

2 Related Work

In this section, the most important work related to our approach is presented and discussed in detail. First, an overview of previous approaches for static image processing is provided, followed by the introduction of related approaches applying manipulation of video streams. Finally, an overview of recent Mediated Reality techniques will be given.

2.1 Static Image Processing

In the last decade, a wide variety of approaches for image processing directly related to the approach presented in this work have been proposed. Basically, these approaches can be separated into three individual categories with different goals:

- **Texture Synthesis** tries to create a new image entirely using a simple regular texture from a given reference frame. The resulting image aims to provide the same visual structure as the reference frame but has a different frame dimension (see Figure 2.1a).
- **Image Inpainting** tries to seamlessly remove user defined elements in images with complex and non regular visual content. The new created image content is created in such a manner that the final result spoofs observers not knowing that the specific image content is missing in the image (see Figure 2.1b).
- **Image Manipulation** allows individual modifications of image content. For the most part, this topic encompasses image reshuffling and image resizing. However, depending on the approach, algorithms can be used to remove visual content. Image manipulation techniques are the most powerful approaches in the context of image processing. The area of application for image manipulation does not necessarily fit into unique categories (see Figure 2.1c).

Image processing approaches can also be categorized according to their modification unit. Approaches using a pixel unit modify the result pixel by

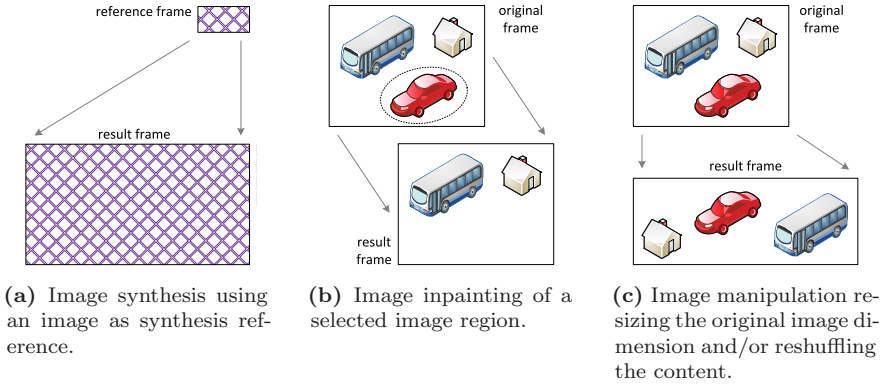


Figure 2.1: Overview of the three individual image processing categories discussed in this work.

pixel. Algorithms using image patches directly modify a set of joined pixels (mostly a squared image patch). It has to be clarified that even approaches applying single pixels as modification unit may use image patches (centered at single pixels) for similarity determination.

In the following subsections, the three main categories of image processing and their techniques are introduced in more detail. Their complexity, usability and performance is discussed.

2.1.1 Texture Synthesis

In computer graphics, textures are used intensively for creating more realistic visual results. They are often applied as simple memory storage for complex operations. The visual content of textures in computer graphics is not restricted to any specific property.

In the context of texture synthesis or image processing, textures are expected to provide simple but mostly regular structures over large image regions as depicted in Figure 2.1a. Examples for regular textures found in nature are images of regularly structured stone walls, large flowerbeds or huge sets of similar small objects. However, the synthesis of textures is not the main focus of our work as our objective focuses on natural visual content derived from arbitrary environments. Nevertheless, texture synthesis uses several basic concepts also applied in approaches suitable for images with

more complex content. An overview of the main approaches for texture synthesis is presented for a better understanding of past research. In the following, if not explicitly stated, the term texture is used in the context of texture synthesis and thus is expected to denote only the above described properties.

First, approaches applying pixel units will be introduced, followed by approaches using sets of pixels (patches):

