Improved Image Quilting

Jeremy Long
Department of Computer Science
University of Saskatchewan
jsl847@mail.usask.ca

David Mould
Department of Computer Science.
University of Saskatchewan
mould@cs.usask.ca

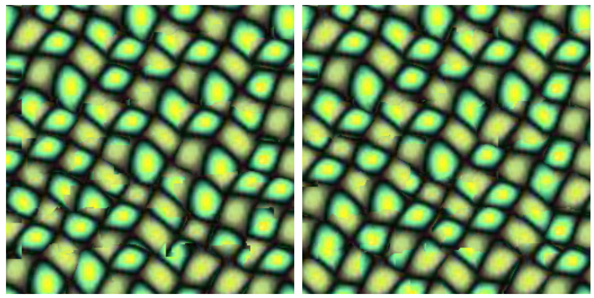

Figure 1: Image Quilting (left) and Improved Image Quilting (right).

ABSTRACT

In this paper, we present an improvement to the minimum error boundary cut, a method of shaping texture patches for non-parametric texture synthesis from example algorithms such as Efros and Freeman's Image Quilting [4]. Our method uses an alternate distance metric for Dijkstra's algorithm [3], and as a result we are able to prevent the path from taking shortcuts through high cost areas, as can sometimes be seen in traditional image quilting. The resulting artifacts tend to be less pronounced than those created using the traditional cumulative distance metric. Post-process methods such as pixel re-synthesis [8] can easily be modified and applied to our minimum error boundary cut to increase the quality of the results. This technique is not only useful for texture synthesis, but also seems promising for applications such as Wang Tiles [12, 13], or other contexts in which the minimum error boundary cut is applied.

Keywords: texture synthesis, non-parametric synthesis from example, patch-based texture synthesis, minimum error boundary cut

1 INTRODUCTION

Texture synthesis from example is the process of taking an input texture chip and using it as a basis for generating an arbitrary quantity of 'similar' texture, without obvious repetition. One way this can be accomplished is by copying parts (pixels or patches) from the input and pasting them together as an output image.

Though many methods have been developed using a combination of pixel and patch-based approaches, there remains much active research in this area. In this paper, we review several methods that are used to remove seams and discontinuities in patch-based synthesis, and propose our own method, which is a modification on the minimum error boundary cut first introduced into texture synthesis by Efros and Freeman [4].

First, we survey related work in the field of non-parametric texture synthesis from example. Because there has been so much research done in this area, we focus very specifically on the domain where our contribution is most relevant. As such, our emphasis is on patch-based synthesis and on previous methods for avoiding boundary artifacts along the seams of the patches. We also survey several other domains where improvements to the minimum error boundary cut might prove fruitful, such as Wang Tiles [12, 13].

Once we have a background for comparison, we then describe the alternate non-scalar distance metric that our improved image quilting algorithm uses. This metric encourages paths to avoid peak cost areas, even if it means taking long circuitous routes. We show in this paper how this property is desirable for texture synthesis and other applications where the minimum error boundary cut is applied. We also show how this metric can be approximated to fit with a user's preferences.

Finally, we compare results obtained using several different methods for shaping the texture patches. We demonstrate the effectiveness of our method through a qualitative survey, wherein respondents were asked to pass judgement on textures generated using the traditional minimum error boundary cut [4] and our improved version. We discuss error counts supplied by our respondents which seem to indicate that our method produces fewer discernable errors in general. We also note that our participants tended to prefer our results over ones generated using conventional image quilting (table 1).

The main contribution of our paper is a new distance metric for generating the minimum error boundary cut that reduces visual discontinuities and can be controlled by modifying the size of the plane upon which the pathing is calculated. We show the application of this method to texture synthesis from example, but we believe it also has implications in other related areas such as Wang Tiles.

2 RELATED WORK

There have been two major branches of research with regards to non-parametric synthesis from example. One set of methods are pixel-based, where each pixel in the output is copied individually from the input based on the nearby neighborhood of already synthesized pixels [5]. Although initially an exhaustive process, many clever optimizations have been implemented to increase the efficiency of this method and to generalize a framework based upon it [14, 1, 16, 6]. However, because the algorithm focuses only on a small neighborhood and a single pixel, it does not tend to preserve global features within the input texture.

