Computational Aesthetics in Graphics, Visualization, and Imaging
Angus Forbes and Lyn Bartram (Editors)

EXPRESSIVE 2016

Painted Stained Glass

Lars Doyle [†] and David Mould [‡]

Carleton University, Ottawa, Canada

Abstract

We propose a new region-based method for stained glass rendering of an input photograph. We achieve more regular region sizes than previous methods by using simple linear iterative clustering, or SLIC, to obtain tile boundaries. The SLIC regions respect image edges but provide an oversegmentation suitable for stained glass. We distinguish between important boundaries that match image edges, and unimportant boundaries that do not; we then resegment regions with unimportant boundaries to create more regular regions. We assign colors to stained glass tiles; lastly, we apply a painting layer to the simplified image, restoring fine details that cannot be conveyed by the tile shapes alone. This last step is analogous to the overpainting done in real-world stained glass. The outcome is a stylized image that offers a better representation of the original image content than has been available from earlier stained glass filters, while still conveying the sense of a stained glass image.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Since the early middle ages, people have looked up into the diffused colored light filtering in from windows high in the church walls and felt inspired as much by the quality of the light as by the stories and moral lessons depicted in the images. The ever-changing sky brought new emphasis to different parts of a collection of images that never remained static. Glass was regarded as special in these times not only for its ability to transmit light, but also for the mysterious alchemy that transformed base elements into this new form.

Looking beyond the colored glass, one can see that stained glass is actually a hybrid art. Both the design of the glass panes and the painting imposed on the glass play a role in the art form. Painting on glass was done using a using a range of dark tones which, depending on the era, could be used mainly to outline shapes or to achieve the complete modelling of figures in a full tonal range. For this reason, stained glass could also be seen as a branch of painting. In fact, the name of the art form is derived from the silver stain that was painted on to the glass, which transformed into a yellow or gold color when fired [Kha].

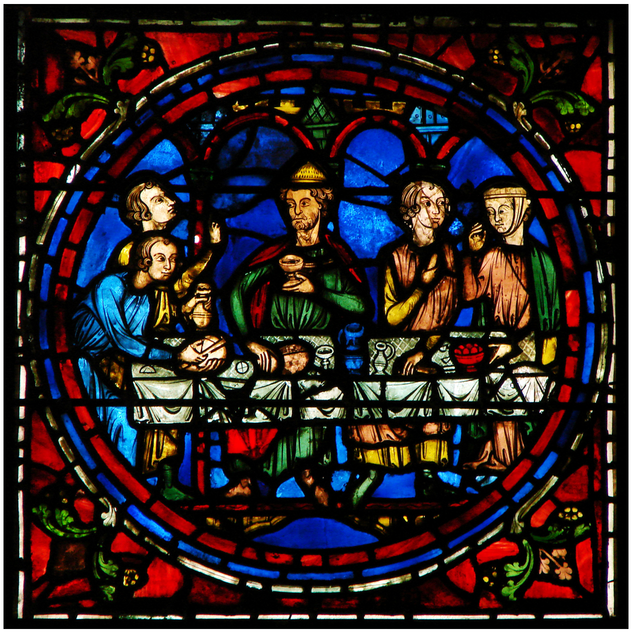

Figure 1: *Detail of a panel from Chartres Cathedral illustrating our intent*

To make a stained glass window, craftsmen would select chosen colors and cuts of glass to match a section of a widow design pattern. Details that were too fine to be shown by individual cuts of glass, such as features, drapery and ornamental motifs, were opaquely painted onto the glass; glassmakers used a relatively colorless vitreous enamel composed of dense, dark metal oxides and ground glass, mixed with a glue binder, such as gum arabic. The painted glass was then placed in a kiln and heated to a tempera-

[†] larsdoyle@cmail.carleton.ca
[‡] mould@scs.carleton.ca

ture sufficient to fuse the enamel to the glass. Strips of lead, with grooves on either side, were then fitted around all the pieces of glass and the joints between them soldered to create a single window [Sow81]. An example of a stained glass panel is shown in Figure 1. The thousand-year-old process of stained glass making still survives in a similar fashion today.

In the tradition of non-photorealistic rendering, we propose an image filter that will take an input photograph and produce an output image in the style of a medieval stained glass window. Our major contribution is to add an explicitly defined painting layer to the tiled glass representation of the image. The painting layer adds back some of the image details that are impossible to communicate with the large-scale tiles. Additionally, we base our image segmentation on simple linear iterative clustering (SLIC). We also present a method based on vertex cover to distinguish between SLIC regions that we wish to keep intact and those that we recalculate to achieve a more regular structure. Finally, we use the results of our vertex cover to attempt foreground-background segmentation and apply different tile stylings to highlight the foreground.

2. Previous Work

In computer graphics, as well as historically, stained glass was preceded by mosaics. Hausner's notable work on mosaic creation [Hau01] has influenced non-photorealistic rendering considerably. Stained glass rendering was first attempted by Mould [Mou03], who used mean-shift segmentation to obtain approximate tiles and then morphological operations to clean up the tile boundaries. Subsequent work on stained glass by Brooks [Bro06] and Setlur and Wilkinson [SW06] similarly relied on automated segmentation, with more attention to the textures within tiles. Brooks's work is the most similar to ours, as the author provides synthetic texture and retains some texture details in each glass tile; we propose incorporating more of the original image into our paint layer, more fully preserving image contents.

Segmentation is a tool in widespread use for image stylization, helpful for more prosaic effects such as watercolor rendering [BKTS06] as well as dramatic alterations of the canvas such as the stylized photographs of DeCarlo and Santella [DS02] or the "arty shapes" of Song et al. [SRHC08]. Manual segmentation is sometimes employed; in this work, we had hoped to achieve a fully automated rendering process. A recent, fast approach to oversegmentation is given by the simple linear iterative clustering algorithm (SLIC) [ASS*12], with applications in NPR such as pixel art [GDA*12].

SLIC uses k-means to cluster pixels using a combined image-plane and colorspace distance metric. Cluster centers are initialized at regular grid steps with spacing $S = \sqrt{N/K}$, where K is the user-selected number of superpixels and N is the number of pixels in the image. SLIC operates iteratively: a single iteration searches a $2S \times 2S$ square neighborhood around each cluster center and assigns pixels to their closest center, where "closest" is with respect to the combined distance metric. Cluster centers are then recomputed, taking on the average position and color of their associated cluster pixels. Iterations continue for a fixed number of steps, or halt when movement of cluster centers between iterations slows below a given threshold. In a final step, small isolated regions are absorbed into their larger surrounding regions. SLIC's basic formulation produces a set of clusters of approximately uniform size, where the cluster boundaries correspond roughly to image edges. A medium-scale oversegmentation from SLIC seems ideal for stained glass.

