Poisson Blended Exemplar-based Texture Completion

Hoang M. Nguyen **Burkhard C. Wünsche** **Patrice Delmas**
Christof Lutteroth

Computer Science Department
The University of Auckland,
Email: justin.nguyen@auckland.ac.nz
{burkhard, p.delmas, lutteroth}@cs.auckland.ac.nz

Abstract

Image inpainting is the process of correcting undesirable changes to an image in an unobtrusive way. The existing literature in this research field describes predominantly techniques designed to correct narrow missing regions, which thus often produce undesirable results when the damaged region is large.

This paper presents a novel exemplar-based image inpainting technique for automatic filling-in missing region of an image. Our solution offers two major improvements compared to existing techniques. Patches for filling in missing regions are identified using an appearance space vector, which not only encodes pixel colours, but also colour gradients, feature distances and other measures for computing image similarity. In order to speed up the search for a matching patch we use a Principal Component Analysis to reduce the size of a feature vector used for patch comparison.

The second major improvement is the technique used combine patches filling in a missing region. In order to avoid visible seams we use a Poisson-guided interpolation to blend patches.

Our evaluation and comparison with existing techniques demonstrates significantly improved performance for inpainting missing image regions.

Keywords: texture inpainting, texture reconstruction, image information recovery

1 Introduction

The problem of modifying an image to revert deterioration in a non-detectable way has long been an intensive research field in computer graphics. Generally, two classes of image inpainting techniques have been explored and studied: *pixel-based* and *exemplar-based* texture inpainting. Pixel-based inpainting methods attempt to reconstruct missing or damaged regions one pixel at a time. This class of methods is often fast and produces good results for small regions. *Pixel-based* approaches work by propagating pixel values along contours of equal luminance and computing the value of a "missing" pixel based on its surrounding "good" pixels. The method tends to produce blurred outputs for larger regions.

Exemplar-based inpainting attempts to construct the damaged regions by searching for the best-fitting patches and copying them over to the missing region. These methods are often not as efficient as pixel-based techniques due to the time consuming process of finding best-fitting patches. However, they are most suitable for dealing with larger missing regions. Efficiently and accurately determining the patch best fitting a missing region is one of the key problems of this class of methods. Furthermore, in most applications it is not possible to fill a missing region with a single patch. Hence multiple patches must be copied, which results in partial overlaps and consequently visible seams along patch boundaries.

In this paper, we present a new Poisson-exemplar-based method for inpainting a missing region in an image. In order to improve the efficiency and accuracy of the best-fit patch finding stage, we forgo the conventional *Sum of Squared Differences (SSD)* score technique and employ so-called appearance space attributes to help with this task. For each pixel, the appearance space attribute contains not only the RGB color value, but also its signed feature distance, gradient in both directions and *HSB* values. This provides far more accurate information about each pixel and its neighbourhood, and hence makes it possible to find better matching image regions. Selected patches are fused and blended together using a Poisson interpolation technique, significantly reducing visible seams.

The remainder of this paper is organised as follows: After a brief discussion on the state-of-the-art of image inpainting in section 2, we describe our inpainting algorithm in section 3. Section 4 presents some of our results. Section 5 concludes our paper.

2 Related Work

An analysis of the literature reveals two key classes of algorithms for image in-painting. The first group of methods approaches the problem of texture inpainting from a pixel-based perspective. These methods reconstruct a missing region by computing color values for each pixel one at a time starting from the missing region's boundary and processing inward until the entire region is filled.

Exemplar-based techniques fill missing regions by searching for patches matching the region boundary and inserting these texture patches such that discontinuities with the valid image region are minimised. Both approaches are able to produce high-quality results.

The arguably best known and successful algorithm amongst pixel-based inpainting methods was proposed by Bertalmio *et al.* (2000). The authors attempt to replicate the manual inpainting by propagating the known color values into a missing region along so called isophotes, representing the smallest spatial change of color values and structures.

