The Development of a Fragment-Based Image Completion Plug-in for the GIMP

Cathy Irwin

Supervisors: Shaun Bangay and Adele Lobb

Abstract

Recent developments in the field of image manipulation and restoration have seen the merging of techniques that reconstruct texture and those that reconstruct structure to create algorithms that are more effective for a far greater variety of image completion scenarios. Few of these innovative and useful algorithms are available to ordinary users however. We describe the implemention of a version of one such technique, the fragment-based image completion algorithm developed by Drori, Cohen-Or and Yeshuran (2003) as a plug-in for the GIMP making it freely accessible and easy to use in conjunction with other image manipulation tools. Results thus far are comparable with those of the original authors.

1 Introduction

The manipulation and modification of images in an undetectable manner has in the past required skillful and time-consuming artistic endeavour. Applications for such a process include the removal of scratches, defects or writing from photographs or films, the restoration and reconstruction of damaged images, and the removal of objects from images. Computer technology has made this process significantly easier with many widely available image-editing packages such as Adobe PhotoShop or the Gnu Image Manipulation Package (GIMP) which provide tools such as erasing, blurring and clone brushing. These manual techniques still require skillful application however and bring their own set of problems with them. Recent developments in the field of image manipulation and restoration have focused on automating the image completion process. The particular technique this paper implements is called fragment-based image completion and is based on the work of Drori, Cohen-Or and Yeshuran (2003). Their general algorithm is used in the development of an automatic image completion plug-in for the GIMP.

While we describe a re-implemention of the techniques described in the original paper, the approach is novel because of its incorporation into the GIMP. Although there will inevitably be some differences in the results achieved due to the choice of implementation language, development environment and coding style, the general claims of the authors can be verified and the algorithm tested to see whether it works in all the scenarios that the original authors claim it does.

This paper outlines various techniques related to the field of image completion and gives an overview of the fragment-based image completion algorithm. This is followed by a detailed description of how steps in this algorithm have been implemented as a plug-in for the GIMP. The paper concludes with an overview of results and outline of the forthcoming steps.

1.1 Related Work

The broad influences on the work of DRORI et al. (2003) are those of texture synthesis, image inpainting, the recent combination of these two methods, related types of image manipulation and some of the more purely mathematical models of predicting completion fields.

Texture Synthesis generates a new texture of potentially unlimited size that is perceived to be the same texture as an original. Approaches include a technique of incrementally building up new texture based on similar neighbourhoods in the sample texture (WEI and LEVOY 2000), using a given mask to replace regions with synthesized texture (IGEHY and PEREIRA 1997), producing new texture by stitching together blocks of example texture (EFROS and FREEMAN 2001), producing new textures by means of pyramid-based texture analysis (HEEDER and BERGEN 1995), and creating fundamentally new textures by means of applying editing operations to the original texture (BROOKS and DODGSON 2002). Some of these methods are more suited to reproducing stochastic (random) textures (HEEDER and BERGEN 1995; IGEHY and PEREIRA 1997) while others work better for more structured, deterministic textures (EFROS and FREEMAN 2001).

Image inpainting is based on the way professional art restorators retouch paintings. Inpainting algorithms aim to smoothly propagate the information on the boundary areas of the selection inwards to fill the gap (BERTALMIO et al. 2000). This reproduces the structure of the region but does not take into account the texture of the region at all. The simultaneous filling in of texture and structure into missing regions has also recently been explored (BERTALMIO et al. 2004).

Closely related to the topic of texture synthesis and transfer are other techniques such as image analogies (HERTZMANN et al. 2001) and transferring colour to greyscale images (WELSH et al. 2002). Common concepts utilized by these techniques and the fragment-based image completion algorithm are their use of pyramids and the methods of getting neighbourhood samples in order to manipulate target pixels.

Other relevant investigations into image completion include those which focus on shape and curve completions across gaps (SHARON et al. 2000; WILLIAMS and JACOBS 1997) and those which explicitly relate shape completion to image restoration (HIRANI and TOTSUKA 1996).

