EUROGRAPHICS 2018 / D. Gutierrez and A. Sheffer
(Guest Editors)

Volume 37 (2018), Number 2

Watercolor Woodblock Printing with Image Analysis

A. Panotopoulou[1], S. Paris[2] and E. Whiting[3]

[1]Dartmouth College [2]Adobe Research [3]Boston University

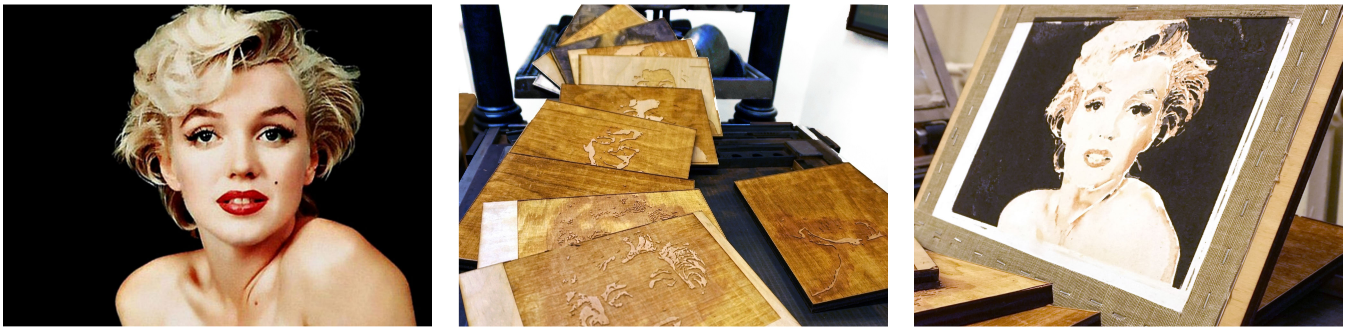

(a) *Input photograph* **(b)** *Laser cut woodblocks* **(c)** *Resulting watercolor print*

Figure 1: *A Marilyn Monroe photograph was used as an input to our pipeline (a), with our hybrid fabrication process constituted by watercolor paint and lasercut woodblocks (b). We created the final painting (c) through layered prints of the individual woodblocks.* (*Photograph by Milton H. Greene ©2018 Joshua Greene www.archiveimages.com*)

Abstract

Watercolor paintings have a unique look that mixes subtle color gradients and sophisticated diffusion patterns. This makes them immediately recognizable and gives them a unique appeal. Creating such paintings requires advanced skills that are beyond the reach of most people. Even for trained artists, producing several copies of a painting is a tedious task. One can resort to scanning an existing painting and printing replicas, but these are all identical and have lost an essential characteristic of a painting, its uniqueness. We address these two issues with a technique to fabricate woodblocks that we later use to create watercolor prints. The woodblocks can be reused to produce multiple copies but each print is unique due to the physical process that we introduce. We also design an image processing pipeline that helps users to create the woodblocks and describe a protocol that produces prints by carefully controlling the interplay between the paper, ink pigments, and water so that the final piece depicts the desired scene while exhibiting the distinctive features of watercolor. Our technique enables anyone with the resources to produce watercolor prints.

CCS Concepts

•***Human-centered computing*** → ***Interface design prototyping;*** •***Computing methodologies*** → *Image processing; Image representations;* •***Applied computing*** → *Fine arts;*

1. Introduction

Watercolor is a popular form of painting that mixes delicate color gradients with finely detailed contours and diffusion patterns that are immediately recognizable. This unique, abstract look is created by diluting the pigment in water and using a brush to deposit the paint on absorbent paper. Several complex physical phenomena occur during the painting process such as the diffusion of the pigments through the wet paper and the progressive absorption and drying of the water, which combined together generate the visual features characteristic of watercolor. Unfortunately, mastering this technique is challenging and typically out of reach of most people. Further, even for a trained artist, producing several copies of a painting is a tedious task because one needs to manually reproduce all the steps.

Several options have been proposed to overcome this challenge. For instance, one can make copies by scanning an actual painting and using a standard 2D printer. However, this approach creates identical replicas that have lost the characteristic uniqueness of actual paintings. Further, one still has to create the first painting using the standard process. Another approach is to rely on watercolor simulation software, which brings the convenience and controllability of digital authoring to the creation process but sacrifices its physicality. The latter may limit the ability to reproduce all the features of watercolor and, even if one can introduce variation through randomization, each copy is unique only insofar as one does not print each piece more than once. Closer to our work, lithography is a process where the artist first creates an etched stone that is later used to print several copies. Since the print process is manual, each print is subtly different from the others, thereby preserving the unique nature of the work. However, lithography does not apply to watercolor that is the focus of our paper.

In this paper, we propose a new solution in this design space. We enable users to create watercolor prints using a photograph and minimal guidance as input. We first introduce an image analysis technique to determine the main color regions of the image and their level of sharpness. We design this step so that users can easily provide guidance if needed, e.g., to control the level of abstraction of the final print. Based on these data, we then fabricate woodblocks and define the paints and the presence of water on each of them. Our use of woodblocks replaces brushes in the traditional watercolor painting technique, providing a higher level of automation. Finally, we describe a protocol to realize the final print. In our prototype, this last printing step is done by the users themselves but it is fully specified, i.e., it does not involve subjective judgment during the process, and we envision that it could be performed by a robot in the future. We provide a variety of examples created with our approach to demonstrate that it successfully reproduces the characteristic look of watercolor without requiring any painting skills from the users. Recognizable characteristics of watercolor paintings are present in our results: the paper texture, the variability of the thicknesses on the deposited pigments, and the broad gamut of the produced colors.

1.1. Related Work

Various techniques have been developed for the computational fabrication of paintings. Researchers have used a robotic arm to paint strokes for a given input image [DLPT12, LPD13, TL12]. Iarussi et al. [IBT13] presented a guidance drawing tool. Yao et al. [YS05] focused on robot controlled Japanese brush painting techniques specific to bamboo imagery. Prevost et al. [PJJSH16] proposed a system for painting large-scale images using spray paint based on a photograph. Our technique is on a smaller scale and aims to create effects specific to watercolors such as diffusion between colors. Flagg et al. [FR06] propose another system of guided painting, this time for oil painting, which incorporates a color mixing model. Their system is intended as a learning tool and its results rely upon the painter. We propose a technique that produces results purely contingent upon the procedure itself.

Related to fabrication of paintings, non-photorealistic rendering aims to generate stylized compositions that reflect artistic intent [Bot09, BNTS07, HE04, WWF*14, LC14, CAS*97, SLKD16]. Projects about simulation of different painting media, including watercolor, create a virtual painting or investigate the properties of the media [CKIW15, LBDF13]. We aim to fabricate an artistic version similar to the non-photorealistic rendering based on the input photograph. At the same time, we try to understand the properties of the media we use in order to use them efficiently.

Printing and pressing techniques have also been used in computational fabrication research for broader applications. Hydrographic printing has been used to physically texture map free form 3D objects [ZYZZ15, PDP*15]. Similarly, thermoforming transfers predistorted images onto a plastic sheet, which are then deformed in to a 3D shape [SPG*16]. More generally, fabrication research has led to creative control in a variety of domains, such as flyability [UKSI14], floatability [WW16], fabricating musical instruments [UMII10], or mechanical parts [MLB12].

1.2. Background

Our work builds upon watercolor painting and woodblock printing, two well established techniques. We summarize their main components below before describing our methodology in the next section.

Woodblock Printing. The technique of woodblock printing originated from East Asia as a form of relief printing [Bul] and has been used for well-known pieces like The Great Wave off Kanagawa by Hokusai. The artist carves the negative of the desired image on the surface of a flat piece of wood called *woodblock*. Then ink is applied to the raised area and the woodblock is pressed on a paper canvas. The ink used in the traditional East-Asia technique is water-based while western woodblock printing inks are often oil-based. In this work, we build upon the former and extend it to generate watercolor prints. We propose a computational approach to the design of the woodblocks. We also describe how to generate color gradients using water diffusion instead of painting them manually on the woodblocks as done traditionally.

