Depicting Stylized Materials with Vector Shade Trees

Jorge Lopez-Moreno, Popov Stefan, Adrien Bousseau, Maneesh Agrawala, George Drettakis

▶ **To cite this version:**

Jorge Lopez-Moreno, Popov Stefan, Adrien Bousseau, Maneesh Agrawala, George Drettakis. Depicting Stylized Materials with Vector Shade Trees. ACM Transactions on Graphics, 2013, SIGGRAPH Conference Proceedings, 32 (4). hal-00828067

HAL Id: hal-00828067

https://inria.hal.science/hal-00828067

Submitted on 31 May 2013

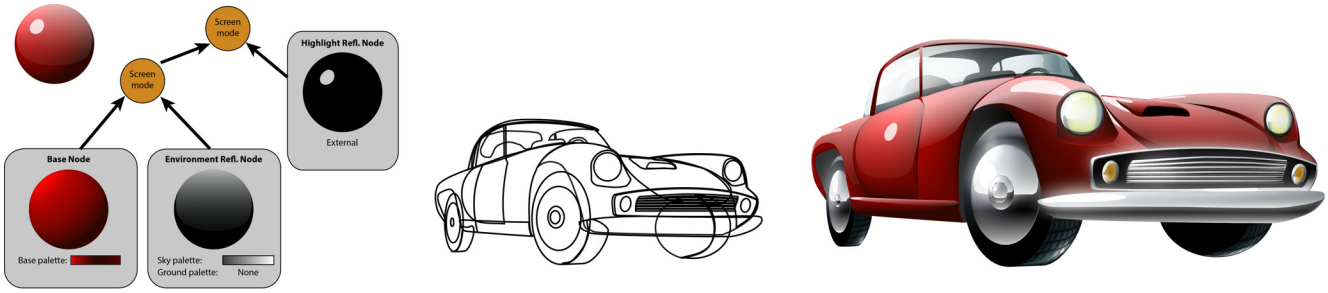
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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.

Depicting Stylized Materials with Vector Shade Trees

Jorge Lopez-Moreno[1] Stefan Popov[1] Adrien Bousseau[1] Maneesh Agrawala[2] George Drettakis[1]

[1] REVES / INRIA Sophia - Antipolis [2] University of California Berkeley



(a) Vector Shade Tree (b) Vector line drawing (c) Our material depiction

Figure 1: *We describe* Vector Shade Trees *that represent stylized materials as a combination of basic shade nodes composed of vector graphics primitives (a). Combining these nodes allows the depiction of a variety of materials while preserving traditional vector drawing style and practice. We integrate our vector shade trees in a vector drawing tool that allows users to apply stylized shading effects on vector line drawings (b,c). This pdf contains png versions of vector art to avoid viewer compatibility problems; full vector versions of our results are in supplemental material. Original line drawing from koconmus, openclipart.org©, colored version ©the authors .*

Abstract

Vector graphics represent images with compact, editable and scalable primitives. Skillful vector artists employ these primitives to produce vivid depictions of material appearance and lighting. However, such stylized imagery often requires building complex multi-layered combinations of colored fills and gradient meshes. We facilitate this task by introducing *vector shade trees* that bring to vector graphics the flexibility of modular shading representations as known in the 3D rendering community. In contrast to traditional shade trees that combine pixel and vertex shaders, our shade nodes encapsulate the creation and blending of vector primitives that vector artists routinely use. We propose a set of basic shade nodes that we design to respect the traditional guidelines on material depiction described in drawing books and tutorials. We integrate our representation as an Adobe Illustrator plug-in that allows even inexperienced users to take a line drawing, apply a few clicks and obtain a fully colored illustration. More experienced artists can easily refine the illustration, adding more details and visual features, while using all the vector drawing tools they are already familiar with. We demonstrate the power of our representation by quickly generating illustrations of complex objects and materials.

Keywords: Material Depiction, Artistic Guidelines, Vector Graphics, Shade Trees, Gradient Mesh

Links: DL PDF WEB VIDEO DATA

1 Introduction

Vector graphics have many advantages over bitmaps: they are compact, scalable and allow high-level manipulation of shapes rather than pixels. Artists often use the distinctively clean and sharp look of vector graphics to create stylized illustrations. Yet, despite the advantages of vector art, creating high-quality illustrations using vector graphics software is tedious and complex. For example, to color a line drawing artists must carefully apply color brushes, paths and gradients to each closed region. Moreover, to depict materials such as chrome, plastic or glass, artists must usually layer multiple colored fills and gradients in each region. Even if the line drawing is provided beforehand, only expert illustrators are even capable of creating illustrations with rich stylized materials.

In contrast, in 3D graphics, adding materials to geometry is simple. Early work such as Shade Trees [Cook 1984] provided a way to easily construct the appearance of complex materials based on a combination of a few basic nodes. Such approaches have now become commonplace in 3D modeling software, where users can specify and edit trees which define a complex appearance. These trees contain nodes, e.g., colors, texture or vertex/pixel shaders that can be combined to create materials such as wood, metal etc. However, directly applying the 3D shade tree pipeline to vector graphics would be require a complete change of artists' workflow, since the primitives and controls in each case are very different. We take inspiration from the flexibility of 3D shade trees, but strive to fully respect vector artists tools and workflows.

We automate the process of stylized material depiction for vector art and allow inexperienced users to take a line drawing like the one in Fig. 1b, and obtain a fully colored figure like Fig. 1c by applying a few clicks. Our solution also allows more experienced artists who are familiar with vector graphics software to easily refine the illustration, adding more details and visual features using the tools they already know.

