Image Object Removal

Alastair Crowe

March 21, 2008

Figure 1: *From left to right: the original image, the image with the target region highlighted, the constructed image using AIEI, the constructed image using CIEI. [3]*

Object removal is concerned with the problem of replacing a user selected area, termed the target region Ω, in an image Υ with a new computer generated plausible image (see figure. 1) . This may be used to remove unwanted people or objects from a photograph. The most successful methods for doing use exemplar based texture synthesis which is using textures from the remainder of the image, termed the source region $\Phi = \Upsilon - \Omega$, to fill in the target region. This is done one patch at a time and in most algorithms the user must determine the size of patch to be used. The algorithm selects a patch to be filled in the target region called the current patch Ψ_p, which is centered on pixel p. The most similar patch from the source region, termed the candidate patch Ψ_q , is then copied onto the current patch.

The main differences in object removal algorithms are the methods used to define the best candidate patch to copy onto the current patch. Also, as the target region is filled previously inserted patches contribute to the future chosen patches, therefore the order in which current patches are chosen to be filled also effects the results.

Criminisi et al. [1] describe an algorithm (AIEI) which orders the sequence of patches to be filled by two variables. The first is a confidence factor which describes how much information is available to aid in current patch selection. The second is a data factor which indicates the linear structure intersecting the target region at the pixel where the current patch is centered. The algorithm encourages patches which contribute to linear structure entering the target region while also encouraging patches with a high amount of information available to be filled in first. The candidate patch is selected by minimizing a simple distance function. This algorithm is described in more detail in the following section.

Wu et al. [3] edit the data factor in the previous algorithm to encourage patches which show intruding linear structure while discouraging patches which are parallel to linear structure (CIEI). This modification improves the algorithm and a comparison of the results is shown in figure 5 and in figure 1.

Along Isophote Exemplar-based Inpainting

The algorithm presented in [1] uses the order in which the patches are placed to preserve linear structure in the image. This is termed fill order and uses

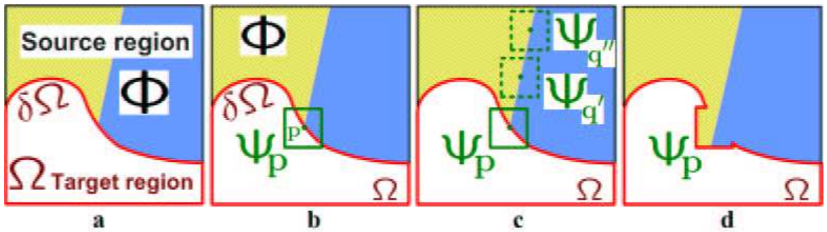

Figure 2: *a) Shows the source region Φ, target region Ω, and the boundary separating them $\delta\Omega$.b) The current patch Ψ_p has been selected on pixel p. c) The candidate patch $\Psi_q$is most likely selected from along the same texture boundary in the image. d) The optimal candidate patch is copied onto the target patch* Ψ_p. [1]

confidence values and data values for each pixel along the perimeter of the target region to order patches by giving them priority values. These priority values are used to select patches which contribute to the linear structure while remaining consistent to the source region.

The pixel with the maximum priority is then found and the most similar patch in the source region is centered over it. The patch must be sampled entirely from the source region and a distance function is used to calculate the similarity of the candidate patch and the current source patch. The entire source region is raster scanned to find the optimal candidate patch before it is copied onto the current patch. The new confidence factors are then assigned, the new priorities calculated and the next iteration proceeds until the entire target region is filled in.

Initialization

Initially the algorithm needs an image Υ separated into the source region Φ and an target region Ω (see figure 2). Additionally the user specifies a template window Ψ size, which indicates the size of patch to be copied during each iteration. By default this is a 9×9 pixel window however the it should be set slightly larger than the largest distinguishable texture element in the source region.

Each pixel in the image has a colour value and a confidence value which indicates our confidence that the pixel has been filled in correctly. On initialization the colour value is assigned 'empty' for pixels in the target region and the confidence values for all the pixels non-empty pixels (i.e those in the source region) is 1. Once pixels are filled in confidence values cannot be changed.

Finding the maximum pixel priority

Each pixel along the boundary $\delta\Omega$ (see figure 3) of the target region Ω has a priority value defined a

$$P(p) = C(p)D(p) \tag{1}$$

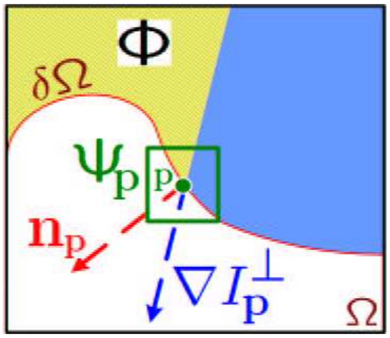

Figure 3: *Shown is the source region* Φ*, target region* Ω*, the target boundary* $\delta\Omega$*, current patch* Ψ_p *centered on pixel* p*, the normal* n_p *to the target boundary at pixel* p *, and the isophote* $\nabla I_p^{\perp}$ *at pixel* p*.* [1]

where $C(p)$ is the confidence value and $D(p)$ is the data value for pixel p and defined as

$$C(p) = \sum_{q \in \Psi_p \cap \Phi} C(q) / \mid \Psi_p \mid$$

$$D(p) = (\nabla I_p^{\perp} \cdot n_p)/\alpha$$

where $\mid \Psi_p \mid$ is the number of pixels in the current patchΨ_p, α is a normalization factor (i.e $\alpha = 255$ for a typical gray scale image) , n_p is a unit vector normal to the perimeter at this pixel p, and $\nabla I_p/$ change of illumination at this pixel. Therefore $\nabla I_p^{\perp}$ is the magnitude and direction of the strongest isophote.