Watercolor Painting. Compared to other painting techniques, watercolor stands out by its use of pigments that are suspended in water. This enables characteristic visual effects based on the flow of the pigments in the water and on the diffusion of the water on the paper. One way painters control the flow of water is with *masking fluid*, a paint-like liquid that stops the water flow. Artists also use pigment diffusion to create color gradients. They brush a thick layer of water on the paper and deposit paint of different colors at different locations. The progressive diffusion of the pigments in the water creates a smooth color gradation and at the same time produces subtle patterns due to the interplay with the paper. Our work is inspired by these two techniques and we describe a systematic approach for deciding where to put masking fluid and for creating color gradients using water and pigment diffusion.

Color Formation. Watercolor painting follows a subtractive color model, similar to standard CMYK printing. Watercolor painting uses freeform strokes and varies the concentration of paint in water to achieve different transparency levels and colors. Unlike standard printing that relies on halftoning to generate a large spectrum of colors from a small set of inks, we continuously mix the inks using water diffusion, which also generates the gradient patterns characteristic of watercolor.

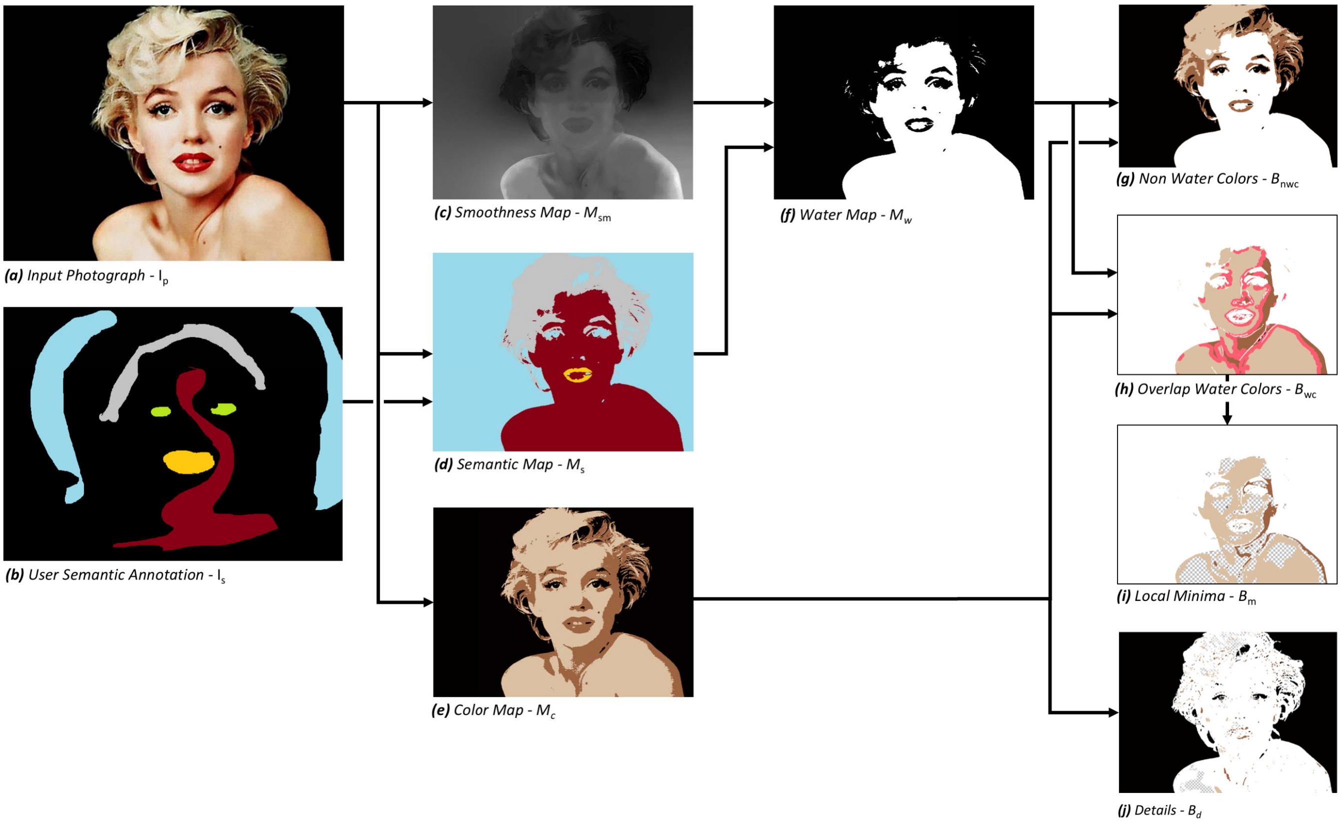

(a) Input Photograph - I_p
(b) User Semantic Annotation - I_s
(c) Smoothness Map - M_sm
(d) Semantic Map - M_s
(e) Color Map - M_c
(f) Water Map - M_w
(g) Non Water Colors - B_nwc
(h) Overlap Water Colors - B_wc
(i) Local Minima - B_m
(j) Details - B_d

Figure 2: *Image analysis Pipeline. The inputs are the original photograph*[†] I_p *(a), and the user semantic annotation* I_s *(b). The colors in the output images represent print colors depicted in (d), the white and the chess pattern in (g)-(j) correspond to non printed areas.*

1.3. Overview

The input to our technique is a digital image, I_p, and user-provided annotations, I_s, of the process. We designed a pipeline having in mind the abstraction of watercolor paintings. First, we analyze the input image to determine how each region should be represented in the final print. We compute a palette composed of a few paint colors and determine where each color will be applied. We also estimate the local smoothness of the image and use it to locate where to put water. We combine these pieces of information to define a set of regions where we will place paint or masking fluid. We give the shape of these regions as input to the laser-cutter to carve the woodblocks. Finally, we use these blocks to apply the paints and masking fluid onto white paper. While this step is manual, it is fully deterministic and could be performed by a robot. However, the design of such robot is beyond the scope of our work. Instead, we describe a step by step procedure, in which no subjective evaluation is required at any point

2. Image Analysis

This section describes our image analysis methodology to generate the components of our watercolor prints. We later use the output to fabricate the woodblocks for printing.

2.1. Overview of our Image Analysis Pipeline

The input to our technique is a digital image, I_p, and user-provided annotations, I_s, indicating the main semantic regions that the user would like the final print to retain. For example in Figure 2(a), the user selects distinct regions for the background, hair, skin, eyes and lips. Using these inputs, the pipeline of our image processing algorithm consists of the following steps (Fig. 2):

1. Decomposition of the input image into a small set of color regions that we use to determine the color of the paint at each location.
2. Estimation of the local smoothness of the input image.
3. Decomposition of the image into semantic regions based on the user's annotation. We use this information and the local smoothness to locate where to put water.
4. Detailed analysis of the variations of color and intensity in the water regions, which we use to generate watercolor diffusion patterns that approximate the content of the input image.
5. Optional: characterization of the fine details not represented by the previous step so that we can print them over the diffusion areas once the water has dried.

[†] Photograph by Milton H. Greene ©2018 Joshua Greene www.archiveimages.com

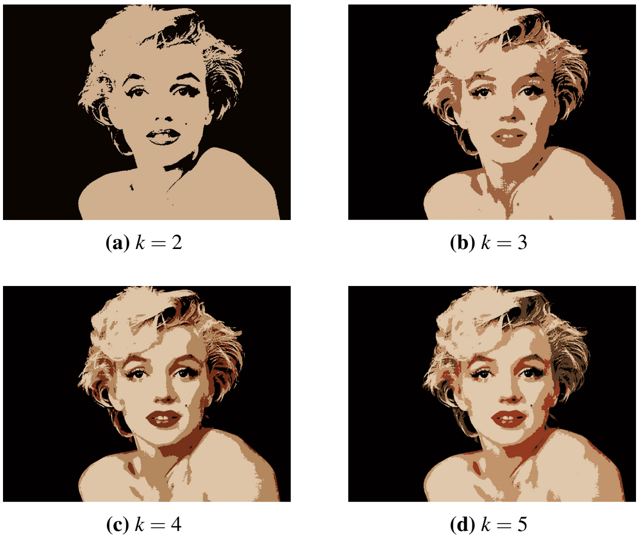

(a) $k = 2$ **(b)** $k = 3$ **(c)** $k = 4$ **(d)** $k = 5$

Figure 3: *The user can control the level of abstraction by varying the number of colors, k, in the color palette extraction.*

Throughout this process, we create *maps* that store intermediate data, e.g., a *color map* for the representation with a small color palette and a binary *water map* to indicate where to put water. The output of our image analysis is a set of 2D shapes in the image plane describing the woodblocks that we will use for creating the print. We will send these shapes to a laser-cutter to fabricate the woodblocks.

