Isoluminant Color Picking for Non-Photorealistic Rendering

Trân-Quân Luong† Ankush Seth† Allison Klein† Jason Lawrence‡

†School of Computer Science
McGill University

‡Department of Computer Science
Princeton University

Abstract

The physiology of human visual perception helps explain different uses for color and luminance in visual arts. When visual fields are isoluminant, they look the same to our luminance processing pathway, while potentially looking quite different to the color processing path. This creates a perceptual tension exploited by skilled artists. In this paper, we show how reproducing a target color using a set of isoluminant yet distinct colors can both improve existing NPR image filters and help create new ones. A straight-forward, geometric technique for isoluminant color picking is presented, and then applied in an improved pointillist filter, a new Chuck Close inspired filter, and a novel type of image mosaic filter.

Key words: Nonphotorealistic rendering, color halftoning, artistic dithering.

1 Introduction

In her book *Vision and Art: The Biology of Seeing*, Margaret Livingstone writes

> The elements of art have long been held to be color, shape, texture, and line. But an even more fundamental distinction is between color and luminance. Color (in addition to reproducing objects' surface properties) can convey emotion and symbolism, but luminance (what you see in a black and white photograph) alone defines shape, texture, and line. [18]

Physiologically, humans see the world in two different ways. Initially, color and luminance are processed separately by our visual system, and then this information is integrated to give us our perceived final image.[6] When two fields are isoluminant (i.e. possessing the same luminance values), they look the same to our luminance processing pathway, while potentially looking quite different to the color processing path. This creates a perceptual tension that skilled artists can exploit for quite astonishing effects. For example, the sun in Monet's painting *Impression Sunrise* is often perceived by viewers to pulse or shimmer because the sun's luminance is the same as that of the surrounding clouds even though the sun's hue is quite different! [18] (See figure 1 a and b)

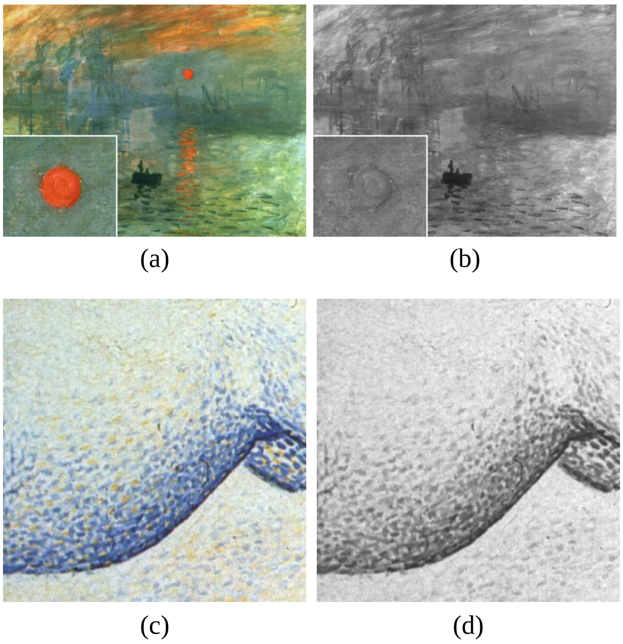


Figure 1: Skilled artists can use isoluminant colors to produce perceptual tension between the luminance and color processing pathways of the human visual system. In the sky of Impression Sunrise, Claude Monet selects colors with radically different hues(a) but near constant luminance(b). The result is a sun that appears to shimmer in the morning mist. [18] In (c), one region of Le Cirque, Georges Seurat uses isoluminant blue and yellow strokes(d) to both achieve the desired average color and to convey changes in surface orientation along the belly of a horse.

Skilled artists aside, most people find it difficult to detect or choose isoluminant colors with differing hues, probably due to the complexity of separating out luminance differences from those of hue and saturation.[13] However, computers can both detect and choose isoluminant colors quite easily. Therefore it is surprising that more work on incorporating isoluminance into computer graphics, and more specifically non-photorealistic rendering (NPR), has not been done. Often, the color picking in many NPR filters is just point or area sampling with some random perturbation. No specific measures are taken for maintaining input luminance values. Conversely, more general color picking algorithms in computer graphics usually try to maximize color fidelity to an input image given a fixed number of inks. Input luminance is maintained as a byproduct of color fidelity. Yet strict color fidelity is often not desired in NPR filters.