2 Fragment-Based Image Completion

The fragment-based image completion algorithm (DRORI et al. 2003) uses the visible parts of an image as a training set to infer the unknown parts when a portion of the image is removed. Firstly, an inverse matte that contains the entire extracted region is defined by the user. This is used to define a confidence level for each pixel (called the confidence map) with those pixels that are closer to the known regions having higher confidence values. All the confidence values increase during the completion process. An approximation of the low confidence areas is generated using a simple smoothing process known as fast approximation. This rough region is then augmented with familiar details taken from areas of higher confidence. At each step a target fragment, which consists of a circular neighbourhood around the pixel, is completed by adding more detail from an appropriate source fragment which has higher confidence. The source fragments are selected from the immediate vicinity of the unknown region. As the process continues the average confidence of the pixels converges to one, completing the image in a manner which takes both texture and some sense of structure into account.

2.1 Fast Approximation

The fast approximation algorithm involves first multiplying the original image *C* (Figure 1a) by its inverse matte $\bar{\alpha}$ (Figure 1b) to mask off the region to be removed . This produces an image $\bar{C}$ (Figure 1c) where the region of the offending object is replaced by a white gap. The image is then repeatedly blurred, shrunk, expanded and blurred again and then has the known values $\bar{C}$ reintroduced at each iteration as well. This has the effect of making the boundaries of the gap creep inwards until the missing region is filled with roughly the same colours as appeared on its borders (Figure 1d and e). The boundaries are considered to have converged sufficiently when the convergence error dips below a certain threshold called the convergence maximum. The convergence error is calculated on each iteration of the loop by taking the average difference between the image before the blurring process and after.

(a) (b) (c) (d) (e)

Figure 1: (a) Original Image. (b) Inverse Matte $\bar{\alpha}$. (c) Image $\bar{C}$. (d) Image $\bar{C}$ after two blur iterations. (e) Final result after fast approximation phase with blur radius set to 9 pixels.

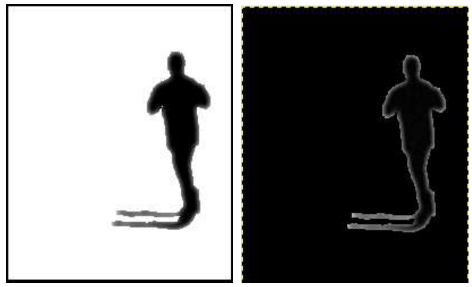
2.2 Confidence Map and Calculating candidate regions

The second phase of the algorithm is to use the inverse matte $\bar{\alpha}$ to calculate a confidence map β of the image to define the level of certainty with which the colour value of each pixel *i* is known. The confidence map is calculated according to the following formula (DRORI et al. 2003):

$$\beta_i = \begin{cases} 1 & if \bar{\alpha} = 1 \\ \sum_{j \in N(i)} g_j \bar{\alpha_j^2} & otherwise \end{cases} \tag{1}$$

Here *N* is the size of the neighbourhood region around the pixel and *g* is a Gaussian falloff function which decreases the level of confidence toward the centre of the unknown region. Figure 2a shows the initial confidence map β that is calculated. During the completion process the black areas gradually turn to white as the confidence in those pixel values increases.

This confidence map β is then passed as input to the function which calculates the set of candidate positions v to use in the search phase of the algorithm. The pixel with the maximum value in the candidate map is then concluded to be the next most appropriate pixel, *T*, to be used in the search phase. Figure 2b shows the initial map of candidate postions.



(a) (b)

Figure 2: (a) Confidence map. (b) Candidate Position Map.

2.3 Searching

The search function takes the target pixel T and iterates through each pixel in the vicinity to find an appropriate source pixel S to use in the compositing phase of the algorithm where the detail from the known region is used to fill in the unknown regions. At this stage the pixel values used at the specified coordinates T and S are those from the image obtained in the fast approximation step. For each potential source pixel S, the difference d between the colour of T (in the unknown region) and S (in the known region) is used together with the difference between the values in the confidence map at the corresponding coordinates according to the following formula:

$$r = \min \sum_{s=S_r(i),t=T(i),i\in N} \left(d\left(s,t\right)\beta_s\beta_t + \left(\beta_t - \beta_s\right)\beta_t\right) \tag{2}$$

According to the original authors (DRORI et al. 2003) this formula finds the pixels which have a higher confidence value in the source than in the target regions and correspond well in terms of colour.