Traditional stained glass used a reduced palette. Mould [Mou03] used an ad-hoc palette inspired by the heraldic tinctures, while other stained glass rendering systems discounted the need for palette reduction. Recent work in reduced-palette NPR, such as that of Semmo et al. [SLKD15] could be adapted to stained glass. In this paper, we present a simple quantization-based technique for tile coloring, not always effective; we note that strict adherence to historical colors is limiting, with more appealing results arising simply from averaging the original image colors over the tiles.

We drew inspiration from the base-plus-detail formulation of Bae et al. [BPD06]. These authors created texture-enhanced photographs by separating a low-frequency base layer from a high-frequency detail layer and applying different enhancements to each. In our work, we have three layers: a colored layer with the stained glass tiles, and a two-part paint layer, separated into a base layer showing a smooth version of the input image tones and a detail layer containing sharp features. Our approach approximately mimics the image creation process of historical image creation in stained glass. We used the powerful eXtended Difference of Gaussians process (XDoG) [Win11] to extract some details from the original image for subsequent inclusion in our paint layer.

3. Algorithm

With our image filter we aim to reproduce effects which relate back to the techniques, technical limitations, and design objectives of the medieval craftsmen who produced stained glass windows. First, we describe our method for segmenting the input photograph into regions corresponding to stained glass tiles. Next, we show how a glass overpainting effect can be created from this same input photograph. Finally, we combine the glass tiles and the paint layer in order to produce a stained glass representation of the original input photograph. Figure 2 illustrates the pipeline of our method.

3.1. Segmentation

We oversegment our input image with simple linear iterative clustering (SLIC). The resulting clusters are compact, relatively uniform in size, and adhere well to image boundaries. These are qualities which we observe to be inherent in medieval stained glass tiles, with the exception that real stained glass potentially has more variability in tile size. Also, the cluster boundaries are fairly irregular. This is desired in regions where the cluster boundaries closely follow the image boundaries. Yet, in other regions, the resulting irregularity has no semantic meaning; real stained glass tiles tend to be convex and somewhat regular in shape. The unmodified SLIC clusters are not wholly suitable as stained glass tiles. The following paragraphs describe how we recalculate selected regions and boundaries to create tiles that better conform to the expectations of stained glass.

Our process of region modification considers the original SLIC

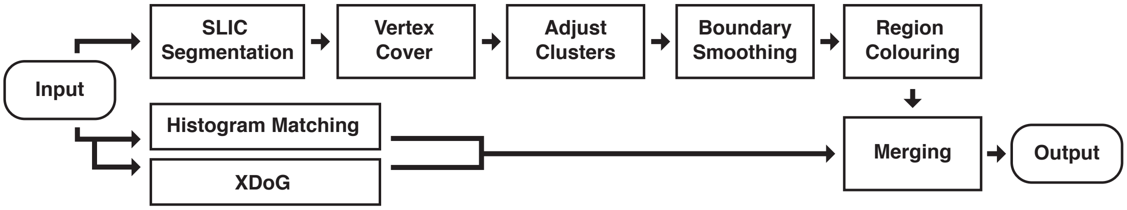

Figure 2: *Schematic of our processing pipeline.*

clusters as belonging to one of two classes – those that describe the important edges in the image which we want to preserve, and those whose shape is less important for describing the image content. An edge map produced with a Difference of Gaussians (DoG) filter over a greyscale version of the image provides a means of making this classification. A cluster boundary pixel is considered an edge pixel if it corresponds to a pixel in the DoG output above a threshold $t \in [0,1]$. We will use the edge pixels to help us distinguish between SLIC cluster boundaries that define the image content and those that do not. Boundaries on edges will be left intact with only minor smoothing, while other boundaries will be recomputed completely; details follow.

A graph $G(V,E)$ is constructed with V being the set of vertices defined as the clusters produced by our initial segmentation. The set E of edges is then built by considering the neighbouring clusters n_i of each cluster v. If the ratio of edge pixels to total boundary pixels between two clusters is above a threshold then (v, n_i) is an edge in E. We then find a vertex cover in the graph. By leaving the clusters represented by our vertex cover unmodified we can preserve all the cluster boundaries that accurately define the edges of our original input image. As vertex cover is NP-complete, we used an approximation which selects vertices in a greedy method. Algorithm 1 describes this method.

Algorithm 1 TrustedClusters

```
procedure TrustedClusters($G(V,E)$)
    $result \leftarrow \emptyset$
    while $E \neq \emptyset$ do
        $(u,v) \leftarrow$ random edge from $E$
        $w \leftarrow$ vertex u or v with the largest ratio of edge pixels to total boundary pixels
        $result \leftarrow result \cup w$
        $E \leftarrow E \setminus$ {all edges incident to w}
    return $result$
```

Our greedy approach returns a set of vertices that correspond to the clusters whose boundaries best define the original image. We call these our *trusted clusters*, and leave them unmodified; some trusted clusters are shown in Figure 3, bottom left. It is our goal to achieve as small a vertex cover as possible, since unnecessary irregular boundaries detract from the stained glass look we seek to achieve. We perform several iterations of our randomized algorithm, and then select the smallest result as input to the next stage of the processing pipeline.

After finding trusted clusters, the unmarked clusters may be modified as we wish. We will refer to non-trusted clusters collectively as the *modification set*. Our method then recalculates the clusters in the modification set using the original SLIC cluster centres, adjusting the parameters to use solely Euclidean spatial distance. The resulting clusters are Voronoi-like on all boundaries except where they are adjacent to a trusted cluster.