In this paper, we show how isoluminant color picking can be utilized to both improve existing NPR image filters and to create new ones. A straight-forward, geometric technique for isoluminant color picking in linear color spaces is presented and then applied in a pointillist filter, a new Chuck Close inspired filter, and a novel type of image mosaic filter.

Specifically, contributions of this paper include:

- A discussion of how isoluminant color picking can be used to improve NPR filters
- A method for isoluminant color picking
- An improved pointillist filter
- A new NPR filter modeled after a particular style of paintings by the artist Chuck Close
- A new filter for stylized image mosaics.

The remainder of this paper is organized as follows. In section 2, we discuss previous research. Section 3 presents our basic technique for picking isoluminant colors, and then demonstrates how this technique can be used to improve an existing pointillist filter. In sections 4 and 5, we apply isoluminant color picking to create a Chuck Close inspired filter and a new type of image mosaic filter, respectively. Finally, section 6 presents conclusions and possible areas for future work.

2 Related Work

This paper explores the role of color reproduction based on fixing luminance while allowing hue and saturation to vary. The primary applications presented here are in NPR. However, because we exploit the eye's tendency to integrate color over small spatial regions, our work also relates to multi-color halftoning and artistic screening – topics mainly studied in the context of offset printing.

Most painterly NPR image or video filters (e.g. [8, 10, 11, 17, 26]) focus on algorithms for producing realistic brush strokes or paint textures, choosing colors for these strokes by sampling the underlying image and then sometimes adding small, random perturbations to these sampled values. Automatic NPR renderings of 3D models (e.g. [4, 14, 19]) usually follow the same strategy, relying on the underlying material properties and stylized transformations of traditional analytic shading models to dictate color.

One exception is the work of [7, 9]. The authors develop a shading model for technical illustration based on the artistic concepts of *tone* and *temperature*, noting that artists often use gradients in tone (i.e. variations in hue but not luminance) to indicate changes in surface orientation under a limited luminance range. Their approach, however, does not provide a fully general, automatic method for selecting a finite set of isoluminant colors, but instead focuses more specifically on continuous tone variation. Artistic and art-historical literature[1, 18] more directly discuss the importance of a color's luminance and the role it plays in an artist's ability to convey "shape, texture and line," offering evidence of the benefits an automatic isoluminant color picking algorithm could provide NPR applications.

The work of Sloan *et al.*[28] focuses on developing a shading model based on a particular artist or artistic genre, and therefore relates to our Chuck Close and pointillist filters. However, [28] focuses on providing a convenient interface for specifying non-realistic shading models without directly addressing the benefit of constant luminance color selection.

Like previous work in print reproduction of color images[3, 21, 23, 29], this paper exploits the eye's ability to spatially integrate a multi-colored region. Our criteria for color selection, however, is different. With color picking for offset printing, the goal is often to select a small set of colors which can be combined to reproduce the entire image, and with the objective of minimizing visual artifacts in the final printed output. In our case, we select colors locally (i.e. within image regions) to intentionally produce visual artifacts at close range while converging to the desired image when viewed from a distance.

Two example filters in this paper (sections 4 and 5) are related to previous work in artistic screening[2, 21, 31] because we also use images or texture maps to control the spatial distribution of color within atomic image regions and we seek to combine traditional halftoning techniques with more aesthetic or expressive goals. However, our work differs in that we select colors in each distinct cell to intentionally disrupt the near visual field. Moreover, we select colors locally and independently for each region. In contrast, with multi-color halftoning and artistic screening, the fixed palette of colors is chosen such that both the local and global color matching is as close as possible to the original image.

The work of [24, 25] is an exception. They also present geometric color picking algorithms (based on pairs of colors) for linear color spaces. However, they intentionally avoid isoluminance for aesthetics in achromatic regions and for legibility in watermarking applications. Additionally, instead of maximizing color differences for aesthetic effect as we do, [24, 25] try to make these differences as uniform as possible across the entire image in order to achieve uniform readability. Consistent distance isoluminant pairs are an interesting area for future work with our technique.

Finally, a third category of related work is that of image mosaics[5, 15, 27] in which a target image is replicated by judiciously selecting from a library of images or textures based on shape attributes. These previous works also use color proximity as a selection criteria, and seek to minimize any modifications to the library images. This contrasts with our approach wherein we explicitly modulate our textures with colors that are chosen to be far from those of the target.

3 Isoluminant Color Picking and Pointillism

In this section, we present a formal analysis of isoluminant color selection in the context of pointillism. A general, target-driven method for choosing isoluminant colors is presented and then applied to improve existing pointillist filters.

