



**HAL**  
open science

# A generic framework for the structured abstraction of images

Noura Faraj, Gui-Song Xia, Julie Delon, Yann Gousseau

► **To cite this version:**

Noura Faraj, Gui-Song Xia, Julie Delon, Yann Gousseau. A generic framework for the structured abstraction of images. Expressive - NPAR 2017, Jul 2017, Los Angeles, United States. 10.1145/3092919.3092930 . hal-01587259

**HAL Id: hal-01587259**

**<https://hal.science/hal-01587259>**

Submitted on 13 Sep 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

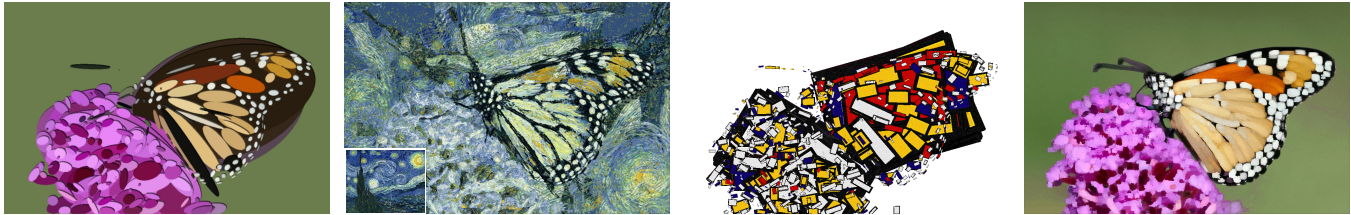
# A generic framework for the structured abstraction of images

Noura Faraj  
CNRS, MAP5, Université Paris Descartes

Gui-Song Xia  
State Key Lab. LIESMARS, Wuhan University

Julie Delon  
MAP5, Université Paris Descartes

Yann Gousseau  
LTCI, Télécom-ParisTech, Université Paris-Saclay



**Figure 1:** The geometrical structure of an input photograph is analyzed. Then, each shape of the resulting hierarchical structure is replaced either (from left to right) by an ellipse, a shape extracted from a full dictionary (Van Gogh painting), or a segmented one (Mondrian). Finally, a brush like effect can be obtained by "shaking" all the shapes from the input.

## ABSTRACT

Structural properties are important clues for non-photorealistic representations of digital images. Therefore, image analysis tools have been intensively used either to produce stroke-based renderings or to yield abstractions of images. In this work, we propose to use a hierarchical and geometrical image representation, called a topographic map, made of shapes organized in a tree structure. There are two main advantages of this analysis tool. Firstly, it is able to deal with all scales, so that every shape of the input image is represented. Secondly, it accounts for the inclusion properties within the image. By iteratively performing simple local operations on the shapes (removal, rotation, scaling, replacement...), we are able to generate abstract renderings of digital photographs ranging from geometrical abstraction and painting-like effects to style transfer, using the same framework. In particular, results show that it is possible to create abstract images evoking Malevitch's Suprematist school, while remaining grounded in the structure of digital images, by replacing all the shapes in the tree by simple geometric shapes.

## CCS CONCEPTS

•Computing methodologies → Non-photorealistic rendering; Image processing;

## KEYWORDS

Image Processing, Image Representation, Hierarchical, Morphological, Image abstraction, Picture/Image Generation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

NPAR'17, Los Angeles, CA, USA

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-5081-5/17/07...\$15.00  
DOI: 10.1145/3092919.3092930

## ACM Reference format:

Noura Faraj, Gui-Song Xia, Julie Delon, and Yann Gousseau. 2017. A generic framework for the structured abstraction of images. In *Proceedings of NPAR'17, Los Angeles, CA, USA, July 28-29, 2017*, 11 pages. DOI: 10.1145/3092919.3092930

## 1 INTRODUCTION

In the field of non-photorealistic image rendering, the term *image abstraction* refers to the process of generating simple, stylized version of photographs [DeCarlo and Santella, 2002, Orzan et al., 2007]. This task usually requires the omission of small details that are unimportant for our understanding of a scene. While doing so, one might want to keep in mind that the high-level structural properties of images, such as interactions between objects through inclusion or occlusion, play an important role for the artistic representations of scenes. For instance, Cubism or Futurism heavily rely on the structural properties of the image objects.

In this paper, we propose to analyze the geometric structure through a *topographic map* [Caselles et al., 1999] of an image. All connected components of level sets, the *shapes* of the image, are organized in a hierarchical manner to form a tree accounting for inclusion properties. The resulting representation of the image is both complete (that is, contains all the image information) and morphological (that is, invariant to contrast changes). The input image can then be edited by modifying, filtering, or replacing these shapes in a global-and-local manner. Eventually, a new image is created from the resulting tree. The geometric shapes alterations proposed in this paper include simple random displacements of shapes, creating painting-like effects, as well as replacement of every shape of the image by shapes from a dictionary, according to some replacement rules, yielding drastic alterations of images. These alterations are only some of the many possibility offered by the proposed generic framework.

Since the image is represented using a collection of shapes, we have a direct control on geometric manipulations. Furthermore, the

topographic map accounts for *all* scales in the image, enabling interplays between textures and macroscopic structures of the image. Lastly, the hierarchical structure of the chosen representation allows inclusions properties between shapes to be partially preserved in the synthesized image, yielding highly structured abstract images. We believe that the generic image representation structure, advocated for in this paper, namely the topographic map, is a powerful tool. Indeed, by iteratively performing simple local operations on the shapes (such as rotation, scaling, random displacement...), we are able to generate abstract renderings of digital photographs ranging from geometric abstraction, painterly effect and style transfer using the same framework (see Figure 1). A main singularity of our method is to offer geometric abstractions of digital photographs. In particular, we suggest a way to abstract images by replacing *all* geometric shapes by one or a few reference shapes. As a result, we obtain images that evoke the abstraction painting schools of the beginning of the 20th century, while giving a faithful account of an example image structure.

## 2 RELATED WORK

The origins of non-photorealistic rendering (NPR) take place in interactive painting and drawing systems [Haerberli, 1990, Winkenbach and Salesin, 1994], which describe techniques for creating numerical paintings by sequentially adding brush strokes on an image. Many work have since been dedicated to the automatization and refinement of stroke-based rendering, e.g. [Brooks, 2007, Collomosse and Hall, 2002, Gooch et al., 2002, Hertzmann, 1998, 2001, Litwinowicz, 1997]. Most of these works rely on a preliminary image analysis step which encodes the image structure either completely or partially: edge detection [Litwinowicz, 1997], linear scale-space [Hertzmann, 1998], region extraction [Gooch et al., 2002, Mould, 2013], saliency map [Collomosse and Hall, 2002], multi-scale edge detection [Hays and Essa, 2004], Laplacian pyramids [Brooks, 2007], etc... These systems follow the abstraction paradigms proposed by Gomes *et al.* [Gomes and Velho, 1995], offering general principles on how to model, represent and build abstraction systems in computer graphics.