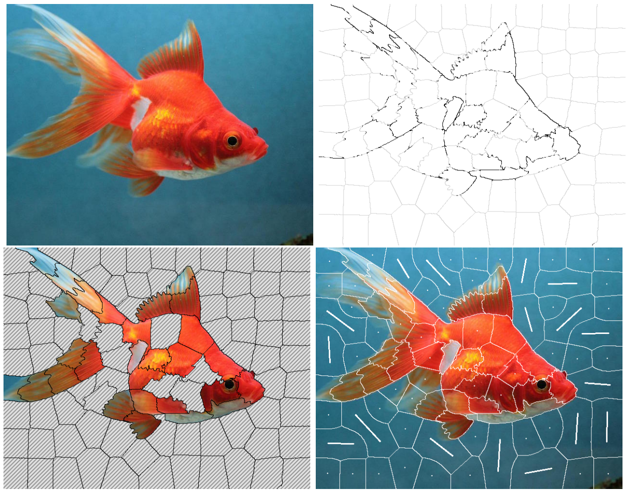

Figure 3: *Upper left: input image. Upper right: original SLIC segmentation showing pixels that align with image edges in black. Lower left: result of vertex cover. Lower right: adjusted modification set*

To obtain a slight degree of cluster diversity, we merge those clusters from the modification set that exhibit sufficient color similarity in the input photograph. The procedure is as follows. First, we compute the average color of each cluster. Second, we compute the color distance between each cluster and its neighbouring clusters. If the color distance between a cluster and its closest neighbour is below a threshold then these two clusters are selected for merging. Finally, we set a flag preventing the newly merged cluster from further merging. Clusters are merged by defining a line segment between the two cluster centres and using it as a cluster centre in a final SLIC pass, where the spatial distance might be from a point or from a line. Figure 3 illustrates the result of merging and recalculating the modification set.

Finally, we smooth the regions using the erosion and dilation operators from mathematical morphology [GW08]. Following Mould [Mou03], we simultaneously erode all regions with a circular structuring element; to be precise, a pixel is marked as eroded if any pixel in the structuring element belongs to a different region than the centre pixel. We add a small modification to this process so as to preserve edges: whenever the structuring element covers an edge pixel, no erosion takes place. Subsequently, we simultaneously dilate all regions to relabel the eroded pixels: each eroded pixel is labeled with the region of its nearest non-eroded neighbour, where *nearest* is with respect to Euclidean distance in the (x, y) plane.

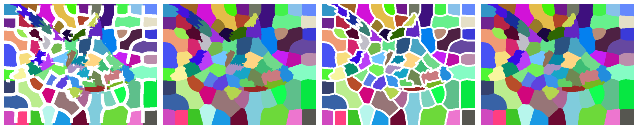

Figure 4: *Smoothing the segmented regions. Left to right: constrained erosion of the clusters; subsequent dilation; universal erosion; final dilation*

3.2. Glass Painting

A typical method for painting stained glass would involve applying a succession of thin layers of vitreous paint using a wide brush to build up the shading of the image. As a final step the main outlines and dark solid regions of the image were painted in using a thicker paint. We represent this process by splitting the painting algorithm into two stages – the base layer and the detail layer.

In order to imitate the above-mentioned painting methods, the base layer in the painting filter will first convert the input image to greyscale and then match its histogram to a target image. We perform histogram matching by first computing the normalized cumulative histograms of both the target and input image. We then determine new values for our base layer by finding the minimum grey value in the target cumulative histogram which contains at least as many pixels as the cumulative histogram of the input image at its observed grey level. The target image for the histogram matching is a series of distinct grey value steps, as illustrated in Figure 5, top. Each grey step in the target image corresponds to one wash of thin paint, creating the effect of an image being painted in a series of washes. This method removes all extraneous detail from the image and leaves only its basic forms. Finally, a Gaussian blur is applied, completing the effect of thin washes of paint blending into each other. The base layer is shown in Figure 5, middle left.

To recreate fine details that are lost in the histogram matching, we used the extended difference of Gaussians (XDoG) filter. Using XDoG requires selecting a few parameters: the threshold ε, affecting sensitivity to noise; parameter φ, controlling the width of intensity transitions; and p, controlling the strength of edge sharpening. We used $\varepsilon = 50$, $\varphi = 0.05$, and $p = 5$ for all images, settings which we found gave consistently good results for our application: the resulting edge enhanced image with soft thresholding provided a suitable detail layer. In a final step, the base and detail layers were combined using a multiplicative blend: see Equation 1, where intensity ranges are assumed to be normalized to the interval [0, 1]. Figure 5 shows the base, detail, and blended paint layers.

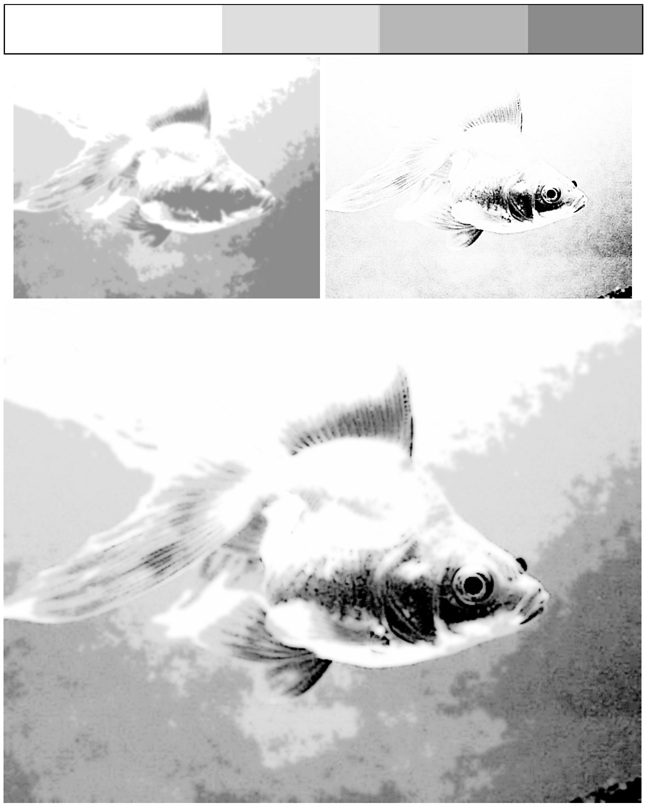

Figure 5: *Progression of an image through the painting filter. Top: grey steps used for histogram matching. Middle left: base layer. Middle right: detail layer. Bottom: merged result*

$$I_{blend} = I_{base} \times I_{detail} \tag{1}$$

3.3. Rendering Procedure

Now that the main algorithms for segmenting and painting the image have been described, what remains is to color the glass, apply the painted effect, and then render the lead cames.

Working in the HSV colorspace, we color the glass tiles with the average hue and saturation of the region's color while keeping a constant lightness in the value channel. The missing tonal range will be reintroduced through the painting layer that is later blended with the glass colors.