Additive color theory was a primary motivation for pointillist (and also divisionist) artists. The pointillists believed that the "optical mixing" achieved by placing small points of color with strong, differing hues near each other resulted in brighter, more lively paintings than could be obtained through physical pigment mixing, a subtractive color process. [1, 16] In addition, the colors chosen not only averaged to the desired target *color*, but also often had the crucial property of all being close to the target *luminance*.[18] We can see an example of this technique by looking closely at the horse's belly in Georges Seurat's famous painting *Le Cirque.* (See figure 1 c & d)

Our technique is based on this pointillist approach of choosing a set of isoluminant yet visually distinct colors to reproduce a desired color. Given a target color C in a linear color space and a set of weights $\{w_1, w_2, \ldots, w_n\}$ that sum to 1, we wish to find some set of colors $\{C_1, C_2, \ldots, C_n\}$ such that

$$C = \sum_{i=1}^{n} w_i C_i \qquad (1)$$

where $\forall i, Lum(C_i) = Lum(C)$. Geometrically, the above equation means that for a given target color C, we find an n-gon on the same luminance plane as C such that the barycentric coordinates of C with respect to this n-gon match the weights $\{w_i\}$.

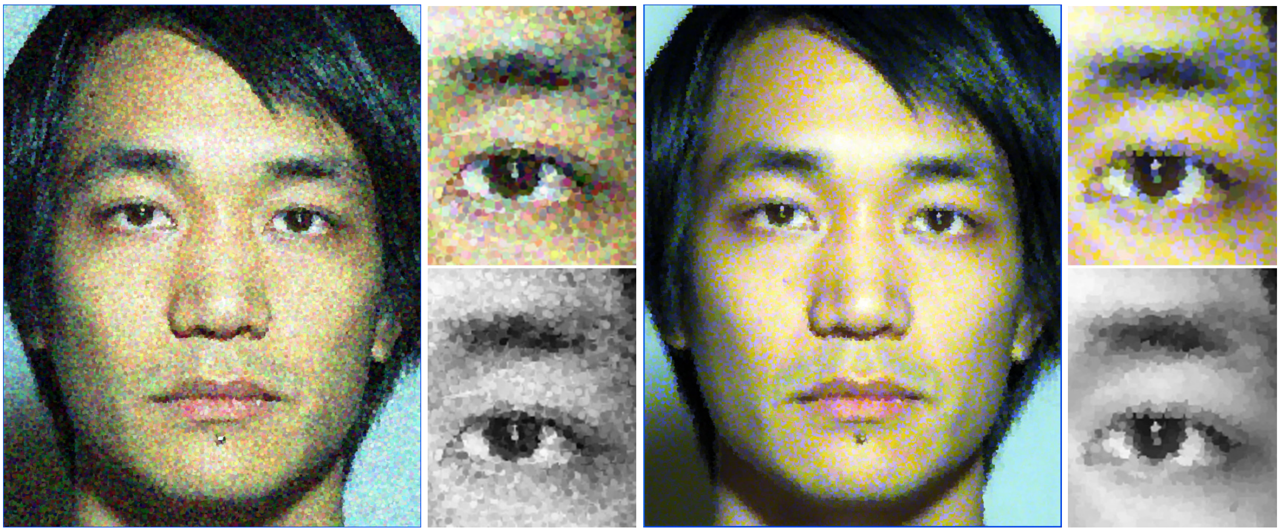
For any set of vertices that satisfies the above constraints, uniformly scaling or rotating the polygon within the luminance plane around C will result in a new polygon whose vertices also satisfy these constraints. (Care must be taken to guarantee these new vertices stay within the region of visible light.) Restated, this isoluminant condition alone yields a problem that is underconstrained – there is, in general, no single solution for the set $\{C_i\}$. A particular application might impose additional constraints or adopt a stochastic approach for generating sets of isoluminant colors.

This discussion naturally leads to a novel isoluminant pointillist filter where instead of performing point sampling in the input image and perturbing the colors randomly[10], one can generate clusters of points that average to the target while still maintaining isoluminance. In a prototype filter, we chose to use euclidean distance in the RGB cube as a reasonable approximation of perceptual distance – that is, we assume the size of the polygon with vertices $\{C_i\}$ will correlate with how much the colors differ both from each other and from the target color. We could have used other linear color spaces such as YIQ or LEF [20] that may have allowed for a more efficient implementation or better correlation with perceptual distance, but we leave such evaluation to future work.