Several papers focus on image *simplification* or *abstraction*. These works tend to simplify a photograph by highlighting important edges and smoothing regions which are considered as perceptually negligible. A natural way to achieve this kind of effect is also to use image analysis tools. In order to render the geometric structure of images at multiple scales, several methods rely on a linear scale space [DeCarlo and Santella, 2002, Orzan et al., 2007, Santella and DeCarlo, 2004]. De Carlo and Santella [DeCarlo and Santella, 2002, Santella and DeCarlo, 2004] use a hierarchical segmentation and a pruning of the corresponding structure to create an adaptive segmentation of the initial image. The choice of the level of details in each part of the image relies on the local contrast and on a user-driven measure of visual attention. Alternatively, Orzan *et al.* [Orzan et al., 2007] follow Canny edges through a linear scale space in order to define a multi-scale edge-based structure and to create a simplified image by solving a Poisson equation. Targeting a different application, other methods rely on mean-shift segmentation to put together texture patches and emulate stained

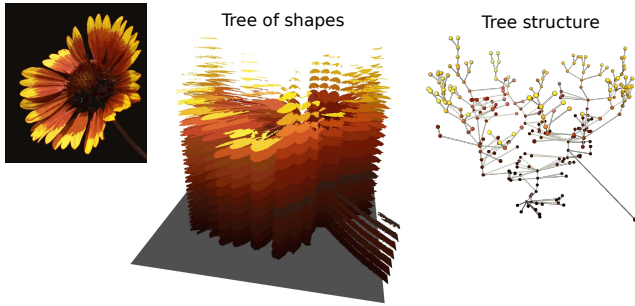
glass rendering [Mould, 2003, Setlur and Wilkinson, 2006]. An alternative direction, suggested by Bangham *et al.* [Bangham et al., 2003], is to use non-linear scale spaces to produce artistic effects, a path further explored by Kang *et al.* [Kang and Lee, 2008]. Mi *et al.* [Mi et al., 2009], propose a structural image abstraction method, through the segmentation and organization of image shapes into perceptual parts. These shapes are simplified, i.e. removed, using this hierarchical decomposition. Zhao *et al.* [Zhao and Zhu, 2010] use a meaningful image decomposition to generate painterly effects on photographs using brush superimposition. A segmentation, guiding the abstraction process, is performed and annotated by the user and the resulting shapes are organized by the user in a binary tree. While generating esthetically pleasing results, this methods necessitates heavy user interaction. Most recently, methods were proposed to mimic the styles of specific artists, such as Picasso [Lian et al., 2016] and Pollock [Zheng et al., 2015], using artistic layers approaches, and Miro [Xiong and Zhang, 2016], using random generation.

There are relatively few works concerned with the abstraction of images obtained through radical geometric modifications of shapes. The simplest approach to such a geometric abstraction is available in popular plug-ins aiming at the creation of cubist-like images. These usually superimpose to the original image simple geometric shapes over which the colors are averaged. A more involved approach was long ago suggested in the inspiring work of Haerberli [Haerberli, 1990], where geometric shapes are superimposed to the image in order to minimize a given energy. Such approaches involve complex optimization procedures, usually relying on heuristics. A more structured approach developed to achieve the same goal is presented by Song *et al.* [Song et al., 2008], where a segmentation of the image (in this case a normalized cut) is used to replace shapes by geometric primitives such as disks or rectangles, resulting in complex abstract images.

An alternate trend, for image stylization, is to use style transfer by applying an artistic style from an existing painting or photograph. Patched-based approaches transfer to an image entire patches or pixels extracted from an exemplar image [Frigo et al., 2016, Hertzmann et al., 2001, Lee et al., 2010]. These methods take inspiration from copy-paste methods developed for texture synthesis [Efros and Freeman, 2001, Efros and Leung, 1999, Hertzmann et al., 2001, Wang et al., 2004].

More recently, neural networks systems [Gatys et al., 2016] are used to learn the features of the exemplar images' style to generate a stylized image from a given input photograph. These methods use the deep Convolutional Neural Networks representation combined with an iterative optimization [Johnson et al., 2016]. We refer to [Kyprianidis et al., 2013, Rosin and Collomosse, 2012] for a more complete survey of NPR methods.

In this work, we propose to use a hierarchical structure accounting for the inclusion of shapes, allowing to produce a variety of NPR effects. Indeed, our versatile framework allows to obtain geometric abstractions of images, add a painting-like effects to the image and transfer style from existing images. This flexibility stems from the image decomposition into a topographic map, described in the next section.



**Figure 2: From left to right: a flower image, its topographic map and the structure of the corresponding tree.**

### 3 TOPOGRAPHIC MAP - TECHNICAL BACKGROUND

The topographic map, introduced by Caselles *et al.* [Caselles et al., 1999], is a hierarchical structure composed of shapes obtained from connected components of level sets. It is a powerful way to represent and manipulate the geometric content of a gray level image [Cao et al., 2005, Caselles et al., 2002, Desolneux et al., 2001]. In particular, we shall see that this map is a flexible and efficient tool to transform or simplify the geometric structure of an image, while preserving the inclusion properties of shapes.

Recall that the upper level sets of an image  $u : \Omega \mapsto \mathbb{R}$  are defined as the sets

$$\chi_\lambda(u) = \{x \in \Omega; u(x) \geq \lambda\}, \quad (1)$$

where  $\lambda \in \mathbb{R}$ . The connected components of these upper level sets correspond to light objects, i.e., lighter than  $\lambda$ . Lower level sets  $\chi_\lambda(u)$  of  $u$ , corresponding to dark objects, are defined similarly by inverting the inequality in Equation 1. Observe that the image is completely described by its upper (or lower) level sets, which form a decreasing (increasing) family with respect to inclusion. Hence, the connected components of upper (or lower) level sets are naturally embedded in a tree structure. While being dual in their definition, these two trees are very different. To obtain a unique hierarchical structure containing both light and dark objects, Monasse and Guichard [Monasse and Guichard, 2000] introduce *shapes*, which refer to connected components of upper or lower level sets whose holes<sup>1</sup> have been filled. Shapes are thus simply connected sets (i.e. without holes). Monasse and Guichard [Monasse and Guichard, 2000] show that the set of all shapes of an image has a tree structure, in which parent-child relationships correspond to set inclusion. In this paper, we refer to this tree structure as the topographic map and we use the Fast Level Set Transform (FLST) [Monasse and Guichard, 2000] to compute it efficiently. Alternatively, one might choose contour tree computation methods, widely used in the scientific visualization community [Carr et al., 2003, Gueunet et al., 2016]. A schematic representation of the topographic map is displayed in Figure 2.