Drori *et al.* (2003) present a similar approach using

Copyright © 2014, Australian Computer Society, Inc. This paper appeared at the Thirty-Seventh Australasian Computer Science Conference, ACSC 2014, Auckland, New Zealand, January 2014. Conferences in Research and Practice in Information Technology, Vol. 147. Bruce H. Thomas and David Parry, Eds. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

adaptive circular fragments to operate on different scales to capture both global and local structures and approximating the missing region.

Ignecio *et al.* (2004) present an inpainting technique based on a fast marching method for level set applications. The method is simple and considerably more efficient than other pixel-base methods.

As pixel-based methods synthesize texture information of a pixel by examining only its neighborhood information, these methods only yield good results for small and narrow missing regions. For larger holes, the reconstructed image regions tend to be blurry and visually obtrusive.

Exemplar based methods are hence becoming increasingly popular for generating large missing texture patches.

Criminisi *et al.* (2003) propose a method that reconstructs missing texture regions by iteratively selecting a "best-fit" rectangular patch and copy it over to the target region. The order in which boundary pixels of the missing region are processed is based on the amount of information available for that pixel and whether it has any prominent features.

In this paper we present a new algorithm, which uses a similar patch search and insertion concept, but offers two key improvements. First, instead of using only pixel colours for the patch finding process, we employ an appearance space which encapsulates much more information. Second, Criminisi's method does not handle seams along patches. Inevitably, their results often look unrealistic. In contrast, our method smoothly fuses patches together to remove all visible seams.

Cheng *et al.* (2005) updated the priority equation of Criminisi *et al.* (2003) and made it adjustable to the structural and textural information specific to an image. Ignecio *et al.* (2007) extended the concept of Criminisi's method and applied it in the wavelet domain. Their method computes the fill-priority by first transforming the image and the provided binary mask and then use wavelet coefficients and a similarly defined priority to define the fill-order.

3 Algorithm

3.1 Fundamentals

In order to facilitate understanding of our technique and comparison with alternative techniques we adopt the notation used in the image inpainting literature. Let Ω indicate the *target region* to be inpainted. Note that there is no restriction imposed on the topology of Ω. Let $\delta\Omega$ denote the boundary of the target region. The boundary is sometimes referred to as *"fill front"* since this contour evolves inward as the algorithm progresses. The *source region* is denoted as Φ, which in our algorithm remains unchanged throughout the processes. Let Ψ_p be a window centered at the point p.

The main principle behind exemplar-based methods is simple. As with all exemplar-based texture inpainting methods (e.g. Efros *et al.* (1999), Criminisi *et al.* (2003)), the size of the template patch (window) must be specified in advance. In our algorithm, the default patch size is empirically set to 11×11 (refer to (Nguyen *et al.* 2013) for a more detailed analysis of the algorithm parameters). To synthesize the missing region, the following procedure is repeated until all pixels are filled.

For a given pixel p on $\delta\Omega$, find a patch Ψ_q where $q \in \Phi$ such that Ψ_q is most similar to those parts that are already filled in Ψ_p. The missing texture information is then transfered from Ψ_q to Ψ_p. Figure 1 illustrates this process.

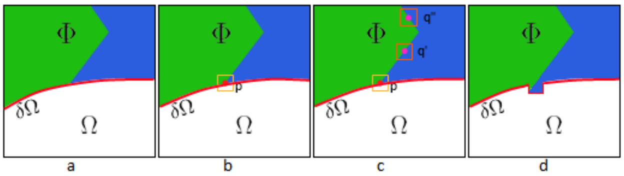

Figure 1: Exemplar-based texture inpainting. a) The original input image with the source region Φ, the target region Ω and the boundary $\delta\Omega$. b) Attempting to reconstruct an area around pixel p. c) Several likely candidate matches are found in the source region. d) The content of the best patch is copied over, resulting in a partial filling of Ω (adapted from Criminisi *et al.* (2003)).

The filling order is critical for inpainting techniques in general, and even more so for non-parametric texture synthesis. Traditionally, the most well-known method has been *"onion peel"*, where the inpainted region is synthesised in concentric layers inwardly (Criminisi *et al.* 2003). Therefore, in our method we iteratively shrink the gap of the inpainted region by continuously transferring colours from source regions to patches centered at boundary pixels.