2.1.1.1 Pixel Unit Synthesis

One of the first texture synthesis approaches was proposed by Efros and Leung [32]. They created synthesized texture from a given reference frame providing the visual structure that the new synthesized frame had to adopt. The approach created the new frame pixel by pixel starting at an arbitrary seed point in the synthesis frame. Efros created the new texture using a Markov Random Field (MRF) [59] based on the assumption that the pixel intensity value of each pixel in the texture is only dependent on a local neighborhood. The color of a new synthesized pixel p was determined by the highest probability $P(p|\omega(p))$ of the probability density function (PDF) depending on the given neighborhood $\omega(p)$ corresponding to the point p . The local neighborhood was represented by a square pixel region. Efros and Leung approximated the PDF by the application of a color intensity histogram. The histogram was composed of the center pixels from best matching patches between the reference frame and the synthesis frame. Each possible patch in the reference frame has been checked upon similarity (see Figure 2.2a) and the corresponding center pixel from the reference frame was added to the histogram if the similarity was better than a defined threshold (see Figure 2.2b). Patch similarity was determined by the sum of squared differences (SSD) of pixel color values. The approach worked well for simple and regular textures as shown in Figure 2.3. Performance and visual result directly depend on the size of the neighborhood $\omega(p)$. The size had to be chosen carefully to cover at least the lowest visible frequency (equal to the largest visible feature) in the reference frame to allow a sufficient result.

Similar to the work of Efros and Leung, Wei and Levoy [101] proposed a modified approach avoiding PDF and histogram calculation. Instead of determining a set of best matching patches between result and reference frame, they make direct use of the center pixel of the one unique best matching patch. Instead of starting at an arbitrary seed point, Wei applied a scanline algorithm in combination with an L-shaped neighborhood. This L-shape only allows for the matching of already assigned pixels between

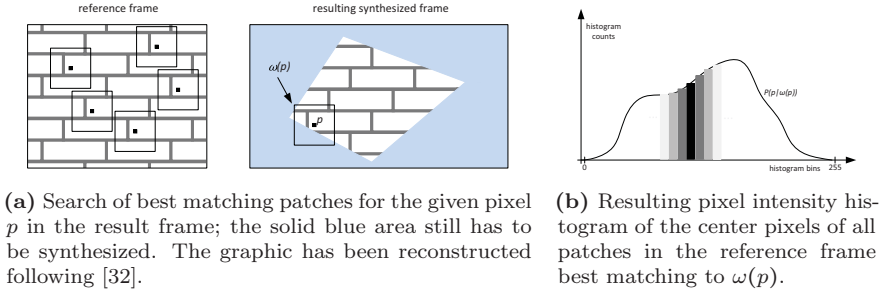


Figure 2.2: The two steps of the texture synthesis approach of Efros and Leung.

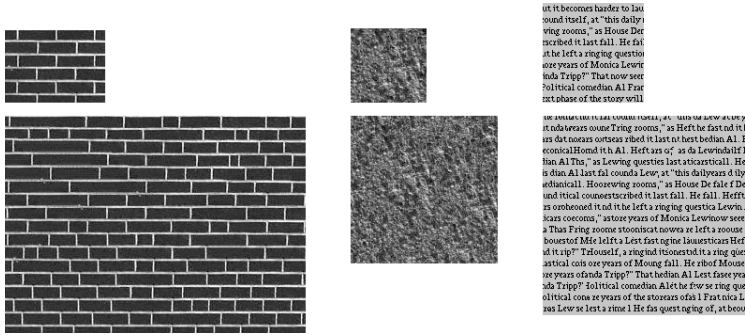


Figure 2.3: Three results of the texture synthesis approach by Efros and Leung. Top row: reference frames, bottom row: final synthesis result with ideal neighborhood $\omega(p)$. The texture images are taken from [32], kindly authorized by the author, ©1999 IEEE.

reference and result frame. (see Figure 2.4). Wei proposed the application of a multiresolution synthesis to avoid the explicit selection of the size of the neighborhood $\omega(p)$. Several pyramid layers of the reference and result frames are created to apply the synthesis to each layer. The algorithm starts on the most coarsest pyramid layer and ends on the finest image resolution. A Tree Structured Vector Quantization (TSVQ) [37, 100] is used to avoid the brute force search of best matching pixels. However, TSVQ creation is in itself time-consuming, memory intensive and cannot guarantee ideal matchings.

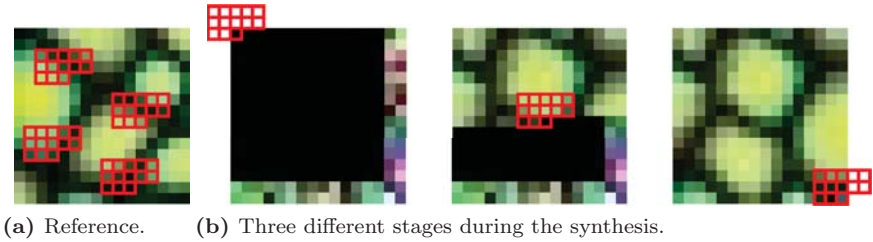


Figure 2.4: Texture synthesis by Wei and Levoy for the lowest frame resolution. The images are taken from [101], kindly authorized by the author, ©2000 ACM.

