Image Inpainting and Its Application in Video CAPTCHA Text Removal

Huaijin Chen, Richard Zanibbi, Advisor

Rochester Institute of Technology

OBJECTIVE

The motivation of this senior project is to improve the reliability of the current Video CAPTCHA by implementing an image inpainting algorithm to eliminate visible text within the video, thus reduce the risk of being bypassed using OCR attacks while avoiding visual artifacts to be shown to the end-users. We completed the following work to prototype a pipeline that we take in images or decomposed video frames that contain text, inpaint the specified text region, and eventually evaluate the inpainting performance.

1. Investigated four major inpainting algorithms, and chose Bertalmio's Image Inpainting [1] for our inpainting step
2. Concluded a proper stop point of the iteration and a optimal dilation size for the text region mask for our particular case of inpainting, because those are not specified in the Bertalmio's paper
3. Developed framework to pre-process the inpainted images for the OCR attack
4. OCR attacked the pre-processed images and measured the attack success rate using the string edit distance.
5. Assessed the inpainting through perceptual appearance of the inapinted image.

Background

- **Inpainting**

In art world, the word "inpainting" referred to the process of precise restoring of the lost or damaged part of paintings. This technique has been widely used by skilled art conservators to recover valuable paintings. In a typical inpainting process, the inpainting region is defined first, after that, contour line intercept at the boundary containing structure information near the boundary are extended into the inpainting region, and then different areas inside the inpainting region segmented by the contour lines are filled with the corresponding colors matched at the boundary, eventually textural details are extended into the region. [8]

- **Digital Image Inpainting**

Bertalmio's Imaging Inpainting is arguably the first paper on digital image inpainting. The algorithm proposed in the paper exactly follows the fundamental inpainting principles stated above. In such algorithm, Given the inpainting region and boundary of the inpainting region, the goal of such algorithm is to prolong the contour lines of different gray-scale level arriving at the boundary, while maintaining the angle of arrival [1]. Shown in Figure 1.

Figure 1: Extending contour lines into the inpainting region

- **Video CAPTCHA**

Kluever and Zanibbi [9] developed a novel video-based CAPTCHA system which will take in a video clip from user-contributed video sites, i.e. YouTube1, then ask the user to watch the clip and type in three keywords about the video. The CAPTCHA test is passed if the typed in keywords match the retrieved keywords.

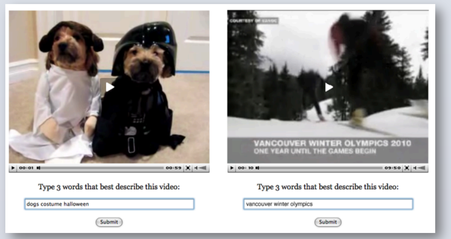


Figure 2: Video CAPTCHA

METHODS

- **Inpainting**

We examined four inpainting algorithms, including the Image Inpaint- ing [1] and Simultaneous structure and texture image inpainting [2] by Bertalmio, et al, Region filling and object removal by exemplar-based image inpainting [6] by Criminisi, et al, and Total Variation-based method proposed by Chambolle [3], but eventually chose to incorporate Bertalmio's original image inpainting algorithm to our text removal pipeline because of its assumption of distinguishable contour lines agrees with the fact that text natural scenes or artificially imposed are usually of bold color on a smooth background.

This iterative algorithm is basically consisted of two steps: Inpainting and Diffusion.

Inpainting Step

Given a 2D gray-scale image I of size M × N, the inpainting process can be generally described using

$$I^{n+1}(i,j) = I^n(i,j) + \Delta t I_t^n(i,j)$$

where (i,j) are the pixel coordinates that needs to be inpainted within the inpainting region Ω, $I^n(i,j)$ is the input image at each iteration and $I_t n(i,j)$ is the update to the input image, and Δt is the rate of improvement.

Since we need to smoothly propagating information $L^n(i,j)$ from outside the inpainting region Ω into the region Ω across the inpainting boundary δΩ along the direction $\vec{N}^n(i,j)$, the update $I_t^n(i,j)$ needs to be

$$I_t^n(i,j) = \delta \vec{L}^n(i,j) \cdot \vec{N}^n(i,j)$$

where $\delta L^n(i,j)$ is a measure of the change in the information L(i,j). To smoothly propagate information inside, $L^n(i,j)$ should be a image smoothness estimator. Thus we can use the discrete Laplacian

$$L^n(i,j) = I_{xx}^n(i,j) + I_{yy}^n(i,j)$$

where the subscripts represent the derivatives. Therefore, the change $\delta L^n(i,j)$ of the discrete Laplacian can be expressed as

$$\delta \vec{L}^n(i,j) = (L_n(i+1;j) - L_n(i-1,j), (L_n(i;j+1) - L_n(i,j-1),$$

Meanwhile, the direction of the inpainting $\vec{N}^n(i,j)$ is defined as the direction of the contour lines, therefore it can be calculated by

$$\frac{\vec{N}^n(i,j,n)}{|\vec{N}^n(i,j,n)|} := \frac{-I_y^n(i,j), I_x^n(i,j)}{\sqrt{(I_y^n(i,j))^2 + (I_x^n(i,j))^2}}$$