Patch-based approaches attempt to synthesize whole patches of pixels at once, usually using an overlap region to determine which patch to choose. Implementations of this method [15, 4, 7, 8, 9, 11] tend to preserve features in the image, assuming the chosen patch size is large enough to encompass them. However, the naive patch-based approach can leave discontinuities along the seams of the patches, unless some method is applied to avoid or repair them. Because that is the focus of this paper, we delve more deeply into previous work in this area.

Several methods for removing the seams on the boundaries of patches have been investigated, including the use of irregularly shaped patches [4, 7] and post-process pixel re-synthesis methods [8, 9]. Our contribution falls into the former category, so we shall describe some of the existing methods for creating irregularly shaped patches in detail.

One of the first methods for shaping irregular patches was Liang et al.'s feathering approach [7]. This technique uses alpha blending across the overlap region to mix the new patch with the results synthesized in previous iterations. While feathering is effective for preserving global structure, it comes at the cost of significant blurring along the edges of the patches. This defect can be subtle in high frequency input, but it is much more apparent when applied to textures with strong features or sharp edges.

Efros and Freeman [4] proposed a different method for obtaining irregularly shaped patches in their Image Quilting paper. They transform the overlap region between a new patch and the already synthesized texture into an error map, and then use Dijkstra's algorithm [3] to find a lowest cost path through the error map from one boundary to another. This is called the minimum error boundary cut, and it attempts to shape the patches so as to minimize the error along the boundary path. In so doing, the minimum error boundary cut helps to preserve localized high-frequency structure such as edges within the input texture. This makes it a useful technique for approaching textures with distinct, localized features.

Overlap repair methods have also been proposed that use a combination of the patch and pixel-based approaches [8, 9]. Nealen and Alexa proposed a hybrid method that uses patch sampling for initial synthesis, followed by a post-process stage where individual pixels are re-synthesized in a carefully calculated sequence. These sorts of methods can yield very compelling results, and can be applied to any of the patch shaping methods described earlier. For the sake of completeness, it is important to note that pixel re-synthesis could be adapted and applied to our new method as a post-process in order to achieve comparable results. However, it is not our aim to improve on the end results of hybrid synthesis - we are more interested in the applications of a superior minimum error boundary cut, which could lead to new avenues for texture synthesis, in addition to having implications in other areas.

We propose an alternative that is based on the idea of the minimum error boundary cut, but uses a different distance metric for obtaining the shortest path. We believe our algorithm reduces discontinuities in the result, as it makes greater attempts to avoid high cost regions in the error maps of the overlap regions. We also find that our method benefits more from a larger overlap region, as it will use this extra space to maneuver around high error regions, while the conventional minimum error boundary cut will adhere mostly to paths with fewer edges in keeping with its priority to minimize cumulative distance.

The distance metric that we use for our improved image quilting was first introduced by Pai and Reissell [10] as a means of improving path planning for robots. Their non-scalar distance metric attempts to model the roughness of terrain, under the premise that robots are safer by avoiding peak edge costs, even if it means taking an alternate route with a higher total cost.

As our contribution is an improvement to the minimum error boundary cut, we also discuss several other contexts in which it has been or could be used, and where we believe our method could also yield some improvements. Two of the applications discussed here are Wang Tiles and lapped textures.

Wang Tiles are defined as a set of squares in which each edge of each tile is colored, and these colored edges are matched and aligned to tile the plane [12, 13, 2]. Textures can be mapped to Wang Tiles in order to perform a form of synthesis. Cohen et al. have presented a novel means of generating Wang Tiles based on deploying the minimum error boundary cut on an input texture [2].

Another area of interest is lapped textures [11], where texture patches are placed repeatedly onto an arbitrarily shaped surface. While some methods use alpha blending to mix the patch with the surface [11], we believe that using the minimum error boundary cut could potentially be useful for the class of textures for which image quilting is well suited, such as those with strong and distinct edges.

3 The Non-Scalar Distance Metric

3.1 Motivation

The minimum error boundary cut described by Efros and Freeman [4] is a useful means of forming irregularly shaped patches for texture synthesis. However, we find that there are certain undesirable side effects associated with the cost metric traditionally used in the path planning step.