Ashikhmin [2] changed the approach of Wei by simply replacing the TSVQ application by an information propagation technique. Ashikhmin found that the usage of TSVQ is still too slow and at the same time may produce undesired visual results due to the reduced matching quality. He observed that information already found for the ideal matching of the previous pixel in the scanline order can also be used for the current pixel. The algorithm tests shifted pixels of already assigned matches in direct neighborhoods. A detailed description of a similar propagation technique is explained in Subsection 2.1.4.2. Propagation of already gathered information allows a significant performance increase compared to the performance of previous approaches. Finally, Ashikhmin introduced a possibility to create user-controlled visual results implementing the application of predefined target frames (giving a rough suggestion of the final result).

2.1.1.2 Patch Unit Synthesis

A more efficient synthesis approach was introduced by Xu et al. [105]. In contrast to a pixel-based approach explicitly synthesizing each pixel, they propose to apply an approach directly using image patches. The approach can be subdivided into three simple steps. First, Xu et al. create a tiling image by copying the reference frame as often as necessary to fill the entire new image content. Second, patches (slightly smaller than the original reference frame) are copied from one frame position to another position inside the new synthesized frame. Positioning is determined by a chaos function. Finally, the borders between copied patches are removed by cross-edge filtering within a small area around the patch borders. Figure 2.5 shows synthesis results using their approach. The approach is relatively fast due to

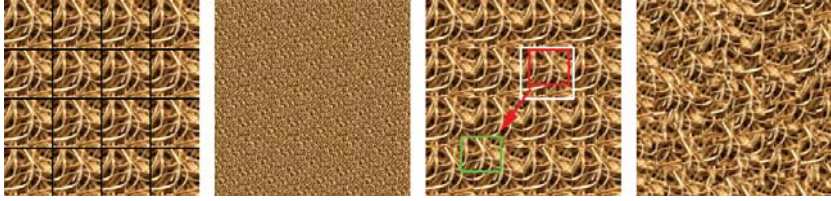


Figure 2.5: Synthesis results by Xu et al. From left to right: Tiling image, chaos function structure, copying of patches, final result after filtering. The images are taken from [105], kindly authorized by the author.

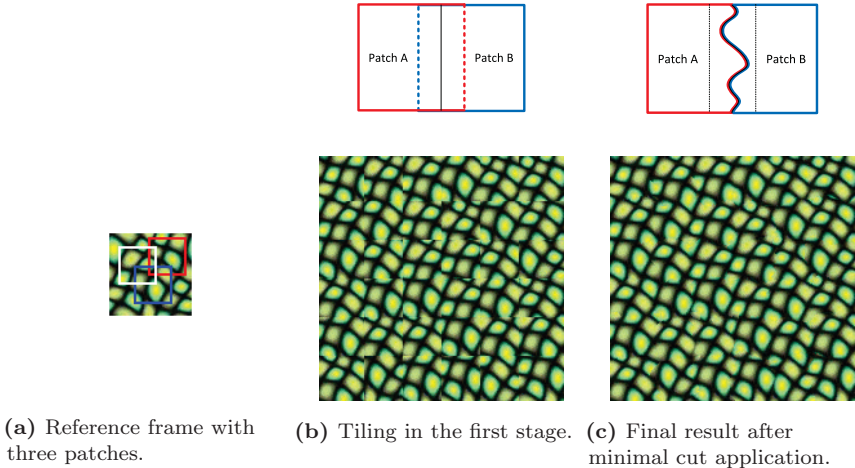


Figure 2.6: Texture synthesis by Efros and Freeman. The texture images are taken from [31], kindly authorized by the author, ©2001 ACM. The overlapping scheme images have been reconstructed following [31].

the application of image patches, however, the border filtering needs noisy textures for visually convincing results.

A slightly different patch-based synthesis approach has been developed by Efros and Freeman [31]. In contrast to Xu, Efros does not use edge filtering at the patch boundaries but applies a minimum error boundary cut between overlapping patches. The visual result can be improved as blurring and interpolation at patch boundaries can be avoided. Figure 2.6 shows the approach of Efros and Freeman for a simple and regular texture.