For any source image S, we first divide it into a regular grid of small cells. For each cell, we calculate its average color, C, and its luminance, $Lum(C)$. We pick n colors with corresponding weights to approximate this cell's target color. For the pointillist filter, we choose four colors, each with equal weight $(1/4)$. In the RGB cube, this corresponds to an isoluminant square centered at C. The first color is chosen by picking a point isoluminant with C and at a some user-defined distance d from C. Given this initial starting point, the remaining three vertices of the square are fixed and can be calculated directly. We then uniformly scale this square, until any vertex reaches a boundary of the RGB cube. In most cases, this scale operation will increase the square's area, and thereby provide more color variation in the resulting vertices. However, for target colors near the boundary of the RGB cube, the initial square may need to be compressed to accomodate the limited range of isoluminant colors.

The initial starting point can either be chosen in a random direction, or it can be chosen at a fixed angle relative to C. We used the latter as this tended to produce the appearance of a fixed palette of colors within fields of near constant hue, and is therefore more similar to what artists usually do. (As an example, see Seurat's use of only two colors for the horse's belly in figure 1.)

For the pointillist images presented in this paper, the input images were divided into cells of four-by-four pixels. The initial distance d was $\sqrt{2}$, and we used points that were 5 pixels wide. When drawing our dots, we ensure there is neither significant overlap nor gaps in order to match our weight choices. Our implementation of the filter takes about one second to process the input image.



(a) Perturbed Sampling

(b) Isoluminant Color Picking

Figure 2: Pointillist filter comparison. Here we show (a) the perturbed sampling approach based on [10] and (b) our isoluminant technique described in section 3. Observe that in our approach, individual points are distinctly colored (top right), yet the isoluminance criteria are still met (bottom right). The latter is not true for the perturbed sampling approach. (Readers of the pdf are encouraged to zoom in on both images.)

Some results can be seen in figure 2.

We can make some initial observations. Haeberli's algorithm[10] does not adhere to luminance constraints nor does it focus on choosing colors that are far from individual point samples while still blending to give the desired colors.[1] In figure 2, we can observe that without local luminance control, the resulting imagery possesses a "grainy" quality. In contrast, because our technique targets both local luminance fidelity and strong color variation, it is closer to the pointillist approach. In most areas of the image, we achieve both nice color variation and luminance fidelity. However, in areas where the target color is close to the RGB cube boundary, our technique only maintains the luminance constraint, yielding dot colors that are very near the target. In these areas, performing a linear interpolation between two colors or relaxing the luminance constraint would produce larger color variation. We notice, though, that the pointillists sometimes chose lower color variation in highly saturated areas, so our current approach may be consistent with original art.

Related to the ideas of [25], controlling the d parameter directly could be useful for specifying the visual divergence of isoluminant color sets at near viewing distances. Also, the algorithm will work with greater or fewer colors and varying weights.

4 Chuck Close

In this section, we extend the isoluminant color picking algorithm from the previous section to create a novel image filter that renders portraits in the "blobby" style of Chuck Close.

Close's portraits are created by photographing the subject, subdividing both the source photograph and the destination canvas into diamond-shaped grid cells, and then reproducing each cell in the source photograph as a collection of colored, concentric blobs. The canvases are quite large (e.g. 8.5 feet high by 7 feet wide). When viewed from far away, they resemble the original photo due to spatial integration of color and the perceptual tendency to concentrate on the larger shape of the face, not the smaller shapes of the individual blobs.[22] However, when viewed from a near distance, Close's use of color, isoluminance, and texture clutter the visual field [18], invoking "a competition between the face and its constituent blocks to engage our perception of shape from shading."[22]

Part of the magic in Close's work is the knowledge that these paintings are created by hand, with no aids other than Close's creativity and talent. A Chuck Close original will always be unique and valuable. However, the creation of an automatic filter for this style allows us to both better understand Close's work and to produce our own portraits, which is appealing given that most of us will never be the subject of an original Close painting.

[1] The Adobe Photoshop pointillist filter suffers from these same drawbacks. It is our belief that the Adobe Photoshop pointillist filter is an implementation of [10], though we cannot be positive since it is proprietary.

If we had a library of source-destination pairs, we could use a machine learning technique such as [12] to produce such a filter. However, [12] has difficulty replicating large scale structures like those found in Close's grid cells, particularly joined cells. In addition, a library of source-destination image pairs is not readily available.