Three aspects of this map make it particularly suitable for manipulating the geometric content of an image. First, it constitutes a **complete** representation of the image content, without any loss

<sup>1</sup>A hole of a set in an image is defined as a connected component of the complementary of this set that does not intersect the border of the image.

of information, contrarily to edge-based descriptions. This map is therefore well adapted for fine image modifications, even at very small scales. Second, this representation is **multi-scale** and provides ways to edit simultaneously all scales of an image in different ways. Finally, since the topographic map is a hierarchical structure of shapes, which are simply connected, it is especially well adapted to render the **inclusion properties** of objects in images. For instance, if an image represents a small white disk over a large black disk, the black disk will appear only once in the topographic map, as the parent of the white one. In comparison, a hierarchical segmentation of the image would code the large disk twice, both at a large scale (as a disk), and at a small scale (as a ring around the small disk), making the interpretation of the image as a stack of different object layers more difficult. This particularity of the topographic map will appear essential in Section 4.

Several approaches were proposed to compute the topographic map of color images. Caselles *et al.* show that most of the geometric content of color images is contained in the topographic map of its gray level channel, and therefore advise to use the shapes of this map, associated with their average saturation and hue, to describe color images [Caselles et al., 2002]. Coll *et al.* propose to define a total order in the HSV space with a good correspondence to the human perception of geometry. The topographic map of a color image is then computed on this channel [Coll and Froment, 2000]. Finally, Gouiffès *et al.* [Gouiffès and Zavidovique, 2008] propose a color topographic map representation relying on a dichromatic reflectance model. Since these last two methods are time consuming and show minor improvement in most images, we chose to use the simplest approach [Caselles et al., 2002]. For more precision and results on the topographic map, we refer to the monograph [Caselles and Monasse, 2008].

### 4 A FLEXIBLE IMAGE MANIPULATION FRAMEWORK

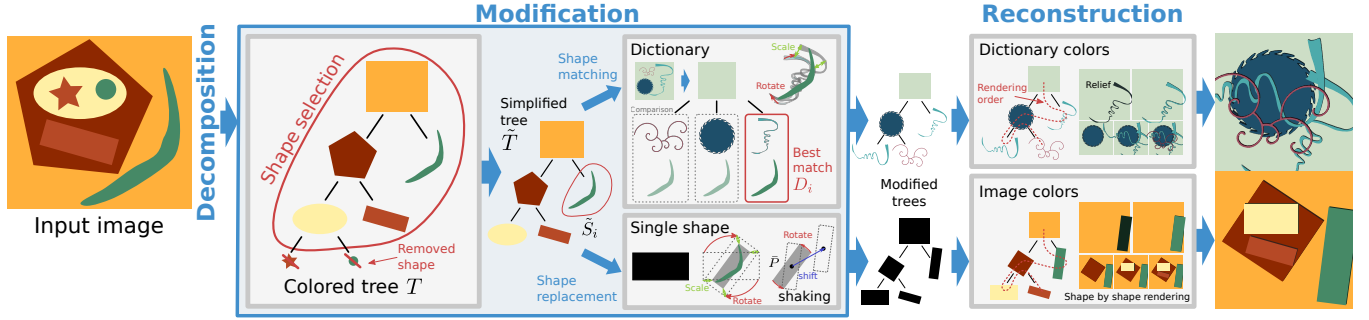
We present our general framework in section 4.1 and its details in sections 4.2 and 4.3. This generic procedure is illustrated in Figure 3.

#### 4.1 General outline

*Decomposition.* The image is transformed into a topographic map, that we call *Tree of Shapes (ToS)*, using the luminance channel.

*Modification.* As a second step, the input tree can be modified using different processes such as:

- **shape selection:** shapes are kept in the tree using the three following modalities of selection:
  - according to geometric properties, such as area or elongation,
  - according to the contrast of shapes,
  - interactively, where only shapes meeting a user defined region are kept.
- **shape displacement:** a random shift and/or rotation can be added to each shape.
- **geometric transformation:** each shape is replaced by a new shape (simple shapes - disks or ellipses -, hand drawn shapes or shapes extracted from another image, etc.). The



**Figure 3: The proposed framework for image abstraction.** **Decomposition:** we first decompose the input image into a colored tree (left). **Modification:** the tree is then modified: the small shapes, which have an area lower than a given threshold, are removed (shape selection). The remaining shapes are then replaced either by shapes extracted from a dictionary (top) or by a single similar shape (bottom). Shapes can be further modified by adding a random displacement (shaking): the shapes are rendered one by one with additional stylization effect - relief in the example - into the result image with, either the color of the input shape (top), or the color of the new shape (bottom).

new shape can be chosen according to the geometric or color properties of the input shape.

- **color modification:** the color of each shape is modified by using some color palette, possibly inherited from another tree.

*Reconstruction.* To reconstruct an image from the modified tree of shapes, it is traversed in a given order and each shape is rendered into the final image with possible additional effects such as transparency, shadow or blur.

Note that the user can perform these processes iteratively since the output of each operation is a modified ToS that can be used as an input. Observe also that these rules are only a subset of the possibility offered by the approach, which provides a versatile way to stylize images. Indeed, the flexibility of our approach potentially allows for a much wider range of tree or shape processes.

## 4.2 Analysis and abstraction

The images are abstracted or stylized using the generic framework presented in the previous section and detailed here.

We define here the shape descriptors and the distance functions between shapes that are used to process the tree.

*Colored tree.* The decomposition of the input image is implemented by the *Fast Level Set Transformation* [Monasse and Guichard, 2000]. First, for a color image  $I$ , its luminance channel is computed in HSV color space. Then, the topographic map  $T = (S_i)_{i=1, \dots, N}$  of the luminance image is computed. Eventually, for each shape  $S_i$ , a color  $c_i \in \mathbb{R}^3$  is obtained by averaging in RGB space the color values of all pixels included in  $S_i$ . We call the resulting structure  $(S_i, c_i)_{i=1, \dots, N}$  a *colored tree* (Figure 3 left).

*Shape description.* In practice, a shape  $S_i$  is made of a set of pixels noted  $(P_j)_{j=1, \dots, M}$  with  $M$  the shape area and  $(x_j, y_j)_{j=1, \dots, M}$  their associated positions. We compute its center of mass  $\bar{P}$  whose coordinates are noted  $(\bar{x}, \bar{y})$ .