Eventually the discrete implementation of the update $I_t^n(i,j)$ becomes

$$I_t^n(i,j) = \left(\delta \vec{L}^n(i,j) \cdot \frac{\vec{N}^n(i,j,n)}{|\vec{N}^n(i,j,n)|}\right) |\nabla I^n(i,j)|$$

The last term in the equation is the slope-limited version of the norm of the gradient of the image that

$$|\nabla I^n(i,j,n)| = \begin{cases} \sqrt{(I_{xbm}^n)^2 + (I_{xfM}^n)^2 + (I_{ybm}^n)^2 + (I_{yfM}^n)^2}, & \text{when } \beta^n > 0 \\ \sqrt{(I_{xbM}^n)^2 + (I_{xfm}^n)^2 + (I_{ybM}^n)^2 + (I_{yfm}^n)^2}, & \text{when} \beta^n < 0 \end{cases}$$

where

$$\beta^n(i,j) = \delta \vec{L}^n(i,j) \cdot \frac{\vec{N}^n(i,j,n)}{|\vec{N}^n(i,j,n)|}$$

and the sub-indexes b and f denote backward and forward differences respectively, and m and M denote the minimum or maximum, respectively, between the derivative and zero.

Diffusion Step

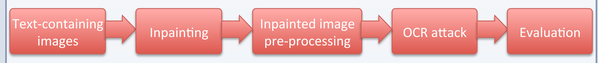
To further propagate information into the region and ensure the smoothness of the propagation, a diffusion step is suggested to be applied after each iteration of inpainting by the paper. Anisotropic diffusion is used in order to achieve goal without losing sharpness in the reconstruction. The discrete anisotropic diffusion can be described using the following equation.

$$\frac{dI}{dt}(x,y,t) = g_\epsilon \kappa(x,y,t) |\nabla I(x,y,t)|$$

The paper suggested a total iteration T = 3000, with inpainting iteration t=17 and diffusion iteration d=2.

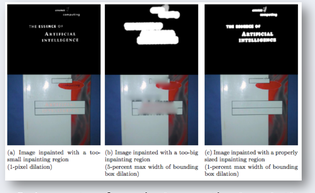
Experiments

Our pipeline for the experiments is shown below



- **Testing datasets**

The testing datasets we used for our experiments were the ICDAR robust reading and text locating testing datasets, which contain 251 images of natural scenes that contains in total 517 lines of text (five images without text were also included) We resize the oversized image (longest size > 480 pixels) to 480p resolution which resembles the resolution of a typical online video. The text regions are provided by [15] and dilated with a disk that has radius of 1 % of width the of the biggest text bounding box within in the image. The dilations were made to deliver the optimal inpainting result. (Figure 3)



(a) Image inpainted with a too-small inpainting region (1-pixel dilation) (b) Image inpainted with a too-big inpainting region (5-percent max width of bounding box dilation) (c) Image inpainted with a properly sized inpainting region (1-percent max width of bounding box dilation)

Figure 3: Impacts of mask size on the inpainting result

- **Inpainting algorithm**

Our pipeline starts with a text-containing image and the defined corresponding text regions. We inpaint the text region with the modified Bertalmio's algorithm in which the algorithm stops when the difference in MSE between the current iteration image and the original image is smaller than 0.1 × 10−4 compared to the previous iteration or after 1500 iterations. Such threshold was made due to the fact that we found the inpainting algorithm converges around that point. (Figure 4)

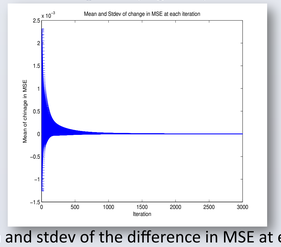


Figure 4: Mean and stdev of the difference in MSE at each iteration

- **Pre-processing of the inpainted image**

The following pre-processing procedures were applied to the inpainted images to ensure OCR success rate. (Figure 5)

1. Extracting the text regions of the image to smaller images
2. Mean shift filtering to reduce the noise and segment the text out [4]
3. Convert the segmented color images to gray-scale
4. Image binarization using Otsu's method [12]

- **OCR attacks**

Finally, we ran OCR attacks on the pre-processed images. We chose the Tesseract OCR engine3 for its better performance among the open source OCR engines. For the parameters of the OCR, we set the language to English and page segmentation to treat the image as a single text line.

