Example-Based Art Pattern Synthesis Using Level Sets

Ruobing Wu Wenping Wang Yizhou Yu
The University of Hong Kong

Abstract

Line drawings and digital arts appear everywhere, from simple icons and logos to cartoons, maps, and illustrations. We define art patterns as the subset of line drawings and digital arts that are comprised of repeated elements. Inspired by recent success of exemplar-based texture synthesis, in this paper, we focus on synthesizing art patterns with curvilinear features from exemplars, which we cast as a global optimization problem. Our energy function for this problem measures both the appearance similarity of color patterns and shape similarity of curvilinear features between an input exemplar and a synthesized image. We develop an overall EM-style algorithm for minimizing this energy function. The shape similarity part of the energy is minimized through an innovative application of the level set method. We further generalize our energy function and optimization algorithm to multilayer pattern synthesis. Our generalized optimization can effectively handle multiple layers and synthesize valid instances of interaction.

Keywords: Line Drawing, Vector Arts, Texture Synthesis

1 Introduction

Line drawings and digital arts appear everywhere, from simple icons and logos to cartoons, maps, illustrations, and storyboards. The creation of line drawings and digital arts is typically a time-consuming process that requires skilled artists. Most of us appreciate their simple and aesthetic appearances, but do not have the necessary skills to create them. There exist a subset of line drawings and digital arts that exhibit the essence of textures, i.e. a spatial arrangement of repeated elements. We use the phrase, *art patterns*, to call both such line drawings and digital arts.

Art patterns have their unique characteristics. They are typically dominated by curves and shape contours. Spaces among the curves are typically filled with constant colors or color gradients. In this paper, we would like to investigate techniques for synthesizing art patterns with curvilinear features from exemplars.

Synthesizing the aforementioned art patterns is challenging for the following reasons. First, given the relatively wide span of a curve, local synthesis is unlikely to work well, often producing fragmented appearances. And it is also unclear how to place this synthesis task in a global optimization context. Third, elements in an exemplar may have mutual interactions, giving rise to layers and occlusions. It is unclear what mechanisms need to be in place to create valid instances of interactions in a synthesized image.

In this paper, we present a novel method for example-based synthesis of art patterns. It is based on the level set method, a popular numerical technique for tracking and transforming dynamic curves and surfaces. Introduction to the level set method can be found in

Figure 1: *Closed feature curves of an input exemplar and their signed distance function.*

[Sethian 1999; Osher and Fedkiw 2003]. The level set method embeds a closed curve as the zero level set of a level set function whose domain covers the entire 2D image space, and performs curve tracking indirectly through the level set function, which makes it possible to perform local image synthesis. Furthermore, the level set method offers a global optimization capability for our task. As long as a global energy function and its derivative with respect to level sets are well defined, this energy function can be minimized through the level set method. To enable effective synthesis of complex scenarios, such as mutual occlusions among image elements, we make use of multiple level set functions, each of which is assigned a distinct layer, and further generalize our synthesis technique to accommodate multiple layers of level set functions.

1.1 Related Work

There exists much work on developing computer algorithms for generating floral ornaments [Wong et al. 1998] and planar patterns [Kaplan and Cohen 2003]. Such algorithms do not need exemplars. They can generate patterns following either mathematical (geometric) principles or design principles observed by human artists. In this paper we take an exemplar-based approach instead.

Kwatra *et al.* [2005] present an EM-style synthesis-by-optimization scheme. Their technique gives robust performance and well supports constrained synthesis. More recent work along this line includes [Kopf et al. 2007; Wei et al. 2008; Rosenberger et al. 2009], Our technique is a natural generalization of Kwatra *et al.* [2005] in terms of global optimization. By applying the level set method in the optimization stage and involving layer information in the similarity metric, our technique achieves more accurate patch matching and better curvilinear feature alignment for art patterns.

2 Basic Algorithm

The energy function in our global optimization consists of two parts. The first part evaluates the local appearance similarity between the synthesized image and the input exemplar. The second part of our energy function measures the local similarity between signed distance transforms of curvilinear features in both the synthesized image and the input exemplar. One of our main goals is to measure the local shape similarity between these curvilinear features in the input and output. Since every set of features uniquely define a signed distance function, computing local shape similarity can be cast as computing the local similarity between two signed distance functions [Barrow et al. 1977].