3.2 Determining the Filling-Order

Given a set of boundary pixels, the objective is to determine the order or priority of the pixels to be processed. This task is accomplished as followed. For each boundary pixel p, let Ψ_p be a patch centered around p. The priority of p is defined as by Criminisi *et al.* (2003):

$$Priority(p) = Confidence(p) * Data(p) \quad (1)$$

The *confidence term*, which quantifies the amount of reliable information in the pixel's neighborhood, is defined as:

$$Confidence(p) = \frac{\sum_{q \in \Psi_p \cap \Omega} Confidence(q)}{|\Psi_p|} \quad (2)$$

where $|\Psi_p|$ is the area of the patch Ψ_p and Ω denotes the target region to be inpainted. The function $Confidence(q)$ returns 1 if q is already filled and 0 otherwise. The confidence term aims to boost the priorities of patches that have more already-filled pixels, allowing them to be synthesized first.

The *data term*, which defines the strength of the isophotes arriving at the boundary, is defined as:

$$Data(p) = \frac{|\bigtriangledown I_p^{\perp} \cdot n_p|}{\alpha} \quad (3)$$

where $\bigtriangledown I_p^{\perp}$ represents a vector that is orthogonal to the gradient vector at p, n_p is the normal at p, and α denotes a normalisation factor ($\alpha = 255$ for RGB-colour images). The purpose of this data term is to find matching patches preserving linear texture features, such as straight lines or curves, and therewith extending the linear features gradually inwards.

The *confidence values* for all boundary pixels are computed and the pixel with the highest confidence value is processed first.

3.3 Candidate Patch Identification

The next task is to search for a patch in the image that retains the highest resemblance to the processed patch. This is achieved by iteratively traversing through each pixel of the image outside the missing region and computing the similarity of the patch centered around that pixel and the original patch. Instead of using the standard *SSD* to measure the similarity of two given patches, we employ *appearance space attributes*, which provide much more information and thus improve the search result.

When searching for a matching patch we consider for each pixel an 11×11 pixel neighbourhood. For each pixel of this neighborhood we consider *RGB* colours, the *gradient vector* as well as the signed Euclidean distance to the closest dominant feature to the original texture. The entire information is encapsulated into an $11 \times 11 \times (3 + 2 + 1) = 726$-dimensional vector.

Determining the similarity of two given patches by comparing two 726-dimensional vectors is not efficient. In order to make the appearance space more practicable, the 726-dimensional vectors are projected into low-dimensional vectors using *principal component analysis (PCA)* (Lefebvre *et al.* 2006, Manke *et al.* 2009). In our method, the dimensionality is reduced to 12, which from our experiments on different types of images produced the best results.

The clear advantage of attribute space over the conventional SSD is that the attribute space approach permits any meaningful information about the pixels and their surrounding to be embedded for matching purposes. By reducing the dimensionality, the computation time can be kept manageable.

3.4 Patch Fusion

The final step is to replicate the content of the candidate patch and smoothly blend it with the target region. We employ a *Poisson-guided interpolation* approach proposed by Perez *et al.* (2003) for this task. The principle behind this is fairly straightforward.

Suppose Ψ_B is the candidate patch to be copied and fused over the target patch Ψ_A, and let ∂_A and ∂_B be the boundaries of the target and candidate patches respectively. The goal is to adjust the colour information of Ψ_B, while preserving the *relative information* (image gradient) as much as possible, so that the transition between the newly modified patch Ψ_C and the rest of the image is gracefully blended. This is accomplished as follows:

First, the values of the boundary pixels of Ψ_C are initialised to be equal to the corresponding values of the boundary pixels of Ψ_A. This is to ensure that the isophotes arriving at the boundary are properly maintained.

$$\Psi_{C_{(x,y)}} = \Psi_{A_{(x,y)}} \qquad \forall (x,y) \in \partial_B \tag{4}$$

