Efficient Editing of Aged Object Textures

Olivier Clément* Jocelyn Benoit† Eric Paquette‡

Multimedia lab
École de technologie supérieure
Montreal, Canada


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

Figure 1: *Example of removing (b), reproducing (c) and combining (e) aging effects from the source images (a)(d).*

Abstract

Real objects present an enormous amount of detail, including aging effects. Artists need an intuitive control when they iteratively review and redesign their work to achieve a specific aging effect pattern but physically based and empirical simulations rarely provide an appropriate control. Our motivation comes from simplifying the redesign step by providing appropriate tools. In our system the user interactively identifies aging effects in a source image or photograph. The user then designs a target aging mask presenting the wanted aging effects pattern. Our system then synthesizes the output texture within a few seconds using a texture synthesis approach adapted to aged object texture editing. Thus, the user can quickly redesign the aging mask to achieve better results or test new configurations.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

Keywords: realistic rendering, weathering, aging effects, blemishes, deterioration, texture, texture synthesis

1 Introduction

Whether it is for a better immersion in a virtual reality environment, for impressive special effects or for highly realistic video games, computer graphics scenes need to present an increasing amount of detail. In this quest for realistic image synthesis, many aspects have to be considered such as weathering or aging of synthetic objects as can be seen in Figure 1. The problem in adding aging effects is that there are many such effects and they need to be added on a multitude of objects, making it difficult for production studios to keep up with such a labor-intensive task. Thus, artists need to have tools to semi-automatically add aging effects. One way to add such effects is by using physically based simulation. Such simulation methods often result in objects of impressive realism but at the cost of long computation times and non-intuitive physical parameters that are hard to understand for artists or non-specialists. Other methods take an empirical approach to creating aging effects. Such methods are often easier to control by artists but still require the artists to learn how to control the different parameters. Furthermore, it is difficult to control physically based simulations and empirical approaches to create aging effects that match the desired shape and texture.

Artists often create synthetic objects in a review/redesign loop where the results are reviewed (by the artist himself, a lead artist, an artistic director, *etc.*) and the artist redesigns the synthetic objects in an iterative manner. Artistic goals and constraints often require the artist to modify the appearance of an object many times before it is adequate. This would require changing parameters of a physically based simulation or an empirical method to hopefully get the desired result. An easier control and tools closer to the ones used in production studios are needed.

The traditional path to adding details to objects is through texture mapping. An artist can find a photograph of an object aged in a way similar to the wanted result. But still, as for physically based simulation and empirical approaches, the aging effects of the texture often will not match the desired pattern. Thus, an artist has to manually redesign the texture several times in a review/redesign loop. This redesign of the texture is time-consuming since aging effects often present patterns and textures that are non-trivial to reproduce by hand (see Figures 1 and 11). Texture synthesis methods can be used to ease the reproduction of the aging effects. Most texture synthesis algorithms are amenable to hole-filling but many acceleration methods are based on the particularity of using an L-shaped context and filling the whole image in scan-line order. In the context of hole-filling, this produces discontinuities on the boundary of the region to fill.

Our approach is based on eliminating the need to learn aging methods with their parameters and adapting texture synthesis approaches to the reproduction of aging effects. Photographs are used as input to capture textures of the desired aging effects. The artist identifies

---

*e-mail: Olivier.Clement.1@ens.etsmtl.ca

†e-mail: Jocelyn.Benoit.1@ens.etsmtl.ca

‡e-mail: Eric.Paquette@etsmtl.ca

the aged and non-aged regions of the surface in a semi-automatic interactive system. The artist can then change the location of the aging effects in the texture while the system fills in the missing aged or non-aged texture regions with a texture synthesis approach. This allows adding, modifying, and removing aging effects to achieve the desired artistic effect. The system even allows the user to combine aging effects from many images onto a separate image, as can be seen in Figure 1. The redesign phase of the review/redesign loop only requires that the artist edits a mask specifying where the texture should be aged/non-aged. This can be done by the artist in a matter of a few minutes in an image editing tool. Finally, since the system is interactive and results in a texture that can be used in the traditional production pipeline, it is appropriate and efficient for artists. The main contributions of our work can be summarized as follows:

- A framework to reduce the amount of work needed in the redesign iterations.
- An interactive system to identify features of the source image and synthesize the novel aging texture.
- An adapted texture synthesis approach to take into account the constraints of synthesizing aged and non-aged texture based on the goals specified in a mask image.
- An efficient method to handle hole-filling texture synthesis designed to reduce discontinuities and preserve transitions between aged and non-aged regions.
- A process to transfer many aging effects from different images onto a seperate image.

2 Previous Work

Our work is in the area of aging effects reproduction and simulation but also has tight links to texture synthesis. This section thus first presents an overview of aging methods with a selection of representative works. It then describes image based aging methods and it concludes with a survey of relevant texture synthesis methods.