The cumulative distance metric that is classically used for the minimum error boundary cut tries to minimize the total cumulative cost along the path. This can sometimes lead to very visible discontinuities when the path takes a shortcut through cost peaks in order to avoid having to travel long distances to circumvent them. This can lead to sharp discontinuities along the seams of the texture patches, as seen in figure 2. This figure shows that our method's non-scalar distance metric avoids cutting through the cost peak, resulting in a much less prominent discontinuity.

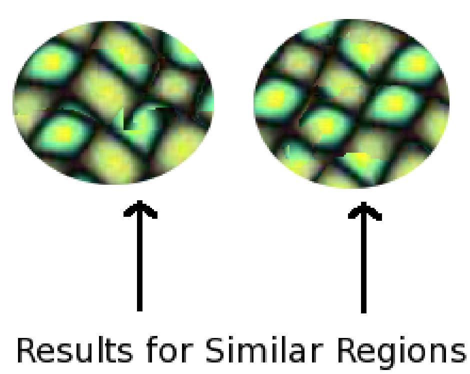

Figure 2: Discontinuities in image quilting (left) and improved image quilting (right).

3.2 The Partial Non-Scalar Distance Metric

The main contribution of our work is the implementation of a non-scalar distance metric for the minimum error boundary cut. This metric discourages paths from travelling over cost peaks. It is based on a pathing metric for robotics first introduced in 1998 [10] to simulate roughness along a path. Even if the cumulative distance of a given path is low, we do not want the robot to traverse an extremely high cost edge, as this could prove hazardous to the robot's health.

This non-scalar metric can be implemented by storing the list of edges for a prospective path in the order of decreasing cost. When two prospective paths are being compared, the corresponding element from each array of edges is selected. As soon as one differs from the other, the path with the smaller value is chosen. In essence, this metric can be seen as minimizing the maximum edge costs in the path.

Sorting the edges along a path in descending order is not a cheap operation. The best sorting algorithms are nlog(n), which is fairly prohibitive in cases where many path computations must be done. Fortunately, we find that we can achieve results similar to the full non-scalar Pai metric using only a partial version.

The partial non-scalar distance metric implemented in this paper keeps track of only the maximum edge cost traversed along the path. Comparisons between paths are first made by comparing these two maximal cost values. In the case where there is a tie, we then rely on cumulative path distance to determine the shorter path.

```
partialNonscalarCompare(Path p1, Path 2): path
{
  if (p1.maxEdgeCost == p2.maxEdgeCost)
  {
    return min(p1.distance, p2.distance);
  }

  else
  {
    return min(p1.maxEdgeCost, p2.maxEdgeCost);
  }

}
```

Pseudo code for the comparison between paths for the partial non-scalar metric is shown above. It returns the 'shorter' path by first comparing the maximum edge values in each, and using cumulative distance only to break ties. As shown in figure 3, even this very reduced version of the Pai metric produces markedly different results from the normal path planning method. These paths were generated within identical graphs using the same 28 endpoints. These endpoints are marked in red, while the starting point is marked in green.

We find that using the partial non-scalar distance metric instead of the traditional cumulative distance metric has very little impact on the efficiency of the overall algorithm. The dominant step of the pathing operation is the brusfire algorithm that is used to find all the costs in the graph, and this does not change with our new method. As such, we conclude that texture synthesis methods based on our approach will be comparable in efficiency to the traditional minimum error boundary cut used in image quilting.

3.3 Image Quilting with the Non-Scalar Distance Metric

We briefly describe the process of Image Quilting [4] upon which this work is based. In order to synthesize an output texture of arbitrary size, we divide an input chip into a number of patches. We synthesize the results by copying a patch from the input that best matches the overlap region with the already synthesized portion of the result. In order to shape the patches, we create an error map over the overlap region and find the shortest cost path from the ends of the patch to the other ends. This cut defines which parts of the overlap region will be left as the already synthesized result and which pixels will be considered part of the new patch. An example of the minimum error boundary cut is shown in figure 7, with the path through the error map in red.