Let X denote the synthesized output and Z denote the input exemplar. Let $\mathbf{A_x}$ (or $\mathbf{A_z}$) denote the appearance vector at pixel $\mathbf{x}$ (or $\mathbf{z}$) in X (or Z). In this paper, we define this appearance vector as the concatenated pixel colors within a local neighborhood centered at pixel $\mathbf{x}$ (or $\mathbf{z}$). We further use $\mathbf{D_x}$ (or $\mathbf{D_z}$) to denote the vector of concatenated signed distance values within a neighborhood centered at pixel $\mathbf{x}$ (or $\mathbf{z}$). Let $\mathbf{z_x}$ denote the center of the neighborhood from the input exemplar, that is most similar to the neighborhood centered at $\mathbf{x}$ in X in terms of both color and signed distance under the Euclidean norm. The total energy is defined with respect to a collection of neighborhoods S_X from X:

$$E_t(S_X) = \alpha \sum_{\mathbf{x}\in S_X} \|\mathbf{A_x} - \mathbf{A_{z_x}}\|^2 + (1-\alpha) \sum_{\mathbf{x}\in S_X} \|\mathbf{D_x} - \mathbf{D_{z_x}}\|^2, \tag{1}$$

where $\alpha \in [0, 1)$ is a weight balancing the effects of color and signed distance similarity. Let the size of a neighborhood be w x w. In practice, the energy is evaluated over a subset of local neighborhoods that are $w/4$ pixels apart because it is redundant and computationally expensive to compute the energy over all neighborhoods in the synthesized image.

We adopt the same multiresolution synthesis scheme as in [Kwatra et al. 2005]. At every resolution, similar to [Kwatra et al. 2005], we perform an EM-like optimization by computing synthesized texture patches in the E-step while locating the set of input neighborhoods most similar to the synthesized patches in the M-step. At the very beginning prior to any M-steps and E-steps in our framework, we initialize the pixelwise appearance and signed distance values in the synthesized image using "image quilting" [Efros and Freeman 2001]. The only deviation from the original quilting algorithm is that every patch from the input exemplar is padded with an extra signed distance channel. We further extract an initial zero level set from the synthesized signed distance channel.

The M-step. In the M-step, we search for the best matching neighborhoods in the input exemplar. That is, in (1), we fix $\{\mathbf{A}_x\}$ and $\{\mathbf{D}_x\}$ while updating $\{\mathbf{z}_x\}$ for all neighborhoods in S_X. The similarity metric for matching is based on summed squared differences (SSD) and includes both terms in (1), color and signed distance. We use PatchMatch [Barnes et al. 2009] to find the best matching neighborhoods.

The E-step. At the beginning of every E-step, there already exist an initial appearance map (color image) and signed distance function for the output image, and every neighborhood in S_X has already been assigned a best matching neighborhood from the input exemplar during the latest M-step. During the E-step, we optimize both pixelwise appearance values and signed distance values. That is, in (1), we fix $\{\mathbf{z}_x\}$ while updating $\{\mathbf{A}_x\}$ and $\{\mathbf{D}_x\}$. Since our input exemplars are mostly line drawings and art patterns that have many curvilinear features, producing high-quality geometric patterns of such features is of higher priority than color patterns. Therefore, we optimize signed distance values and appearance values in two sequential stages, and signed distance values are optimized first. In the following, we elaborate our algorithm for optimizing signed distance values. We simply follow the technique in [Kwatra et al. 2005] to optimize color/appearance values.

