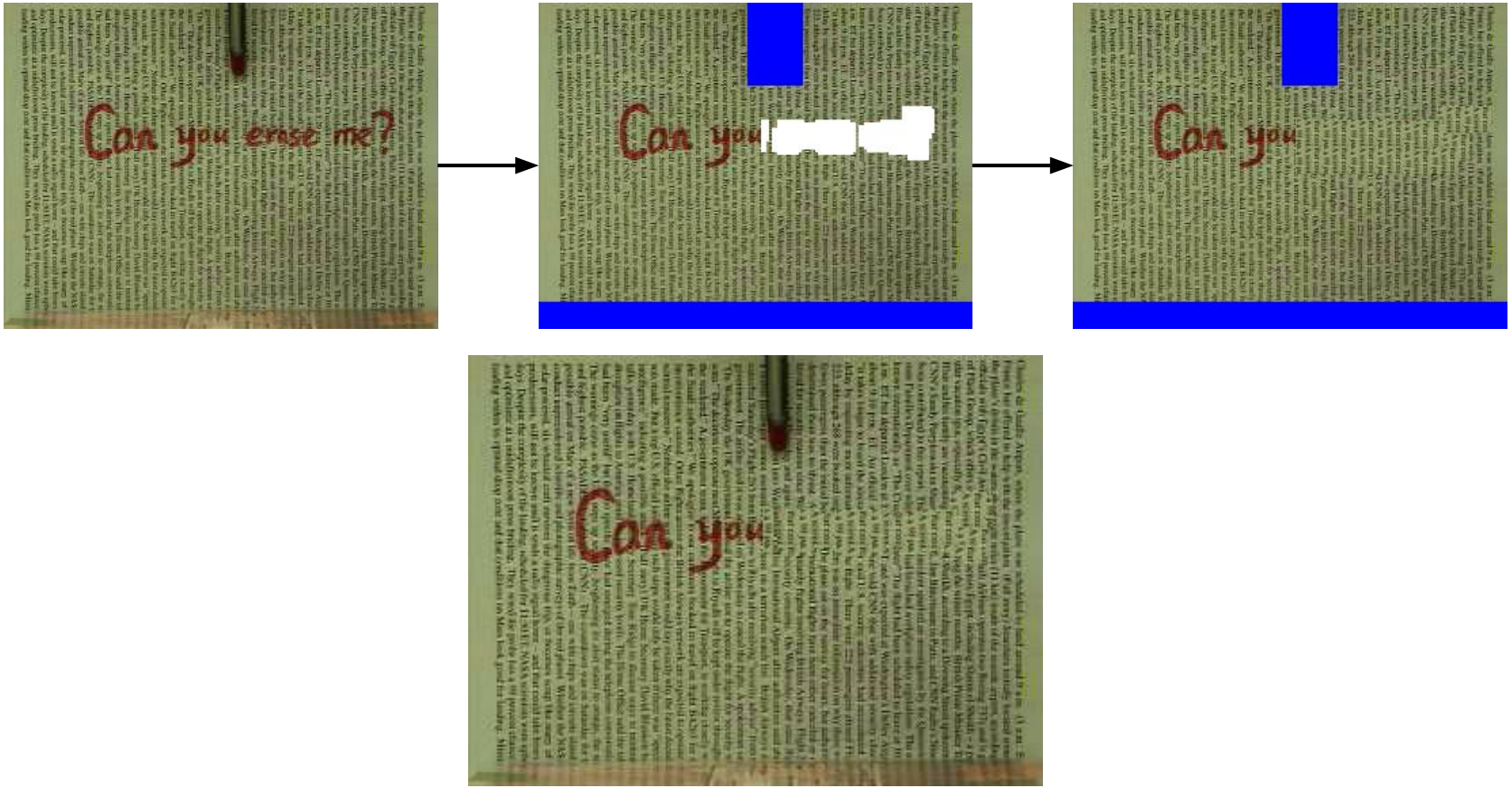
KEY : Marker Text Mask

1. Convert video frame image data from BGR-space to HSV-space for mask generation.
   a. Simplifies grabbing red text and red pen tip.
2. Establish acceptable boundaries for text of interest and use “InRange” function to isolate.
3. Clean up initial mask through erosion and dilation until the mask is…
   a. Devoid of any noise.
   b. Sufficiently large to catch all text of interest.
4. Remove the contributions of the marker or other image artifacts on the cleaned-up mask.
5. Pass the mask to functions that will perform erasing based on marker position and to the “tile” synthesis code.

KEY : Texture Synthesis Algorithm

1. Randomly generate a “Tile” texture sample from the frame.
    a. Need to make sure that the sample does not include any of the erased portion, the pen, or the graphical error at the bottom of the frame.
2. Select an overlap width and scan the left and upper edges of the erased area and the left side of the sample at a depth of the overlap width.
    a. For each pair of pixels perform the “sum of square difference” (ssd).
3. If the “sum of square difference” is a low number then the two overlap areas are similar. If not a low number, then the overlap areas are not similar.
4. Choose a value on the lower end of the total space of “ssd” values.
    a. If the sample image’s “ssd” is below that value, then the tile is placed into the erased area.
5. Repeat until the entire erased area is retextured with the newspaper background.

Results



Areas for Improvement

1. Equalize shading around synthesized textures
    a. Can clearly pick out some synthesized areas based on large changes in contrast. Better if those regions are equalized.
2. Improve isolation stability of marker location
    a. Towards the end of the video, the position of the marker (as detected by the code) jumps around and could cause issue with the display of the texture synthesis results.
3. Change structure to allow for external parameterization of values such as…
    a. Mask colors (what if the marker was blue?)
    b. Texture synthesis parameters
4. General optimizations
    a. Reduced CPU needs
        i. Fewer CPU cycles needed means code can resolve sooner, raising the framerate cap for the Magic Eraser process as applied to live video.
    b. Reduced memory footprint
        i. Generates a lot of “copies” of frame image arrays.