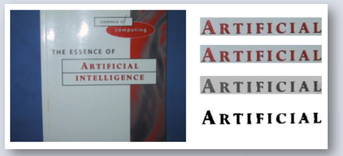


Figure 5: Sample pre-processing result, left column: original image, right column from top to bottom pre-processing results in the order as described

Result

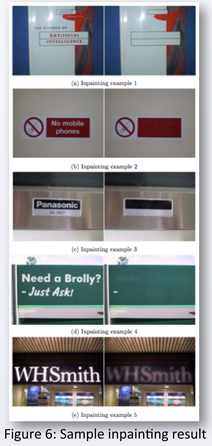


Figure 6: Sample inpainting result

We used Levenshtein distance [10] as the metric for OCR attack assessment. Here we showed five samples of inpainting results in Figure 6, and the corresponding OCR attack results, including the recognized text and the Levenshtein with and without inpainting are shown in Table 1.

Inpainted Image	Text Ground Truth	OCR Before Inapinting	Distance Before	OCR After	Distance After
Example 1	ARTIFICIAL	ARTIFICIAL	0	-	10
Example 2	No mobile	No mobile	0	-	9
Example 3	Panasonic	Panasonic	0	-	9
Example 4	Need a Bolly?	Need a Bolly?	0	-	14
Example 5	WHSmith	WHSmith	0	n HSmith	4

Table 1: OCR attack result

Plotting out the histograms of the Levenshtein distance between the ground truth text and OCR results with and without inpainting for all 517 text lines in the 250 testing images in Figure 7, we can clearly see that inpainting successfully reduced the OCR attack success rate from 38.29% to 3.8%, if we count exact match between OCR result and ground truth as a successful attack.

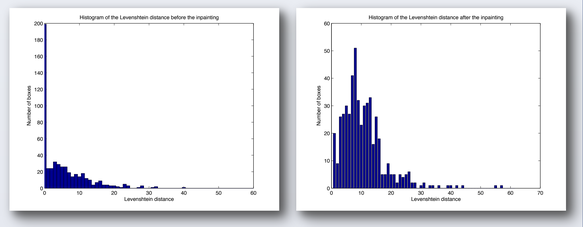


Figure 7: Histogram of the Levenshtein distance between the ground truth text and the OCR text before (left) and after (right) the inpainting.

REFERENCE

[1] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pages 417– 424. ACM Press/Addison-Wesley Publishing Co., 2000.

[2] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. Image Processing, IEEE Transactions on, 12(8):882–889, 2003.

[3] Antonin Chambolle. An algorithm for total variation minimization and applications. Journal of Math- ematical imaging and vision, 20(1-2):89–97, 2004.

[4] Yizong Cheng. Mean shift, mode seeking, and clustering. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 17(8):790–799, 1995.

[5] James A Clarkson and C Raymond Adams. On definitions of bounded variation for functions of two variables. Transactions of the American Mathematical Society, 35(4):824–854, 1933.

[6] Antonio Criminisi, Patrick P´erez, and Kentaro Toyama. Region filling and object removal by exemplar- based image inpainting. Image Processing, IEEE Transactions on, 13(9):1200–1212, 2004.

[7] Jim Douglas and HH Rachford. On the numerical solution of heat conduction problems in two and three space variables. Transactions of the American mathematical Society, 82(2):421–439, 1956.

[8] G. E´mile-M'ale. La Restauration Des Peintures de Chevalet (Restoration of easel paintings). Office du Livre, 1976.

[9] Kurt Alfred Kluever and Richard Zanibbi. Balancing usability and security in a video captcha. In Proceedings of the 5th Symposium on Usable Privacy and Security (SOUPS '09), 2009.

[10] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In Soviet physics doklady, volume 10, page 707, 1966.

[11] Antonio Marquina and Stanley Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. SIAM Journal on Scientific Computing, 22(2):387–405, 2000.

[12] Nobuyuki Otsu. A threshold selection method from gray-level histograms. Automatica, 11(285-296):23– 27, 1975.

[13] Shuman David Vandergheynst Pierre Perraudin Nathanal and Puy Gilles. UNLocBox: Short User Guide. LTS2 EPFL, 2012.

[14] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60(1):259–268, 1992.

[15] Dave Snyder. Text detection in natural scenes through weighted majority voting of DCT high pass filters, line removal, and color consistency filtering. PhD thesis, Rochester Institute of Technology, 2011.

[16] Luminita A Vese and Stanley J Osher. Modeling textures with total variation minimization and oscil- lating patterns in image processing. Journal of Scientific Computing, 19(1-3):553–572, 2003.

[17] Wikipedia. Total variation - wikipedia, the free encyclopedia, 2013. [Online; accessed 10-May-2013].

[18] Luke Wroblewski. A sliding alternative to captcha. http://www.lukew.com/ff/entry.asp?1138.

[19] Hector Yee, Sumanita Pattanaik, and Donald P Greenberg. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. ACM Transactions on Graphics (TOG), 20(1):39–65, 2001.

ACKNOWLEDGEMENT

The author wish to thank Profs. Richard Zanibbi, Carl Salvaggio and Jeff Pelz and Susan Farnad for precious guidance and advice, and members of DPRL for help on the project

RESEARCH POSTER PRESENTATION DESIGN © 2012
www.PosterPresentations.com