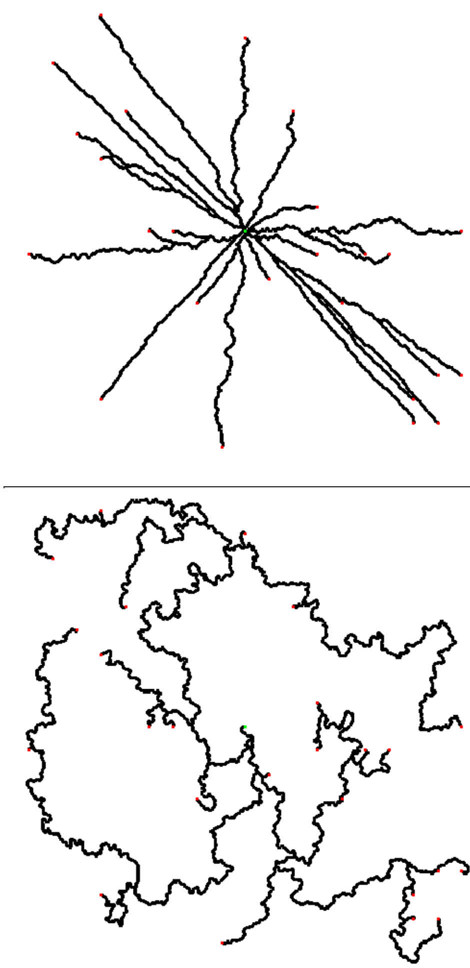

Figure 3: The conventional distance metric (top) and the partial non-scalar metric (bottom) using the same set of 28 endpoints in the same weighted graph.

As described above, our method differs from traditional image quilting in how the shortest path through the error map is determined. Figure 4 shows patches shaped using the traditional cost metric against those calculated using our non-scalar method.

The patches that emerge from improved image quilting display a more winding characteristic consistant with the metric's attempts to avoid high cost areas. As such, we have noticed that increasing the overlap size tends to greatly improve the quality of results. In our examples, we found that best results could be achieved by using an overlap region roughly equivalent to half the patch size. This is somewhat contrary to regular image quilting, where cumulative distance paths do not tend to wander far from the routes with fewer edges, meaning that adding greater overlap space does not seem to have much impact upon the quality of results. The increase in quality of the non-scalar metric as the size of the overlap region is increased is shown in figure 5.

Figure 4: Uncut patch (left) and the same patch after the minimum error boundary cut generated with the conventional distance metric (left) and the non-scalar distance metric (right).

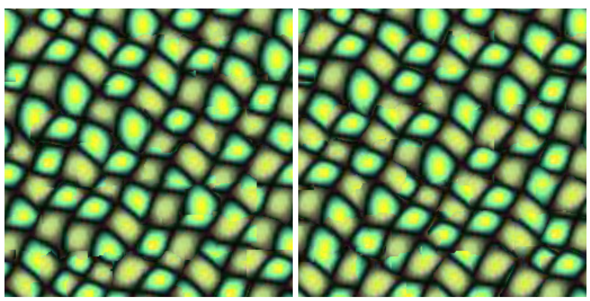

Figure 5: Non-scalar metric with an overlap of 10 (left) and an overlap of 14 (right).

By closely examining the error profiles of two patches being cut with identical constraints along the overlaps, we can see that the partial non-scalar metric features a noticeably lower peak, as seen on at the bottom of figure 6, while the top part of figure 6 shows the conventional metric. Winding around the peak in the error map causes the non-scalar metric to generate a longer path, but this allows it to avoid the high error areas that the conventional metric cuts right through. In essence, the non-scalar metric spreads the error out over a long distance, making it less sharp and apparent. In contrast, the traditional distance metric tries to take a short, compact route, even when that means travelling over an error peak.

We built our implementation for Improved Image Quilting using the Matlab framework for Hybrid Texture Synthesis [8]. This allowed us to take advantage of several advanced features such as adaptive patch sizes. This technique can be used to split patches into smaller ones when certain user defined error thresholds are exceeded.

The Hybrid Texture Synthesis framework uses feathering by default as a means of shaping patches. We first added functionality for the traditional minimum error boundary cut by creating a heap within Matlab and populating it with paths across the overlap surface. We then added functionality for the partial non-scalar metric [10]. As with feathering, we still believe that pixel re-syntehsis can be modified and applied as a post-process pixel in order to reduce errors in our results.