The goal of signed distance optimization is to reduce the second energy term in (1). This problem can be conveniently solved using the level set method [Sethian 1999; Osher and Fedkiw 2003]. Since the neighborhoods in S_X are overlapping, every pixel in the synthesized image is actually covered by multiple best matching neighborhoods from the input exemplar. If we break every neighborhood into a set of pixels, the second energy term in (1) can be rewritten as

$$\sum_{\mathbf{x}\in X} \sum_{j\in s_\mathbf{x}} \left(\Phi^o(\mathbf{x}) - \Phi^i_j(\mathbf{x})\right)^2, \tag{2}$$

where Φ^o is the signed distance function for the output image that we need to optimize, $s_\mathbf{x}$ is the set of neighborhoods (from the input exemplar) covering pixel $\mathbf{x}$, and $\{\Phi^i_j | j \in s_\mathbf{x}\}$ represent the set of signed distance functions associated with the neighborhoods in $s_\mathbf{x}$. Note that Φ^o is also the level set function we use in our level set algorithm.

To derive the velocity field for the level set method, we need to redefine the energy term in (2) in a continuous image domain, where the gradient of Φ^o is well-defined, as follows,

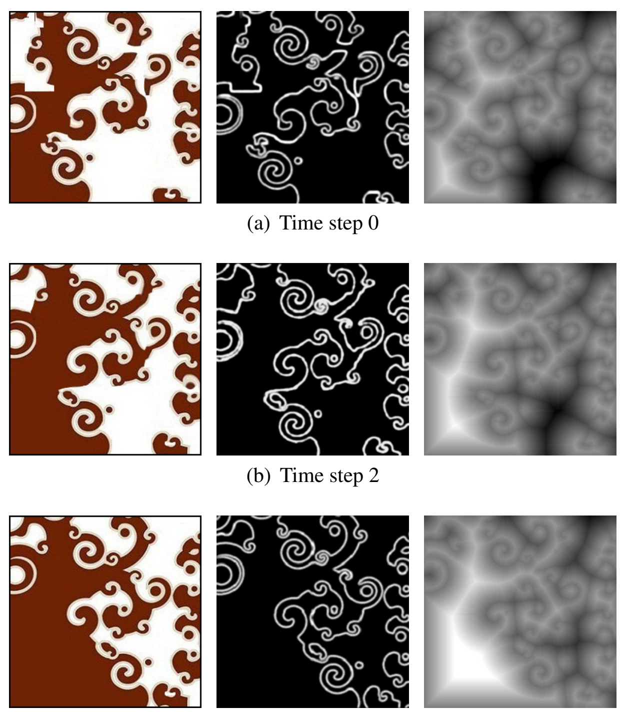
$$E_s(X) = \int_X \sum_{j\in s_\mathbf{x}} \left(\Phi^o(\mathbf{x}) - \Phi^i_j(\mathbf{x})\right)^2 d\mathbf{x}. \tag{3}$$

Taking the derivative of this energy with respect to the level set passing through $\mathbf{x}$, we obtain the following velocity component,

$$\mathbf{V}_s = 2 \sum_{j\in s_\mathbf{x}} \left(\Phi^o(\mathbf{x}) - \Phi^i_j(\mathbf{x})\right) \frac{\nabla\Phi^o(\mathbf{x})}{\|\Phi^o(\mathbf{x})\|}. \tag{4}$$

This equation indicates that the total "force" to deform and optimize the zero level set of Φ^o is a weighted average of the "forces" from individual overlapping neighborhoods, and the magnitude of the "force" from the j-th overlapping neighborhood should be set to $2\left(\Phi^o(\mathbf{x}) - \Phi^i_j(\mathbf{x})\right)$, which can be either positive or negative. When the weight is positive (negative), the zero level set of Φ^o will be pushed outward (inward) to the zero level set of Φ^i_j. Note that the gradient of the level set function can be estimated robustly using the upwind scheme [Courant et al. 1952].

Although neighborhoods in $s_\mathbf{x}$ are overlapping, curvilinear features therein do not necessarily align with each other very well. Minimizing the energy term in (2) alone may give rise to undesired discontinuities in the zero level set of Φ^o. To make the zero level