2.2. Color Map - M_c

While watercolor exhibits a broad spectrum of colors, they are all produced with a small set of paints — the color diversity comes from mixing, diffusion, and dilution. The first stage of our image analysis pipeline determines this small set of paint colors, *the color palette*. We use the palette extraction algorithm of Chang et al. [CFL*15] which takes the photograph bitmap file and the desired number k of colors in the palette as input, and outputs the actual palette colors. This variant of the k-means algorithm uses an acceleration data structure, where k is the number of colors in the output palette. In our context, the number of colors k controls the level of abstraction of the final watercolor print. A low k value represents the input image with few colors and generates a more abstract output, whereas a high k value preserves more details. We typically set k between 3 and 5 (Fig. 3). Next, we compute the *color map* M_c by assigning each pixel to its closest color in the palette in the CIE-Lab color space. This map defines the color of the paint at each location of the final print. The next step is to determine where we will apply water.

2.3. Dry and Wet Areas

Now that we have computed the color of the paint at each point, we decide where to apply water to create diffusion gradients. In "dry" areas, we use watercolor paint with a minimal amount of water so that no diffusion occurs. This is the desired behavior for printing fine details like the eyes of a portrait. In comparison, "wet" areas favor diffusion and are well suited for smooth color gradients. Because of the diffusion process, controlling the output in these regions is more challenging and we introduce a dedicated multi-step process for that purpose. In the rest of this section, we first describe how we determine where to put water, then briefly discuss how we handle the dry areas, and finally discuss in detail the more challenging case of the wet areas.

2.3.1. Where to Put Water

We wish to use water diffusion in smooth regions of the input. To do so, we first estimate a *smoothness map*, M_{sm}, that quantifies the local image smoothness and then describe how we guide users to decide where to put water based on this information. We call the output of this step the *water map*, M_w.

Smoothness Map – M_{sm} We use the algorithm of Zhang et al. [ZWJ*16] to estimate the local smoothness in the input image. This technique has been designed for characterizing defocus blur and we found that it generalizes well to locate smooth regions in general, e.g., it successfully identifies the Marilyn Monroe's skin as smooth (Fig. 2c, brighter values are for smoother regions). The algorithm analyzes the gradients in the image and outputs a continuous sharpness value at each pixel, we call this result the *smoothness map* M_{sm}. We discuss the parameter settings in the section about implementation details.

For our application, we seek to convert these continuous values into binary ones that indicate where to put water. This is a critical decision because the transition between a dry and a wet region creates a visible edge that, if misplaced, generates unsightly results. We experimented with options like per-pixel thresholding and superpixel-based thresholding, and found that they do not prevent such unsightly edges. These early tests also suggested that the location of the diffusion gradients is in part a subjective choice and that some degree of user controllability is desirable. The next section describes our solution motivated by these observations.

Semantic Map – M_s To ensure that wet–dry edges appear at visually adequate locations, we let the user define the meaningful regions of the input image, e.g., the sky and the trees in a landscape, and later only allow dry–wet edges to appear between two such regions. In the previous example, this guarantees that no edge can appear in the middle of the sky for instance, but allows for an edge between the sky and the trees. For this purpose, we use the Dense-CRF algorithm [KK11]. This technique takes the input image and a set of user-provided scribbles, one for each region they wish to define, and it returns a dense map where each pixel is assigned to a region according to criteria like color similarity and image distance between pixels. In our experiments, we found that regions representing semantic image elements such as skin, eyes, and mouth for a portrait, or sky, trees, and ground for a landscape yields satisfying results. Based on this observation, we call this output of the Dense-CRF algorithm the *semantic map*. We refer to the original article for the details of the algorithm and to Section 4.1 for our parameter settings. Semantic image segmentation has also been incorporated in previous work in rendering [ZZ10] and physical painting [LSD16].

Water Map – M_w Using the semantic map, we next define a binary map that indicates where to put water, the *water map*. To ensure that dry–wet edges only appear along semantic boundaries, we assign a single smoothness value to each semantic region by averaging the smoothness map M_{sm} over each region. Without the Smoothness Map, there would be no indication on which semantic regions to put water and users would have to provide this information. On complex images, this may be a challenging task for novices. We let the users define the water map with a threshold on this per-region value. By varying the threshold, one can choose a look from a sharp rendition with details everywhere to an abstract print composed mostly of water diffusion gradients.

2.3.2. Dry Woodblocks

Now that we have computed the color and water maps, we define the shape of the woodblocks that we will later fabricate with a laser-cutter. We first discuss the "dry woodblocks", that is, the blocks that we will use to deposit paint on paper that is only lightly moistened (in comparison to the areas where it is fully wet). We create one such block for each color of the color map by computing the intersection between the corresponding color region and the inverse of the water map (i.e., the dry regions).

2.3.3. Wet Woodblocks

In this section, we discuss the woodblocks that we will use on wet paper. These require more sophistication because of the water diffusion. First, we create a *masking woodblock* by inverting the water map M_w and adding edges (Section 2.3.4). We will use this block to apply masking fluid on the regions that should remain dry. Then, we create woodblocks representing each color in the color map. We first describe how to design the blocks to produce a smooth color gradation and later refine this process to also handle brightness variations.

Color Woodblocks – B_{wc} In our early experiments, we created a wet block for each color by intersecting the corresponding color region in M_c with the water map M_w. This applies paints of different colors next to each other and, with water diffusion, creates a smooth transition between them. However, the extent of this transition is limited. We tested different amounts of water and paint, but the result never exhibited the large gradients typical of watercolor. We tried to shrink the shape of the woodblocks before fabricating them. We hoped that this would leave a gap between the colors that diffusion would fill, thereby creating larger gradients, but instead this created gradients with a white band in their middle. The opposite solution succeeded. We dilate the color regions of the color map with a disk of radius 1.75 mm, before computing the intersection with the water map. This creates areas where colors overlap, essentially introducing an intermediate color between any two adjacent colors. Coupled with water diffusion, this produces large smooth gradients. By varying the amount of dilation, the size of the overlap changes, which in turn affects the extent of the gradient. The resulting color regions correspond to B_{wc} as shown in Figure 2(h), the red areas indicate where the adjacent colors overlap. See Figure 10 for a comparison of gradient effects.

Shading Woodblocks. Overlapping blocks allow us to create a smooth gradation between two colors. We also observed that gradients in the input often exhibit an intensity variation, i.e., they go from dark to bright to suggest shadows and highlights. To reproduce this effect, we refine our woodblock design. First, we compute the block shape as previously described using the *water map* (2.3.1) and consider the corresponding image region in I_p. We convert the image data to grayscale and generate a *grayscale map* in the same way as we have created the color map, i.e., using the k-means algorithm [CFL*15]. Since we operate within a region of the color map, we seek a finer-grain decomposition and generate a larger number of clusters for k=50 grayscale values. We call superpixels the groups of adjacent pixels with the same grayscale value. First, we identify the brightest superpixels using non-maxima suppression, that is, we select the superpixels that are brighter than all their adjacent superpixels. We modify the block shape by removing these bright superpixels from it. As a consequence, these regions will not receive paint directly and color pigments will only reach them in a diluted form via diffusion, which will reveal more of the underlying white paper and create a brighter spot as desired. These brightest superpixel regions are denoted by the checkerboard pattern in B_d in Figure 2(j).