4 Results and Survey

This section compares results generated using the two different methods for patch shaping described in detail within this paper: the traditional minimum error boundary cut and our improved image quilting method. We decided not to include feathering in this survey because it does not work as well for the sorts of textures in

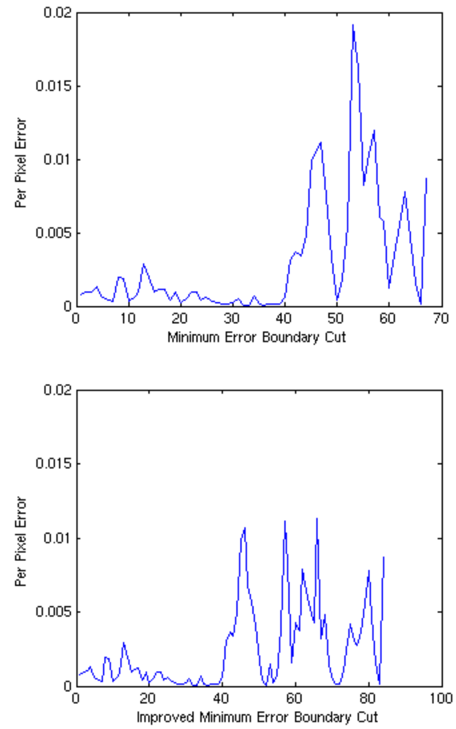

Figure 6: Per pixel error profile along the minimum error boundary cut (top) and the partial non-scalar boundary cut (bottom).

Figure 7: Minimum error boundary cut through the error map.

which image quilting is typically most effective - those where we wish to preserve strong, localized details. Also, feathering is not often used as a standalone process for other contexts in which the minimum error boundary cut is used.

We chose a set of 3 textures, and applied each method to them. The results were shown to a test group of 8 participants, who were asked to count the grievous visible defects in the output, in addition to stating their preferences as to which result best captured the essence of the input texture. We chose to perform this test using more difficult input textures with highly localized features, both because counting errors is much more strenuous in examples that synthesize well with both methods or that have a great deal of high frequency detail, and because these are the sorts of textures where image quilting is the method of choice. As we show in figure 11, our method is still perfectly capable of producing compelling results with high frequency input textures.

Figures 8, 9 and 10 show the input texture used and the output generated from conventional image quilting and our algorithm. The initial patch placement was hardcoded, and the best matching patch was chosen at each iteration. This means that the results initially share the same patch sequence, and that they are easily reproducible. We generated two test sets with each texture using a different arbitrarily chosen initial patch.

The results from our survey are discussed in two parts. First, we consider the errors that were counted by our respondents. We asked them to look for grievous synthesis defects in results generated by traditional image quilting and our new version. For five of the six texture sets, respondents reported fewer discernable errors within the ouput synthesized using our method. The exception to this was the second soybean test set, where we believe that poor initial patch placement led to many defects. More importantly, even in cases where the error counts were comparable, respondents noted that the defects created by our method were much less noticeable to casual observation, and could sometimes only be found through close examination.

Perhaps more interesting was the overall qualitative assessment performed by each participant. We asked them to decide which output seemed like a better synthesis of the input. This required them to take into account the magnitude of the errors present in addition to the number. Without having any knowledge of the methods used to generate the outputs, the respondents prefered our improved image quilting results 73% of the time. This percentage includes the second soybean test set which met with universally poor reviews. Discounting that one set increases the overall approval to 88%. These results are shown for the individual textures in table 1.

Table 1: Texture Survey Qualitative Assessment

Texture	Image Quilting	Improved Quilting
Green 1(Figure 8)	0%	100%
Green 2(Not shown)	0%	100%
Peppers 1(Figure 9)	37%	63%
Peppers 2(Not shown)	25%	75%
Soybeans 1(Figure 10)	0%	100%
Soybeans 2(Not shown)	100%	0%

We do not believe that our survey is a rigorous proof of this technique; rather, we see it as further anecdotal evidence that human observers tend to believe that our method yields fewer errors, and also less grievous ones. This seems to support the concept of the non-scalar distance metric within the context of texture synthesis, and suggests that it often succeeds in avoiding the most serious defects that are created using the traditional cumulative distance pathing metric.