(a) Time step 0

(b) Time step 2

(c) Time step 10

Figure 2: *The evolution of the zero level set over multiple time steps.*

set of Φ^o sufficiently continuous and smooth while minimizing the energy in (2), it is a common practice to add a curvature-based velocity component, which achieves similar effects of a regularization term. Thus, the overall velocity field is defined as

$$V = \lambda V_s + \mu V_c, \tag{5}$$

where $\lambda, \mu \in [0,1]$ are weights of the two velocity components, V_s is defined in (4), and $V_c = \kappa(\mathbf{x})\frac{\nabla\Phi^o(\mathbf{x})}{\|\nabla\Phi^o(\mathbf{x})\|}$, where $\kappa(\mathbf{x})$ is defined as $(\Phi^o_{xx}\Phi^{o\,2}_y - 2\Phi^o_x\Phi^o_y\Phi^o_{xy} + \Phi^o_{yy}\Phi^{o\,2}_x)/(\Phi^{o\,2}_x + \Phi^{o\,2}_y)^{\frac{3}{2}}$, and denotes the curvature of the level set passing through $\mathbf{x}$ [Osher and Fedkiw 2003].

During every E-step, we run the level set method multiple time steps until it converges. During each time step, we follow (5) to compute an overall velocity field, which is used for driving the zero level set. In practice, we use the Narrow-Band method [Adalsteinsson and Sethian 1995] for acceleration. Fig. 2 shows the evolution of the zero level set and the level set function over multiple time steps. It can be observed that misalignments and unnatural local shapes in the initial zero level set gradually disappear as the levelset-based optimization goes on.

When the zero level set of Φ^o is being deformed and optimized, the entire synthesized image should be deformed consistently. To address this issue, at the end of each level set time step, we generate a deformation vector field for the entire image from the velocity vectors within a narrow band around the zero level set of Φ^o using the image morphing technique in Lee *et al.* [Lee et al. 1995]. This deformation field is then used for warping the colors inside the entire synthesized image. The subsequent color optimization stage is actually performed over the warped image.

3 Multi-Layer Synthesis

In the previous section, we have introduced an image synthesis algorithm that relies on a signed distance function (SDF) to optimally synthesize the shape of curvilinear features. There exist many scenarios where we cannot model the features using a single SDF. For example, if an image consists of elements on multiple layers and there exists occlusion between elements on different layers, a single SDF is insufficient to simultaneously delineate the contours of all elements. This limitation can be overcome by increasing the number of SDFs.

To accommodate multiple SDFs in our optimization framework, we allocate a distinct channel for each SDF. To construct multiple SDF channels for an input exemplar, we interactively separate curvilinear features into multiple layers (Fig. 3 bottom row), and compute a signed distance function for the features on each layer. If part of a visually important feature from the bottom layer is occluded by a top layer, we can manually recover the occluded portion to obtain more reasonable SDF distribution on that bottom layer. Each SDF channel has an associated foreground mask, which encloses all curvilinear features on that layer. Pixels inside the mask are called foreground pixels of the corresponding SDF channel. To make the SDF channels behave like layers, they are associated with a predefined order, which will be useful in various scenarios. For example, to correctly handle occlusions among image elements during image synthesis, an occluding layer should precede an occluded layer in this predefined order. Note that in our multilayer representation, only the SDF channel is separated into multiple layers while the three color channels remain intact. The color of a foreground pixel in a SDF layer is still defined by the three color channels at the same pixel location in the original input exemplar.

Suppose there are L SDF channels. Let ϕ^i_m and ϕ^o_m be the m-th SDF channel associated with the input exemplar and synthesized output, respectively. Let $\mathbf{D}^{\mathbf{m}}_{\mathbf{x}}(\mathbf{D}^{\mathbf{m}}_{\mathbf{z}})$ be the vector of concatenated m-th channel SDF values within a neighborhood centered at pixel $\mathbf{x}$(or $\mathbf{z}$). The energy function in (1) is revised as follows:

$$\alpha \sum_{\mathbf{x}\in S_X} \|\mathbf{A_x} - \mathbf{A_{z_x}}\|^2 + (1-\alpha)\sum_{m=1}^{L}\sum_{\mathbf{x}\in S_X}\|\mathbf{D}^m_{\mathbf{x}} - \mathbf{D}^m_{\mathbf{z_x}}\|^2 \tag{6}$$

Note that all SDF layers are optimized at the same time in every iteration. To accommodate multiple SDF layers, our EM-based optimization needs to be revised as follows. During the M-Step, when we search for the best matching neighborhoods in the input exemplar, the SSD-based similarity metric needs to consider all SDF channels as well as all color channels altogether.