2.1 Aging Methods

As stated in the previous section, there are two major approaches to adding aging effects to virtual objects: physically based simulation and empirical approaches. Physically based methods have been developed to address various effects such as deterioration caused by rain and other such flow of water on objects [Dorsey and Hanrahan 2000], weathering of stones [Dorsey and Hanrahan 2000], corrosion of metal [Merillou et al. 2001], and fracture of objects [O'Brien et al. 2002]. These approaches often result in highly realistic results but require lengthy calculations, require the user to enter non-intuitive physical parameters, and do not provide the artistic control required by production studio environments. Like physically based methods, many empirical methods have been developed to address effects such as "stain-bleeding" [Chen et al. 2005], surface cracks [Gobron and Chiba 2001], and peeling of paint [Paquette et al. 2002]. These methods usually provide more intuitive parameters but do not necessarily allow the kind of control required by artists. It is like trying to fracture an object in a very specific pattern by trial and error. Even for someone who has a good understanding of the underlying process, it is difficult to achieve the wanted pattern.

Furthermore, physically based as well as empirical methods typically target a single aging effect. Few works address the need to add many aging effects at the same time. The method of Wong *et al.* [1997] uses geometric properties of the surface and user defined aging sources to determine the regions where the object will be more or less deteriorated. They present results with dust, peeling and patinas. The main problem of this method is controlling the shape and location of the aging effects. The user only has an indirect control of where the aging effect occurs through aging sources that are used to compute a probability on the surface of the object. Cutler *et al.* [2002] present a general framework where not only the surface is aged, but the interior of the object is modeled and deteriorated. By combining aging effects such as fractures, dirt and erosion while aging the surface and the interior of objects, they show results of an increased level of detail and realism. Nevertheless, the control over the shape, size, and location of the aging effects is even more difficult than in the work of Wong *et al.*

2.2 Image Based Aging Methods

Recently, approaches using image capture have been proposed. Gu *et al.* [2006] capture a 7D Time Space Variant BRDF using a sophisticated dome composed of 16 cameras and 150 light sources. With this setup, they are able to capture both the texture, reflection, and time-varying aspects of the material. Then, through a factorization, the appearance and time-varying aspects are separated so that novel patterns can be synthesized. Wang *et al.* [2006] capture the appearance of an aging effect at a single point in time and infer the evolution of the aging effect from more to less aged regions in the captured data. From that, they separate the time variant and texture data in order to synthesize new weathered textures and also evolve these textures through time in a manner coherent with the captured data. These two works are close to ours. They generate a sequence of textures representing the aging effect evolving through time while we generate textures that are fixed in time. Another main difference is the simplicity of our capture process: a single photograph of the aging effect compared to high dimensional BRDF acquisition. Finally, we do not rely on a phenomenon that evolves rapidly (as the method of Gu *et al.*) or an evolving phenomenon forming an appearance manifold (as the method of Wang *et al.*). The appearance manifold eases the automatic segmentation of the degree of aging of the object but cannot handle aging effects such as shoe marks or graffitis that our method handles (see Figure 11). The capture process in the dome setup is also impractical for such effects.

2.3 Texture Synthesis

Our approach uses the texture synthesis method of Efros and Leung [1999]. In their work, they present texture synthesis based on finding the best match of a texture window context between the synthesis texture and the input texture. Even though simple, this texture synthesis approach is quite effective. Our technique is not limited to this texture synthesis approach as we use an improved search method [Arya et al. 1998] and we could benefit from the improved synthesis quality and reduced computation times of other texture synthesis approaches such as the method of Lefebvre and Hoppe [2006] or the method of Liang *et al.* [2001].

Some texture synthesis approaches allow for some control over the resulting texture. Improving from the base texture synthesis approach, Image Quilting [Efros and Freeman 2001] allows texture transfer by reproducing image intensity of an input image using the texture of another image. Brooks and Dodgson [2002] globally edit the texture features, allowing global modification with minimal user interaction but not providing the local control needed in our application. In the method of Ashikhmin [2001], the user can quickly draw the high level appearance of the texture and the system bases its patch based synthesis on this information. One major difficulty of this approach is that it relies on a high frequency irregular texture to hide seams between patches of the input image that are copied to the output texture. This is not practical in our aging context where the texture of either the aged or non-aged surface can be regular or low frequency. The graphcut approach of Kwatra *et al.* [2003] reduces the problem of the seams, but requires the user to manually place input texture patches, which is time-consuming and would be part of every redesign iteration. As in our application, object based image editing [Barrett and Cheney 2002] requires rapid feedback and uses texture synthesis to fill the background behind objects. However, the notion of object is not needed in our application while fine control is needed to specify the aging pattern of the resulting texture.

The texture synthesis approach closest to ours is the Image Analogies framework of Hertzmann *et al.* [2001]. The "Texture-by-numbers" and "Interactive editing" applications of Image Analogies are similar to our texture synthesis approach. The main difference is that we use a hole-filling approach with various shapes of neighbourhoods that allows the system to synthesize only specific regions of the output image and to resolve the problem of discontinuities caused by the L-shaped windows (see Section 3.2). Also, the hole-filling approach reduces the computation times and is more accurate than the "Interactive editing" which is based on coherence based search. Finally our approach is dedicated to reproducing aging effects and should be considered as a specialization and an extension of Image Analogies.