5 Conclusion and Future Work

In this paper, we introduced a method for improved image quilting. Our method uses a minimum error boundary cut based on a non-scalar distance metric which attempts to avoid areas with high error, instead of taking shortcuts through them as the conventional distance metric tends to do. This yields fewer discontinuities than conventional image quilting when applied as a method for shaping patches in patch-based texture synthesis. In addition, we find that modifying the size of the overlap region significantly improves the results of our method, whereas it has little effect on the conventional minimum error boundary cut.

One avenue of future work would be a more precise method for determining proper patch placement based on the minimum boundary cut. Rather than choosing a patch based on the total error present over the entire overlap region, it would seem more fruitful to choose the next patch based on the error characteristics of the lowest cost path through the overlap, which might be very different from the overall error. This seems like it would be especially true for our non-scalar distance metric, where the paths can be much longer, but will avoid high cost regions.

As stated in our introduction, we also hope that this method for generating a minimum error boundary cut can be introduced into other contexts where this concept could be useful, such as Wang Tiles [12, 13, 2] or lapped textures [11]. We believe the characteristics of this method - especially the ability to improve quality using overlap size - are promising, of comparable efficiency to existing methods and easy to add on top of existing implementations of the minimum error boundary cut.

6 Acknowledgements

We would like to thank Andrew Nealen and Marc Alexa for making their Hybrid Texture Synthesis [8] code freely available for research purposes. This framework provided an excellent background for the rapid prototyping of our method.

References

[1] Michael Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226, 2001.

[2] Michael F. Cohen, Jonathan Shade, Stefan Hiller, and Oliver Deussen. Wang tiles for image and texture generation. In *ACM Transactions on Grapics (TOG)*, volume 22, pages 287–294, July 2003.

[3] J. Davis. Mosaics of scenes with moving objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 354, 1998.

[4] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer Graphics and Interactive techniques*, pages 341–346, 2001.

[5] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision - Volume 2*, page 1033, 1999.

[6] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and Interactive techniques*, pages 327–340, 2001.

[7] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. In *ACM Transactions on Grapics (TOG)*, volume 20, pages 127–150, July 2001.

[8] Andrew Nealen and Marc Alexa. Hybrid texture synthesis. In *Proceedings of the 14th Eurographics workshop on Rendering*, pages 97–105, 2003.

[9] Andrew Nealen and Marc Alexa. Fast and high quality overlap repair for patch-based texture synthesis. In *CGI '04: Proceedings of the Computer Graphics International (CGI'04)*, pages 582–585, Washington, DC, USA, 2004. IEEE Computer Society.

[10] Dinesh K. Pai and L.M. Reissell. Multiresolution rough terrain motion planning. *IEEE Transactions on Robotics and Automation*, 14(1):19–33, February 1998.

[11] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 465–470, 2000.

[12] H. Wang. Proving theorems by pattern recognition ii. *Bell Systems Technical Journal*, pages 1–42, 1961.

[13] H. Wang. Games, logic and computers. *Scientific American*, pages 98–106, November 1965.

[14] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured quantization. In *Proceedings of the 27th annual conference on Computer graphics and Interactive techniques*, pages 479–488, 2000.

[15] Y. Xu, B. Guo, and H. Shum. Chaos mosaics: Fast and memory efficient texutre synthesis. Tech Report MSR-TR-2000-32, Microsoft Research, 2000.

[16] Steve Zelinka and Michael Garland. Jump map-based interactive texture synthesis. *ACM Trans. Graph.*, 23(4):930–962, 2004.