We begin by identifying some conventions in Close's work:[2]

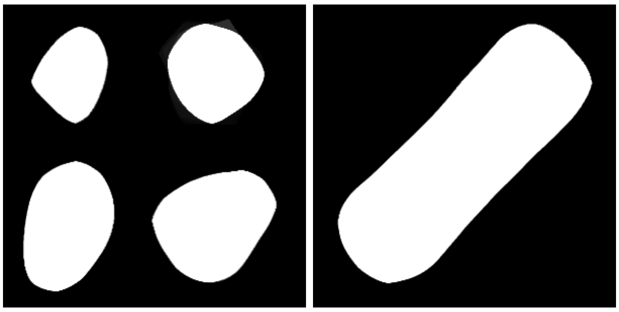
1. Each target cell contains one or two nested quads and then three to ten nested blobs.
2. Close often starts by filling the grid cell (the first quad) with an arbitrary color, and then successively adding layers to move closer to the target color.[30] These nested quads and blobs often alternate in luminance and/or color, having fewer luminance levels than color levels, typically only two or three.
3. If there is no strong edge in a source cell, the target cell usually contains round blobs. Otherwise, the target cell holds elongated vertical, horizontal, or diagonal blobs depending upon the underlying edge information in the source.
4. Diagonal blobs in adjoining cells that have similar average colors may merge across the shared flat edge. In general, two to three cells at a time may be joined this way, forming a single, larger diagonal blob, or, occasionally, a "vee." Vertical and horizontal blobs are never joined.
5. Within large areas of little or no edge information (such as the portrait background), Close often joins diagonal cells in the above manner to eliminate uniformity in the visual field.

We approximate these properties using steps described in the next two subsections, with the complete algorithm described in section 4.3.

4.1 Joins and Shape Assignment

We begin by filtering the image with a Canny edge detector. We then compare each cell's average edge magnitude with a fixed threshold to decide if the cell has a strong enough edge in order to address rules 3 and 4. We also randomly sample cells to see if they are in large, flat areas, and thus should be joined according to rule 5.

Using this edge information, we assign five layers (rule 1) of the appropriate shapes (rule 3) to each cell. The first two layers are quads and the 3 remaining layers are nested blobs, rendered as alpha-blended texture-mapped quads (see figure 3). At run time, each quad will be modulated with the appropriate layer colors and rendered from outermost to innermost.



Example Round and Elongated "Blob" Masks

Figure 3: To capture the stroke style of Chuck Close we modulate scaled and rotated texture masks with different colors in each diamond cell. Elongated blobs can be used for rendering joined cells.

4.2 Calculating Layer Colors

For each cell, we choose the five layers colors such that layers 1 and 3 are isoluminant, as are layers 2 and 4 (rule 2). Because the layers vary in size, we calculate their relative areas in each cell as follows: For each layer L in a given cell, we render it modulated with white, while rendering all the other layers in black. Then, we sum up the intensities of all pixels in this cell. The percent of the cell covered by L is simply the current summed intensity divided by the cell's total area in pixels. This percent coverage is L's weight in the color picking algorithm.

Using these layer weights, we can compute the layer colors in the following manner: given a target color C and a set of five weights $\{w_i\}$, we determine two colors C_1 and C_2 that satisfy the following:

$$C = w_0 L_0 + (w_1 + w_3)C_1 + (w_2 + w_4)C_2,$$

where L_0 is the randomly chosen color for the cell's outermost layer (rule 2).

Then we modify the technique from section 3 to generate isoluminant pairs for unequal weights. Specifically, for colors C_1 and C_2, we wish to generate layer colors $\{L_i\}$ such that:

$$\begin{aligned} C_1 &= (w_1 L_1 + w_3 L_3)/(w_1 + w_3) \\ C_2 &= (w_2 L_2 + w_4 L_4)/(w_2 + w_4) \\ Lum(L_1) &= Lum(L_3) \\ Lum(L_2) &= Lum(L_4), \end{aligned}$$

where the distance between the isoluminant pairs is maximized. The implementation is to randomly choose L_0 and test if a solution exists by checking if $(C - w_0 L_0)(1 - w_0)$

[2] In order to respect copyrights, we are unable to reproduce any Chuck Close imagery in this paper. Readers are encouraged to view the Lyle portrait or other portraits in [30] in order to verify these rules for themselves.

is a valid color. If the test fails, then a new L_0 is randomly chosen. Similarly, we randomly choose a new C_1 and then test if $C - w_0L_0 - (w_1 + w_3)C_1$ is inside the RGB cube. For each C_1, we iterate over all possible L_1's. Each L_1 uniquely determines an L_3, and we choose the maximally distant isoluminant pair that is still valid. Since each C_1 uniquely determines C_2, we can proceed similarly to determine the remaining colors. We continue iterating over choices for C_1 until we find a valid set of colors. It is also possible to maximize over *all* color pair distances, but in practice we found the results to be visually similar with the drawback of significantly longer computation times.