Our proposed method distinguishes itself from previous work since it provides an effective system specifically for editing of aged object textures; it does not rely on simulation or empirical aging approaches; it needs only a photograph or image of the effect; it can combine many aging effects from different images onto a seperate image; it focuses on reducing the time needed in each redesign phase; and its texture synthesis approach is efficient and produces good results in the context of aged object textures.

3 Aged Texture Automated Editing

The goal of our method is to edit aging effects. To do this, the system uses a *source image* (a photograph or any other image) that contains aging effects and a *target aging mask*. The mask is a binary image representing the pattern of the desired aging effects. The user can create the target aging mask in the image editing software of is choice or directly in our interactive application. The result of the aging editing process is the *reproduction image*: the source image without the initial aging effects not identified in the target aging mask and with new aging effects identified in the target aging mask. This process is summarized in Figure 2 and in the video sequence[1] provided with this paper.

Editing the aging effects is done in two phases: elimination and reproduction (see Figure 3). Our system divides the process such that only the reproduction phase is done every review/redesign iteration. The elimination first removes the aging effects. For the system to eliminate aging effects from the source image, it must know which regions contain the aging effects. These regions can be interactively identified in our system by standard image processing segmentation methods. This step results in the creation of a binary image called the *source aging mask*. With the source image and its aging mask, the system is able to eliminate all the aging effects. The result of this elimination phase is the *elimination image*: the source image without the aging effects identified in the source aging mask. Then, in the second part, the system reproduces novel aging effects. To reproduce the aging effects, the system needs to know the regions in which to add the aging effects. The user specifies these regions with the target aging mask. Starting with the elimination image and considering the aging effects pattern of the target aging mask, the system reproduces the aging effects found in the source image and identified by the source aging mask resulting in the reproduction image. To do that, the system uses a hole-filling approach that produces more realistic aging effect boundaries than standard scan-line approaches.

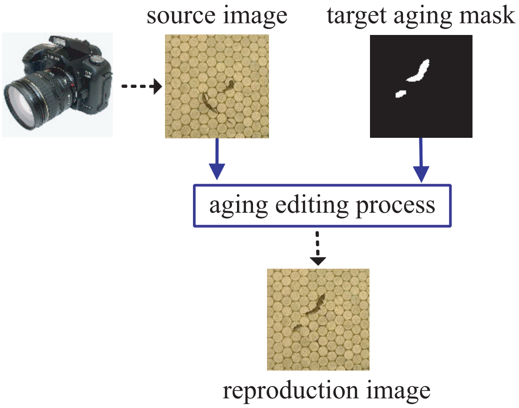

Figure 2: *Overview of the aging editing process.*

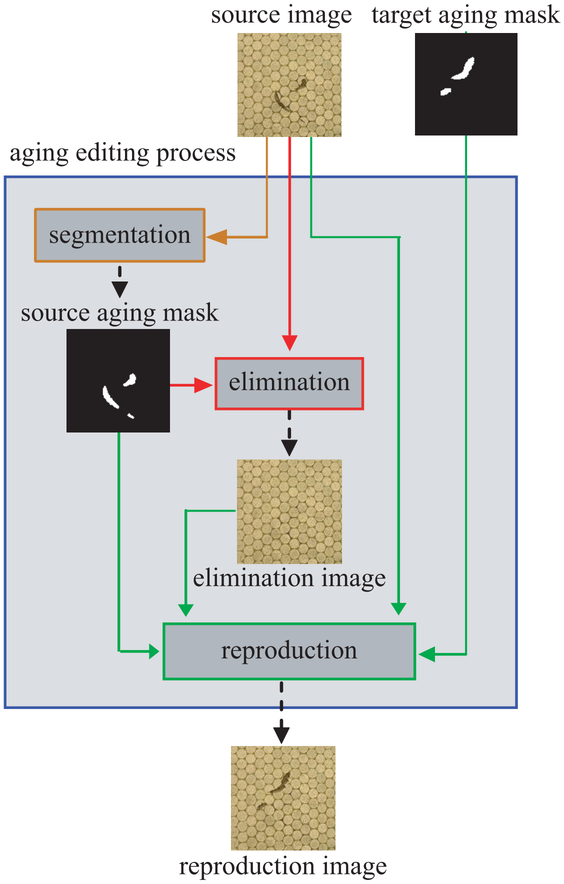

Figure 3: *Overview of the elimination and reproduction.*

---

[1] The video can be found here:
http://profs.logti.etsmtl.ca/paquette/Research/Papers/Clement.2007/

3.1 Interactive Segmentation Phase

The interactive segmentation phase identifies the aged regions of the source image and creates the source aging mask. Like explained in the overview section, the interactive segmentation phase is important; the quality of the results directly depends on the precision of the source aging mask. If the source aging mask does not contain all of the aging effects, the elimination phase will leave unwanted aging effects.

The identification of the aged regions using a binary mask could be done outside of our system using segmentation tools or an image editing software. Nevertheless, we preferred to integrate segmentation tools in our system since it is much easier for the user. The next paragraphs explain the segmentation techniques offered by the system and concludes with insights on which approach proved to be the most efficient for the user.