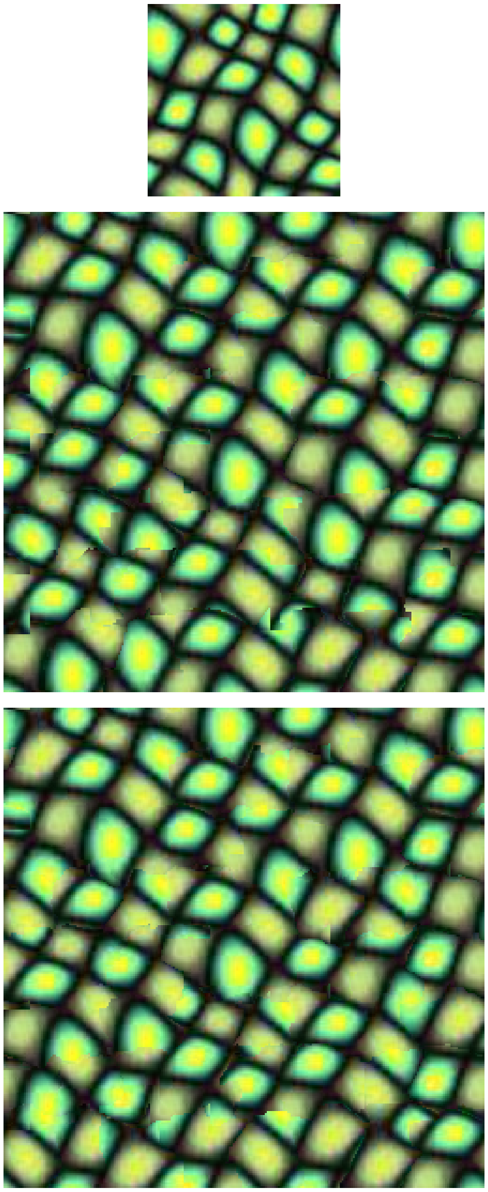

Figure 8: Input texture (top), image quilting (middle) and improved image quilting (bottom). Both outputs were generated with a patch size of 32 x 32 and an overlap size of 14.

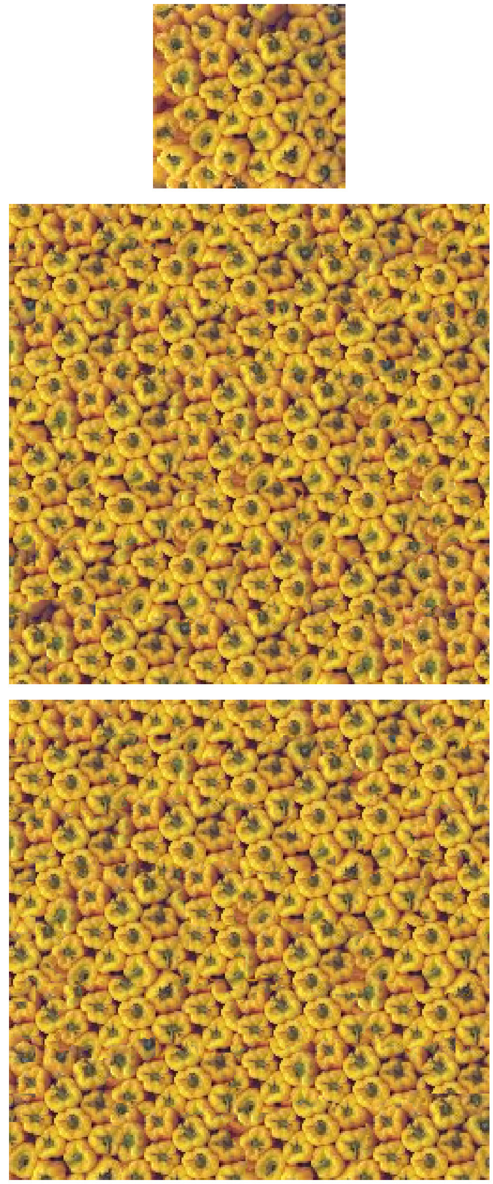

Figure 9: Input texture (top), image quilting (middle) and improved image quilting (bottom). Both outputs were generated with a patch size of 32 x 32 and an overlap size of 14.

Figure 10: Input texture (top), image quilting (middle) and improved image quilting (bottom). Both outputs were generated with a patch size of 32 x 32 and an overlap size of 14.