Part of the difficulty in depicting stylized materials stems from the fact that the techniques artists use to draw materials are based on accumulated artistic knowledge. These *guidelines* they use to de-

pict appearance are spread across a variety of books and tutorials, and are often intertwined with information about how to draw the shape of an object. To build a tool facilitating stylized material depiction we first select, classify and generalize a small and coherent set of guidelines that artists combine together to generate a variety of appearances. Our guidelines describe artists' current workflows (the tools they use, their vocabulary), and decompose these workflows into components suitable for integration into a functional vector drawing system.

Our analysis directly informs the design of a *vector shade tree* representation that allows the instantiation of the guidelines in vector graphics systems. Using our guideline classification we derive a compact set of shade nodes, that are sufficient to express a wide range of materials while respecting artistic workflows. These nodes are based on primitives commonly used by vector artists, such as gradient fills, gradient meshes and paths. We provide high-level user controls over each node, such as light direction or elevation of horizon line, as well as finer level controls using existing vector drawing tools.

In summary, our work makes two main contributions:

- We distill and unify guidelines on material depiction from a variety of sources.
- We propose *vector shade trees* to express these guidelines using a small yet expressive set of *nodes*. These nodes rely on standard primitives and controls that vector artists are already familiar with.

We have implemented our solution as an Adobe Illustrator plug-in. Several professionally trained artists were able to use our tool, providing strong indications that our solution has significant potential in accelerating the artistic process, or alternatively providing much higher quality results for the same amount of time spent on an illustration.

2 Related Work

Non photorealistic rendering. Artists often depart from realism to depict the appearance of an object in a simplified or exaggerated manner. While numerous methods have been proposed to render images in a particular style (e.g., painting [Hertzmann 1998], pen and ink [Winkenbach and Salesin 1994], watercolor [Curtis et al. 1997]), few of these methods focus on non-photorealistic rendering of material appearance.

Several methods, such as the *Lit Sphere* [Sloan et al. 2001] and *XToon* [Barla et al. 2006] extend the classic toon shader to encode non-photorealistic shading models in a texture that maps surface normal to color. Artists can produce different appearances by varying the shading texture. However, this representation prevents explicit control over the combination of the different components of material appearance. In addition, while artists can sample shading textures from existing illustrations, drawing an original texture by hand requires knowledge of material depiction guidelines.

Focusing on pen-and-ink rendering of specular objects, Kim et al. [2008] propose an algorithm to orient strokes according to the shape of reflected surfaces rather than the reflector. Anjyo and Hiramitsu [2003] introduce stylized highlights for cartoon rendering that mimic practice in traditional animation, such as squared window-shaped reflections. Gooch et al. [1998] take inspiration from technical illustration to design a non-photorealistic lighting model that conveys metallic appearance. We adopt a similar methodology since we analyze illustrator practice to identify material depiction guidelines. However, we focus on vector graphics rather than 3D rendering and we describe new guidelines for the depiction of other common material properties such as glossy reflections, mirror reflections, transparency and translucence. Combined together in our shade trees, these properties allow the generation of a much larger set of stylized materials.

In the context of photorealistic rendering, Bousseau et al. [2011] take inspiration from lighting design practice to generate environment maps that enhance material depiction. We share a similar approach and adapt some of their principles to the context of vector graphics.

The seminal Shade Trees paper by Cook [Cook 1984] also introduced the concept of shading languages to the realistic rendering community; this approach was a rich source of inspiration for our work. Grabli et al. [2010] adapt this concept to the stylization of strokes in line drawings and describe programmable shaders to control style attributes such as color, texture and thickness. We adopt a complementary approach by focusing on the stylized depiction of materials, giving control to material attributes like shading and reflections.

Vector graphics. For a long time, vector images have been restricted to the flat or limited shading offered by linear and radial gradients. Recently however, researchers have introduced new vector primitives that offer much richer gradient capabilities. Diffusion Curves [Orzan et al. 2008] and its variants [Bezerra et al. 2010; Finch et al. 2011; Boyé et al. 2012] produce smooth gradients by interpolating colors between freeform parametric curves. Instead, *gradient meshes*, as available in Adobe Illustrator and CorelDraw, interpolate colors over the parametric patches of a mesh. We rely on gradient meshes as a core component of our system. We enrich the standard mesh tool with a normal estimation that automates shading, and we also use it as a deformation tool to mimic texture foreshortening and highlight distortions.

While these advanced gradient tools make it a little easier to create high-quality illustrations, it still requires significant expertise to create objects with plausible material appearance. Several methods facilitate this illustration process by automatically vectorizing a photograph [Sun et al. 2007; Lai et al. 2009; Liao et al. 2012] into a gradient mesh. However the resulting gradient mesh is often very complex making it difficult to control and edit. Eisemann et al. [2008] render 3D models into vector graphics, a feature that Adobe Illustrator also provides to create gradient meshes from parametric surfaces. However, these methods rely on cartoon and Phong shading models that are not suitable for visual effects like transparency, translucence and environmental reflections.

3 Material Depiction Guidelines

Books on drawing and illustration present step-by-step instructions explaining how to render common materials (e.g. plastic, wood, metal, glass, etc.) using traditional colored media including airbrushes [Martin 1989a; Martin 1989b; Martin 1989c], markers [McGarry and Madsen 1992; Eissen and Steur 2008], or pastels [Powell 1986; Doyle 2006]. Digital artists have adapted these material depiction techniques to vector drawing software (e.g., Adobe Illustrator) using two main tools: 1) gradient fills and 2) layer blending. While simple linear and radial gradients are well suited to depicting flat and spherical surfaces, more complicated surfaces require gradient meshes which provide parametric patches that interpolate colored vertices. Blending modes (e.g., linear, multiplication, screen blending) allow artists to layer and combine the fills to produce more elaborate shading effects.