To create the source aging mask, the system allows the user to segment the aged regions of the source image with many segmentation methods. Moreover, the user can combine different segmentation methods with logical operators such as union, difference and intersection. When required, the user can manually edit the source aging mask directly in our system by using a paint brush and an eraser. Our implementation provides four distinct segmentation methods: grayscale threshold, color threshold (both HSV and RGB) and an energy minimization method. Also, the user can apply morphological operators like erosion and dilation to change the source aging mask. Other methods can simply be added if needed to ease the segmentation phase.

However, the most used method is the energy minimization method. The energy minimization method offered by the system is an easy to use stroke-based technique inspired by the work of Lischinski *et al.* [2006] used to rapidly identify a specific region in an image. With this technique, the user paints strokes on the aging effects and the system automatically computes the segmentation of the corresponding regions. The method assigns to every pixel a label (aged or non-aged) that minimizes an energy function. The system uses the graphcut optimization approach and software of Boykov *et al.* [2001] to compute the minimization. Refer to their paper for a detailed description of this approach. In our system we redefined the data cost and the smooth cost in order to use the technique with image segmentation. For the smooth cost, a Canny edge detection is computed and the cost is proportional to the number of edge pixels in the 8-neighbor around the pixel. This favors discontinuities in the labeling along edges. The data cost is the smallest difference of color between the pixel color and the pixels color identified by the user stroke. This favors color similarities with the stroked pixels in the labeling.

3.2 Elimination Phase

The system uses the output of the segmentation phase to identify and eliminate aging effects present in the source image. To do this, a Markov Random Field texture synthesis technique was developed, inspired by works of Efros and Leung [1999] and Hertzmann *et al.* [2001]. The system performs constrained texture synthesis from the non-aged pixels to replace the aged regions identified in the source aging mask. The result of this is the elimination image: the source image modified such that it contains only non-aged pixels. This section presents the Efros and Leung algorithm [1999], adapted to the interactive aging editing context. The elimination phase is summarized in Algorithm 1.

The outer loop of the algorithm implements a hole-filling approach that allows our system to synthesize only portions of the image. Since the replacement pixel will be found based on the color information of the non-aged pixels, the algorithm must start with the pixels that have some non-aged pixels in their neighbourhood. Thus, the system will first process the boundary of the aged region as shown in Figure 4 (Unfilled boundary). After the boundary pixels have been processed, the algorithm identifies the new boundary and fills it. The system repeats the process until every pixel is replaced.

```
Input: Source Image, Source Aging Mask
Elimination Image ← Source Image
Region To Fill ← GetRegionToFill (Source Aging Mask)
U ← GetUnfilledBoundaryPixels (Region To Fill)
while Region To Fill is not empty do
    foreach pixel P in U do
        R ← FindBestNon-AgedMatch (P, Source Image,
        Source Aging Mask)
        Elimination Image (P) ← Source Image (R)
        Region To Fill (P) ← Filled
    end
    U ← GetUnfilledBoundaryPixels (Region To Fill)
end
return Elimination Image
```

Algorithm 1: Pseudo-code of the elimination phase.

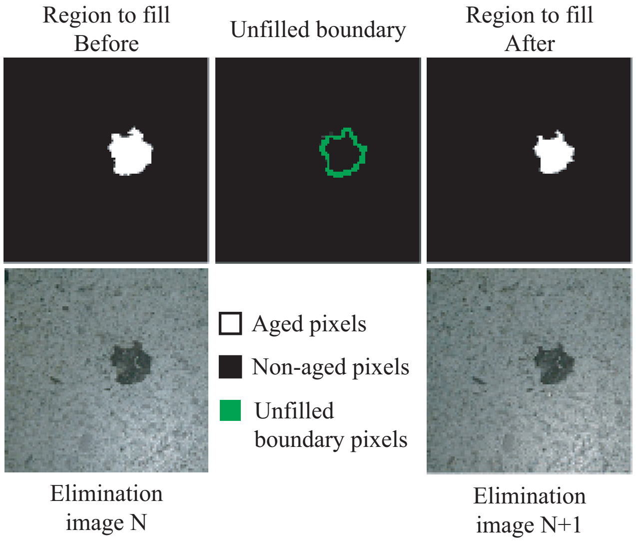

Figure 4: *An iteration of the hole-filling approach.*

From the aged region of iteration N, each unfilled boundary pixel is replaced with an approximation of the best match in the non-aged pixels of the source image. An approximation of the best match is used to reach interactivity. The search is done using the ANN library developed by Arya *et al.* [1998]. This library implements an approximate nearest neighbour searching algorithm based on a kd-tree structure. These search structures are initialized before the system begins the elimination iterations. To find the best matches, the library computes a L_2 norm on a set of features specified when initializing the search structures. Our system uses a feature vector composed of the RGB components of the non-aged pixels of the neighborhood. The size of the feature vector depends on the size of the window around pixel P with which the user wants the system to search for best matches. Thus, the system seeks a replacement pixel that minimizes the following L_2 norm:

$$\sum_{i,j \in W} D(i,j)$$

$$D(i,j) = [S_w(i,j) - R_w(i,j)]^2$$

Where W represents the window shape, S_w the window over the current pixel in the elimination image and R_w the window over the possible replacement pixel in the source image.