Next, each colour channel's value of the remaining interior pixels within Ψ_C are independently adjusted to be consistent with the boundary pixels while constraining the image gradient to be identical to that of Ψ_B.

$$\bigtriangledown C(x,y) = \bigtriangledown B(x,y) \qquad \forall (x,y) \in \Psi_C \backslash \partial_C \tag{5}$$

where $\bigtriangledown(x,y)$ denotes the image gradient at the pixel (x,y), and $\bigtriangledown C(x,y)$ and $\bigtriangledown B(x,y)$ are defined as

$$S_1 = \sum_{(x+\delta x, y+\delta y) \in \Psi_A} C(x+\delta x, y+\delta y) \tag{6}$$

$$S_2 = \sum_{(x+\delta x, y+\delta y) \in \partial_A} A(x+\delta x, y+\delta y) \tag{7}$$

$$\bigtriangledown C(x,y) = |N| \; C(x,y) \; - S_1 \; - S_2 \tag{8}$$

where N is the number of valid pixels. A pixel is considered valid if it is inside the processed patch.

$$\bigtriangledown B(x,y) = \sum_{(x+\delta x, y+\delta y) \in \Psi_A \cup \partial_A} B(x,y) - B(x+\delta x, y+\delta y) \tag{9}$$

δx and δy designate a set of 4-connected neighbours around x and y. The equation 5 can then be expressed in the form of a system of linear equations with $\boldsymbol{i}$ variables ($\boldsymbol{i}$ is the number of pixels in $\Psi_{C_{(x,y)}}$), and can be solved using an *iterative matrix solver* such as the *Jacobi Method*.

4 Evaluation

In this section, we investigate the effect of different algorithm parameters and compare its performance in comparison with popular existing algorithms.

4.1 Evaluation of Appearance Space Attributes

Figure 2: a) The image contains three black square regions where image information was removed in order to fill the regions using image inpainting techniques. b) The original input image. c) Inpainted image using SSD d) Reconstructed image using appearance space attributes.

We have tested different appearance space attributes and found that the best matching patches are found by using a combination of *gradient values*, *signed feature distance* and *RGB* colours. Adding additional information such as *HSB* channels and neighbouhood variance increases the cost and produces no visible improvement. The difference between using standard *SSD* and using *Appearance Space Attribute* is demonstrated in figure 2.

The reconstructed image using SSD has a poor quality. All three missing regions have been filled using patches which do not properly match the boundary of the hole. Reconstructing the image using *appearance space attributes* demonstrates clear improvements. Although there are still some artefacts around the eyebrow region, these will be mended during the blending process.

4.2 Blending versus Non-Blending

We present several examples to demonstrate the effectiveness of our algorithm compared with existing techniques.