At initialization, the confidence values $C(p)$ are set to $C(p)$ $=0 \forall_p \in \Omega$ and $C(p) = 1 \forall_p \in \Phi$.

The confidence value is a measure of reliable information surrounding the pixel. This encourages pixels with more of the surrounding area already filled in to be filled in first, as well as the boundary of target region to be filled in before the center (see figure 4).

The data value is a function of the strength of isophote entering the target region at each perimeter pixel each. This is a scaled measure of the change of illumination along the perimeter. This encourages pixels which contribute to the linear structure of the image to be filled in first.

Once all perimeter pixels have been assigned their priority values the pixel with the highest priority is found and the next candidate patch is to be centered on this pixel.

Finding the best candidate patch

To find the best candidate patch the source region is raster scanned and the candidate patch centered on the pixel which minimized a simple distance func-

Figure 4: *This series of pictures demonstrates the fill order of the AIEI algorithm. Notice, particularly in the third image, that the bounding edges between the building and the foliage have a higher priority.* [1]

tion is chosen. A valid candidate patch must be completely inside the source region.

$$\Psi_{\hat{q}} = \arg \min_{\Psi_q \in \Phi} d(\Psi_{\hat{p}}, \Psi_q) \tag{2}$$

where the distance between two candidate patches and is the sum of squared differences of the already filled pixels in the two patches.

$$d(\Psi_{\hat{p}}, \Psi_q) = \sum_{i,j} (\Psi_{\hat{p}}(i,j) - \Psi_q(i,j))^2$$

where $\Psi_p(i,j)$ is value of the pixel (i,j) in the CIE Lab color space [2]. Once the candidate patch has been found its is copied pixel for pixel into the target region of the current patch.

Updating confidence values

The confidence values for the new pixels are then assigned and frozen.

$$C(p) = C(\hat{p}), \forall p \in \Psi_{\hat{p}} \cap \Omega \tag{3}$$

This will result in confidence values decreasing towards the center of the target region, encouraging the outside to be filled first. The process is then repeated with the new confidence values and the new target region.

$$\Omega' = \Omega + \Psi_{\hat{p}}$$

Cross Isophotes Exemplar-based Inpainting

An extension to the previous algorithm has been proposed by Wu et. al [3] where it is suggested that the data value should be the diffusion of total variation.

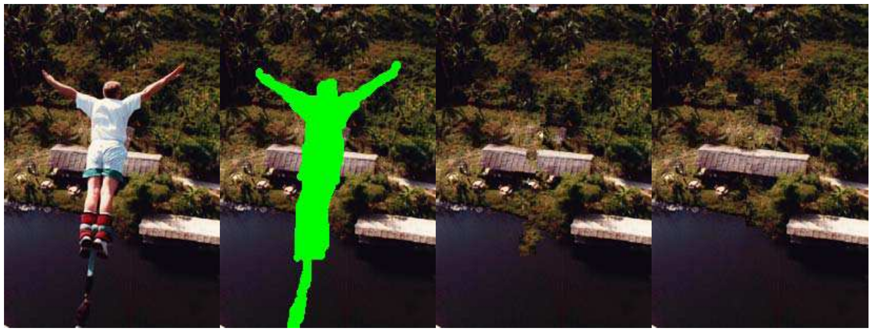

Figure 5: *From left to right: The first image shows the original picture, next has the target region is highlighted, the third shows the results of the AIEI algorithm, and the last the CIEI algorithm.[3]*

$$D(p) = \mid u_{\varepsilon\varepsilon} \mid / \mid \nabla u \mid \tag{4}$$

where $\mid \nabla u \mid$ is the magnitude of change of illuminance along the perimeter $\delta\Omega$ and $\mid u_{\varepsilon\varepsilon} \mid$ is the second order directional derivative of image in the direction along the normal direction. Notice that when the change of illumination along the normal ∇u is large the data value $D(p)$ is small. This discourages pixels to be selected when there is slight linear structure along the perimeter. When the second order derivative $u_{\varepsilon\varepsilon}$ is data value $D(p)$ is large. This encourages pixels to be selected when linear structure is intruding into the target region Ω at this point.

Conclusion

By visually inspecting the newly created images using each algorithm the improvement brought by using the CIEI data value calculation (Figures 1 & 5) can easily be seen. This algorithm successfully creates a plausible target region and currently is one of the best available.

Bibliography

[1] Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting”, IEEE. Trans. Image Processing, vol. 13, no. 9, Sept. 2004.

[2] J. M. Kasson and W. Plouffe, “An analysis of selected computer interchange color spaces,” in ACM Trans. Graphics, vol. 11, Oct. 1992, pp. 373–405.

[3] Jiying Wu and Qiuqi Ruan, ”Object Removal By Cross Isophotes Exemplar-based Inpainting”, The 18th International Conference on Pattern Recognition (ICPR’06)