The standard scan-line method using an L-shaped window [Efros and Leung 1999; Hertzmann et al. 2001] is not effective in combination with the hole-filling approach described earlier. Inevitably, it leads to discontinuities at the bottom right contours of the filled region since the replacement pixels are always synthesized in accordance with the top left color context as shown in Figure 5. The hole-filling algorithm of Bertalmio *et al.* [2000] minimizes such discontinuities but is not suitable to our application since the texture is not properly reproduced.

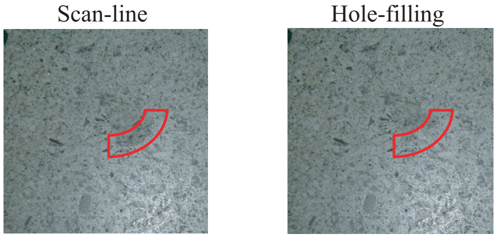

Figure 5: *The scan-line approach and the resulting discontinuities compared to the hole-filling approach.*

An exhaustive search would allow the system to use a variable context window shape for each pixel of the region. However, processing times required by such an exhaustive search are far from being interactive. To solve this issue, the system pre-processes a series of ANN search structures for different window shapes. Thus, replacement pixels are synthesized using different window shapes, depending on their non-aged context, as shown in Figure 6. While these windows contain fewer pixels than the standard L-shaped window, they significantly reduce the discontinuities caused by the standard scan-line approach (see Figure 5). Also note that each of these windows overlaps with two others, reducing possible discontinuities as well. When synthesizing pixel P, the algorithm tries to find a window where all of the elimination image pixels are non-aged. In the examples presented in this paper, such a window exists more than 70% of the time. However, if all windows contain at least one aged pixel, pixel P will not be filled during the current iteration. Such problematic pixels will be filled a few iterations later, when their neighbors are synthesized with non-aged pixels.

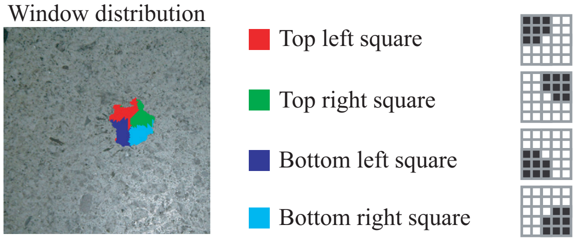

Figure 6: *Window distribution used to select the replacement pixels in the identified region.*

3.3 Reproduction Phase

The algorithm used to reproduce aging effects is a variation of the elimination algorithm. The elimination image resulting from the previous phase is now modified to add the desired aging effects. The regions in which to reproduce these effects are specified by the user in a binary image: the target aging mask. A similar iteration-based hole-filling technique is used to replace the pixels specified by the target aging mask with one of the best matches from the source image. The major difference is that the algorithm now considers both aged and non-aged context pixels. Also, a new set of features is added to the search structures to consider the aged/non-aged context from the target aging mask. When looking for the best matches, the algorithm tries to fit the color information context of the pixel and its aged/non-aged context (see Figure 7).

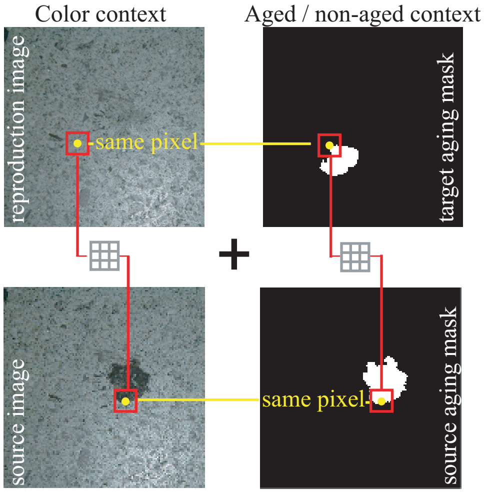

Figure 7: *A visual representation of the new set of features.*

The L_2 norm now seeks a replacement pixel with:

$$\begin{aligned} D(i,j) &= (1-\alpha)\,[S_w(i,j) - R_w(i,j)]^2 \\ &\quad +\alpha\,[S_m(i,j) - R_m(i,j)]^2 \end{aligned} \qquad (1)$$

Where S_m represents the window over the source aging mask, R_m the window over the target aging mask, and α the aging constraint parameter.

The balance between the two terms of Equation 1 is achieved with the aging parameter α. High values of α favor matching the aged/non-aged context while low values of α favor matching the color context. Most of the results have been synthesized using a 50/50 blend ($\alpha = 0.5$). This new set of features allows the reproduction process to consider internal patterns within the aged regions.

In addition, the reproduction phase allows the user to combine several aging effects from different source images. Indeed, by performing multiple reproduction steps while providing different source images and their corresponding source aging masks, it is possible to merge effects from an image to another (see Figure 1 and 8). This feature is achievable since the hole-filling algorithm allows the system to synthesize only specific regions of the output image. Also, the reproduction phase is not forced to start from the elimination image. Consequently, the reproduction image can be iteratively constructed from different parts of multiple source images. However, to achieve good results, the transition between the aged and the non-aged regions must be consistent with both images.