3 Implementation in GIMP

The plug-in is being developed to work in GIMP 2.0. Version 1.3.3 of the freely available plug-in templates from the GIMP Developers' website (http://developer.gimp.org/plug-in-template.html) is used as a base for the implementation. The template is an 'empty' plug-in which just defines the structure needed.

3.1 Fast Approximation implementation

The first phase of the implementation of the fragment-based image completion algorithm is to implement the fast approximation step. Here, advantage is taken of the GIMP concept of a *drawable*. A drawable is anything in an image that can be drawn on with the GIMP painting tools. These include layers, channels, layer masks and selection masks. Having different layers in an image allow that one image to be constructed of several conceptual parts that can be manipulated independently of other parts of the image. In GIMP, layers form a stack with the bottom of the stack being the background. Each layer can have its own attributes including its level of transparency which determines how much of the layer below it is visible. In order to check the results of each phase of the fast approximation, each successive step in the image manipulation is added as a new layer to the original image (Figure 3). This allows the accuracy of intermediate image manipulations to be checked. The visibility of each layer and channel can be turned on or off to view the stages independently or their cumulative effect.

In order to perform some of the common operations necessary for this step, such as the multiplication or addition of images, seperate subroutines are defined that utilise some of the build-in GIMP functions. Some of these built-in functions, such as those which get or set the value of individual pixels, are very computationally expensive however and this has contributed to the relatively slow processing time.

Figure 3: Channels and Layers dialogs show ing how each processing phase can be viewed separately.

3.2 Confidence map and Candidate region map implementation

The confidence map formula is implemented by first checking the value of each pixel in the inverse matte. If the pixel is white (corresponding to 1 in Eq. 1) then it is a known pixel and the corresponding confidence map value is also white. The unknown regions are black in the inverse matte $\bar{\alpha}$ and so the formula is applied there to calculate the corresponding confidence map values. The confidence map (Figure 2a) is added as a new channel to the image instead of as another layer. This is because it represents more of a selection mask that defines how other layers will be manipulated rather than a separate layer which itself needs to be changed

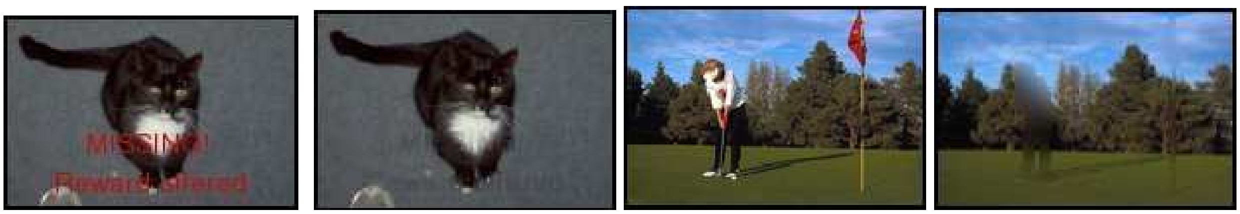
The candidate pixel map v (Figure 2b) is also added to the image as a new channel. The chosen pixel's coordinates are passed as parameters to the function which performs the search of the rest of the image looking for an appropriate source region to sample detail from.

3.3 Search implementation

The way that the search algorithm (Eq. 2) is implemented in the plug-in is a slight simplification from the technique used in the original paper which searched not only over all pixels x and y, but also over five size scales and eight orientations of the image and also compares luminance values as well as colour values of T and S. In order to minimize the processing time for the image, these refinements have not yet been implemented in the plug-in code. Upon visual inspection however, the resulting source pixel S chosen via the simplified search function still seems to be an appropriate match. More testing is needed to confirm this however.