In order to analyze, compare and discriminate between shapes, we chose to use a descriptor, retaining their main features: the **elongation** and **compactness** measure, proposed by Xia *et al.* [Xia et al.,

2010], and inspired by the work of Flusser *et al.* [Flusser and Suk, 1993]. This measure - invariant to scale, translation and rotation - stems from the shape's invariant moments. Note that, the compactness is similar to the ellipticity measure proposed by Rosin *et al.* [Rosin, 2000]. Such moments are used for image registration [Monasse, 1999] and texture recognition [Hamdan and Larson, 2002, Xia et al., 2010]. We extract the invariant moments from the normalized inertia matrix of order 2, defined as :

$$A = \begin{pmatrix} \eta_{20} & \eta_{11} \\ \eta_{11} & \eta_{02} \end{pmatrix},$$

with,  $p$  and  $q$  being two integers,

$$\eta_{pq} = \frac{\mu_{pq}}{M^{(p+q+2)/2}},$$

and  $\mu_{pq}$  the central moment given by

$$\mu_{pq} = \sum_{j=1}^M (x_j - \bar{x})^p (y_j - \bar{y})^q.$$

We compute  $\lambda_1$  and  $\lambda_2$  the two eigenvalues of  $A$  with  $\lambda_1 > \lambda_2$ . Note that, this values correspond to fitting an ellipse to the shape with  $2\sqrt{\lambda_1}$  and  $2\sqrt{\lambda_2}$  as semi-major/minor axis. Finally, we can compute the elongation  $e$  and compactness  $\kappa$  used to describe the shape:

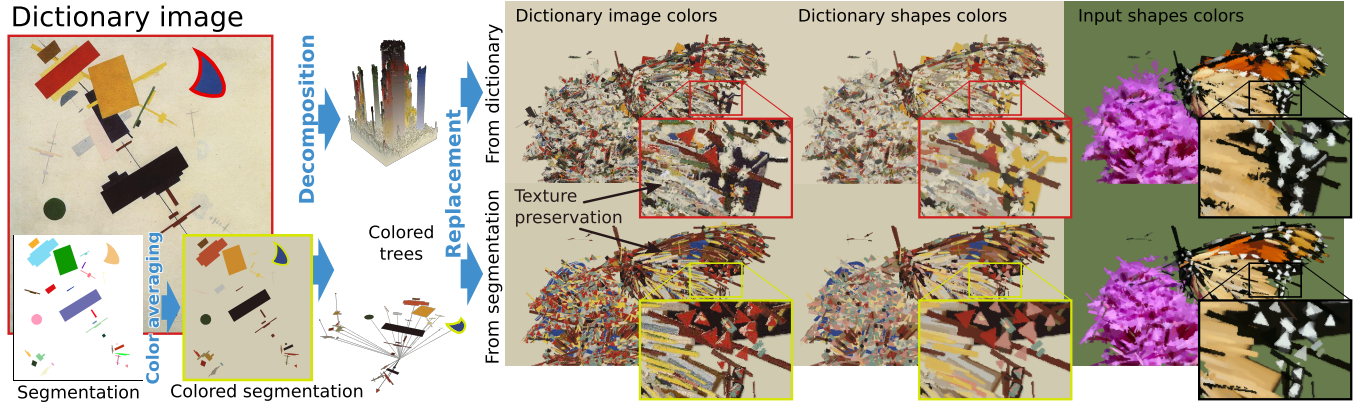
$$e = \frac{\lambda_2}{\lambda_1} \quad (2)$$

and

$$\kappa = \frac{1}{4\pi\sqrt{\lambda_1\lambda_2}}. \quad (3)$$

Note that, the compactness measures the shape similarity to an ellipse and that the values of  $e_i$  and  $\kappa_i$  are between 0 and 1 (see [Xia et al., 2010] for details).

*Shape comparison.* In order to enable the replacement of shapes (either by shapes extracted from a dictionary or by a single shape), the shapes need to be compared and the best match found. To do so, we use a distance  $d_1$  between two shapes  $S_i$  and  $S_k$  that



**Figure 4: Color modification.** The colors can be either transferred from the dictionary image to preserve its texture (left), or from the dictionary ToS (middle) or kept unchanged (right). This process can be performed on a segmented version of the dictionary (bottom).

depends only on the shapes' geometry using their area, elongation and compactness:

$$d_1(S_i, S_k) = (e_i - e_k)^2 + (\kappa_i - \kappa_k)^2 + (1 - \min\{\frac{M_i}{M_k}, \frac{M_k}{M_i}\})^2. \quad (4)$$

We use a normalized area comparison so that the area term does not shadow the others. This distance is preferred when using monochromatic or single image dictionaries. To transfer shapes from a colored dictionary (for instance for style transfer), we minimize the following distance accounting for the shapes' colors:

$$d_2(S_i, S_k) = d_1 + d_{color}(c_i, c_k). \quad (5)$$

with  $d_{color}(c_i, c_k)$  the distance between the shapes' colors defined as  $\frac{1}{3} \sum_{channel=RGB} (1 - \min\{\frac{channel_i}{channel_k}, \frac{channel_k}{channel_i}\})^2$ . The distances between the color channels are normalized so that a channel does not prevail over the other.

### 4.3 Tree processing

We propose a set of possible modifications that can be used jointly or separately.

**4.3.1 Modification.** The proposed shape modifications are illustrated in Figure 3 (Shape selection, displacement and transfer).

**Selection.** In order to simplify the image content only a subset ( $\tilde{S}_i$ ) of the shapes ( $S_i$ ) are kept. The shape description parameters - elongation, compactness and area - are used to perform the discrimination. Additionally, shapes can be filtered with reference to their contrast, using the automatic topographic map as a *sketch* of the image. The resulting tree  $\tilde{T} = (\tilde{S}_i)$  is obtained by defining the parent of  $\tilde{S}_i$  as the first ancestor shape of  $S_i$  in  $T$  that has not been removed (Figure 3 shows such a simplified tree). The root of the tree is kept unchanged.

**Displacement.** The structure of the tree  $T$  is kept unchanged, and for each  $i = 1 \dots N$ , a random translation and/or rotation is added to ( $S_i$ ). We call this effect *shaking* (Figure 3 bottom) The displacements can depend on the properties of  $S_i$ .

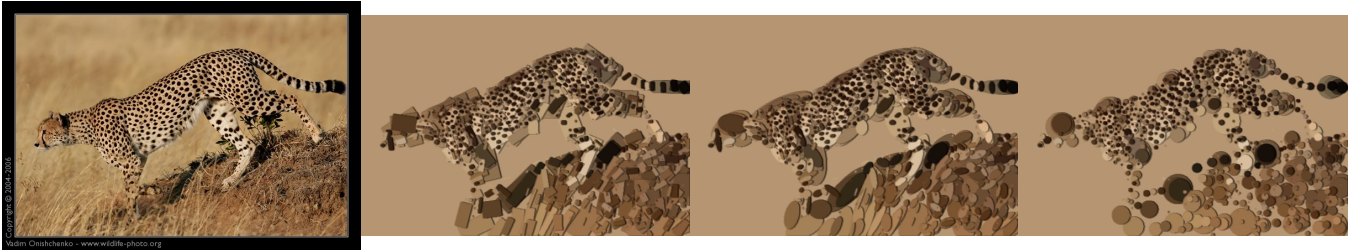
**Transformation and matching.** Each shape from the tree can be replaced by a given shape or a shape extracted from a prescribed dictionary (see modification box Figure 3, resp. bottom and top). The dictionary can be obtained from an exemplar image decomposed into a colored tree. More precisely, for each shape  $S_i$ , a shape  $D_i$  is chosen in the dictionary tree so as to match  $S_i$ . The shape is chosen as the one minimizing the distance  $d_1$  or  $d_2$ .