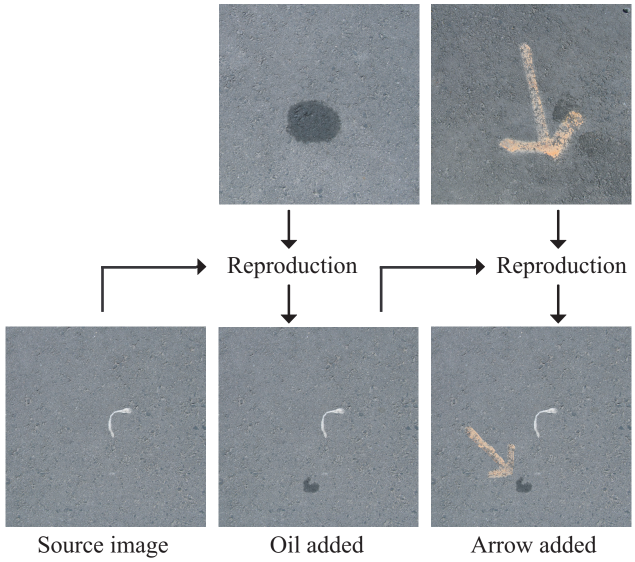

Figure 8: *An example of combining aging effects from different images.*

4 Results

The results presented in Figure 11 and in the video sequence provided with the paper highlight the versatility of the approach. Both the elimination and the reproduction images are realistic for a wide range of surfaces and aging effects. Indeed, the algorithm worked efficiently for repeated textures such as ceramic materials and for stochastic textures such as concrete and marble. The results obtained during the segmentation phase are also convincing. Most of the time, the stroke-based technique used with the energy minimization algorithm produces an excellent segmentation with minimal interaction of the user. Any misidentified regions can be quickly corrected manually by painting or erasing the pixels with a brush tool integrated in the system. We also used simple threshold methods for the examples that contain highly contrasting aging effects. The segmentation phase rarely took more than 3 minutes.

The images presented in Figure 9 show that our approach is flexible and works on a wide range of target aging mask patterns. Indeed, from the source image, it is possible to eliminate and reproduce aged regions of various shapes. Note that the system is also able to handle overlapping source and target aging masks. It is also possible to synthesize aged regions that follow a text pattern.

4.1 Efficiency

Figure 10 presents a summary of the time required to produce the images found in this paper. The colored bars present the average time needed for a specific operation while the black lines show the data dispersion.

The total time the user spends interacting with the system is on the order of a few minutes: one to six minutes to create the source aging mask and then one to four minutes on each redesign iteration to modify the target aging mask. The time required for system computation is relatively insignificant: 4 to 65 seconds to process the elimination image and then less than a second on each redesign iteration to process the reproduction image. Note that the video sequence provided with this paper shows the detailed performance for each image. These computation times are for a PC with a 3.2GHz CPU and 3GB of RAM.

4.2 Limitations of Our Approach

Even if the developed system synthesizes realistic results, a few restrictions still apply. First, the technique presented in this paper is only able to process aging effects that act on the surfaces of the objects. Consequently, aging effects that modify the geometry of the object (*e.g.* fracture, deformation) are not considered.

Also, the approach forces the studied surface to be binary-divided into aged or non-aged regions. Thus, the effects cannot be graduated from slightly aged to highly aged. This implies that the photograph containing the aging effects must be taken at the desired stage of deterioration.

Finally, care must be taken with respect to the photograph and the synthetic object lighting. Like in other camera-based texture acquisition processes, the user must carefully take into account the lighting from the capture to avoid problems with specular effects. Thus, the flash should often be disabled. Also, to avoid problems related to distortions of the texture, photographs of flat objects are often preferred.

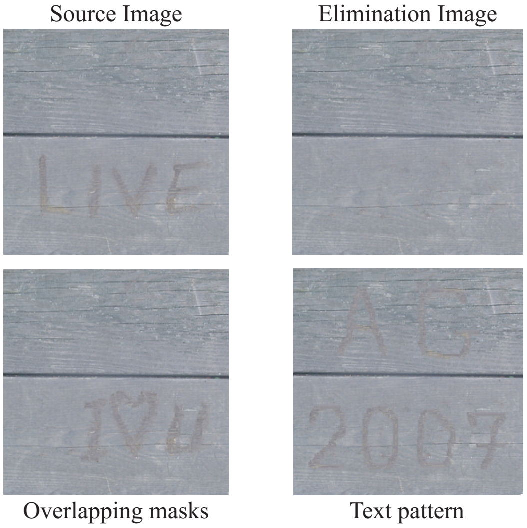

Figure 9: *The system is able to eliminate and reproduce a wide range of patterns.*