For the shadow regions, we identify a set of darkest superpixels using non-minima suppression, on the grayscale k-means map for k=30 grayscale values and for each color block using a procedure that we describe in the supplemental material we expand them. These shadow regions correspond to the colored areas in B_m in Figure 2(i). We will use these blocks to apply a darker shade of paint that will smoothly blend because of the water diffusion.

2.3.4. Detail Woodblocks

So far we have worked with large and medium sized superpixels, and have not captured fine details. While it is common for watercolor imagery not to exhibit such details, artists sometimes add them back using a fine brush. We provide a similar option. From dry areas, we allow users to add details by extracting the edges of the input image using the Canny edge detector and printing them on top of the color regions. For this purpose, we create a dedicated woodblock for each color region and use it to apply a darker paint of the same color. Water areas exhibit even less details because of the water diffusion. First, we add the Canny edges to the masking fluid block to create discontinuities in the diffusion gradient, thereby making these edges visible in the print. Also, the watercolor block expansion stops wherever an edge is present. Then, we extract small image features. We use k-means clustering [CFL*15] on a grayscale version of the image for k=50 grayscale values to create small image regions and we select the dark regions using non-minima suppression. We create a woodblock with these small regions in addition to the edges of the dry region and use it to apply a darker paint of the same color once the water has dried.

3. Fabrication Method

In the previous section, we have computed the shape of the various woodblocks needed to create the final print. We now explain how to fabricate the actual woodblocks and how to create the print.

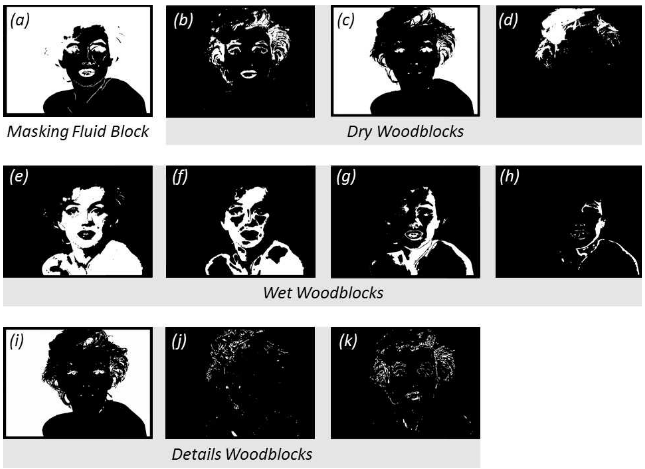


Figure 4: *Images given as input to the laser-cutter, corresponding to the woodblocks in Figure 1.*

Woodblock Cut Pattern Given the image analysis output from Section 2, we translate the output images (Fig. 2(g–j)) into woodblock cut patterns that we send to a laser-cutter (Fig. 4). The images indicate the carved regions in black. While a laser-cutter can carve arbitrary patterns [LLI09], thin wood features are fragile and likely to break under the action of the printing press that we use in the final stage. To prevent breakage, we shrink the carved regions by 0.05 mm, to account for the laser spot radius of 0.05 mm.

Creating the Print We now describe how to use the woodblocks to create the final print. First, we affix the paper onto a flat plywood board to ensure no distortion when we apply water to the paper. Then, we apply the masking fluid using the corresponding woodblock, i.e., cover the woodblock with the fluid, position the block on the paper, and press it against the paper. Once the masking fluid is dry, we spray water on the paper and we print the wet color regions. Then we spray water again to allow diffusion. Once the print is dry, we remove the masking fluid and print the dry color regions and the details. The printing sequence is illustrated in Figure 5 and we reference our equipment in the Implementation section.

Discussion: We observed that the order in which we print the wet color influences the final result. The last color we apply tends to "dominate", e.g., red over yellow gives a redder orange than yellow over red. For consistency, in our experiments, we always applied the colors from brightest to darkest.

4. Implementation Details

We describe the implementation details of our algorithm and list materials and machinery used for fabrication. We further discuss these issues in the supplemental material.

4.1. Parameters of the Image Analysis

To promote a straightforward process in the image analysis stage, the parameters that the user controls are limited to the threshold for the Water Map, the k number of colors in the Color Map, and the Semantic annotation scribble.

For the k-means palette extraction algorithm of Chang et al. [CFL*15], we made modifications so that dark colors are not discarded and shadows are not ignored. We use 10 bins for every color channel.

We use the publicly available implementation [Den] of the DenseCRF algorithm [KK11]. In all our examples, we set the weight and the standard deviation for the smoothness kernel (color independent terms) as $w^{(2)} = 3$ (unitless) and $\theta_\gamma = 3$ pixels, respectively. For the appearance kernel (color dependent terms) the weight, standard deviation and color standard deviation are: $w^{(1)} = 10$ (unitless), $\theta_\alpha = 80$ pixels, and $\theta_\beta = 19$ (RGB channels $\in [0,255]$).

The algorithm for estimating the local image smoothness [ZWJ*16] required more adaptation. We observed that, on our data, producing consistent results was challenging with the settings suggested in the original article. We conducted a parameter search using 19 test images and picked values that performed well on all of them while being robust to perturbations, i.e., results produced with slightly different parameters were also acceptable. We set the standard deviation for blur to $\sigma = 10$ pixels, the Canny [Can86] thresholds to 20 and 100 (for pixel intensities $\in [0,255]$), and $\varepsilon = 10^{-6}$ to compute the defocus value. For the gradient we use the Sobel function of OpenCV. For the guided filter [HST13] we set $\varepsilon = 10^{-9}$ and the radius $r = 20$ pixels. For the KNN matting [CLT13] we set $\lambda = 0.2$, which is a constant controlling the user's confidence on the scribble. We set the level to $l = 5$ which is a unitless constant that controls the amount of spatial coherence, we use 10 neighbors (pixels) with default spatial coherence and 1 neighbor with weak spatial coherence. We consider all the values of the user annotation with a defocus threshold of $t = 10^{-5}$. Figure 6 shows a few examples of smoothness maps computed with these parameters.

We use the OpenCV implementation of the Canny edge detector with parameters 34 and 45 (channel $\in [0,255]$). We applied a morphological dilation of radius 0.5 mm followed by an erosion of the same size (a.k.a. closure) to fill in the small holes that may exist in the edges. We discarded the edges between the wet and dry regions because they are already visible as discussed previously.

4.2. Design of the Woodblocks

The images from the Image Analysis were vectorized using path trace in Adobe Illustrator. We resized them in accordance with the dimensions of the machine press. Marilyn Monroe was printed in 24 cm × 16 cm blocks, while all the other results in 20 cm × 15 cm blocks.

4.3. Fabrication Details

To carve the woodblocks we used a Laser Pro Spirit GLS with 170 pulses/cm or dots/cm, in raster, with 90% power and 35% speed. The maximum laser processing speed and power are: 203 cm/s and 100 W respectively.