The dictionary might be made of a large number of shapes, therefore, performing this process is time consuming since one needs to compute the distance between each shape  $S_i$  and all the ones from the dictionary color tree (dotted box Figure 3). To reduce the computation time we use a kd-tree [Bentley, 1975] to perform an efficient search to match points in a k-dimensional space. Since our distances are not Euclidean, due to the normalizations, we build the tree using six-dimensional points describing the shapes of the dictionary color tree using  $(e_i, \kappa_i, \frac{M_i}{mean(M)}, \frac{r_i}{mean(r)}, \frac{g_i}{mean(g)}, \frac{b_i}{mean(b)})$  with  $mean(M)$  referring to the average value of the shapes areas and  $mean(\{r, g, b\})$  to the average values of each color channel. Then, we get the  $l$  nearest neighbors from the current shape, on which we perform the exact distance calculation and minimization in order to get the best matching shape. In our experiments, a neighborhood of size  $l = 50$  is sufficient.

Once a shape  $D_i$  is selected, an isometry  $\mathcal{T}$  is computed, in a way that the area and orientation of  $\mathcal{T}(D_i)$  matches the ones of  $S_i$ . The corresponding rotation and translation are illustrated in Figure 3.

Note that, when dealing with a single shape, only the isometry needs to be computed. Additionally, when the dictionary is made of few shapes with limited color contrasts, one might want to use  $d_1$  only, accounting for the shape geometry.

**Color modification.** The color  $c_i$  associated with  $S_i$  is usually kept unchanged except, when we perform shape transfer between two images, in which case the color may be transferred from the dictionary image to reproduce its texture, or the ones of the dictionary colored tree's shapes as illustrated in Figure 4. When dealing with extreme abstraction, one might want to limit the number of shapes to extract from the input dictionary. To do so, we use a segmentation



**Figure 5: Shape selection and single shape rendering.** The background of the original image, from [www.wildlife-photo.org](http://www.wildlife-photo.org) courtesy of Vadim Onishenko (left), is eliminated using a segmentation procedure [Felzenszwalb and Huttenlocher, 2004], the resulting image’s colored tree is filtered using a contrast based automated procedure [Desolneux et al., 2001]. All the shapes from the resulting *sketch* are replaced with (from left to right) rectangles, ellipses, or disks.

of the dictionary image. In this case, the colored tree is generated from a color image computed by averaging the colors of the dictionary image in each shape from the segmentation image (as illustrated on the left of Figure 4).

**4.3.2 Reconstruction.** Eventually, we reconstruct an image from the modified ToS. The resulting image is obtained by rendering iteratively the shapes of the final tree. Additional effects such as relief, transparency and blur can be easily added at this stage.

**Ordering.** The shapes are rendered one by one, as illustrated on the right of Figure 3, where the tree traversal order is displayed as a dotted line. Observe that, in a tree resulting from either shape displacements or geometric transformations, the parent-children relationships no longer correspond to the inclusion of shapes, as in the input tree  $T$ . Therefore, one has to make some choice on how to reconstruct images from the tree of shapes. In order to preserve the inclusion of shapes as much as possible, we choose to compute the resulting image by assigning to each pixel the color of the smallest shape containing it. Otherwise, a breadth-first rendering of the shapes is performed since the ToS accounts for shape inclusion.

**Relief Effects.** Relief effects on 2D images are usually achieved by using some 3D models, as in bump mapping [Blinn, 1978]. In the framework of this paper, a different relief effect can be achieved by assigning a shadow to each shape. For each shape  $S_i$  with color  $c_i$ , a slightly translated version of the same shape is added in the tree as the parent of  $S_i$ , with a darker color  $k.c_i$  (first rendered shape in Figure 3 right) with  $k$  a scalar between 0 and 1. This effect is used to render the simple geometric shapes of Figures 1, 5 and 7.

**Blur.** Observe also that the shape manipulation of the previous stage could yield aliasing effects when the new shapes are used for the reconstruction. To overcome this issue, we choose a simple matting technique consisting in adding a transparency channel with Gaussian decreasing around the shape. The parameter of the Gaussian is chosen as 0.4 in all the experiments.

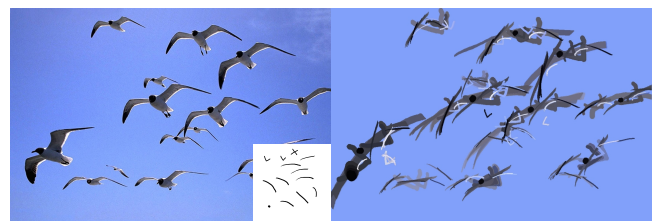
## 5 RESULTS

We apply our generic decomposition-modification-reconstruction framework on a large variety of examples among which photographs from the benchmark image set for evaluating stylization [Mould and Rosin, 2016]. We demonstrate the validity of our approach for image abstraction in Figures 5, 6, 7 and 8 and stylization in Figure 9.