2.1.1.3 Discussion

In the previous subsection, the most relevant texture synthesis approaches have been described. Some approaches apply single pixels, other apply image patches to create the final result. Approaches that do not use a multiresolution image pyramid rely on the manual adjustment of patch size with respect to texture frequency. The lower the frequency of interesting image content, the larger the patch size has to be in order to cover enough visual information for measuring similarity. The application of a multiresolution approach allows automated image synthesis without multiple adjustments by the user. Still, texture synthesis is obviously restricted to mainly non-natural images displaying regular patterns.

2.1.2 Image Inpainting

Image inpainting does not create new images by using a given reference texture, but instead seamlessly removes undesired content in images. The objective of image inpainting is to create a visual result spoofing an observer not knowing the original image. As long as observers do not notice any image manipulation although a significant portion of the original visual information is missing in the inpainted image, the objective is achieved. Several completely different visual solutions for a given inpainting image may exist, as long as the overall impression of each result allows observers to be spoofed. The manipulated image content must not disturb important visual structures such as strong borders or edges. Also, textured and blurred areas have to be sufficiently reconstructed.

In the research community, image inpainting has not yet been uniquely defined and thus also the term *image completion* is used for the same context as visual object removal. Often, the terms image inpainting and image completion are used likewise although some authors distinguish between both according to the manipulation approach used. The technique denoted image inpainting is used in combination with algorithms manipulating the image by iteratively shrinking the size of the undesired image region. In contrast, image completion is used for approaches removing (or optimizing) all undesired pixels in parallel (see Figure 2.7).

However, in the following, the notation *image inpainting* will be used for both cases, because, as a rule, the algorithms cannot be uniquely separated to denote either image inpainting or image completion. If a unique separation is necessary, the inpainting process iteratively reducing the size of the undesired image content will be denoted by *shrinking* inpainting. The

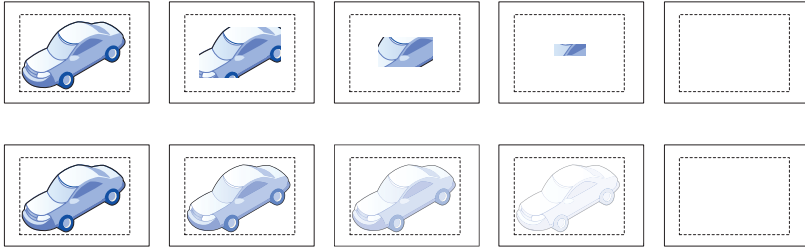


Figure 2.7: Two individual techniques to remove undesired objects in a frame. Top row: a local optimization as the manipulation starts at the boundary (sometimes mainly denoted with *Image Inpainting*, bottom row: a global optimization as the entire area is manipulated in parallel (sometimes mainly denoted with *Image Completion*). The increasing transparency stands for an abstract successive optimization of the visual result.

approach concurrently improving the entire inpainting area will be denoted by *vanishing* inpainting. Furthermore, the undesired pixel region in inpainting images will be denoted as (*inpainting*) *mask* or (*inpainting*) *hole* and the border between desired and undesired image content will be described as (*inpainting*) *boundary* or (*inpainting*) *border*.

In the next subsections, the most important inpainting approaches over the past years are introduced. Comparable to texture synthesis algorithms, two main categories of inpainting approaches exist. The first category uses pixels as the inpainting unit, approaches of the second category apply patches as the inpainting unit. However, we rather present the individual inpainting approaches in chronological order than distinguishing between those two types.

2.1.2.1 Bertalmio et al. 2000

One of the first pixel-based inpainting approaches has been proposed by Bertalmio et al. [13] to restore images with small visual errors. This approach applies a pixel-based mask filling algorithm iteratively propagating image content from the direct inpainting boundary towards the center of the hole. However, compared to a simple erosion filter [94], the approach of Bertalmio tries to explicitly propagate isophote lines (intersecting with the inpainting boundaries) into the hole to improve the visual result. Pixel values inside the undesired area are updated according to the projected 2D Laplacian [94] of the image intensities. The projection is applied onto the directions of