4 Results

The first two phases of the process are complete and the simplified version of the search algorithm has been implemented. Figure 4 shows the resulting images after the fast approximation phase. As the fast approximation phase works in a similar fashion to inpainting techniques it is not surprising that the 'best' results are for filling in narrow regions where structure is of little concern, such as with the writing on the picture of the cat (Figure 4a). The picture of the golfer (Figure 4b) shows more clearly how the neighbourhood colours blur inwards to fill the gap. It is these approximate colours that are used in the search function to select matching source regions to pick detail from.



(a) (b)

Figure 4: Results after Fast Approximation Phase

While the fast approximation procedure produces good results for smooth regions, highly textured areas such as the trees surrounding the golfer, are unconvincing. This is dealt with in the search and composite steps (still to be implemented) which map details from the textured regions to the unknown areas. The results achieved so far are comparable to those achieved by the original authors of the paper and indicate that the fragment-based image completion algorithm is indeed suited to realistic reconstruction.

The processing time for each image (of size approximately 150 x 100 pixels) is several minutes but will be significantly reduced once the final stages have been implemented and the code can be optimized.

5 Future Work

The next stage of the process is the implementation of the composition step which takes the detail from the selected source regions and incorporates it into the unknown target regions. This will be handled using some of the standard GIMP toolbox functions such as copy and paste in conjunction with methods to feather the selected regions and take alpha values into account in order to make the newly composited region blend in seamlessly with the rest of the background.

Verification of results will entail testing the plug-in on images with a variety of different textures, stuctures and sizes for the missing regions.

6 Conclusion

While plug-in development in general for the GIMP is relatively easy due to the availability of templates and documentation regarding GIMP-specific functions, certain common operations such as multiplying images have proved complex to implement because of the need to iterate through each pixel. While it is not possible to re-implement the original paper exactly within the GIMP environment due to the function differences and extra processing required for certain operations, this paper discusses a plausibly similar implementation which takes advantage of some of the built-in GIMP features that are used to approximate the intentions of the original paper. Results can only be verified properly once the final stages of implemention are complete.

References

BERTALMIO, MARCELO, G. SAPIRO, V. CASELLES and C. BALLESTER (2000). Image inpainting. In *ACM SIGGRAPH* (2000), pp. 417–424.

BERTALMIO, MARCELO, L. VESE, G. SAPIRO and S. OSHER (2004). Simultaneous structure and texture image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition* (2004).

BROOKS, STEPHEN and N. DODGSON (2002). Self-similarty based texture editing. In *ACM Transactions on Graphics* (2002), pp. 653–656.

DRORI, IDDO, D. COHEN-OR and H. YESHURUN (2003). Fragment-based image completion. In *ACM SIGGRAPH* (2003), pp. 303–312.

EFROS, ALEXEI and W. FREEMAN (2001). Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH* (2001), pp. 341–346.

HEEDER, DAVID and J. BERGEN (1995). Pyramid-based texture analysis/synthesis. In *ACM SIGGRAPH* (1995), pp. 229–338.

HERTZMANN, AARON, C. JACOBS, N. OLIVER, B. CURLESS and D. SALESIN (2001). Image analogies. In *ACM SIGGRAPH* (2001), pp. 327–340.

HIRANI, ANIL and T. TOTSUKA (1996). Combining frequency and spacial domain information for fast interactive image noise removal. In *ACM SIGGRAPH* (1996), pp. 269–276.

IGEHY, HOMAN and L. PEREIRA (1997). Image replacement through texture synthesis. In *IEEE international conference on Image processing* (1997), pp. 186–189.

SHARON, EITAN, A. BRANDT and R. BASRI (2000). Completion energies and scale. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000), pp. 1117–1131.

WEI, LI-YI and M. LEVOY (2000). Fast texture synthesis using tree-structured vector quantization. In *ACM SIGGRAPH* (2000).

WELSH, TOMIHISA, M. ASHIKHMIN and K. MUELLER (2002). Transferring color to greyscale images. In *ACM Transactions on Graphics* (2002), pp. 277–280.

WILLIAMS, LANCE and D. JACOBS (1997). Stochastic completion fields: A neural model of illusory contour shape and salience. In *Neural Computation* (1997), pp. 837–858.