4.3 Algorithm

We can now put all of the above pieces together to state the final algorithm:

1. Grid off the source image.
2. For each grid cell, calculate its average color, and dominant edge orientation.
3. Join grid cells with a shared edge if they have the same diagonal edge direction and similar average colors. Also, randomly join cells in large, edgeless areas. (For all remaining steps of the algorithm, joined cells will be treated as a single cell.)
4. For each cell:
   (a) Assign a set of nested blob textures of the appropriate shape.
   (b) Calculate the relative weight of each layer texture as in subsection 4.2.
   (c) Given the relative layer weights and the cell's average color, calculate the layer colors as described in subsection 4.2.
   (d) Render the cell with each layer texture modulated by its corresponding layer color.

For the images presented in the paper, our grids are 24 by 29 cells (chosen based on Close's own numbers), and we use input images cropped to have the same aspect ratios as original paintings. The number of grid cells is a parameter that can be changed by the user, as is the threshold for determining edges in rule 3 above. In practice, this entire computation is fast, taking less than 2 seconds for the images presented in this paper.

Figure 4 shows some of our results. Note that our approach does not do anything special to match blob shapes with features, as Close often does around the eyes or nostrils. Also, since we do not restrict our color palette at all, we sometimes get colors (such as lime green) that do not seem to be used by Close. A machine learning technique could help ameliorate this. Nevertheless, we believe our results support our hypotheses from the beginning of section 4 and are a nice approximation to the real thing. In addition, now that we have distilled some rules about this style, we can apply them to a new hybrid style, presented in the next section.

5 A Novel Image Mosaic Filter

We finish by presenting a novel type of image mosaic that combines ideas from photomosaics[27] and our Chuck Close filter. Specifically, we categorize small images by shape (as in rule 3 of section 4), and then coarsely segment these images to create cell layers. (See figure 4.) Each layer receives a color based on its relative area within the cell. The colors chosen can either all be isoluminant or can have a smaller number of luminance levels than colors, as in the previous section. In the images shown, we used the latter approach, effectively applying the color picking algorithm from section 4 to the segmented image layers. Results can be seen in figure 4.

Because a large library of source imagery is not required for our technique, this is an easy-to-use method for generating complex image mosaics. Both tile selection and color picking are fast, so image generation generally takes less than 2 seconds. Moreover, we believe the results are visually intriguing and fun.

6 Conclusions and Future Work

We have shown how isoluminant color picking can be used to both improve existing NPR image filters and to create new ones. We present a fast, geometric technique for isoluminant color picking in additive linear color spaces, and then apply this technique to an improved pointillist filter, a new NPR filter in the style of Chuck Close, and a novel stylized image mosaic filter.

Because restricting a set of colors to the same luminance plane does not itself constrain the process of color selection, we believe there are many future directions of work that investigate additional constraints (e.g. incorporating a particular artist's palette) and novel algorithms for estimating a "desirable" set of isoluminant colors.

Thus far, we have primarily explored isoluminance as a tool for creating "layered" imagery, i.e. imagery which has visual interest or meaning at various viewing distances. For future work, isoluminant color picking could help augment previous techniques in painterly rendering. Instead of simply sampling the underlying image, colors in neighboring strokes or image segments could be specifically chosen to have more or less isoluminance, directing the reader's attention towards or away from certain features, much as Monet did. Such techniques would also be useful for applications outside of NPR, such as increasing or decreasing the perceived delineation between text and its background.

In addition, because luminance helps convey the surface orientations of 3D objects, this suggests a mechanism for deliberately exaggerating or minimizing the sensation of depth in both NPR and photoreal imagery. Our visual processing mechanisms expect the scene data given by the chrominance and luminance channels to be generated by the same underlying physical scene. By deliberately decoupling these signals, we can achieve new visual styles and effects.

Acknowledgements

We wish to thank Victor Ostromoukhov for introducing us to Dr. Livingstone's book, Soukia Savage for providing input photography, and Neema Moraveji for inspiring conversation about Close's portraiture. This work was supported by an NDSEG fellowship and generous donations from ATI corporation.