During the E-step, when we optimize the SDF channels of an output image, we apply the level set method to every SDF channel independently, and compute a distinct velocity field prescribed by (4) for every SDF channel during each time step according to the corresponding SDF channel of the matching neighborhoods from the input exemplar. Every SDF channel generates its own warped version of the color map from the previous time step. At the end of the current time step, we “collapse” all warped versions into a single image by observing the predefined order among the SDF channels. That means, if the m_i-th SDF channel precedes the m_j-th channel, foreground pixel colors from the m_i-th warped image should overwrite colors at the same pixels from the m_j-th warped image. At the end of the E-step, once levelset based optimization concludes, we further update pixelwise colors in the “collapsed” image.

The pseudo-code in Algorithm. 1 shows the outline of our overall synthesis framework.

Algorithm 1 Multi-Resolution Multi-Layer Pattern Synthesis

```
INPUT ← input exemplar Z, multi-layer SDFs;
INPUT ← weight α, target image resolution R_T;
R_0 ← R_T/16;
for (resolution R_c = R_0 to R_T, R_c ← 2R_c) do
  if (R_c = R_0) then
    Initialize X^0 using image quilting enhanced with SDFs;
  else
    Upsample the resolution of X^0 to R_c;
  end if
  for (neighborhood size S_c = S_UB downto S_LB, S_c ← S_c/2)
  do
    λ ← 0.2, μ ← 0.8;
    for (levelset iteration n = 1 to N) do
      z_x ← the neighborhood in Z most similar to x , ∀x ∈ X^{n−1};
      X^n ← argmin_X E_t(X; {z_x}_{x∈X^{n−1}}), where E_t is
      the energy defined in Eq. (6);
      λ ← λ + 0.75/N, μ ← μ − 0.75/N;
    end for
    X^0 ← X^N;
  end for
end for
OUTPUT ← synthesized texture X
```

4 Synthesis Results

We have fully implemented our algorithms and successfully tested them on a variety of art patterns and layered textures. The running time for generating a 256x256 (512x512) synthesized image from a 128x128 input exemplar is around 3 (11.5) minutes on an Intel 2.00GHz Core i7-2630M processor. In our multiresolution scheme,