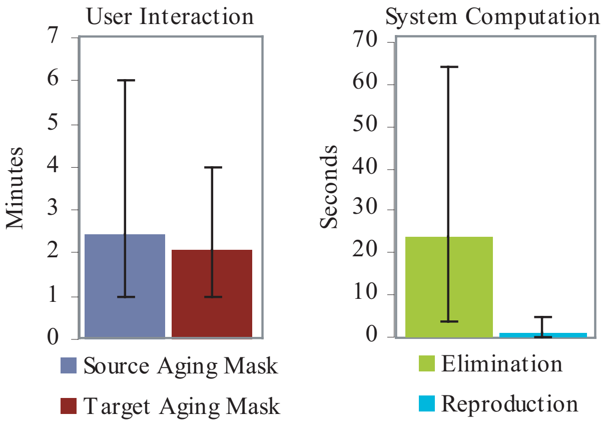

Figure 10: *Time required for each operation of the process.*

5 Conclusion

We presented a framework to reduce the amount of work needed in the redesign iterations when producing realistic objects with aging effects. It uses an interactive system integrating segmentation methods to identify aged regions of the source image. An adapted texture synthesis algorithm for removing and adding surface aging effects was developped. This algorithm takes into account the constraints of synthesizing aged and non-aged texture based on the goals specified in the target aging mask and based on the regions identified in the source aging mask.

During the redesign iterations, the system requires interaction times of one to four minutes followed by computations of only a few seconds. As shown in the examples, the system provides intuitive control over and produces convincing results for a wide range of aging effects and materials.

Such a method is appropriate for artists since it enables easy and rapid capture and editing. Compared to the physically based and empirical methods, our method has few parameters to adjust and minimal training is required to produce textures of a large variety of aging effects. Its characteristics match well with the current production pipeline. It should provide valuable help to production studios by providing an intuitive tool to edit aging effects. Thus, time required to produce aged objects should be considerably reduced and their realism significantly improved.

5.1 Future Work

The current system relies on the user to provide the target aging mask. It should be extended such that it could also synthesize the target aging mask. This would be helpful when editing effects such as scratches. Such effects present numerous regions of a particular shape which would be time-consuming to draw manually.

A difficult case to handle is multiple and combined effects where dirt could be applied on top of rust. This would require a new approach to identify the regions and effects so that it remains easy and fast to specify the target aging mask. Along the same lines, continuous variation of the aging effect cannot be represented with our binary masks. The appearance manifold approach of Wang *et al.* [2006] could be added to the segmentation tools to provide a continuous variation identification of the aging effects.

Acknowledgements

We would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the École de technologie supérieure (ÉTS) for supporting this project with scholarships. We would also like to express appreciation to Stephane Gobron, Stéphane Coulombe and all our reviewers for their constructive critiques and ideas.

References

ARYA, S., MOUNT, D. M., NETANYAHU, N. S., SILVERMAN, R., AND WU, A. Y. 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM 45*, 6, 891–923.

ASHIKHMIN, M. 2001. Synthesizing natural textures. In *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, 217–226.

BARRETT, W. A., AND CHENEY, A. S. 2002. Object-based image editing. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 777–784.

BERTALMIO, M., SAPIRO, G., CASELLES, V., AND BALLESTER, C. 2000. Image inpainting. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 417–424.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (nov), 1222–1239.

BROOKS, S., AND DODGSON, N. 2002. Self-similarity based texture editing. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 653–656.

CHEN, Y., XIA, L., WONG, T.-T., TONG, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2005. Visual simulation of weathering by gamma-ton tracing. *ACM Trans. Graph. 24*, 3, 1127–1133.

CUTLER, B., DORSEY, J., MCMILLAN, L., MÜLLER, M., AND JAGNOW, R. 2002. A procedural approach to authoring solid models. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 302–311.

DORSEY, J., AND HANRAHAN, P. 2000. Digital materials and virtual weathering. *Scientific American 282*, 2 (Feb.), 46–53.

DRORI, I., COHEN-OR, D., AND YESHURUN, H. 2003. Fragment-based image completion. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, ACM Press, New York, NY, USA, 303–312.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of ACM SIGGRAPH 2001*, 341–346.

EFROS, A. A., AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision (ICCV'99) Proceedings*, vol. 2, 1033–1038.

GOBRON, S., AND CHIBA, N. 2001. Crack pattern simulation based on 3D surface cellular automata. *The Visual Computer 17*, 5, 287–309.

GU, J., TU, C.-I., RAMAMOORTHI, R., BELHUMEUR, P., MATUSIK, W., AND NAYAR, S. 2006. Time-varying surface appearance: acquisition, modeling and rendering. *ACM Trans. Graph. 25*, 3, 762–771.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, 327–340. ISBN 1-58113-292-1.

KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph. 22*, 3, 277–286.

LEFEBVRE, S., AND HOPPE, H. 2006. Appearance-space texture synthesis. *ACM Trans. Graph. 25*, 3, 541–548.

LIANG, L., LIU, C., XU, Y.-Q., GUO, B., AND SHUM, H.-Y. 2001. Real-time texture synthesis by patch-based sampling. *ACM Trans. Graph. 20*, 3, 127–150.

LISCHINSKI, D., FARBMAN, Z., UYTTENDAELE, M., AND SZELISKI, R. 2006. Interactive local adjustment of tonal values. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, ACM Press, New York, NY, USA, 646–653.

MERILLOU, S., DISCHLER, J.-M., AND GHAZANFARPOUR, D. 2001. Corrosion: Simulating and rendering. In *Graphics Interface Proceedings*, 167–174.