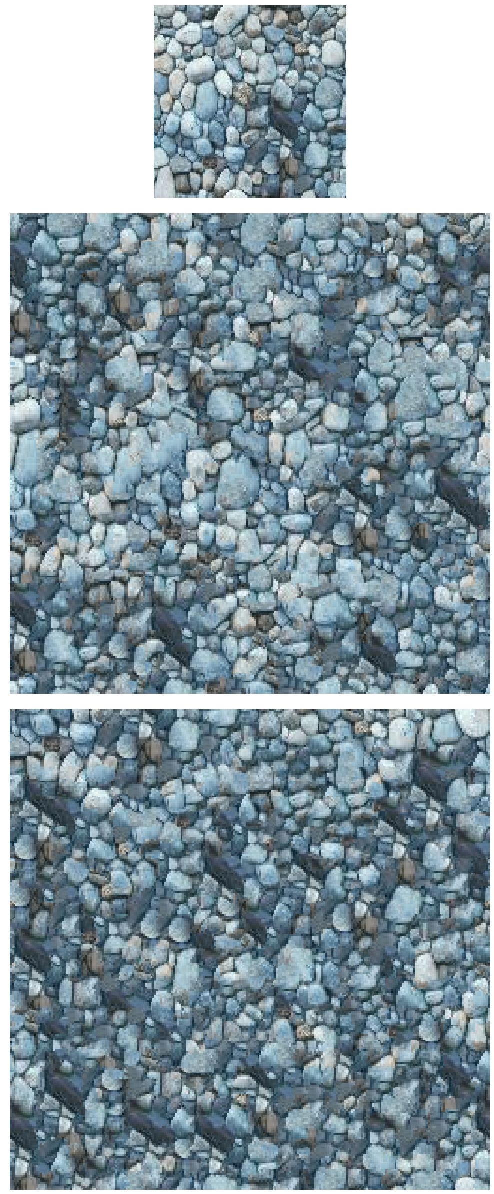

Figure 11: Input texture (top), image quilting (middle) and improved image quilting (bottom). Both outputs were generated with an adaptive patch size starting at 32 x 32 and an overlap size of 10.

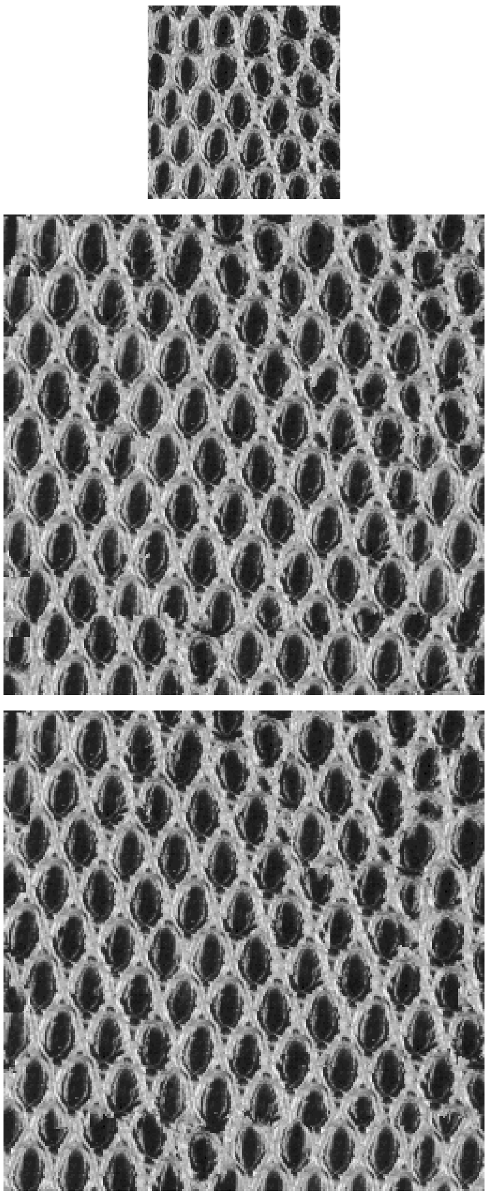

Figure 12: Input texture (top), image quilting (middle) and improved image quilting (bottom). Both outputs were generated with a patch size of 32 x 32 and an overlap size of 14.

Figure 13: Input texture (top), image quilting (middle) and improved image quilting (bottom). Both outputs were generated with a patch size of 32 x 32 and an overlap size of 14.