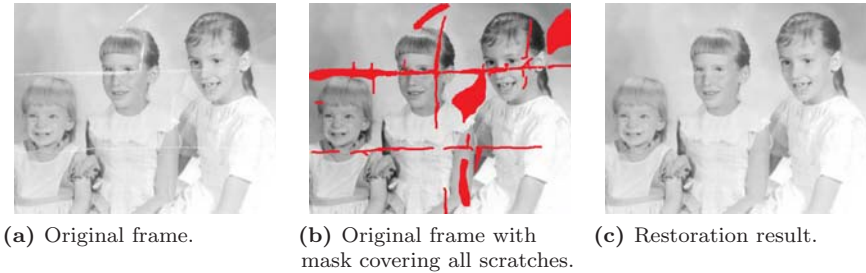


Figure 2.8: Image restoration result for the approach of Bertalmio. This inpainting approach works well with smooth or blurred image content. The images are taken from [13], kindly authorized by the author, ©2000 ACM.

isophote lines determined by the image gradients. Several thousand iterations in combination with a pyramid layer approach are applied to produce a converged result. Figure 2.8 provides a typical restoration result using the Bertalmio approach showing that this approach works well only for images with several scattered but small pixel errors. The approach is best applied to areas with smooth image information as the algorithm basically propagates blurred image content from the inpainting boundary towards the inpainting mask. Thus, this approach is not suitable for manipulation of arbitrary natural images.

2.1.2.2 Demanet et al. 2003

A few years later, Demanet et al. [29] proposed an new pixel-based inpainting approach which iteratively shrinks the undesired pixel area by replacing pixel values at the inpainting boundary. This replacement is repeated until the hole is closed. In contrast to the approach of Bertalmio, this approach uses a correspondence map for all pixels inside the undesired pixel area. The correspondence map stores exactly one reference pixel from the remaining image content for each hole pixel. This map is defined based on the visual appearance of the eight direct surrounding neighborhoods of each pixel. Demanet et al. further introduced several optimizations such as a randomized matching search, mapping propagation for neighboring pixels (like Ashikhmin [2]) and a multiresolution approach to improve the overall performance. These optimizations are also applied in most of the approaches to be introduced. Our approach is also partially inspired by mapping pixel

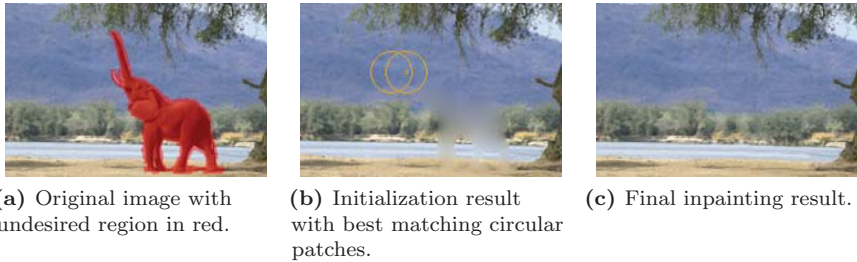


Figure 2.9: Image inpainting results by Drori et al. showing the individual inpainting steps of their approach. The images are taken from [30], kindly authorized by the author, ©2003 ACM.

positions and their introduced improvements. The approach presented by Demanet et al. provides neither an initialization strategy nor a possibility to preserve structural information from remaining image content. The simple hole shrinking strategy cannot produce a global optimum and thus may provide undesired results.

2.1.2.3 Drori et al. 2003

Drori et al. [30] introduced one of the first patch-based inpainting approaches. First, they applied a multiresolution approach using smoothing kernels to create an initial image. The initial image is created within several up- and downsampling iterations diffusing the content of the inpainting boundary into the mask area of the image. The initial process creates images with blurred inpainting holes visually comparable to the inpainting results of Bertalmio et al. [13] (see Figure 2.9b). A following second process then produces a high quality image. Drori et al. use best matching image patches with individual scale and orientations to successively increase the visual result of the initial process. They assign a confidence value to all image pixels. At the start, all pixels in the inpainting mask receive the lowest confidence value. Pixels from the remaining image information hold the highest confidence value. During iterative refinement, best matching (circular) patches with high confidence are merged to areas with lower confidence. This refinement is repeated until all image pixels have reached a specified confidence value. Figure 2.9 shows the visual results of Drori et al. Although the image quality is better than in previous inpainting approaches the algorithm fails in areas with strong structures. Performance is still far from real-time performance.

<http://www.springer.com/978-3-658-05809-8>

Advanced Real-Time Manipulation of Video Streams

Herling, J.

2014, XXIV, 244 p. 150 illus., 20 illus. in color., Softcover

ISBN: 978-3-658-05809-8