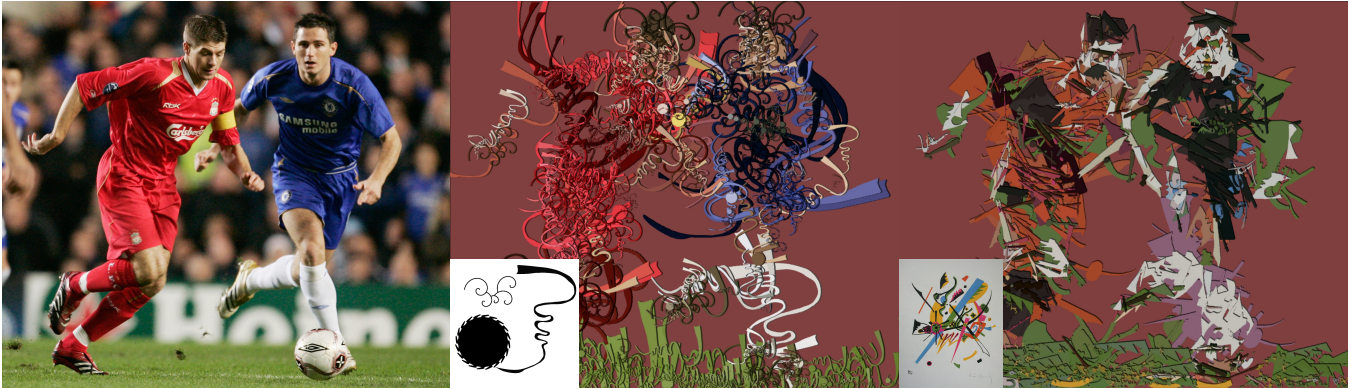
### 5.1 Image abstraction

The main ideathat we propose is to replace all shapes of a given image by shapes from a dictionary. This way, we produce images that at times can evoke the abstract paintings of the Suprematist school, while retaining the overall composition of a given input image. Observe however that if one were to establish a parallel with art, the proposed framework would evoke the approach of the Cubist school. Quoting Picasso, “*There is no abstract art. You must always start with something. Afterward you can remove all traces of reality, there is no hazard since the idea of the object has left an ineffaceable print*” [Elger, 2008].

**Single shape rendering.** All shapes of the image are replaced with a simple geometric primitive. In Figure 8, as well as Figure 5, we display examples obtained using, from left to right, a single rectangle, ellipse then disk as a dictionary. It is interesting, from a perceptual point of view, to observe how much visual information is gained by switching from disks to ellipses with varying elongation. In the Figure 5, and, last example Figure 8, the background was eliminated using a segmentation procedure [Felzenszwalb and Huttenlocher, 2004], shapes are rendered using the relief effect presented at the end of Section 4. For all the abstraction examples, the topographic map has been simplified using a contrast based method [Desolneux et al., 2001]. In short, only the scales of shapes and their spatial relationships are kept. From this point of view, these experiments are intriguing perception experiments, in which one may be surprised to perceive the meaning of the depicted scene.



**Figure 6: Shape transfer from a dictionary.** (left) Original image with a hand drawn dictionary, (right) the resulting shape transfer. Image courtesy of Lynn Ewing.



**Figure 7: Shape transfer from a dictionary. (Left) Original image, (middle) result of shape transfer along with its dictionary created with a drawing software and (right) a second result with a Kandinsky painting as a dictionary. Image credit: image courtesy of Iberico 90.9.**

*Rendering from a more complex dictionary.* Next, we show some examples in which the dictionary is made of more complex shapes. In Figure 6, the dictionary is made of hand-drawn shapes. In Figure 7 the dictionary is made of shapes obtained using a drawing software. This dictionary is also used in Figure 8 along with a segmented one (Mondrian painting). In these examples, the topographic map has again been simplified using the *sketch* [Desolneux et al., 2001]. In Figure 7, a segmentation procedure has been used to get rid of the background. As explained before, shapes from the dictionary are selected so as to minimize elongation and compactness differences with the replaced shape, following Formula (4), and then rotated and scaled to match the area and orientation of the replaced shape. The effects of this simple procedure may be observed in the heads and wings of the birds in Figure 6.

## 5.2 Painting-like effects

In this section, we illustrate the flexibility of the general framework presented above, by showing that it permits to obtain some painting-like effects by modifying the shapes of the colored tree of images. Contrarily to some classical rendering approaches [Gooch et al., 1999, Hertzmann, 2003], these effects are obtained without using any explicit brush model and only involves shape modifications.

*Shape shaking.* Camera blur in photographs usually results in many nested level lines along contours [Caselles et al., 1999]. We suggest to use this redundancy to produce oscillating boundaries, as may be encountered in oil-paintings. This effect is achieved by adding a random displacement to each shape of the colored tree of the input image, thus mimicking small oscillations produced by the painting tool. The resulting effects can be seen on the first column of Figure 9.

*Shape smoothing.* The second painting-like effect proposed in the same framework aims at producing images mimicking water-color paintings (Figure 9, second column). A median filter is applied individually to each shape of the color tree of the input image. This results in boundary motion, the motion being more important at points of high curvature. This approach shares some similarity with

works suggesting that interesting painting-like effects can be obtained by applying some operators from mathematical morphology, such as sequential openings and closings [Bousseau et al., 2007].

*Shape filtering.* Removing small shapes allows to abstract the image since these shapes represent the textures of the input image [Mould, 2012]. This simple filtering gives a painting-like feel to the images as illustrated in the third column Figure 9.

## 5.3 Style transfer

By using a painting as an input dictionary - *i.e.* by replacing the shapes from the input image by the shapes from a painting - we are able to transfer its style to the image. To do so, all the shapes of the colored tree are preserved in order to transfer the texture yielding the features of the painting. In the last two columns of Figure 9, the style from a painting is transferred to the input image. To do so, the distance  $d_2$ , from Formula (5), accounting for the shapes' colors, is used. Note that the color of the actual dictionary image is transferred during the rendering stage, in order to transfer the dictionaries textures. Even though the style transfer results presented here yield more artifacts than the ones produced by CNN methods, our framework produce esthetically pleasing results efficiently and is significantly easier to implement.

## 6 DISCUSSION

The main characteristic of our approach is to offer an image abstraction method using drastic shape simplification. In the same spirit, Song *et al.* propose a geometric image abstraction method using a normalized-cut segmentation in order to partition the image and replace each region with a given shape [Song et al., 2008]. This process results in a mosaic of shapes of a given size. To overcome this issue, the aforementioned process is performed using two segmentations of the image, each representing a given shape scale - *i.e.* a level of detail - that are later combined. In contrast, the topographic map naturally accounts for all scales in the image and their inclusions. As a result, the synthesis procedure proposed in



this paper enables complex interactions between shapes as may be seen in Figure 8.

Our system has some limitations. Indeed, certain abstraction processes are better suited for certain kinds of images. For instance, the style transfer is more appropriate for very contrasted images with a large number of details. Large uniform color regions in the input image will correspond to a single shape in the tree that will reproduce a large portion of the dictionary. Furthermore, the user has little control over the automated contrast-based simplification method used to produce the sketch of the image.

Our framework could be enriched with additional options such as an adaptive abstraction over the image. Either the user could provide a segmentation depicting regions of interest, or an abstraction field defining the intensity of the effect over the image. An example could be to apply less shaking or blurring around the eyes in the results presented Figure 9. Additionally, the shape's depth in the tree can be used to guide the stylization process. Moreover, instead of blurring the new shapes to overcome aliasing issues, we could generate an output image with a large resolution. The possible tree and shape modifications are not limited to the ones presented in this work, for instance, one could perform more drastic shape transformations, for stylization purposes, by adding perturbations or shape changes with reference to the shape's curvature. Finally, our stylization effects can be combined with existing stroke-based methods or region-based ones.