We used Arches hot pressed paper 300 g/m^2. For the paint materials we used Winsor and Newton Colourless masking fluid and

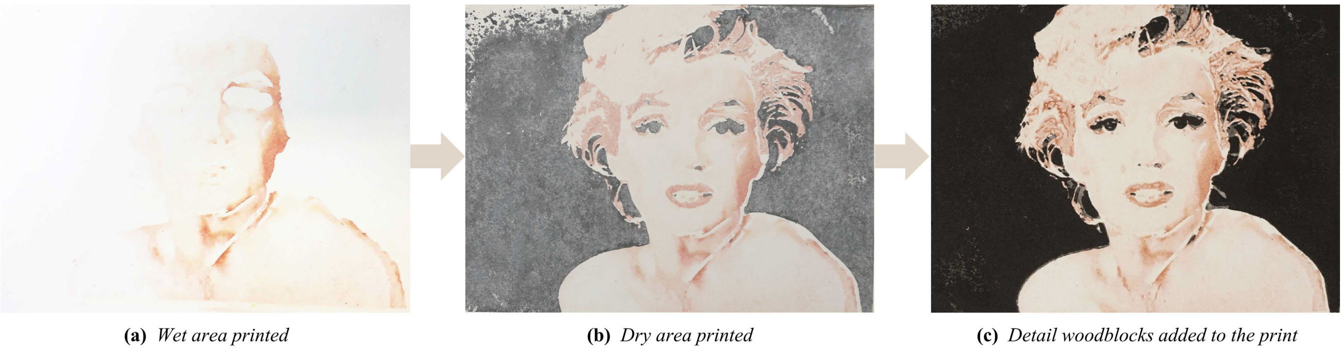

(a) *Wet area printed* **(b)** *Dry area printed* **(c)** *Detail woodblocks added to the print*

Figure 5: *An example of our printing sequence. (a) We start by masking the "dry"areas without color diffusion (white) and printing the "wet"colors with water diffusion first. (b) We then remove the masking fluid and print the dry colors without water diffusion. (c) Lastly we print the details blocks (minima and edges).*

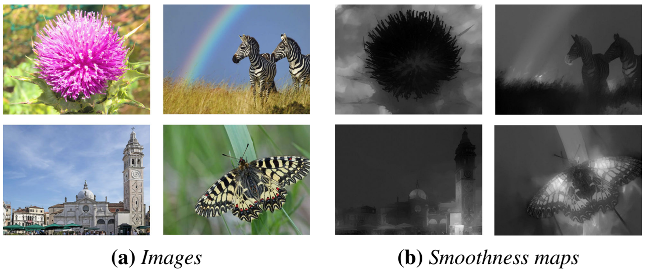

(a) *Images* **(b)** *Smoothness maps*

Figure 6: *4 out of 20 images we used to identify the values for the parameters for the smoothness map algorithm*

Winsor and Newton Cotman paint. In many cases we were able to find paint tints that closely matched the color palette, otherwise we performed color mixing using a guidebook by Wilcox [Wil94]. Light colors used a low ink-to-water ratio and darker parts used a higher ratio (e.g., 1 : 4 for light areas and 1 : 1 for dark areas).

In order to ensure a flat surface for printing, the paper was stretched on plywood. Before printing the wet color regions, the paper must also be moistened to allow absorption of the watercolor paints and permit diffusion effects. We used an aerosol sprayer, and performed a series of spraying and absorption iterations (see the supplemental material for further details). To perform the printing we used an Albion press, which is an early iron hand printing press that operates with a simple toggle action. We experimented with different amounts of pressure. Low pressure was used for the water areas for improved diffusion gradients. We found that high pressure gave sharper results in the non-water areas.

5. Results

We first illustrate the effects of key elements of our approach. In Figure 7, we show that without our complete pipeline, either spurious edges appear, shading is poorly represented, and/or diffusion patterns are inadequately shaped. Figure 8 gives an example of a print with and without using detail woodblocks. Figure 9 shows three prints made with the same process. The variability inherent in our process makes each print unique and at the same time all of them remain qualitatively similar to the input. Figure 10 illustrates the benefits of overlapping woodblocks for creating smooth color gradients.

Figure 12 shows a set of diverse prints. We chose the portrait because artifacts on faces can quickly look unpleasant; this result shows that our technique can accurately combine fine details and smooth gradients. The flower illustrates our ability to produce smooth gradients with both color and intensity gradations. We picked the bird for the 3-color gradient on its body and its thin legs. The lake cabin demonstrates that we can also produce prints with extensive details. We masked out the background region for the bird and flower examples.

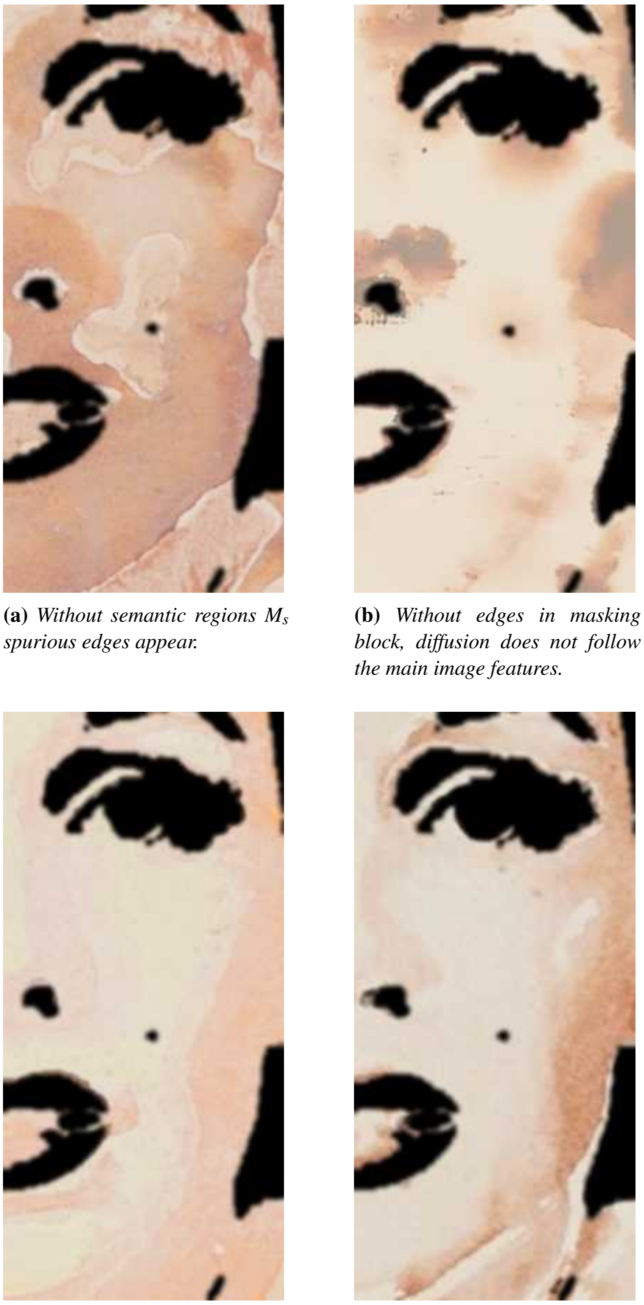
5.1. Informal User Study

We performed an informal user study with four participants from the local art community, where we interviewed college-level art majors, and students that had taken at least one class on printmaking or painting. We asked for feedback on three results including the Marilyn portrait, the flower, and the bird. We did not describe how the paintings were made.

All participants had positive reactions and specifically commented on the color diffusion effects and the shading. They remarked on the "creation of roundness", "illusion of depth", "nice washes of color", "color blending", and "interesting color overlapping effects". All participants liked the color palette selections with two specific exceptions: students would have preferred a darker brown color in the flower print, and in the Marilyn print they would have preferred the black regions in the hair to be lighter since it appears to be background rather than shading. Participants also commented on the sharp edges and use of white lines between color regions (results of the masking fluid). An art major commented on the flower: "the sharp border is very cleverly placed because it makes the stem pop out". Another art major liked the negative space created by the borders. A third student noted "the borders add to the abstraction and simplicity in a good way".

5.2. Discussion and Future Work

As revealed in the informal user study, color selection is an important point, and, in a future version, we would like to add an automatic mixing tool for the watercolor paints, similar to the one presented by Flagg et al. [FR06], to account for the transparency of the medium and layering effects such as glazing that produces new color hues when pigments overlap. Exploring this direction, we also plan to build upon the insights of Babaei et al. [BVF*17] and Tan et al. [TLG16] on color layering. Our use of a small set of paints limits our ability to reproduce color details, which might negatively impact some texture, e.g., the trees in the landscape example. This can be mitigated with more paints at the expense of a longer printing process and more detailed annotation that is more complex to produce. Also, we did not experiment with various sizes, we expect that parameters like the dilation radius need to be scaled proportionally to the print size but more thorough investigation is necessary.