As an alternative approach to color averaging, we emphasize historical glass-making techniques by using a smaller subset of colors. We do this to recreate the simplicity and boldness of early stained glass. We constrain colors to a quantized color scheme that we derive from the available metal oxides used in medieval glass-making.

These are red, yellow, blue, green, and purple. An off-white color is also used to represent plain, uncolored glass. The method for choosing a color for a particular glass pane commences by averaging the colors in the corresponding region of the underlying image and then finding the closest match among the available colors. Most color matches can then be made by considering only the 360^o value in the hue channel of HSV without regard for color saturation or light intensity. The initial color choices are based on a sequence of thresholds: first, bright colors with low saturation are set to white; second, dark colors with low saturation are set to blue; third, remaining colors are set according to a thresholding of the original hues.

Merging the painted image into the colored glass image is done by taking the product of the value channels. We retain the HSV color space from the previous process and perform our image blend in the value channel using $I_{blend} = I_{value} \times I_{paint}$, where we assume that values range in the interval [0, 1]. Only the value channel is modified; the hue and saturation channels remain unaltered.

The strips of lead which hold the panes of glass together are usually seen as thick dark lines when viewing a stained glass window from the interior of a church. For this reason, we chose to render them in a relatively dark color with the inclusion of some diffuse lighting effects to give them a slight 3-D appearance. Of course, it would be easy to render them in a more complex fashion.

The shapes of the lead cames are created by dilating the contours of the region boundaries with a structuring element of a chosen radius. This will define thick black lines that divide the regions of the image. A shading filter is then applied to color the cames in various values of grey and the final stained glass image is complete.

4. Results and Discussion

In this section, we provide additional results from our method. We demonstrate our stained glass image filter on a variety of subjects including animals, landscape, people, and architecture. Figure 7 shows some general results of our method. Unless otherwise noted, we segmented the input image with 120 SLIC superpixels and set the compactness parameter m to 50. After recalculating the modification set, we then smoothed the resulting regions using a 15-pixel diameter structuring element for morphological opening on non-edge boundaries. We followed with by a 9-pixel diameter opening over all region boundaries. Our non-optimized CPU implementation takes about 35 seconds to process a half-megapixel image, about two thirds of which was spent on morphological operations.

By using SLIC as our initial segmentation method we are able to adjust the tile size as desired. Figure 8 shows some results of running our filter with varying numbers of SLIC regions (75, 200, 400, and 800 are shown). We observe that there is a tradeoff between competing considerations. With large numbers of regions, the tiles are more able to capture the image features, but the lead cames bounding the regions begin to visually dominate the image structure. Conversely, with lower region counts, more concave tile shapes emerge, but with few tiles, only the largest features can be shown. We chose 120 as a good balance between these extremes.

Positive results can be seen in images where the salient content is well defined. The fish in the bottom of Figure 14 achieves excellent separation from the background. While the red-blue color contrast of the foreground-background helps in this case, one can also notice that the tile shapes form an abstract pattern in the background while the tiles in the interior of the fish conform to the image content. This dichotomy of foreground-background shapes helps to accentuate the important content as it is selected by our vertex cover. The same effect is also seen in the bird and the butterfly images in the bottom row of Figure 14, and the football image of figure 7. In contrast, the image of the family in on the bottom right of Figure 7 displays a poor segmentation due to the business in the interior of the image. Here, some excessively concave shapes are present which are uncharacteristic of stained glass. Subsequently, our process of region smoothing has little effect here due to the high edge count in this area.

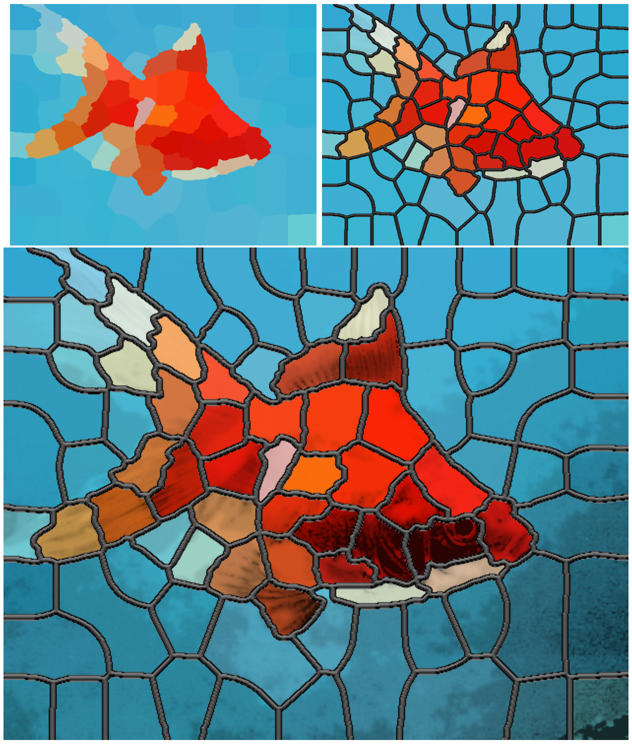

Figure 6: *Illustration of our rendering process. Upper left: colored clusters. Upper right: rendered cames. Bottom: application of painted layer*

We extend from this foreground-background division returned by our vertex cover and attempt to replace the background regions with a new tile pattern entirely. We define backgrounds as regions in our modification set that are part of a connected component with at least one of the four borders of the image. Additionally, in order to prevent true foreground regions from being included in the background we add a condition that a region must contain fewer that four neighbours that belong to the trusted region set to be classed as background. Figure 9 shows the results of this heuristic where the background regions are replaced with diamond shaped tiles. The fish image is successful in that the correct regions are mostly cho-