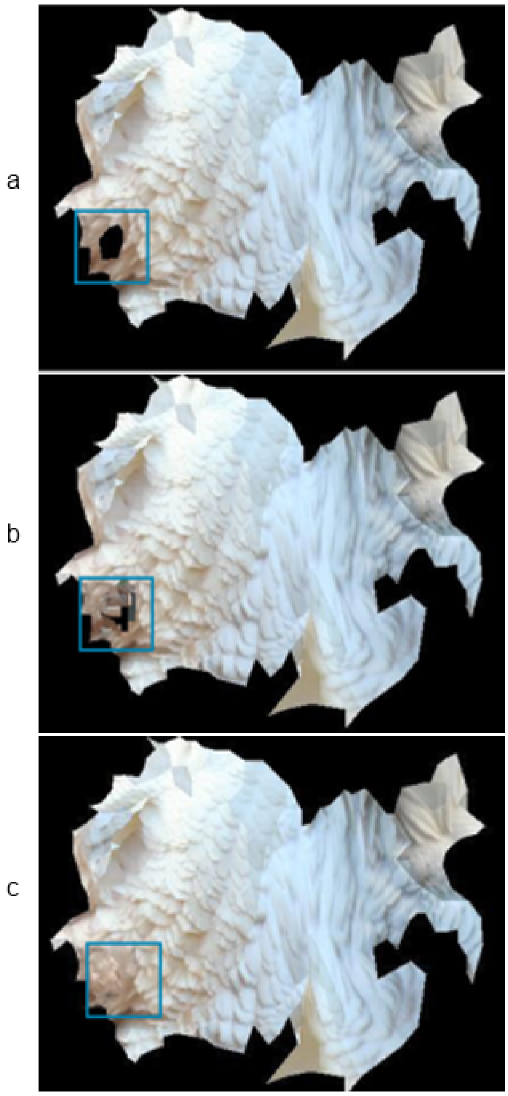

Figure 3: Reconstructed texture with and without blending. a) Input texture. b) Result using the method by Criminisi *et al.* (2003). c) Our result where patches are blended together using Poisson-guided interpolation.

Figure 3 illustrates inpainting results on a part of a texture alas. Note that without blending the overlapping inserted patches have visible seams. Furthermore, not the entire missing region was filled. In contrast, using Poisson-guided interpolation produces a very realistic and complete result.

Figure 4: Inpainted texture without blending (left), and with blending (right).

Figure 4 demonstrates another example of how Poisson-guided interpolation improves the overall inpainting results. Notice how the eyebrow now appears more natural.

4.3 Evaluation against Other Inpainting Methods

In this section, we evaluate our method against some of the best known texture inpainting methods described in the literature.

Figure 5 shows an example in which spots of a cheetah are removed using several well-known inpainting methods.

Ignecio *et al.* (2004)'s method is the most efficient. It took approximately 45 seconds to accomplish the task. However, the result is unsatisfoctory. Most remnants of the cheetah's spot are still evident. Additionally, as some of the spots are relatively large, the inpainted image regions appear blurry.

Bertalmio *et al.* (2000)'s method takes a little more time to process but generate a better result (57 seconds). However, as with Alexandru's method, it fails to remove some of the spots completely. For some large gaps, blurry textures can be seen. Additionally, in some cases (for some spots) colours are not propagated correctly resulting in patches with colours not consistent with the regions neighborhood.

The bottom image in figure 5 shows that our method successfully removes all spots with texture information consistent with the surrounding image region. However, there are some small regions at the tail where the algorithm was unable to reconstruct the fur without spots correctly (part of the tail's texture extrudes to the neighbouring area). This is probably due to the fact that the selected window size was relatively large in this case. Overall, our algorithm works well and produces good results compared to other inpainting methods.

Figure 6 presents another example comparing different inpainting methods. Bertalmio *et al.*'s (2003) method was unable to fill the missing region. This is probably due to the fact that intensity values from the source region are not properly propagated inwardly. Traditional exemplar-based techniques such as Criminisi *et al.* (2003) produce fairly good result, although there is a large artefact in the reconstructed region. Ignecio et al.'s (2004) method produced a reasonably good result, although the inpainted region appears very blurry. Some parts of the window in the image are not reconstructed properly. Our algorithm performs well in this test case. Although the inpainted region still exhibits slight blurriness, the overall structure of different scene components has been correctly reconstructed.

5 Conclusion and Future Work

In this paper we have presented a novel image inpainting algorithm for reconstructing large missing texture regions from digital photographs. The results of this inpainting process is a new image in which the deterioration has been "inpainted" and reverted in such a way that few visible traces of it remain.

The basic idea of our approach is to replicate missing textures by searching for "best-fit" texture patches in the source regions and smoothly insert these patches into the missing region to form the final result. The filling-order is determined using pixels' confidence value, which is defined by the amount of information available for that pixel and the image isophotes. This allows our algorithm to propagate both linear and round texture features into the target region.

Our solution offers two major improvements compared to existing techniques. Patches for filling in missing regions are identified using an appearance space vector, which not only encodes colour differences between regions, but also colour gradients, feature distances and other measures for image similarity. In order to speed up the search for a matching patch we use a Principal Component Analysis to reduce the size of a feature vector used for patch comparison. The second major improvement is the use of Poisson-guided interpolation to blend patches.