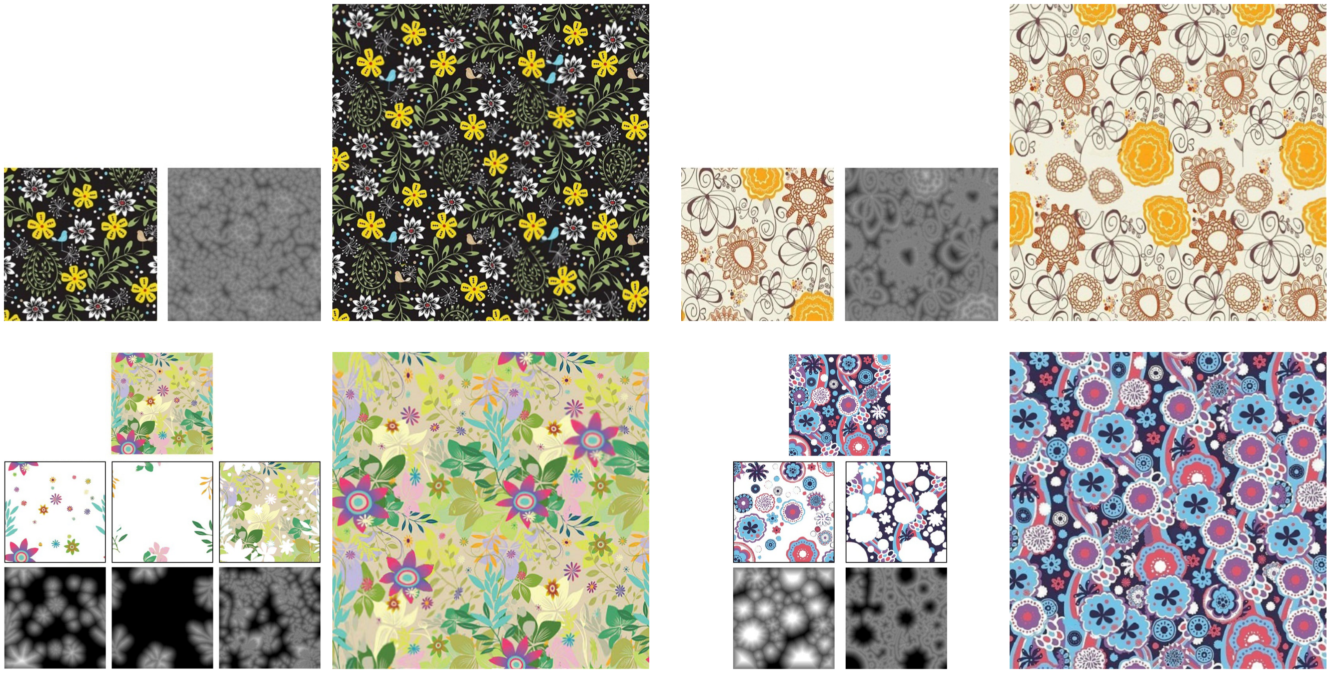


Figure 3: *Art pattern synthesis results from our method. The first row shows synthesis results for single-layer art patterns. The second row shows synthesis results for multi-layer art patterns.*

the output resolution typically increases from 32x32 to 256x256. At each resolution typically 10 EM iterations are performed. The size of image neighborhoods is decreased from 1/4 current image resolution down to 8x8 from iteration to iteration. During each E-step, 5 time steps of the level set method are performed. The α value in (1) and (6) is set to 0.4 for color/grayscale images, and 0.1 for black and white images. λ and μ in (5) are automatically adjusted according to Algorithm. 1.

A gallery of results from our method can be found in Fig. 3.

Acknowledgments

This work was partially supported by Hong Kong Research Grants Council under General Research Funds (HKU719313). The first author was supported by Hong Kong PhD Fellowship (HKPF).

References

ADALSTEINSSON, D., AND SETHIAN, J. A. 1995. A fast level set method for propagating interfaces. *Computational Physics 118(2)*, 269–277.

BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 3 (Aug.).

BARROW, H., TENENBAUM, J., BOLLES, R., AND WOLF, H. 1977. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. 5th Intl. Joint Conf. on Art. Intell.*, 659–663.

COURANT, R., ISAACSON, E., AND REES, M. 1952. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics 5*, 243–255.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, 341–346.

KAPLAN, M., AND COHEN, E. 2003. Computer generated celtic design. In *Proceedings of the 14th Eurographics Workshop on Rendering*, 9–19.

KOPF, J., FU, C.-W., COHEN-OR, D., DEUSSEN, O., LISCHINSKI, D., AND WONG, T.-T. 2007. Solid texture synthesis from 2d exemplars. *ACM Transactions on Graphics 26(3)*, 2:1–2:9.

KWATRA, V., ESSA, I., BOBICK, A., AND KWATRA, N. 2005. Texture optimization for example-based synthesis. *ACM Transactions on Graphics 24(3)*.

LEE, S.-Y., CHWA, K.-Y., AND SHIN, S. Y. 1995. Image metamorphosis using snakes and free-form deformations. In *SIGGRAPH '95*, ACM, 439–448.

OSHER, S. J., AND FEDKIW, R. 2003. *Level Set Methods and Dynamic Implicit Surfaces*. Springer.

ROSENBERGER, A., COHEN-OR, D., AND LISCHINSKI, D. 2009. Layered shape synthesis: automatic generation of control maps for non-stationary textures. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 107.

SETHIAN, J. 1999. *Level Set Methods and Fast Marching Methods*. Cambridge Univ. Press.

WEI, L.-Y., HAN, J., ZHOU, K., BAO, H., GUO, B., AND SHUM, H.-Y. 2008. Inverse texture synthesis. *ACM Transactions on Graphics 27(3)*.

WONG, M., ZONGKER, D., AND SALESIN, D. 1998. Computer-generated floral ornament. In *Proceedings of SIGGRAPH '98*, 423–434.