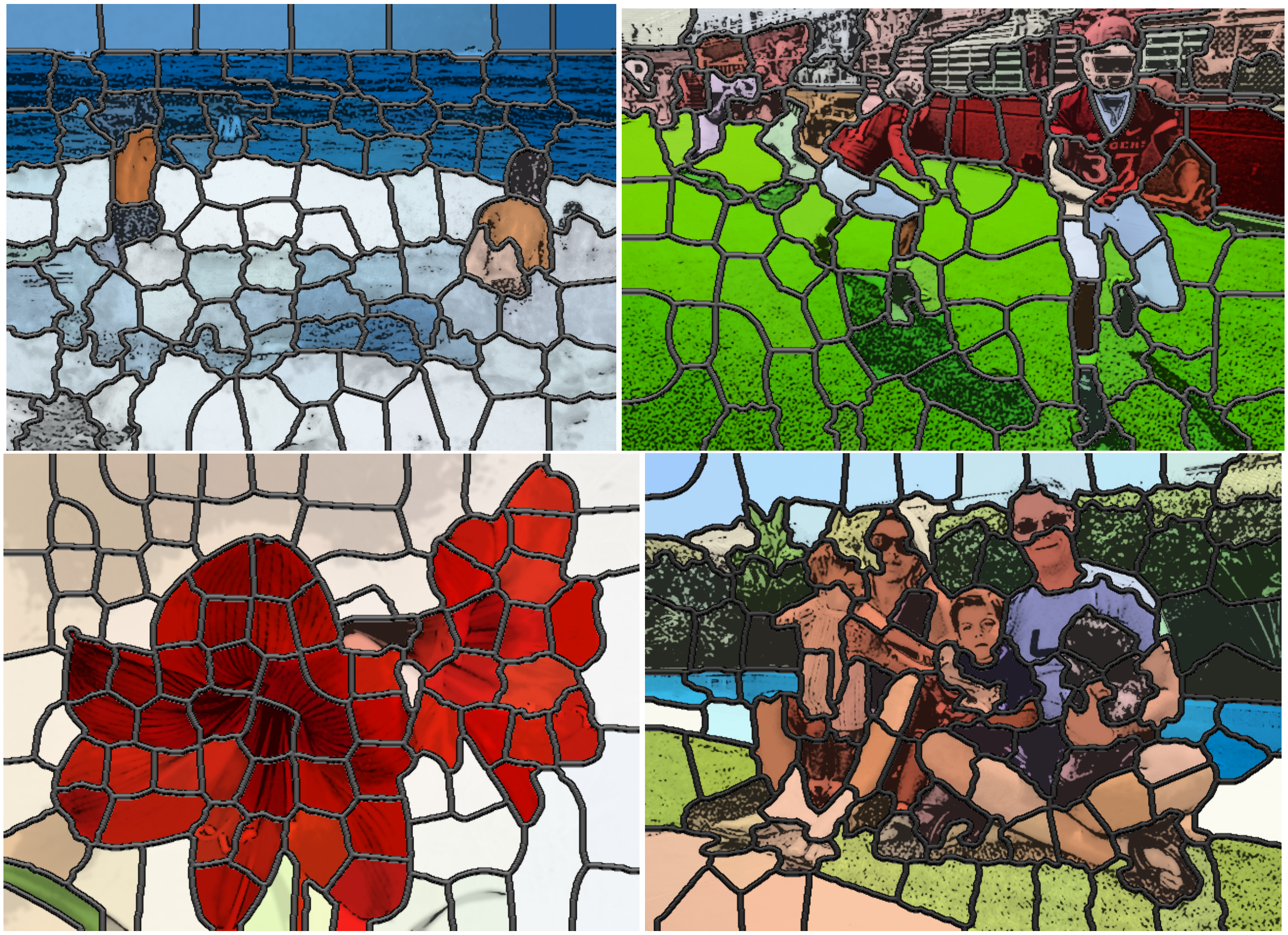

Figure 7: *Results of applying our method.*

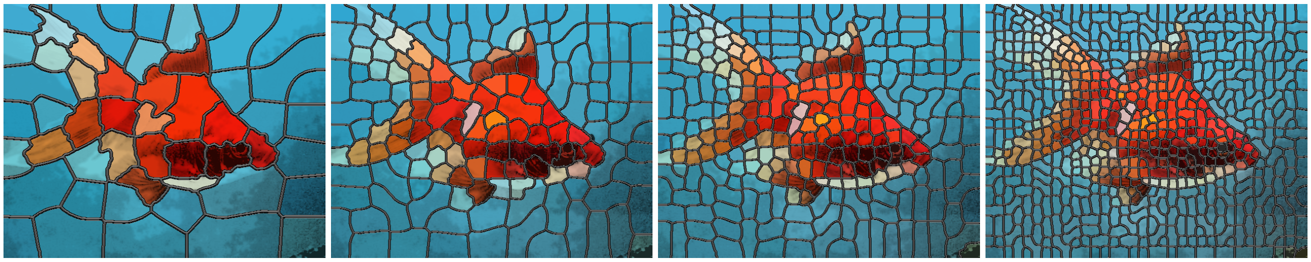

Figure 8: *Comparison of varying superpixel sizes. From left to right: 75, 200, 400, 800*

sen as foreground. The bird image is less so, as some of the true background regions were part of the vertex cover.

Our stained glass filter explicitly uses a painting layer that has not been considered in stained glass work. Figure 10 illustrates the colored tiles of a stained glass image with the addition of either the base, detail, or the combined base and detail of the painting filter. A marked improvement is made to the image with the addition of the smooth histogram matched base layer. Here we see a soft transition of tones that add visual interest to otherwise flat tiles. Our XDoG-produced detail layer, while too harsh to be used on its own, blends seamlessly with the base layer in the final result. Additional results of our painting filter can be seen in Figure 11. The soft color washes produced in the base layer combine with the detail layer to create a painterly impression. The sky above the parliament buildings is reminiscent of watercolor and the high contrast in the flowers creates a richness in tone that is lacking in the full color original. Brooks achieves a painted stained glass filter effect by adding Perlin noise to a colorized input image followed by

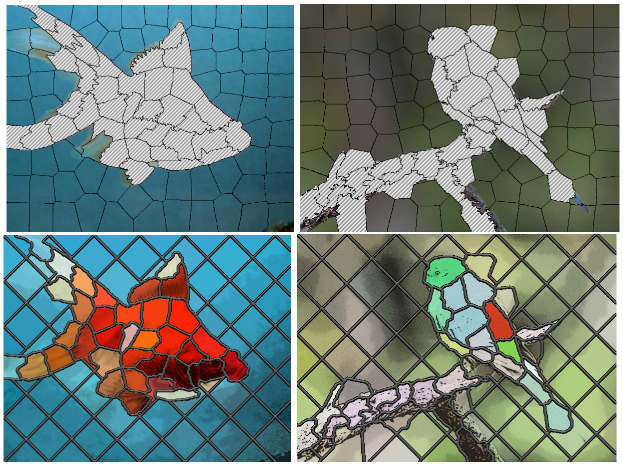

Figure 9: *Results of background detection and replacement.*

image warping (Figure 14: left and middle). This created a subtle glass-like result that blends with the underlying image content. In our filter, we attempt a more deliberate painting effect. The strong solid colors of our detail layer are more representative of the glass painting methods of the medieval period.