References

[1] Josef Albers. *Interaction with Color*. Yale University Press, 1971.

[2] John W. Buchanan. Special effects with half-toning. *Computer Graphics Forum*, 15(3):97–108, August 1996.

[3] S.M. Chosson and Roger D. Hersch. Visually-based color space tetrahedrizations for printing with custom inks. In *Symposium Electronic Imaging 2001, Color Imaging*, pages 81–92, 2001.

[4] Phillippe Decaudin. Cartoon-looking rendering of 3d-scenes. Technical Report INRIA 2919, Universite de Technologie de Compiegne, France, June 1996.

[5] Adam Finkelstein and Marisa Range. Image mosaics. *Lecture Notes in Computer Science*, 1375, 1998.

[6] E. Bruce Goldstein. *Sensation and Perception*. Brooks/Cole Publishing, 1999.

[7] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A non-photorealistic lighting model for automatic technical illustration. *Proceedings of SIGGRAPH 98*, pages 447–452, July 1998.

[8] Bruce Gooch, Greg Coombe, and Peter Shirley. Artistic vision: painterly rendering using computer vision techniques. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 83–ff, 2002.

[9] Bruce Gooch, Peter-Pike Sloan, Amy Gooch, Peter Shirley, and Richard Riesenfeld. Interactive Technical Illustration. *Interactive 3D Conference Proceedings*, April 1999.

[10] Paul Haeberli. Paint by numbers: abstract image representations. In *Proceedings of SIGGRAPH 90*, pages 207–214, 1990.

[11] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of SIGGRAPH 98*, pages 453–460, 1998.

[12] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. *Proceedings of SIGGRAPH 2001*, pages 327–340, August 2001.

[13] Peter K. Kaiser and Robert M. Boynton. *Human Color Vision*. Optical Society of America, 1996.

[14] Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, and Adam Finkelstein. Wysiwyg npr: drawing strokes directly on 3d models. In *Proceedings of SIGGRAPH 02*, pages 755–762. ACM Press, 2002.

[15] Junhwan Kim and Fabio Pellacini. Jigsaw image mosaics. In *Proceedings of SIGGRAPH 02*, pages 657–664, 2002.

[16] Ellen Wardwell Lee. *The aura of neo-impressionism : the W.J. Holliday Collection*. Indianapolis Museum of Art, 1983.

[17] Peter Litwinowicz. Processing images and video for an impressionist effect. In *Proceedings of SIGGRAPH 97*, pages 407–414, 1997.

[18] Margaret Livingstone. *Vision and Art: The Biology of Seeing*. Harry N. Abrams, 2002.

[19] Barbara J. Meier. Painterly rendering for animation. In *Proceedings of SIGGRAPH 96*, pages 477–484. ACM Press, 1996.

[20] Victor Ostromoukhov N. Rudaz, Roger D. Hersch. Specifying color differences in a linear color space (lef). In *Proceedings of IS&T/SID Color Imaging Conference*, pages 197–202, 1997.

[21] Victor Ostromoukhov and Roger D. Hersch. Multi-color and artistic dithering. In *Proceedings of SIGGRAPH 99*, pages 425–432, 1999.

[22] Denis G. Pelli. Close encounters – an artist shows that size affects shape. *Science*, 285:844–846, August 1999.

[23] Joanna L. Power, Brad S. West, Eric J. Stollnitz, and David H. Salesin. Reproducing color images as duotones. In *Proceedings of SIGGRAPH 96*, pages 237–248, 1996.