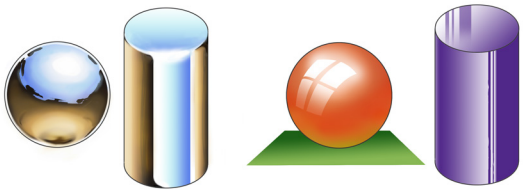


Figure 2: *Artists use an abstract environment to depict reflections. While metallic materials produce strong mirror-like reflections (a), artists depict the weaker reflections over glossy materials by only drawing the brightest light sources such as the sky dome or a window (b). After [Powell 1986](a) and [Martin 1989a](b).*

Vector illustration websites (e.g., vector.tutplus.com, designinstruct.com) contain detailed tutorials describing the gradient fill and layer blending operations necessary to depict a wide variety of materials. However, these tutorials usually explain how to create a complete industrial object (e.g., kitchen pot, hairdryer, etc.) and therefore intertwine geometry creation with material depiction. As a result, guidelines for depicting materials are spread across examples found in many different books and websites.

We have studied this literature to distill a general set of guidelines for depicting several important classes of materials including materials with matte appearance (clay, matte paint), reflective materials (metals, plastic, porcelain, leather), transparent materials (glass, ice), translucent materials (wax, jelly), and textured materials (fabrics, wood). We organize our guidelines into five corresponding categories based on the shading layers artists typically use to represent materials, i.e., base shading, reflections, transparency, translucency and texture. We also present the manual steps required to generate each layer using current vector graphics tools. Note that the Figures 2–9 illustrating these drawing techniques have not been generated with our tool. In Section 4 we show how these guidelines inspire our vector shade tree representation.

3.1 Base shading (clay, matte paint)

Artists usually start depicting a material by defining a "base palette" of colors representing the bright, mid-range and dark parts of the shape. They then color the corresponding regions (bright, mid-range and dark) of the object surface according to its 3D shape and a rough notion of the light position. Finally, they gradually blend these palette colors to fill the in-between regions and generate a complete base shading layer. One traditional approach is to paint these gradients with an airbrush [Powell 1986]. This base shading layer can be used to depict diffuse materials like clay and rough matte paint. It also serves as a base for more complex materials.

In vector graphics, artists use radial and linear gradients to depict base shading on simple objects such as spheres and cylinders. Complex shapes require carefully crafted gradient meshes that properly convey shape variations and lighting. To generate a convincing result, artists commonly sample the bright, mid-range and dark palette colors from photographs of the material and encode them as *stops* in a color ramp.

3.2 Reflections (metals, plastic, porcelain)

Many materials such as metals, plastic and porcelain reflect the surrounding environment to generate strong visible highlights. To avoid cluttering the illustration with lots of highights, artists typically depict reflections with simple abstract environments composed of a sky dome and a ground separated by an horizon line.

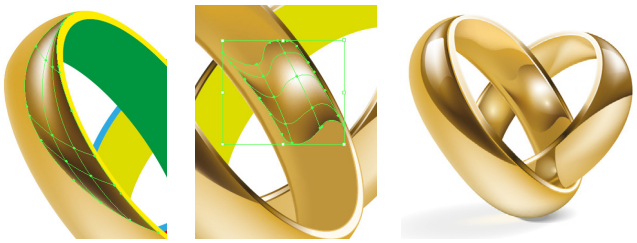


Figure 3: *A tutorial shows how to use gradient meshes to draw a horizon line (a) and highlights (b) over a gold ring (c). Iaroslav Lazunov, Vectorboom.com©.*

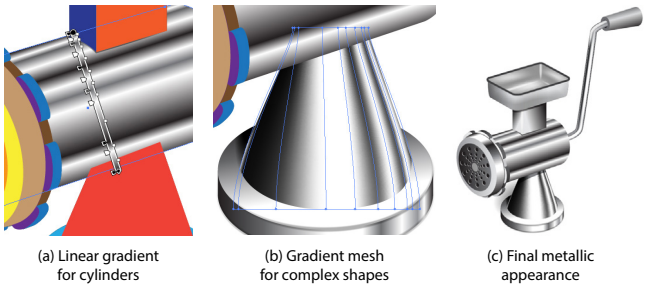
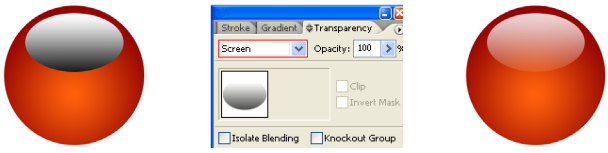


Figure 4: *An online tutorial shows how to illustrate a metallic tube using a multi-stop linear gradient (a) and metallic cone with a gradient mesh (b). The final result convey the appearance of brushed metal (c). Alexander Egupov, vector.tutsplus.com©.*

Small highlights convey the reflection of bright light sources and the shape of the highlight can indicate the type of light source (e.g., four small squares representing a window, a circle representing the sun, etc.), [Martin 1989a]. Artists further distort the horizon line and highlights to convey the curvature of the underlying surface [Powell 1986; Eissen and Steur 2008] . Finally, artists also position the reflection of the horizon line as a cue to depict view direction: a low horizon line depicts a view from above while a high horizon line depicts a view from below.