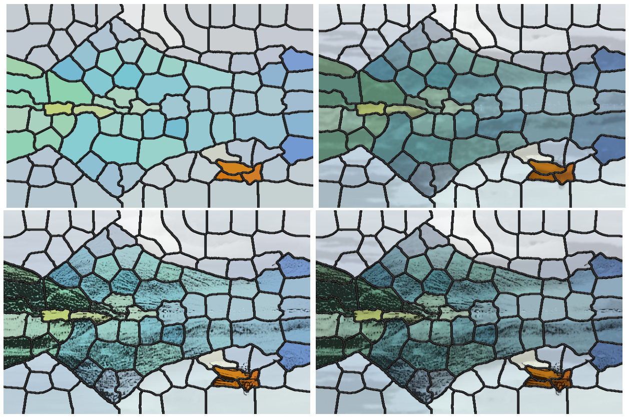

Figure 10: *Variations in the application of the painting layer. Upper left: plain glass. Upper right: base layer. Lower left: detail layer. Lower right: combined base and detail layer.*

Figure 12 shows our attempt at using a limited quantized color matching scheme. While colors do tend to find an appropriate match in our limited palette, some problems do emerge. First, subtle shifts in color are lost, sometimes causing indistinct objects to blend together. This can be seen in the house image where the roof of the house joins with the tree by representation with identical colors. Here, our color averaging scheme performs better. Second, despite our attempts at creating cohesive areas of color by neighbourhood voting at color transition boundaries there are still sporadically colored tiles appearing in our result. The red and yellow split within a single piece of fruit seems too extreme. Conversely, the benefits of the quantized colors can also be seen in the fruit image. The vibrancy in the saturated colors is more representative of medieval stained glass, as opposed to the averaged result which appears dull in comparison. Color planning in art generally requires considering an image as a whole and is unlikely to be solved considering only local regions. More sophisticated methods will be needed to achieve good results.

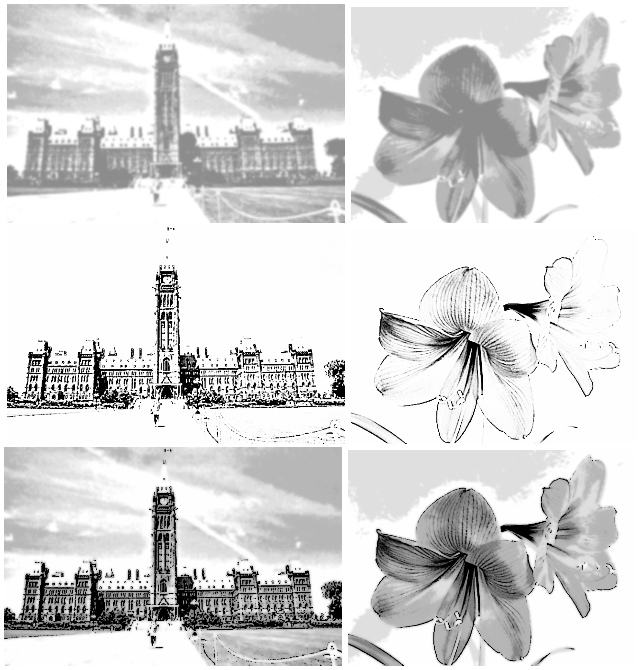

Figure 11: *Progression of additional images through our painting filter*

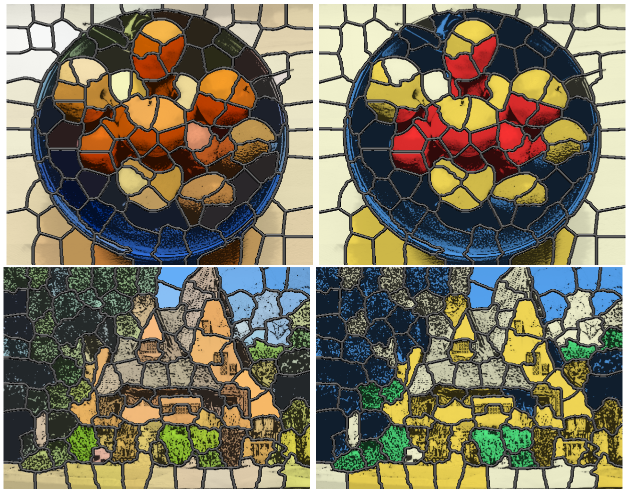

Figure 12: *Left: Average coloring. Right: Results of our quantized color method.*

Figure 13 shows a comparison between our results and those of Mould [Mou03]. In both cases, the clear background allows for a good segmentation. Our tiles are somewhat more regular than those of Mould. The major difference, though, is that Mould used an arbitrary texture in the tile interiors while our paint layer brings to life the image content in the tile interiors. Similary, Figure 14 compares our results with those of Brooks [Bro06], who spent some effort to create lively interior tile colors. His method preserves some interior textural details, and is probably the closest to ours. In our judgement, our paint layer better communicates the image content while still adhering to the style of stained glass. Also, Brooks's tile shapes were created partly manually, and do not have the regularity of size and shape visible in historical stained glass.