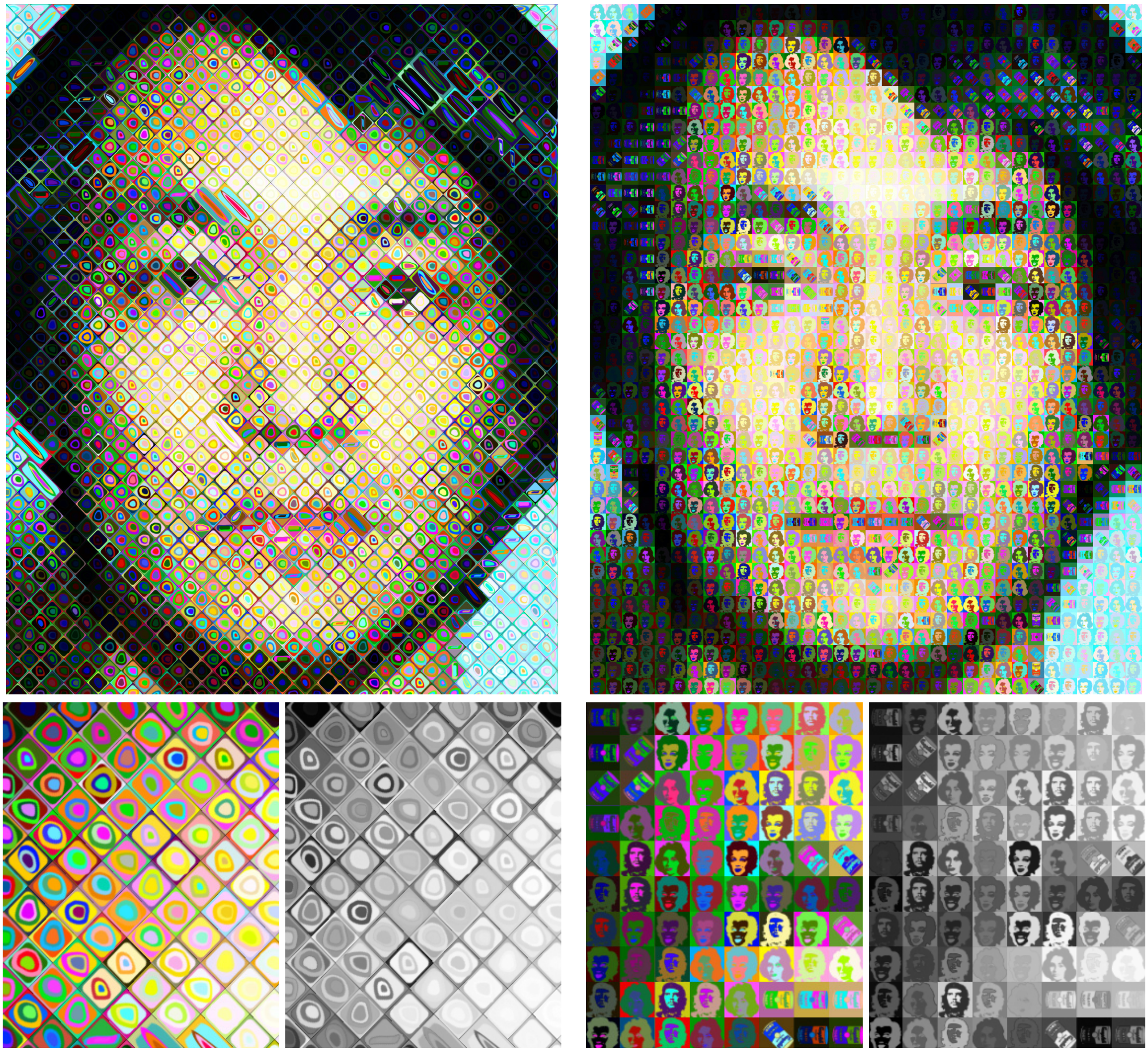
[24] N. Rudaz. *Incrustation De microstructures par Écarts Chromatiques*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, May 2003.

[25] N. Rudaz and R.D. Hersch. Protecting identity documents by microstructure color differences. *Journal of Electronic Imaging*, Vol. 13(2):315–323, April 2004.

[26] Michio Shiraishi and Yasushi Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 53–58, 2000.

[27] Robert Silvers and Michael Hawley. *Photomosaics*. Henry Holt and Co., Inc., 1997.

[28] Peter-Pike Sloan, William Martin, Amy Gooch, and Bruce Gooch. The lit sphere: A model for capturing npr shading from art. *Graphics Interface 2001*, pages 143–150, 2001.



(a) *Chuck Close* Filter Output

(b) Image Mosaic Filter Output

Figure 4: Images created by our Chuck Close filter and our stylized image mosaic filter. Enlargements enable us to see the results of the color picking algorithm, specifically the isoluminant color pairs.

[29] Eric J. Stollnitz, Victor Ostromoukhov, and David H. Salesin. Reproducing color images using custom inks. In *Proceedings of SIGGRAPH 98*, pages 267–274, 1998.

[30] Robert Storr. *Chuck Close*. Museum of Modern Art, 1998.

[31] Oleg Veryovka and John W. Buchanan. Halftoning with image-based dither screens. *Graphics Interface '99*, pages 167–174, June 1999.