## 7 CONCLUSION

In this work, we have shown that a highly structured representation of images, the topographic map, allows to easily create a variety of NPR of digital photographs. Indeed, all the stylization effects presented in this paper - geometric abstraction, shape shaking and filtering, style transfer - are produced using simple operations (translation, scaling, duplication...) on each shape of the tree. Since the processes are performed iteratively on the colored tree, the user can easily adjust the intensity of the desired abstraction. Furthermore, we provide a single interface that allows for novice users to generate stylized images.

## ACKNOWLEDGEMENT

Gui-Song Xia is supported by the projects of National Natural Science Foundation of China (Grant No. 41501462 and No. 91338113). We would like to thank Jean-Marc Thiery for his insights and for providing the kd-tree code.

## REFERENCES

- J. Andrew Bangham, Stuart Gibson, and Richard Harvey. 2003. The art of scale-space. (2003), 569–578 pages.
- Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM* 18, 9 (1975), 509–517.
- James F. Blinn. 1978. Simulation of wrinkled surfaces. 12, 3 (1978), 286–292.
- Adrien Bousseau, Fabrice Neyret, Joëlle Thollot, and David Salesin. 2007. Video Watercolorization using Bidirectional Texture Advection. *ACM Transaction on Graphics* 26, 3 (2007).
- S. Brooks. 2007. Mixed media painting and portraiture. *IEEE Transactions on Visualization and Computer Graphics* 13, 5 (2007), 1041–1054.
- Frédéric Cao, Pablo Musé, and Frédéric Sur. 2005. Extracting Meaningful Curves from Images. *J. Math. Imaging Vis.* 22, 2-3 (2005), 159–181.
- Hamish Carr, Jack Snoeyink, and Ulrike Axen. 2003. Computing contour trees in all dimensions. *Computational Geometry* 24, 2 (2003), 75–94.
- V. Caselles, B. Coll, and J.-M. Morel. 1999. Topographic Maps and Local Contrast Changes in Natural Images. *International Journal of Computer Vision* 33, 1 (1999), 5–27.
- Vicent Caselles, Bartomeu Coll, and Jean-Michel Morel. 2002. Geometry and Color in Natural Images. *J. Math. Imaging Vis.* 16, 2 (2002), 89–105.
- Vicent Caselles and Pascal Monasse. 2008. *Geometric Description of Topographic Maps and Applications to Image Processing*. Springer.
- Bartomeu Coll and Jacques Froment. 2000. Topographic Maps of Color Images. (2000), 3613 pages.
- J. P. Collomosse and P. M. Hall. 2002. Painterly rendering using image salience. (2002), 122–128 pages.
- Doug DeCarlo and Anthony Santella. 2002. Stylization and abstraction of photographs. (2002), 769–776 pages.
- Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. 2001. Edge Detection by Helmholtz Principle. *J. Math. Imaging Vis.* 14, 3 (2001), 271–284.
- Alexei A. Efros and William T. Freeman. 2001. Image Quilting for Texture Synthesis and Transfer. (2001), 341–346.
- Alexei A. Efros and Thomas K. Leung. 1999. Texture Synthesis by Non-parametric Sampling. (1999), 1033–1038 pages.
- D. Elger. 2008. *Abstract Art*. Taschen.
- Pedro F Felzenszwalb and Daniel P Huttenlocher. 2004. Efficient graph-based image segmentation. *International journal of computer vision* 59, 2 (2004), 167–181.
- Jan Flusser and Tomas Suk. 1993. Pattern recognition by affine moment invariants. *Pattern recognition* 26, 1 (1993), 167–174.
- Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. 2016. Split and match: example-based adaptive patch sampling for unsupervised style transfer. (2016), 553–561 pages.
- Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. (2016), 2414–2423 pages.
- Jonas Gomes and Luiz Velho. 1995. Abstraction paradigms for computer graphics. *The Visual Computer* 11, 5 (1995), 227–239.
- Bruce Gooch, Greg Coombe, and Peter Shirley. 2002. Artistic Vision: painterly rendering using computer vision techniques. (2002), 83–90 pages.
- Bruce Gooch, Peter-Pike J. Sloan, Amy A. Gooch, Peter Shirley, and Richard Riesenfeld. 1999. Interactive Technical Illustration. (1999), 31–38 pages.
- Michèle Gouffès and Bertrand Zavidovique. 2008. A color topographic map based on the dichromatic reflectance model. *J. Image Video Process.* 2008, 2 (2008), 1–14.
- Charles Gueunet, Pierre Fortin, Julien Jomier, and Julien Thiery. 2016. Contour Forests: Fast Multi-threaded Augmented Contour Trees. (2016).
- P. Haeberli. 1990. Paint by numbers: Abstract image representations. (1990), 207–214 pages.
- Hazem M Hamdan and Lee M Larson. 2002. Texture classification through level lines. (2002), 937–940 pages.
- James Hays and Irfan Essa. 2004. Image and video based painterly animation. (2004), 113–120 pages.
- Aaron Hertzmann. 1998. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. (1998), 453–460 pages.
- A. Hertzmann. 2001. Paint By Relaxation. In *Proc. of the International Conference on Computer Graphics*. IEEE Computer Society Washington, DC, USA.
- Aaron Hertzmann. 2003. A Survey of Stroke-Based Rendering. *IEEE Computer Graphics and Applications* 23, 4 (2003), 70–81. Special Issue on Non-Photorealistic Rendering.
- Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. 2001. Image analogies. (2001), 327–340 pages.
- Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. (2016), 694–711 pages.
- H. Kang and S. Lee. 2008. Shape-simplifying Image Abstraction. In *Computer Graphics Forum*, Vol. 27. Blackwell Publishing Ltd, 1773–1780.
- Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenber. 2013. State of the Artfi: A Taxonomy of Artistic Stylization Techniques for Images and Video. *IEEE transactions on visualization and computer graphics* 19, 5 (2013), 866–885.
- Hochang Lee, Sanghyun Seo, Seungtaek Ryoo, and Kyunghyun Yoon. 2010. Directional Texture Transfer. (2010), 6 pages.
- Guanyu Lian, Yanli Wang, Kang Zhang, and Li Yao. 2016. (2016), 53 pages.
- Peter Litwinowicz. 1997. Processing images and video for an impressionist effect. (1997), 407–414 pages.
- Xiaofeng Mi, Doug DeCarlo, and Matthew Stone. 2009. Abstraction of 2D shapes in terms of parts. (2009), 15–24 pages.
- Pascal Monasse. 1999. Contrast invariant registration of images. (1999), 3221–3224 pages.
- P. Monasse and F. Guichard. 2000. Fast Computation of a Contrast Invariant Image Representation. *IEEE Trans. Image Processing* 9, 5 (2000), 860–872.
- D. Mould. 2003. A stained glass image filter. (2003), 20–25 pages.
- David Mould. 2012. Texture-preserving abstraction. (2012), 75–82 pages.
- David Mould. 2013. Region-based abstraction. In *Image and Video-Based Artistic Stylisation*. Springer, 125–147.
- David Mould and Paul L Rosin. 2016. A benchmark image set for evaluating stylization. (2016), 11–20 pages.