We have evaluated our method's performance against some of the best known inpainting methods described in the literature and found that our results are superior.

References

Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles and Coloma Ballester (2000), Image Inpainting, *in* 'Proceeding SIGGRAPH '00 Proceedings of the 27th annual conference on Computer graphics and interactive techniques', pp. 417–424.

Iddo Drori, Daniel Cohen-Or and Hezy Yeshurun (2003), Fragment-Based Image Completion, *in* 'ACM Transactions on Graphics', Vol. 22, pp. 417–424.

A. Criminisi, P. Prez and K. Toyama (2003), Object Removal by Exemplar-based Inpainting, *in* 'ACM Transactions on Graphics', pp. 721–728.

Wen-Huang Cheng, Chun-Wei Hsieh, Sheng-Kai Lin, Chia-Wei Wang and Ja-Ling Wu (2005), Robust algorithm for exemplar-based image inpainting, *in* 'In Proceedings of CGIV', pp. 64–69.

Ubirate Ignecio and Cleudio R Jung (2007), Block-based image inpainting in the wavelet domain, *in* 'Visual Computing', pp. 733–741.

Alexandru Telea (2004), Journal of Graphics Tool, *in* 'An image inpainting technique based on the fast marching method', Vol. 9, pp. 23–34.

Patrick Perez, Michel Gangnet and Andrew Blake (2003), ACM TRANSACTIONS ON GRAPHICS, *in* 'Poisson image editing', Vol. 9, pp. 23–34.

Alexei Efros and Thomas Leung (1999), In Proceeding of ICCV, *in* 'Texture synthesis by non-parametric sampling', pp. 1033–1038.

Felix Manke and Burkhard Wunsche (2009), Image and Vision Computing New Zealand 2009. IVCNZ 09, *in* 'Analysis of appearance space attributes for texture synthesis and morphing', pp. 85–90.

Sylvain Lefebvre and Hugues Hoppe (2006), ACM SIGGRAPH 2006 Papers, *in* 'Appearance-space texture synthesis,', pp. 541–548.

Hoang Minh Nguyen, Burkhard Wunsche, Patrice Delmas and Christof Lutteroth (2013), Image and Vision Computing New Zealand 2013. IVCNZ 13, *in* 'Enhanced Patch-Based Image Inpainting with Poison Interpolation', 2013 [Accepted for publication].