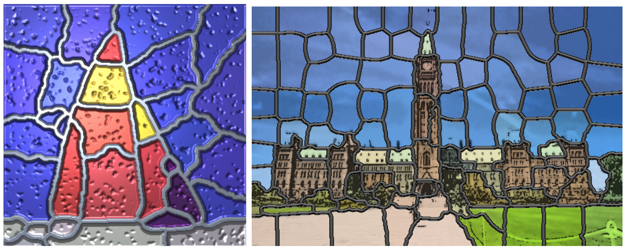

Figure 13: *Comparison of our method with Mould. Left: Mould. Right: ours.*

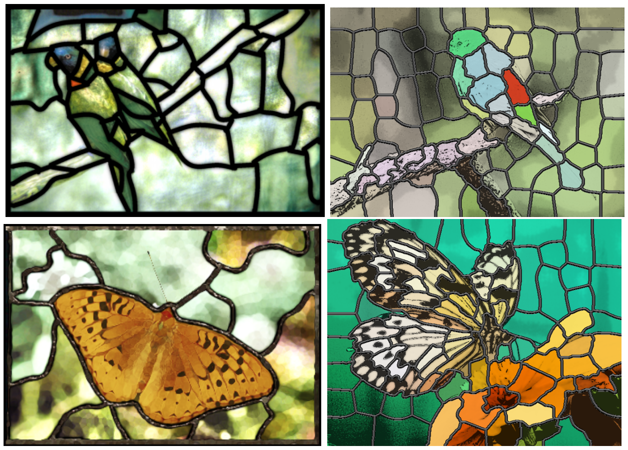

Figure 14: *Comparisons of our method with Brooks. Left: Brooks. Right: ours.*

Finally, we compare with Setlur and Wilkinson [SW06], who used mean shift segmentation and texture synthesis to create a stained glass effect. We can see the shortcomings of the mean shift for a stained glass application, as the tile sizes vary wildly and impractical shapes are suggested. Although we think our result is superior in this example, the comparison also serves to highlight some limitations in our approach. Difficulties arise in complicated regions of our image related to basing our automatic segmentation process on SLIC. These can include busy and highly textured regions, thin structures, and areas where segmented regions cut across elements of the image in a way that makes little sense from a human perspective. These problems become very noticeable in human subject matter. The family image in Figure 7 also displays some of these problems. For example, the splitting of the faces into multiple tiles is unnatural. Given the tile size, it would be more appropriate if each of the human heads were enclosed within its own tile. Here we can see the strength of the painted layer which helps create a coherent stained glass image even when the segmentation is unable to determine the content on its own. Portraits present an additional challenge; see Figure 15 and 16. Humans are well tuned to be critical of the visual presentation of faces. With our automatic segmentation, the glass tile boundaries can seem arbitrary – especially where there are no strong edges, as in the interior of the face. Typically, medieval stained glass did not attempt portraiture; faces were usually part of a larger scene and created at a scale such that they could be cut from a single piece of glass.

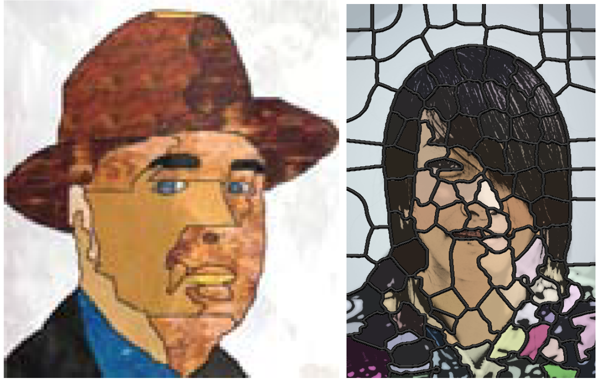

Figure 15: *Comparison of our method with Setlur on faces. Left: Setlur. Right: ours.*

Two interesting partial failures are presented in Figure 16. On the left we see some of the limitations of using SLIC as our initial segmentation method. While some of the tiles make sense, weak boundaries such as the shoulder and leg of the athlete on the right are not incorporated into the tile edges. The tile boundaries can potentially be improved by adjusting the compactness parameter m in SLIC to favor color over spatial distance. However, adjustments of this kind can result in SLIC clusters that are too irregular to be useful in stained glass. Figure 16, right illustrates again the difficulties with portraiture. In this case, colour averaging does not produce a uniform color over the tiles making up the face. Instead, the inclusion of darker hair and shadow pixels in the averaging process darkens the tiles, resulting in a patchwork appearance. We speculate that better results would be achieved by ignoring outliers, requiring a more sophisticated examination of the distribution of values within a tile.

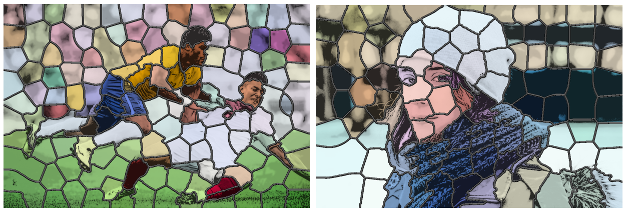

Figure 16: *Two failure cases. Left: athletes. Right: toque.*

Comparing our work to actual stained glass from the medieval period (see Figure 1) we can see some characteristics that we did not achieve. For instance, by relying on SLIC for our initial segmentation, we cluster pixels based on color and spatial relationships, yet ignore the semantics and the contextual relationships amongst the resulting tiles. Interactive content labeling could be useful in this area to provide greater control over where tile boundaries appear. Higher level design is also another aspect to consider. Often, windows are adorned with decorative elements such as floral or other thematic motifs. These are often painted onto simpler geometric tiles of glass and arranged in highly symmetric patterns. We made some initial steps in this direction by replacing the background with diamond shapes in Figure 9. However, there is still much left to do. Incorporating background window templates into our tiled image is one option.

5. Conclusion

We have presented a method for stained glass rendering, where colored tiles provide a coarse impression of the image content, and a separate paint layer shows details. The idea of painting on top of the glass was inspired by historical practice, where craftsmen commonly used paint to add needed details, but has not previously appeared in the NPR literature.

We used SLIC superpixels to create stained glass tiles. The SLIC clusters are much better suited to stained glass than the segments from the mean shift algorithm, as was previously used. The paint layer is created in two phases: a loose approximation to the input image is created by blurring a histogram-matched quantization of the input, and details are obtained from the XDoG filter and then added on top. Combined with colored tiles from the SLIC segmentation, we have an output image that both resembles stained glass and conveys the content of the original image.

Our algorithm is most effective when applied to images with a clear distinction between foreground and background, so that the segmentation boundaries usefully communicate some of the image content. For busier images, the paint layer is able to convey the image content to some degree. Still, the SLIC oversegmentation does not capture weaker or semantically inferred edges, and it tends to produce undesirably irregular tile boundaries in background regions of the image.

Some possibilities for future work remain. We can still adjust the paint layer somewhat; we uncritically took the XDoG output as our detail layer, but for some images, XDoG filtering produces large dark areas, unnecessary in this application. We would like to get tile shapes that are simpler geometrically so as to better reflect real stained glass images. Higher-order patterns and symmetry in the tile arrangement would be worthwhile, especially now that we are not relying so much on the tile shapes to communicate image content. At the same time, we would like to prevent tile boundaries from crossing through semantic features; for example, faces are rarely split by tile boundaries in historical stained glass. Adding more sophisticated image analysis and postprocessing of the tile shapes would be desirable; an interactive application for image labeling and tile editing would also be worthwhile.

Finally, we hope that the ideas in this paper can find applications beyond stained glass. In our observation, NPR has usually stuck to one medium at a time, but artists often create images by mixing several techniques, e.g., painting plus drawing. By allocating responsibility for different aspects of the final look to algorithms for different artistic media, we hope to be able to create more elaborate and attractive synthetic images.