Metallic reflection. Artists convey the mirror-like appearance of metals by coloring the object with the colors of the sky and ground of the environment (Fig. 2a). The sky is usually drawn with a gradient that gets brighter at the horizon while the ground gets darker, which mimics atmospheric effects and enhances contrast [Robertson 2003]. Airbrush artists from the 70s often used a "desert" environment with a brown gradient representing the ground, a light blue gradient for the sky, and a bumpy horizon line for dunes [Powell 1986]. Today, artists often use a high-contrast gray environment with a straight horizon to produce a colder, more modern metallic look. Colored metals like gold produce colored reflections. To produce such colored reflections artists modulate the hue of a gray environment by the base hue of the metal [Martin 1989b]. In drawing metallic tubes and cones, which are highly curved in one direction and flat in the other, artists create elongated reflections. They alternate bright and dark bands aligned with the direction of least curvature [Eissen and Steur 2008; Gooch et al. 1998] (Fig. 2a, right).

In vector graphics, artists use gradient meshes to draw horizon lines and highlights, distorting the mesh to convey curvature, as illustrated in Fig. 3. Metallic cylinders are drawn with multi-stop linear gradients that produce dark and bright bands (Fig. 4a) while conic shapes require the use of gradient meshes to capture the more complex curvature lines (Fig. 4b).



(a) Highlight shape filledwith linear gradient (b) Screen blending (c) Final glossy appearance

Figure 5: *Glossy reflections are drawn with a gray-to-white gradient (a) blended in screen mode (b). The gradient mimics stronger reflections at grazing angle due to the Fresnel effect, while the screen blending mode lets the base color appear through the weaker part of the reflection (c).*

Glossy reflection. Glossy materials such as plastic and porcelain are not as reflective as metals. As a result, artists only depict the reflection of the strongest light sources in the environment such as the sky dome or the rectangular shape of a window [Martin 1989a] (Fig. 2b). These highlights are often painted in white to avoid ambiguity between the color of the light and the color of the object. Artists also exaggerate the strength of the reflections at grazing angle to convey the Fresnel effect [Robertson 2003]. Outside of the highlights, the object is rendered using the base shading palette [Eissen and Steur 2008].

To depict glossy reflections, vector artists first fill the reflection shape with a black-to-white gradient and then blend the reflection over the underlying color region with the "screen" blending mode, as shown in Fig. 5. This blending mode makes the bright area of the reflection opaque and the dark area transparent, revealing the base color underneath. The gradient makes the reflection stronger towards the edge of the object, mimicking the Fresnel effect.

3.3 Transparency (Glass, Ice)

Artists often depict a transparent object by drawing another object behind it and ensuring that this background object is visible through the foreground object. In the absence of other objects, a colored background is often enough to suggest transparency [Eissen and Steur 2008] (Fig. 6a). In addition, colored transparent materials attenuate the background color. Empty objects like bottles attenuate more light along their silhouette where they get thicker. Airbrush artists reproduce this effects by attenuating the background with semi-transparent layers of pigment [Martin 1989c].

Bright highlights are another important visual cue for depicting the shape of transparent objects. Artists often place highlights in multiple locations due to multiple scattering within the object. For simple shapes, these internal highlights are often placed symmetrically to the external highlights with respect to the center of the shape (Fig. 6). In addition, internal highlights are more faint than external ones since they are attenuated by the material.

In addition to bright internal highlights, artists often draw dark bands near the silhouette of transparent objects to depict total internal reflections. These bands delineate the contours of the object with a strong contrasting edge and the width of the bands suggest the thickness of the material.

Vector artists combine multiple abstract shapes to convey external and internal reflections. As for other glossy materials, artists draw reflections with paths or gradient meshes and fill them with dark-to-bright gradients composited in screen mode (Figure 7b). Paths are also used to draw sharp total internal reflections that delineate the overall silhouette of the object (Fig. 7a).

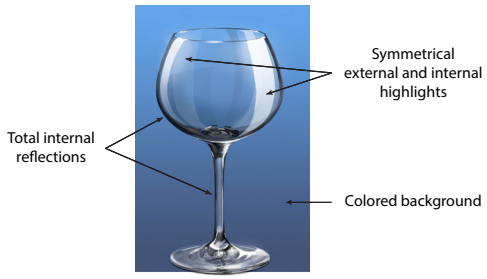
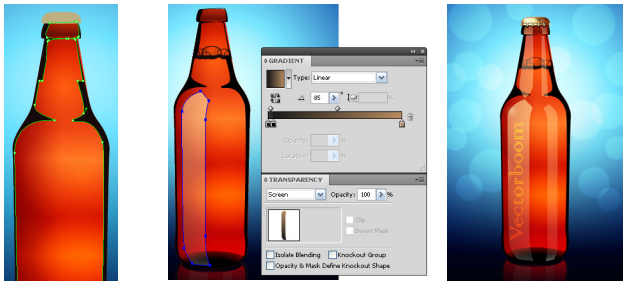


Figure 6: *A colored background helps conveying the transparency of glass. Multiple internal and external highlights, combined with total internal reflections, delineate the shape of transparent objects. After [Powell 1986] and [Martin 1989c].* © *J. Lopez-Moreno.*



(a) Total internal reflections (b) Highlights blended in screen mode (c) Final glass appearance

Figure 7: *Online tutorial to create a beer bottle. First, dark total internal reflections are drawn along the silhouette of the object (a). Highlights are then added by blending dark-to-bright gradients in screen mode (b). Iaroslav Lazunov, Vectorboom.com*©.