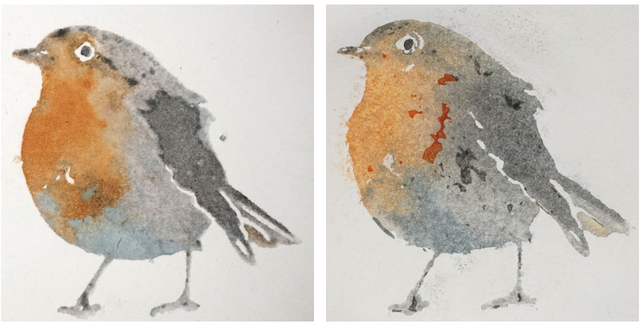
(a) *Without semantic regions M_s spurious edges appear.*

(b) *Without edges in masking block, diffusion does not follow the main image features.*

(c) *Without shading blocks, shadows are weak.*

(d) *Our result.*

Figure 7: *Removing stages in our pipeline negatively affects the results of our algorithm. In this figure, non-skin areas are digitally masked out for clarity.*



(a) *Print without details*

(b) *Print with details*

Figure 8: *We let users choose whether the print has fine details, e.g., the feathers in this case. The difference in the color gradient on the bird body and wing comes from the inherent variations that occur between prints.*

Additionally, a direction of future work could be predicting the faithfulness of the fabricated prototype as a function of the number of the colors, the amount of water, and the printing order, for example using a simulation interface. Similarly, augmenting the abstraction of the print is of interest. For example, simplification of the input image using non-photorealistic rendering or water-colorization techniques can modify the abstraction and uniqueness of the print.

Another direction for future work is refining our diffusion model

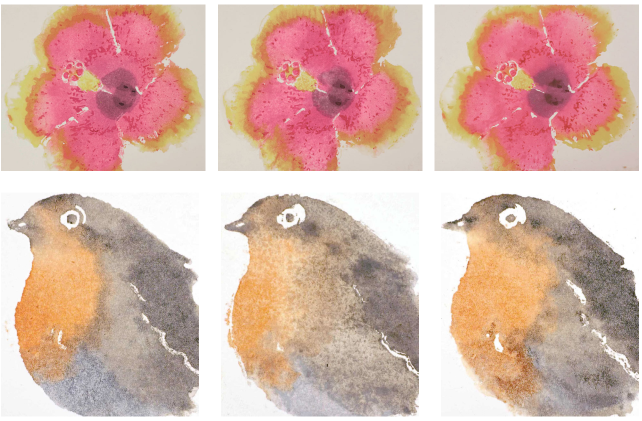
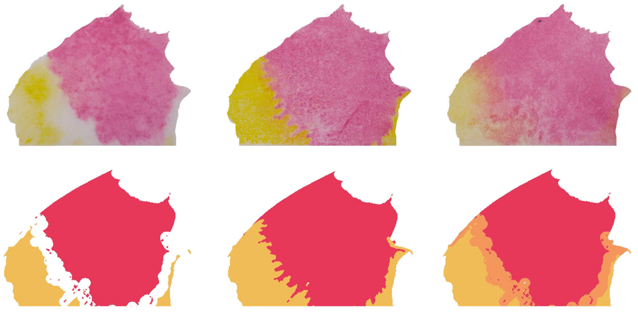


Figure 9: *Two examples of repeated prints to illustrate the variations inherent in our approach.*



(a) *Shrunken color regions* (b) *Adjacent color regions* (c) *Overlapping color regions*

Figure 10: *Print of two color blocks. Different strategies to create color gradients. The top row shows actual prints and the bottom row shows a visualization of the two blocks that we fabricated. Overlapping blocks (c) produce the smoothest gradients. This is the option that we use in our technique.*

to support more levels between our current 'dry' and 'wet' options, which would require a finer-grain analysis of the local image smoothness. Finally, we have experimented with using woodblocks for more than one color to simplify the printing process. Our early experiments done by hand are promising (Fig. 11) and suggest that it should be possible to design an algorithmic solution.

Our image processing algorithm is simple to implement and successfully extracts the corresponding maps and elements. Other techniques could be used and may be able to improve the results, we keep such experiments as future work. For instance, Lin et al. [LH13] create color palettes based on a dataset of human-extracted color themes. Some defocus estimation techniques create pixel-level defocus maps [ZWJ*16, ZS11]. Several other options exist for computing a semantic map [VS12, NHH15]. Other image morphology algorithms can be used for edge extraction and vectorization [MSF02, WD98, Mar98]. Specifically, for the edge guided polygon expansion a clipping algorithm such as [MOJR13] guiding a curve offset [YD15] which takes into account self-intersections could be used.

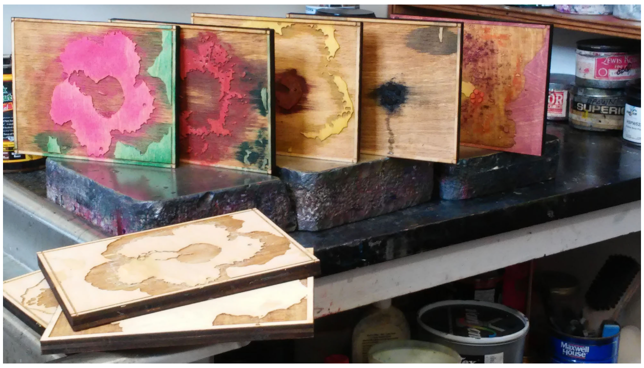


Figure 11: *Manual combination of two (here) or more colors per block, result in a smaller number of blocks and the automation is left as a direction for future work.*

6. Conclusion

We have presented a novel technique of fabricating original watercolor prints given an input photograph using image processing. Our technique generates woodblocks based on the photograph's segmentation according to color values, user-annotated semantics, smoothness values and luminosity. We provide a pipeline to fabricate the prints using watercolor paints by taking advantage of the color diffusion in the water. Our technique fabricates prints that match different aesthetic preferences. We hope that it will stir the interest of people from different backgrounds, who will consider using it as a tool to create their own designs driving our idea of water-woodblock to other directions, or experimenting and creating other innovative art media.

7. Acknowledgement

We thank the anonymous reviewers for their feedback; Ayan Azmat and Sunil Hadap for insightful discussions; the Dartmouth Thayer Machine Shop and its crew (Kevin Baron, Pete Fontaine, Jason Downs and Dan Denauw) for laser-cutter resources; Sarah M. Smith and Katie Baldwin and the Dartmouth Book Arts Workshop for providing printing resources and knowledge; and the volunteer art students (Phoebe, Chicaiza, Iliana, Karen) for participating in the informal study. Photograph sources: Hibiscus Flower: Petr Kratochvil CC0 Public Domain 2018, www.publicdomainpictures.net; Lake Cottage House: Pexels CC0 Public Domain 2018, www.pexels.com; European Robin: Themis Nasopoulou 2012, floraolympus.blogspot.com; Church of Santa Maria Formosa: Didier Descouens 2012, commons.wikimedia.org; Zebra Backgrounds Free Pixels Talk: CC0 Public Domain 2018, Wallpaperget.com; The flower and the butterfly: dataset of Shi et al. [SXJ14]. This project is partially supported by the National Science Foundation under Grant No. 1464267.