O'BRIEN, J. F., BARGTEIL, A. W., AND HODGINS, J. K. 2002. Graphical modeling and animation of ductile fracture. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 291–294.

PAQUETTE, E., POULIN, P., AND DRETTAKIS, G. 2001. Surface aging by impacts. In *Graphics Interface 2001 Conference Proceedings*, 175–182.

PAQUETTE, E., POULIN, P., AND DRETTAKIS, G. 2002. The simulation of paint cracking and peeling. In *Graphics Interface 2002 Conference Proceedings*, 59–68.

SUN, J., YUAN, L., JIA, J., AND SHUM, H.-Y. 2005. Image completion with structure propagation. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, ACM Press, New York, NY, USA, 861–868.

WANG, J., TONG, X., LIN, S., PAN, M., WANG, C., BAO, H., GUO, B., AND SHUM, H.-Y. 2006. Appearance manifolds for modeling time-variant appearance of materials. *ACM Trans. Graph. 25*, 3, 754–761.

WONG, T. T., NG, W. Y., AND HENG, P. A. 1997. A geometry dependent texture generation framework for simulating surface imperfections. In *Eurographics Workshop on Rendering 1997*, Springer Wien, 139–150.

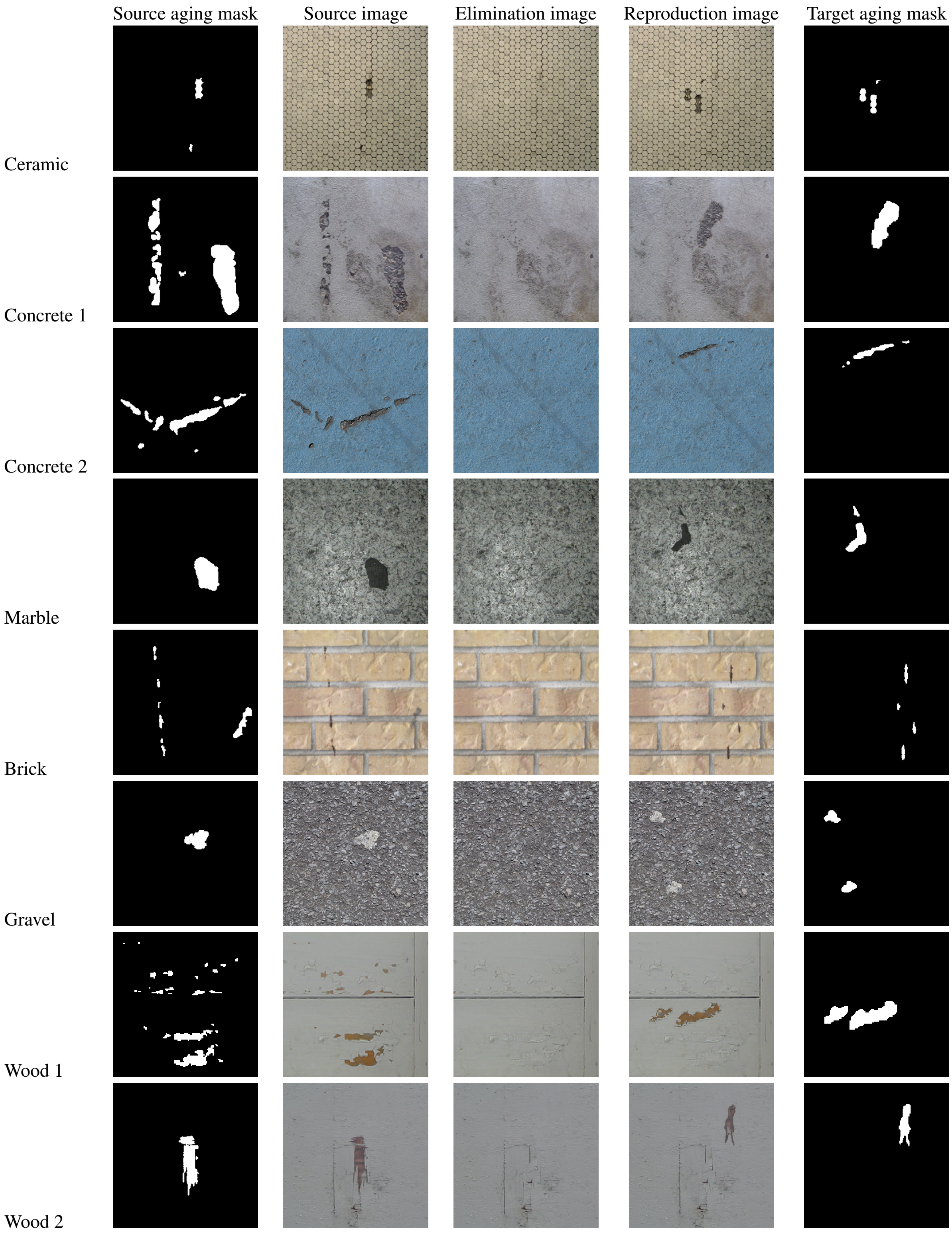

Figure 11: *Source images and results of our method for various materials, aging effects, and aging patterns.*