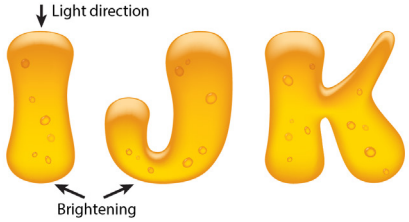
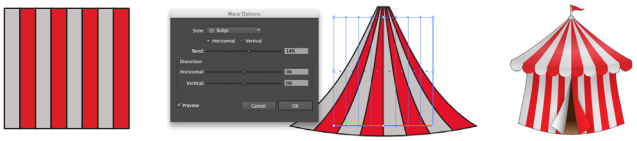


Figure 8: *Artists depict light scattered through translucent material by brightening the parts opposite to the light direction. In this example of translucent letters, the highlights convey light coming from the top while the shading makes the bottom part brighter. Maisei Raman, shutterstock.com*©.

3.4 Translucency (Wax, Jelly)

A distinctive visual feature of translucent objects is their ability to scatter light through multiple internal reflections. Artists convey light exiting the object after internal scattering by brightening the parts of the object that are directly opposite from the point at which the light strikes the surface (Fig. 8).

In vector graphics, artists depict translucency with the same gradient tools as base shading to convey the smooth color variations produced by light scattering. However in this case the brightest palette color appears where the light exits the object, directly opposite from the point where the light enters the object. They then create additional reflections and highlights using the same approach as for transparency. Artists also usually apply a blur to the internal reflections to convey the scattering of light through the material.



(a) Texture (b) Warping tool (c) Textured object

Figure 9: *Warping tools allow the distortion of vector primitives to mimic texture foreshortening. Andrei Marius, vector.tutsplus.com*©.

3.5 Textured Objects (fabrics, wood)

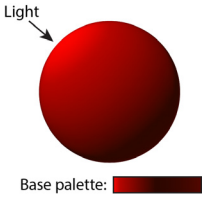
Artists often use textures to add realism to a drawing with little effort and to provide context and scale cues [Eissen and Steur 2011]. Artists also distort textures to convey foreshortening [Eissen and Steur 2008].

Users of vector graphics tool often rely on warping to distort texture patterns. Different warping features exist, either based on cages or on parametric functions (inflate, squeeze, twist, bend), as illustrated in Figure 9.

4 Vector Graphics Shade Trees

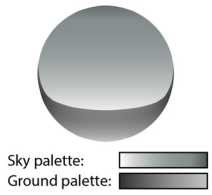
Our trees and their constituent nodes are based on the material depiction guidelines we presented in Section 3. We define a small set of leaf nodes, which are combined together with *blending* operations, represented as interior nodes in our trees. Blending modes can be simple alpha blending, screen mode, multiply mode or additive mode. The order of blending is important; in the trees we show, the right-hand node is blended on top of the left-hand node.

We define a set of leaf nodes, which correspond to the five classes of guidelines we presented in Section 3: *Base*, *Reflections* – including *Environment*, *Highlight* and *Total Internal* reflections – *Transparency*, *Translucency* and *Texture*. Each of these nodes is represented as a vector art primitive – either a gradient mesh or a path – and one or several color ramps. The vector art represents the shape of the material effect while the ramps encode its color palettes. We also enrich the gradient mesh with normals, as detailed in Sec. 6. Artists can interact with the shape and color palettes of a shade node via the user interface described in Sec. 5. Some nodes also have specific parameters that are either defined during the tree creation or manipulated with the user interface. Finally, artists can specify a light source direction that is treated as a global parameter for all the nodes. We set the default light direction to the upper left, following common practice in design illustration.

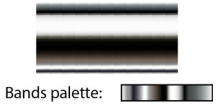


Base node. This node corresponds to the *base shading* guideline of Sec. 3.1. We represent the smooth shading with a gradient mesh and we use our normal estimation to compute the brightness of each vertex with respect to the light direction. This brightness is then modulated by the color palette of the node which contains the bright, mid-range and dark base colors.

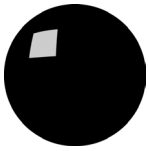
Reflection nodes. We define three types of reflection nodes that represent *environment reflections*, *highlight reflections*, and *total internal reflections* (Sec. 3.2). Similarly to vector drawing practice, the environment and highlight reflections are blended in screen mode to brighten base nodes, while the total internal reflection node is blended in multiply mode to darken the contour of an object.



The Environment Reflection node corresponds to an abstract environment with a sky, ground and horizon line. We represent the horizon line with a path and the ground and sky color gradients with color ramps. The ground gradient covers the object from the bottom to the horizon line, and the sky covers the other part, from horizon to the top. The ground gradient is optional for glossy materials that only reflect the sky dome.

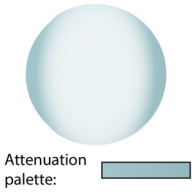


For anisotropic shapes like cylinders, we replace the abstract environment by a gradient mesh that covers the object (see Sec. 6 for implementation details). In this case, the palette of the node represents bright and color bands similar to the ones used by traditional artists. Users can indicate if a shape is anisotropic with a dedicated interface (Sec. 5).

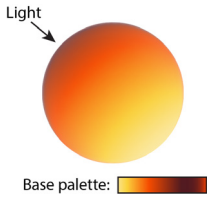


The Highlight Reflection node represents reflections of bright light sources such as spot lights or windows. We represent the shape of an highlight with a path that we optionally blur to mimic rough reflections. We then derive from our normals and the light direction the location of the highlights over the shape (see implementation in Sec. 6). Highlights can optionally be considered as *internal*, in which case we compute their location using the inverted light direction. We set this parameter when creating the tree, as in the Glass example in Figure 10.