References

[BNTS07] Bousseau A., Neyret F., Thollot J., Salesin D.: Video watercolorization using bidirectional texture advection. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. URL: http://doi.acm.org/10.1145/1275808.1276507, doi:10.1145/1275808.1276507. 2

[Bot09] Botkin I.: Painting with polygons: Non-photorealistic rendering using existing tools. In *SIGGRAPH 2009: Talks* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 22:1–22:1. URL: http://doi.acm.org/10.1145/1597990.1598012, doi:10.1145/1597990.1598012. 2

[Bul] Bull D.: Encyclopedia of woodblock printmaking. http://woodblock.com/encyclopedia/topics/toc_process.html. Accessed: 2017-09-18. 2

[BVF*17] Babaei V., Vidimče K., Foshey M., Kaspar A., Didyk P., Matusik W.: Color contoning for 3D printing. *ACM Trans. Graph. 36*, 4 (July 2017), 124:1–124:15. URL: http://doi.acm.org/10.1145/3072959.3073605, doi:10.1145/3072959.3073605. 8

[Can86] Canny J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*, 6 (Nov 1986), 679–698. doi:10.1109/TPAMI.1986.4767851. 6

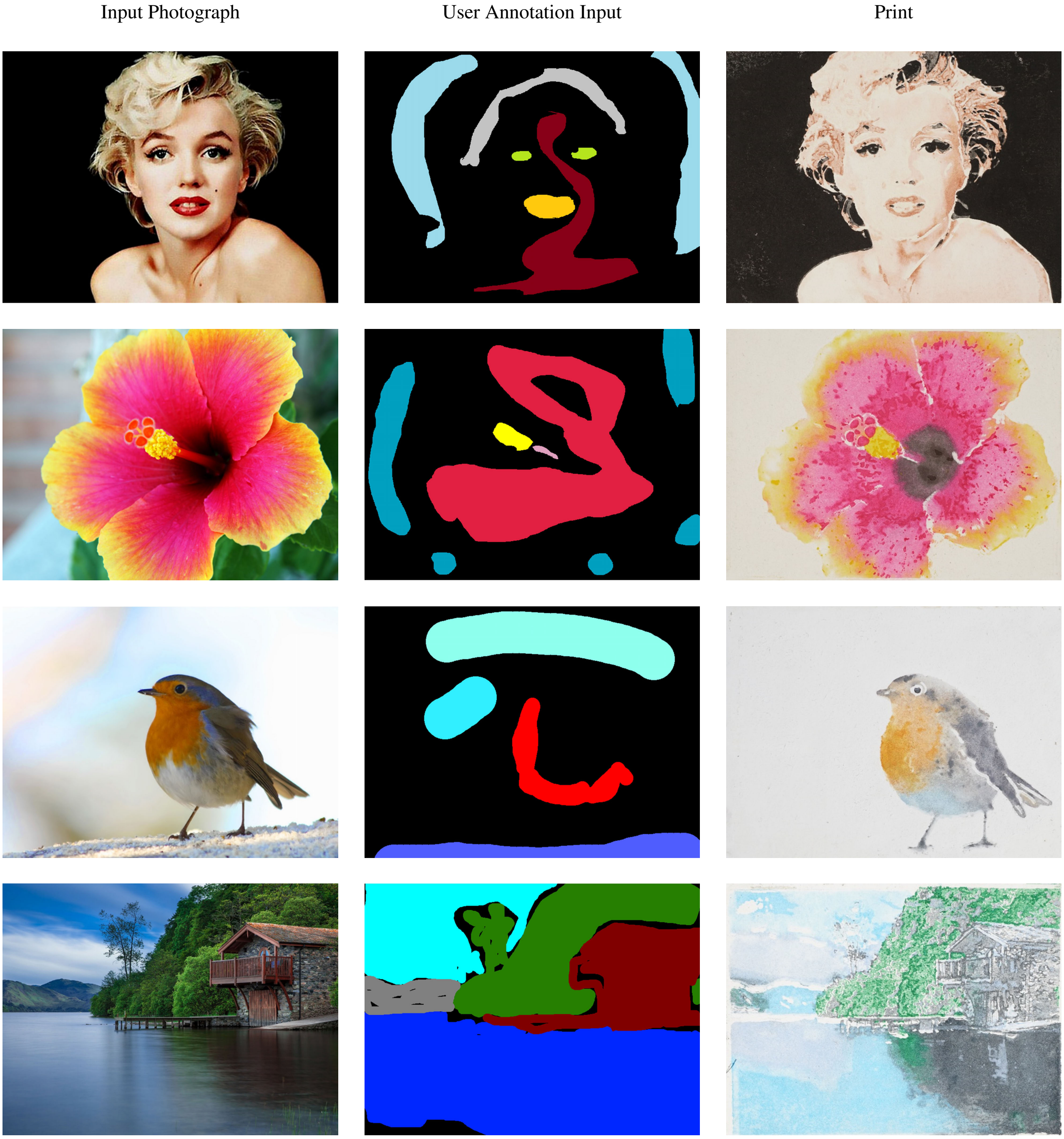


Figure 12: *Examples of watercolor prints with a variety of input photographs. (Row 1) Portrait of Marilyn Monroe*[‡]*. (Row 2) The flower is a representative example of creating a color gradient. (Row 3) The bird has diffusion of more than two colors (Row 4) The landscape contains detailed textures which are produced with the detail woodblocks printed on top of the dry blocks.*

[‡] *Photograph by Milton H. Greene ©2018 Joshua Greene www.archiveimages.com*

[CAS*97] Curtis C. J., Anderson S. E., Seims J. E., Fleischer K. W., Salesin D. H.: Computer-generated watercolor. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 421–430. URL: http://dx.doi.org/10.1145/258734.258896, doi:10.1145/258734.258896. 2

[CFL*15] Chang H., Fried O., Liu Y., DiVerdi S., Finkelstein A.: Palette-based photo recoloring. *ACM Trans. Graph. 34*, 4 (July 2015), 139:1–139:11. URL: http://doi.acm.org/10.1145/2766978, doi:10.1145/2766978. 4, 5, 6

[CKIW15] Chen Z., Kim B., Ito D., Wang H.: Wetbrush: GPU-based 3D painting simulation at the bristle level. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 200:1–200:11. URL: http://doi.acm.org/10.1145/2816795.2818066, doi:10.1145/2816795.2818066. 2

[CLT13] Chen Q., Li D., Tang C. K.: KNN matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 9 (Sept 2013), 2175–2188. doi:10.1109/TPAMI.2013.18. 6

[Den] DenseCRF implementation. https://github.com/cvlab-epfl/densecrf. Accessed: 2017-05-21. 6

[DLPT12] Deussen O., Lindemeier T., Pirk S., Tautzenberger M.: Feedback-guided stroke placement for a painting machine. In *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging* (Aire-la-Ville, Switzerland, Switzerland, 2012), CAe '12, Eurographics Association, pp. 25–33. URL: http://dl.acm.org/citation.cfm?id=2328888.2328894. 2

[FR06] Flagg M., Rehg J. M.: Projector-guided painting. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2006), UIST '06, ACM, pp. 235–244. URL: http://doi.acm.org/10.1145/1166253.1166290, doi:10.1145/1166253.1166290. 2, 8

[HE04] Hays J., Essa I.: Image and video based painterly animation. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering* (New York, NY, USA, 2004), NPAR '04, ACM, pp. 113–120. URL: http://doi.acm.org/10.1145/987657.987676, doi:10.1145/987657.987676. 2

[HST13] He K., Sun J., Tang X.: Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 6 (June 2013), 1397–1409. doi:10.1109/TPAMI.2012.213. 6

[IBT13] Iarussi E., Bousseau A., Tsandilas T.: The drawing assistant: Automated drawing guidance and feedback from photographs. In *ACM Symposium on User Interface Software and Technology (UIST)* (2013), ACM. URL: http://www-sop.inria.fr/reves/Basilic/2013/IBT13. 2

[KK11] Krähenbühl P., Koltun V.: Efficient inference in fully connected CRFs with gaussian edge potentials. In *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc., 2011, pp. 109–117. URL:"http://papers.nips.cc/paper/4296-efficient-inference-in-fully-connected-crfs-with-gaussian-edge-potentials.pdf",. 4, 6

[LBDF13] Lu J., Barnes C., DiVerdi S., Finkelstein A.: Realbrush: Painting with examples of physical media. *ACM Trans. Graph. 32*, 4 (July 2013), 117:1–117:12. URL: http://doi.acm.org/10.1145/2461912.2461998, doi:10.1145/2461912.2461998. 2

[LC14] LÃij C., Chen X.: Image watercolorization based on visual weight-map. In *2014 7th International Congress on Image and Signal Processing* (Oct 2014), pp. 233–237. doi:10.1109/CISP.2014.7003783. 2