- Alexandrina Orzan, Adrien Bousseau, Pascal Barla, and Joëlle Thollot. 2007. Structure-preserving manipulation of photographs. (2007), 103–110 pages.
- Paul Rosin and John Collomosse. 2012. *Image and Video-Based Artistic Stylisation*. Vol. 42. Springer Science & Business Media.
- Paul L. Rosin. 2000. Measuring shape: ellipticity, rectangularity, and triangularity. (2000), 952–955 pages.
- Anthony Santella and Doug DeCarlo. 2004. Visual interest and NPR: an evaluation and manifesto. (2004), 71–150 pages.
- V. Setlur and S. Wilkinson. 2006. Automatic stained glass rendering. *Lecture notes in computer science* 4035 (2006), 682.
- Yi-Zhe Song, Paul L. Rosin, Peter M. Hall, and John P. Collomosse. 2008. Arty Shapes. (2008), 65–72 pages.
- Bin Wang, Wenping Wang, Huaiping Yang, and Jianguang Sun. 2004. Efficient example-based painting and synthesis of 2D directional texture. *Visualization and Computer Graphics, IEEE Transactions on* 10, 3 (2004), 266–277.
- G. Winkenbach and D. H. Salesin. 1994. Computer-Generated Pen-And-Ink Illustration. (1994).
- Gui-Song Xia, Julie Delon, and Yann Gousseau. 2010. Shape-based invariant texture indexing. *International Journal of Computer Vision* 88, 3 (2010), 382–403.
- Lu Xiong and Kang Zhang. 2016. Generation of Miro's Surrealism. (2016), 8 pages.
- Mingtian Zhao and Song-Chun Zhu. 2010. Sisley the abstract painter. (2010), 99–107 pages.
- Yan Zheng, Xuecheng Nie, Zhaopeng Meng, Wei Feng, and Kang Zhang. 2015. Layered modeling and generation of Pollock's drip style. *The Visual Computer* 31, 5 (2015), 589–600.



Figure 8: Image abstraction. (Left column) Input photographs from the benchmark image set for evaluating stylization [Mould and Rosin, 2016]. We compute the sketch of the input images (contrast filtering of shapes) and replace the remaining shapes by (from left to right): rectangles, ellipses, disks and shapes extracted from segmented dictionaries.

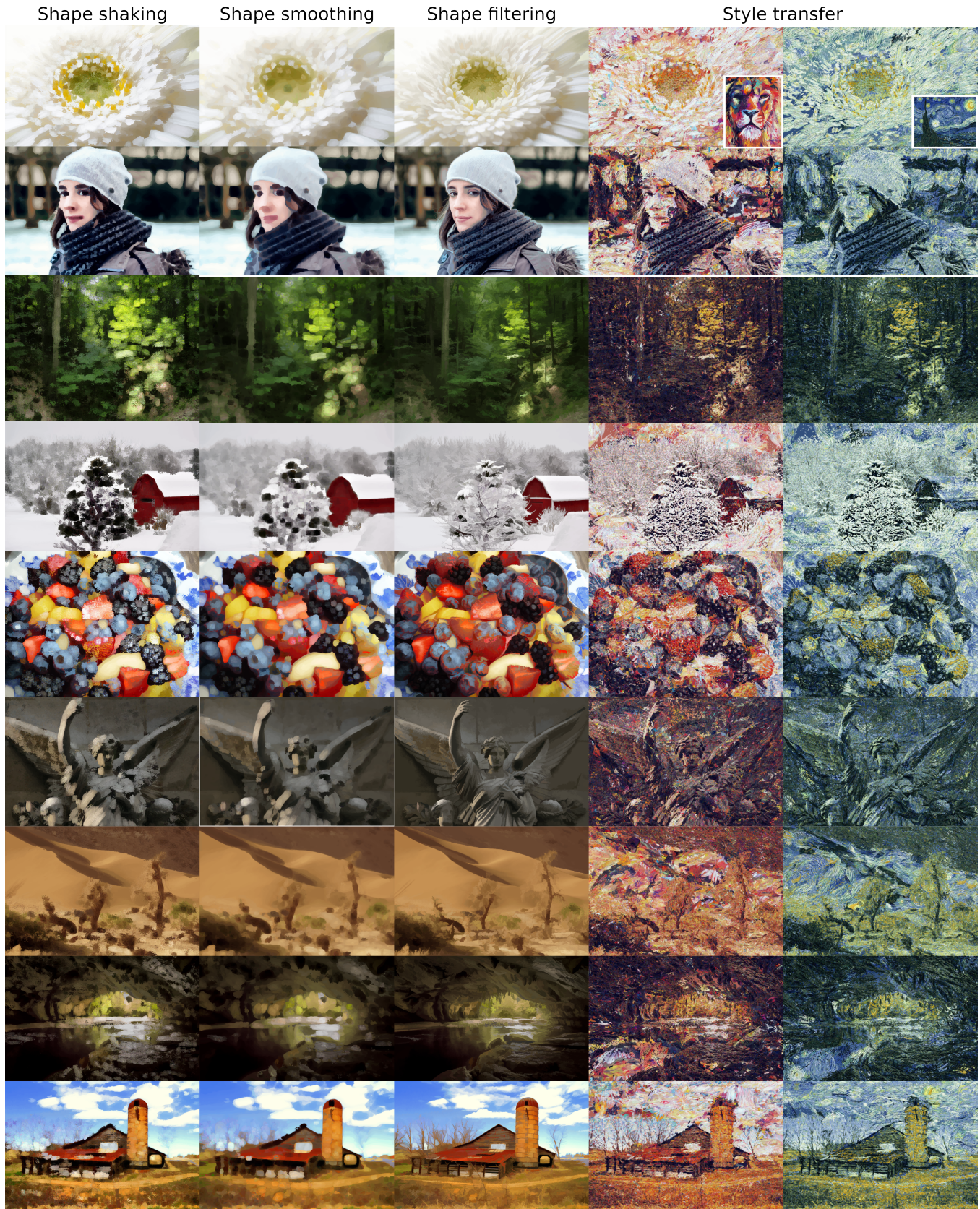


Figure 9: Painting-like and style transfer. (First column) Stylization using shape shaking, (second column) a water color effect is produced by adding a median filter and blur. (Third column) We retain shapes larger than a given threshold, hence, removing small texture details resulting in smoother regions. The two last columns depict style transfer from two different paintings.