Total internal reflections produce dark bands along the edge of a shape. We use a path to represent these bands that artists can distort to suggest a varying thickness along the silhouette.



Transparency node. The transparency node contains a semi-transparent gradient mesh to attenuate the background color (Sec. 3.3). This gradient mesh is more opaque along silhouettes to mimic the way empty objects attenuate more light where they are thicker. The color palette of this node encodes the color of the material function of its opacity.



Translucency node. This node is similar to the base node of opaque objects but encodes an inverted shading computation that makes objects brighter in the direction opposite to the light direction (Sec. 3.4). We use our estimated normals to compute and represent this shading as a gradient mesh. In addition, the translucent node is also slightly transparent to reveal the background.

Texture Node The texture node relies on a gradient mesh to distort a texture according to foreshortening (Sec. 3.5). In our implementation we only use vector textures to maintain the benefits of vector graphics throughout.

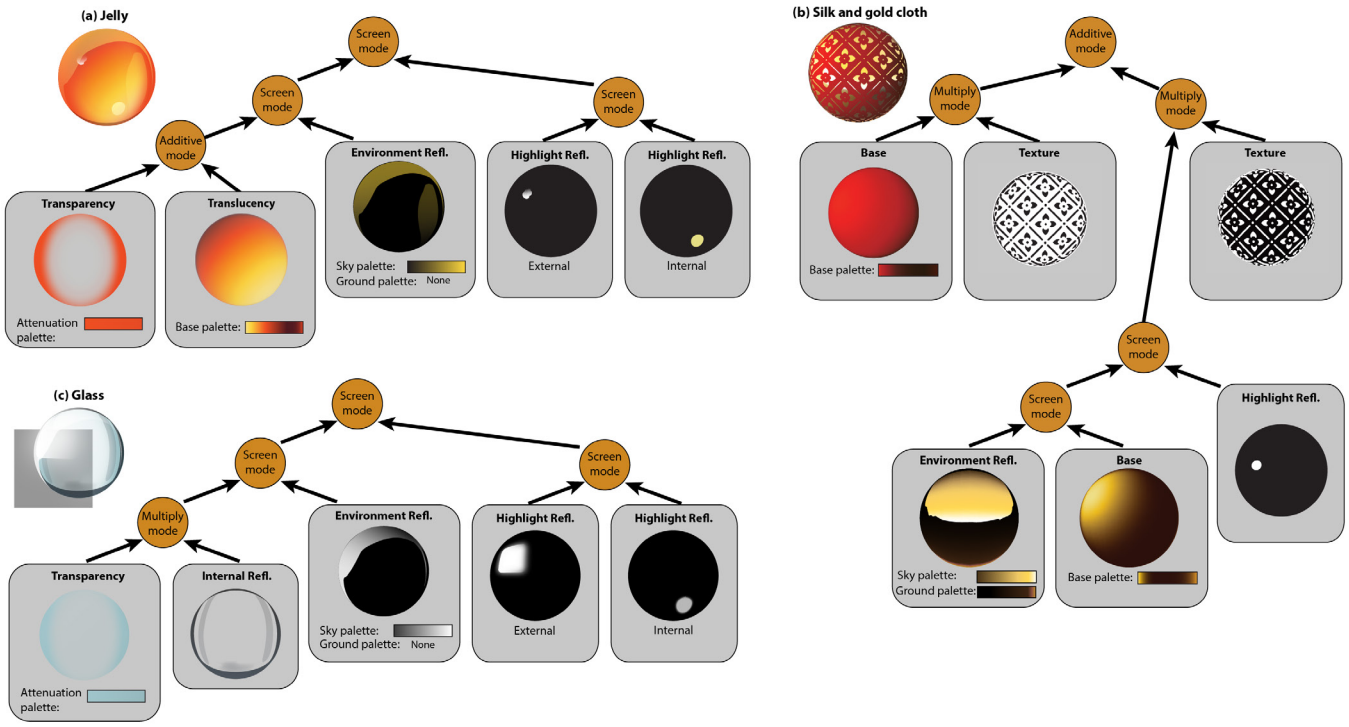


Figure 10: *Vector Shade Trees for three materials: jelly, silk and gold cloth and glass. We use a texture node to combine a red base node with the shade tree of gold and create a rich fabric (b). While the environment reflection greatly contributes to the rendition of gold, we further improve the result by brightening the environment around the light source using a Base node blended in Screen mode. We obtain the complex appearance of glass by combining multiple reflection nodes (c). In particular, we apply highlight nodes with symmetric internal and external reflections, although the internal highlight is more blurry and faint. The jelly material (a) is very similar to glass (b), except that it has a translucency node instead of total internal reflections.*