[LH13] Lin S., Hanrahan P.: Modeling how people extract color themes from images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, ACM, pp. 3101–3110. URL: http://doi.acm.org/10.1145/2470654.2466424, doi:10.1145/2470654.2466424. 9

[LLI09] Leone C., Lopresto V., Iorio I. D.: Wood engraving by Q-switched diode-pumped frequency-doubled Nd:YAG green laser. *Optics and Lasers in Engineering 47*, 1 (2009), 161 – 168. URL: http://www.sciencedirect.com/science/article/pii/S014381660800153X, doi:https://doi.org/10.1016/j.optlaseng.2008.06.019. 6

[LPD13] Lindemeier T., Pirk S., Deussen O.: Image stylization with a painting machine using semantic hints. *Comput. Graph. 37*, 5 (Aug. 2013), 293–301. URL: http://dx.doi.org/10.1016/j.cag.2013.01.005, doi:10.1016/j.cag.2013.01.005. 2

[LSD16] Lindemeier T., Spicker M., Deussen O.: Artistic composition for painterly rendering. Hullin M., Stamminger M., Weinkauf T., (Eds.), The Eurographics Association. doi:10.2312/vmv.20161350. 4

[Mar98] Maragos P.: Morphological signal and image processing. 9

[MLB12] Mueller S., Lopes P., Baudisch P.: Interactive construction: Interactive fabrication of functional mechanical devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2012), UIST '12, ACM, pp. 599–606. URL: http://doi.acm.org/10.1145/2380116.2380191, doi:10.1145/2380116.2380191. 2

[MOJR13] MartÍNez F., Ogayar C., JimÉNez J. R., Rueda A. J.: A simple algorithm for boolean operations on polygons. *Adv. Eng. Softw. 64* (Oct. 2013), 11–19. URL: http://dx.doi.org/10.1016/j.advengsoft.2013.04.004, doi:10.1016/j.advengsoft.2013.04.004. 9

[MSF02] Mahmoudi M. T., Sharifi M., Fathy M.: A classified and comparative study of edge detection algorithms. In *Information Technology: Coding and Computing, International Conference on(ITCC)* (04 2002), vol. 00, p. 0117. URL: doi.ieeecomputersociety.org/10.1109/ITCC.2002.1000371, doi:10.1109/ITCC.2002.1000371. 9

[NHH15] Noh H., Hong S., Han B.: Learning deconvolution network for semantic segmentation. *CoRR abs/1505.04366* (2015). URL: http://arxiv.org/abs/1505.04366, arXiv:1505.04366. 9

[PDP*15] Panozzo D., Diamanti O., Paris S., Tarini M., Sorkine E., Sorkine-Hornung O.: Texture mapping real-world objects with hydrographics. *Computer Graphics Forum (proceedings of EUROGRAPHICS Symposium on Geometry Processing) 34*, 5 (2015), 65–75. 2

[PJJSH16] Prévost R., Jacobson A., Jarosz W., Sorkine-Hornung O.: Large-scale painting of photographs by interactive optimization. *Computers & Graphics 55* (Apr. 2016), 108–117. doi:10.1016/j.cag.2015.11.001. 2

[SLKD16] Semmo A., Limberger D., Kyprianidis J. E., Döllner J.: Image stylization by interactive oil paint filtering. *Comput. Graph. 55*, C (Apr. 2016), 157–171. URL: https://doi.org/10.1016/j.cag.2015.12.001, doi:10.1016/j.cag.2015.12.001. 2

[SPG*16] Schüller C., Panozzo D., Grundhöfer A., Zimmer H., Sorkine E., Sorkine-Hornung O.: Computational thermoforming. *ACM Trans. Graph. 35*, 4 (July 2016), 43:1–43:9. URL: http://doi.acm.org/10.1145/2897824.2925914, doi:10.1145/2897824.2925914. 2

[SXJ14] Shi J., Xu L., Jia J.: Discriminative blur detection features. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014* (2014), pp. 2965–2972. URL: https://doi.org/10.1109/CVPR.2014.379, doi:10.1109/CVPR.2014.379. 9

[TL12] Tresset P. A., Leymarie F. F.: Sketches by Paul the Robot. In *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging* (Aire-la-Ville, Switzerland, Switzerland, 2012), CAe '12, Eurographics Association, pp. 17–24. URL: http://dl.acm.org/citation.cfm?id=2328888.2328892. 2

[TLG16] TAN J., LIEN J.-M., GINGOLD Y.: Decomposing images into layers via RGB-space geometry. *ACM Trans. Graph. 36*, 1 (Nov. 2016), 7:1–7:14. URL: http://doi.acm.org/10.1145/2988229, doi:10.1145/2988229. 8

[UKSI14] UMETANI N., KOYAMA Y., SCHMIDT R., IGARASHI T.: Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM Trans. Graph. 33*, 4 (July 2014), 65:1–65:10. URL: http://doi.acm.org/10.1145/2601097.2601129, doi:10.1145/2601097.2601129. 2

[UMII10] UMETANI N., MITANI J., IBARAKI T., IGARASHI T.: Designing custom-made metallophone with concurrent eigenanalysis. *NIME++* (2010). 2

[VS12] VANTARAM S. R., SABER E.: Survey of contemporary trends in color image segmentation. *Journal of Electronic Imaging 21*, 4 (2012), 040901–1–040901–28. URL: http://dx.doi.org/10.1117/1.JEI.21.4.040901, doi:10.1117/1.JEI.21.4.040901. 9

[WD98] WENYIN L., DORI D.: *A survey of non-thinning based vectorization methods*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 230–241. URL: https://doi.org/10.1007/BFb0033241, doi:10.1007/BFb0033241. 9

[Wil94] WILCOX M.: *Blue and yellow don't make green*. North light books, 1994. 7

[WW16] WANG L., WHITING E.: Buoyancy optimization for computational fabrication. *Comput. Graph. Forum 35*, 2 (May 2016), 49–58. URL: https://doi.org/10.1111/cgf.12810, doi:10.1111/cgf.12810. 2

[WWF*14] WANG M., WANG B., FEI Y., QIAN K., WANG W., CHEN J., YONG J. H.: Towards photo watercolorization with artistic verisimilitude. *IEEE Transactions on Visualization and Computer Graphics 20*, 10 (Oct 2014), 1451–1460. doi:10.1109/TVCG.2014.2303984. 2

[YD15] YAMAN U., DOLEN M.: A gradient-based morphological method to produce planar curve offsets. *The International Journal of Advanced Manufacturing Technology 80*, 1 (Sep 2015), 255–274. URL: https://doi.org/10.1007/s00170-015-6971-z, doi:10.1007/s00170-015-6971-z. 9

[YS05] YAO F., SHAO G.: Painting brush control techniques in chinese painting robot. In *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.* (Aug 2005), pp. 462–467. doi:10.1109/ROMAN.2005.1513822. 2

[ZS11] ZHUO S., SIM T.: Defocus map estimation from a single image. *Pattern Recogn. 44*, 9 (Sept. 2011), 1852–1858. URL: http://dx.doi.org/10.1016/j.patcog.2011.03.009, doi:10.1016/j.patcog.2011.03.009. 9

[ZWJ*16] ZHANG X., WANG R., JIANG X., WANG W., GAO W.: Spatially variant defocus blur map estimation and deblurring from a single image. *Journal of Visual Communication and Image Representation 35* (2016), 257 – 264. URL: http://www.sciencedirect.com/science/article/pii/S1047320316000031, doi:https://doi.org/10.1016/j.jvcir.2016.01.002. 4, 6, 9

[ZYZZ15] ZHANG Y., YIN C., ZHENG C., ZHOU K.: Computational hydrographic printing. *ACM Trans. Graph. 34*, 4 (July 2015), 131:1–131:11. URL: http://doi.acm.org/10.1145/2766932, doi:10.1145/2766932. 2

[ZZ10] ZHAO M., ZHU S.-C.: Sisley the abstract painter. In *NPAR '10: Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (New York, NY, USA, 2010), ACM, pp. 99–107. doi:http://doi.acm.org/10.1145/1809939.1809951. 4