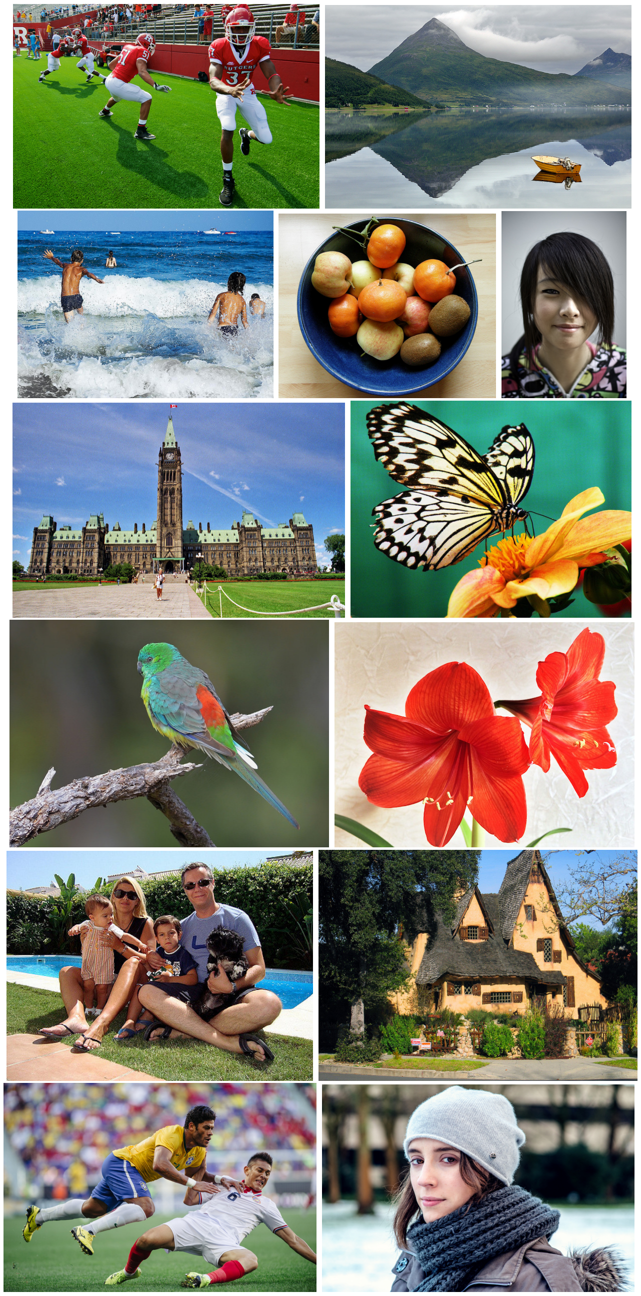

Figure 17: *Original images.*

Acknowledgements

Thanks to other GIGL members for productive comments and discussions, and thanks to the anonymous referees for many insightful comments. Funding for this work was provided by NSERC and by Carleton University.

We used many images from Flickr under a Creative Commons license. Thanks to the numerous photographers who provided material: M. Ignacio Monge García (family), Massimo Ankor (swimmers), Lori Branham (house), Richard Beaulieu (parliament), keepps (fruit), jimmcsweeney (butterfly), Flavijus (landscape), David Jenkins (bird), Elena Penkova (flowers), FLEE (face), rorrrorrrorr (goldfish), vtbrak (football), Nathan Congleton (athletes), and sicknotepix (toque).

References

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SÜSSTRUNK S.: SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 11 (2012), 2274 – 2282. URL: http://ivrg.epfl.ch/research/superpixels. 2

[BKTS06] BOUSSEAU A., KAPLAN M., THOLLOT J., SILLION F. X.: Interactive watercolor rendering with temporal coherence and abstraction. In *NPAR* (2006), pp. 141–149. 2

[BPD06] BAE S., PARIS S., DURAND F.: Two-scale tone management for photographic look. *ACM Trans. Graph. 25*, 3 (July 2006), 637–645. doi:10.1145/1141911.1141935. 2

[Bro06] BROOKS S.: Image-based stained glass. *IEEE Trans. Vis. Comput. Graph. 12*, 6 (2006), 1547–1558. 2, 7

[DS02] DECARLO D., SANTELLA A.: Stylization and abstraction of photographs. *ACM Trans. Graph. 21*, 3 (July 2002), 769–776. 2

[GDA*12] GERSTNER T., DECARLO D., ALEXA M., FINKELSTEIN A., GINGOLD Y., NEALEN A.: Pixelated image abstraction. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2012), NPAR '12, Eurographics Association, pp. 29–36. URL: http://dl.acm.org/citation.cfm?id=2330147.2330154. 2

[GW08] GONZALEZ R. C., WOODS R. E.: *Digital Image Processing*. Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2008. 4

[Hau01] HAUSNER A.: Simulating decorative mosaics. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 573–580. URL: http://doi.acm.org/10.1145/383259.383327, doi:10.1145/383259.383327. 2

[Kha] KHAN ACADEMY: How stained glass is made. URL: https://www.khanacademy.org/humanities/medieval-world/latin-western-europe/gothic1/a/how-stained-glass-is-made. 1

[Mou03] MOULD D.: A stained glass image filter. In *Proceedings of the 14th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), EGRW '03, Eurographics Association, pp. 20–25. URL: http://dl.acm.org/citation.cfm?id=882404.882407. 2, 4, 7

[SLKD15] SEMMO A., LIMBERGER D., KYPRIANIDIS J. E., DÖLLNER J.: Image stylization by oil paint filtering using color palettes. In *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging (CAe)* (2015), pp. 149–158. 2

[Sow81] SOWERS R.: *The Language of Stained Glass*. Timber Press, 1981. 1

[SRHC08] SONG Y.-Z., ROSIN P. L., HALL P. M., COLLOMOSSE J. P.: Arty shapes. In *Computational Aesthetics* (2008), pp. 65–72. 2

[SW06] SETLUR V., WILKINSON S.: Automatic stained glass rendering. In *Computer Graphics International* (2006), pp. 682–691. 2, 8

[Win11] WINNEMÖLLER H.: XDoG: advanced image stylization with eXtended Difference-of-Gaussians. In *NPAR* (2011), Collomosse J. P., Asente P., Spencer S. N., (Eds.), ACM, pp. 147–156. 2