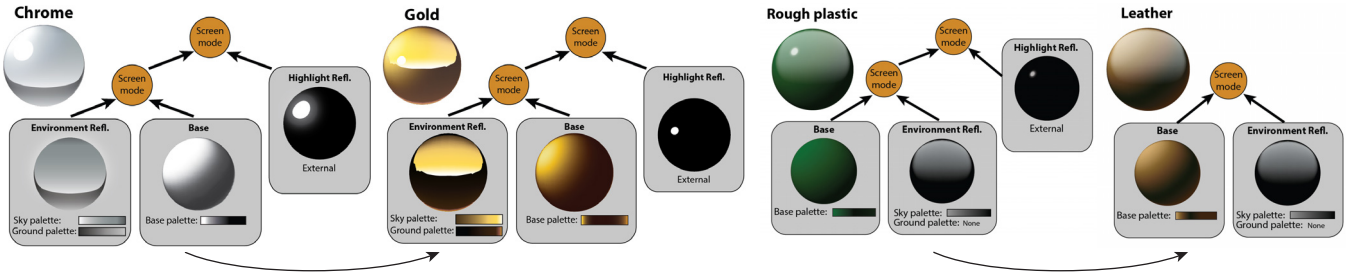


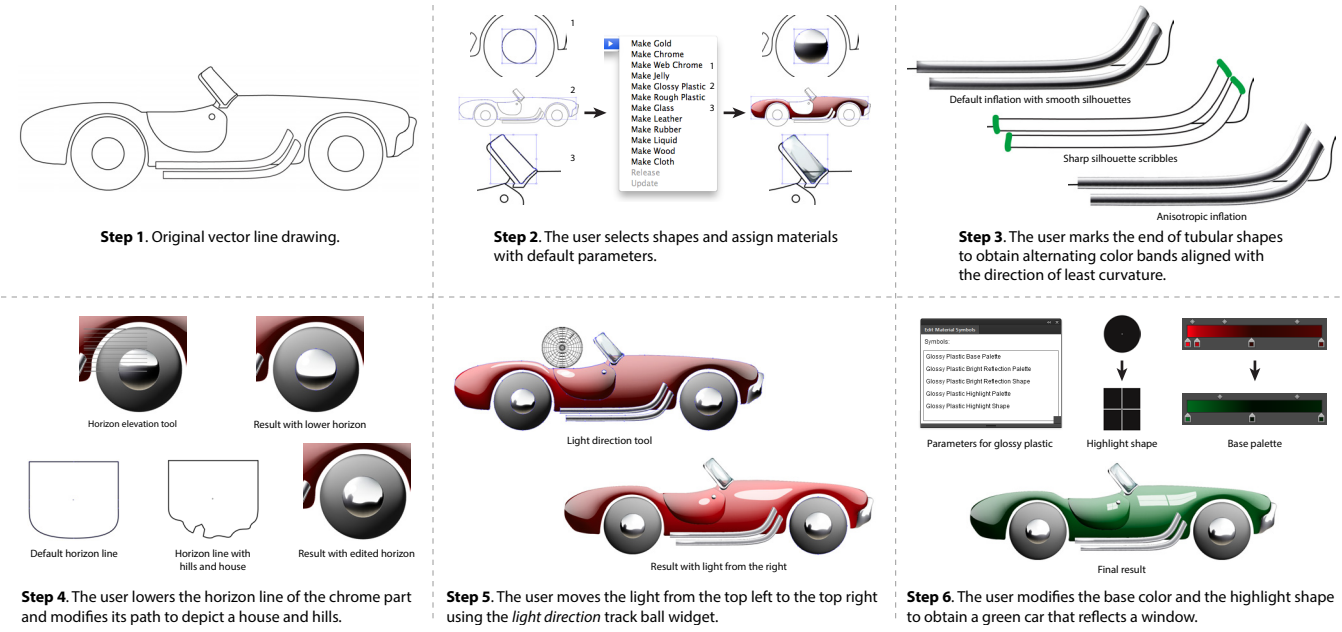
Figure 11: *Shade tree reuse. We created the shade tree for gold by modifying the palettes and the horizon line of chrome. The leather material is similar to rough plastic, without the highlight.*

Shade trees. We combined our basic shade nodes to create a variety of materials including glossy and rough plastic, chrome, rubber, glass, gold, wax, jelly, leather, wood, fabric. Fig. 1a describes the shade tree for a glossy car paint. We use an Environment Reflection node to depict the reflection of the sky dome, where the dark-to-white gradient strengthen the reflection near the top of the object to mimic Fresnel effect. In Fig. 10 we present several other shade trees with their nodes, shapes and color palettes. The set of trees we have created for the other materials in this paper are given as supplemental material.

Similarly to 3D shaders, we expect our vector shade trees to be reused and combined to create libraries of more complex materials. As an example, we derived the shade tree of the leather bag on the motorbike in Fig. 17a from the tree of rough plastic by removing the highlight node and changing the palette of the base nodes (Fig. 11). The cloth in Fig. 10 also reuses the shade tree of gold.

5 Interface for Material Creation and Editing

We have implemented our vector shade trees as part of a plugin for Adobe Illustrator. We illustrate a typical interactive session with this plugin in Fig. 12. The user starts with a vector line drawing that he created himself or downloaded from the internet. He then selects individual shapes and assign materials from a library of shade trees. Our tool automatically generates the vector primitives dictated by the shade trees using a set of default parameters. While these defaults allow artists to quickly depict materials on a line drawing, we also provide interactive controls to adjust the parameters and refine the result. Some controls set parameters that are shared by multiple nodes – light direction and horizon elevation – and some controls apply to individual parameters for a single node – palette and shape. The accompanying video provides additional examples of interactive sessions.



Step 1. Original vector line drawing.

Step 2. The user selects shapes and assign materials with default parameters.

Step 3. The user marks the end of tubular shapes to obtain alternating color bands aligned with the direction of least curvature.

Step 4. The user lowers the horizon line of the chrome part and modifies its path to depict a house and hills.

Step 5. The user moves the light from the top left to the top right using the *light direction* track ball widget.

Step 6. The user modifies the base color and the highlight shape to obtain a green car that reflects a window.