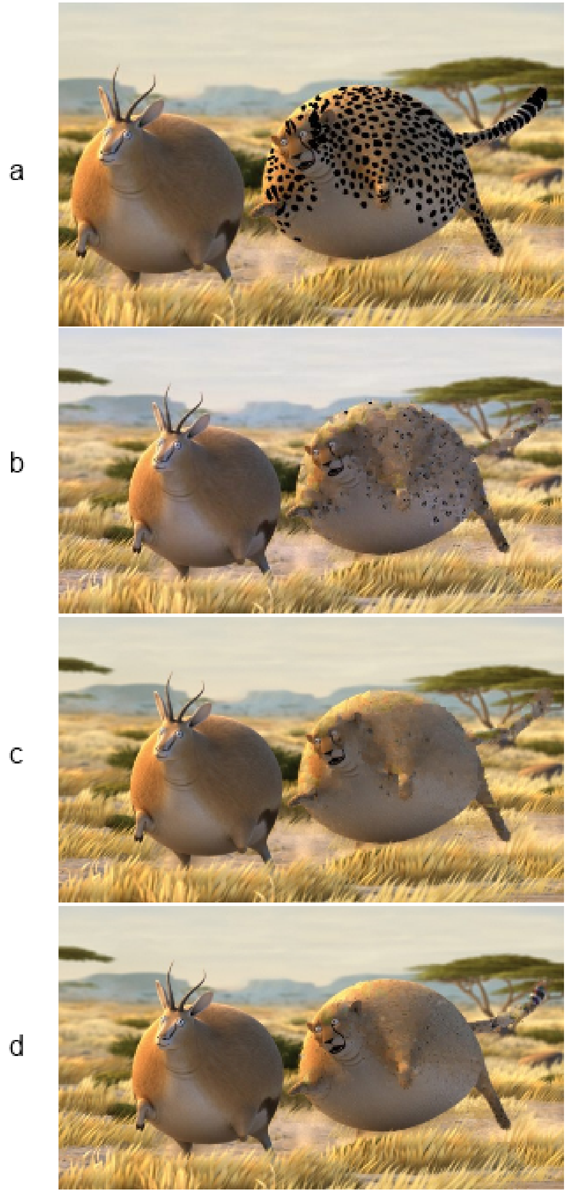

Figure 5: A scene from the animation series - “Rolling Safari”. a) The original input image. All black spots are considered missing regions and we employ image inpainting techniques in order to fill the black spots with colour information consistent with the remaining fur colour of the cheetah. b) The inpainted result using Alexandru’s method Ignecio *et al.* (2004). c) Result obtained with Bertalmio *et al.* (2000) method. d) Result obtained using our image inpainting technique.

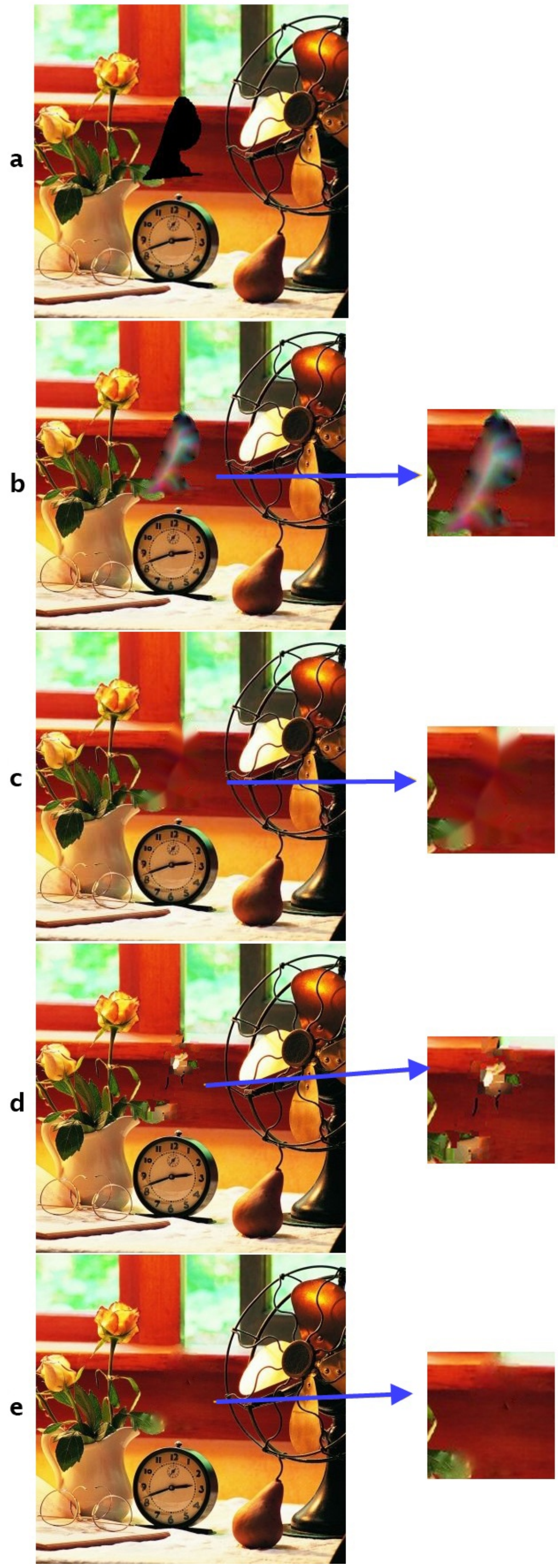

Figure 6: a) The input image. Image inpainting results obtained using the algorithms from: (b) (Bertalmio et al. 2000), (c) (Telea et al. 2004), (d) (Criminisi et al. 2003) and (e) our method.