Figure 12: *A typical editing session with our tool. A non-expert user can obtain a result with default parameters in less than five minutes (step 1 and 2), while the entire editing session took around fifteen minutes to complete. Original line drawing from openclipart.org©.*

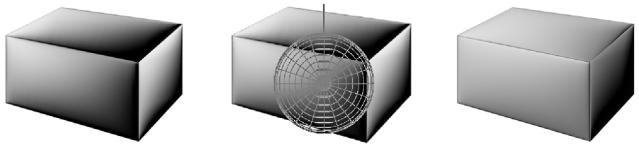
Manipulating the light direction. We set the default position of the light source to the upper left . The artist can select one or more shapes and change the light direction for the group by clicking on a "Manipulate Light ..." button and then dragging the light direction vector using a trackball widget (Fig. 12, step 5).

Changing the palette. Artists can create variations of an object by editing the color palette of nodes in the tree. We give access to all the node parameters of a selected shape via a clickable list. Clicking on a palette opens the corresponding color ramp for editing. Validating the edits triggers the evaluation of the shade tree with the new parameters (Fig. 12, step 6).

Manipulating reflections. Designers commonly use the elevation of the horizon as a cue for view direction (Section 3). We provide a widget to adjust the elevation of the horizon from low to high for the selected shapes (Fig. 12, step 4).

Highlights and reflections in our system are represented using standard vector graphics primitives, such as paths and gradients. Similarly to palettes, artists can modify the shape of a reflection by selecting it from the parameter list. Fig. 12, step 6, illustrates how a user changes the shape of highlights to reflect a window. In Fig. 12, step 4, the user edits the path of the horizon line to create hills and a house in the reflection. The default path delineates the skydome and has rounded corners to mimic the distortion of the horizon near the edge of the shape.

Manipulating normals. By default, our system performs an inflation of the objects based on the silhouette of the vector shape. The user can manipulate the orientation of the surface interactively. In particular, the user can apply a global rotation on the normal field in order to give the impression that the shape points towards another direction (Fig. 13). We also provide the ability to create tube-like geometries, by allowing the user to mark edges that correspond to sharp silhouettes (Fig. 12, step 3). Note that we apply the multi-stop palette guideline in such configurations.



(a) Default inflation oriented toward the viewer

(b) Orientation tool

(c) Result after orienting the three faces of the cube

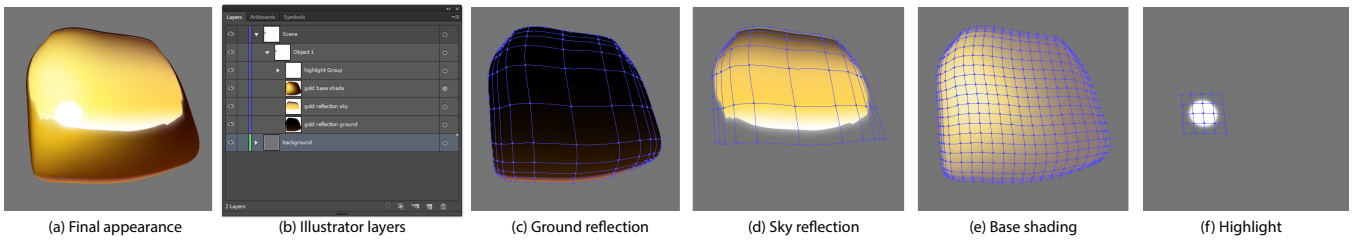
Figure 13: *The default normal inflation produces shapes oriented towards the viewer (a). The user can rotate the normal field (b) to orient the shapes in a different direction (c).*

A major advantage of our system is that it preserves artists workflow by building on existing vector graphics primitives. Our approach gives experienced artists access to the traditional layers, paths and meshes, and perform operations such as changing the color of a mesh vertex in an inconsistent manner, which is sometimes necessary for a given artistic effect.

6 Implementation

We implemented our approach as a plugin in the Adobe Illustrator SDK. Our current system represents shade nodes as classes with palette and path parameters corresponding to Illustrator primitives. We provide a library of shade trees that instantiate the nodes with default primitives. We plan to extend our system to support a graph-based editor to facilitate the type of modifications shown in Figure 11, as is standard practice with 3D shaders.

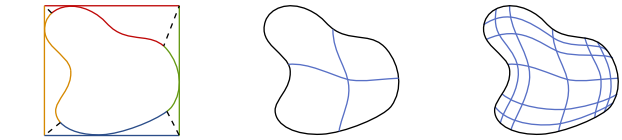
Core to our implementation is the gradient mesh primitive. In addition to smooth color interpolation, the gradient mesh can also be used as a cage to warp other vectorial primitives. We now describe how we generate a mesh from an outline and use it to estimate normals, map textures and position highlights.



(a) Final appearance (b) Illustrator layers (c) Ground reflection (d) Sky reflection (e) Base shading (f) Highlight

Figure 14: *Our shade trees create the vector primitives of each node as Illustrator layers. This golden nugget (a) is composed of four layers (b) that contains the sky and ground reflection (c,d), the base shading (e) and a highlight (f).*

Generating gradient meshes from outlines. Given the outline of a shape we build a gradient mesh following the approach of Lai et al. [2009]. We first divide the outline path into four segments that must map to the four sides of the rectangular domain of the meshm by assigning the vertices of the bounding box of the shape to the closest points along the path (Fig. 15a). We then subdivides this rectangular domain to create the mesh, each subdivision producing vertical and horizontal patch boundaries by interpolating the corresponding sides of the domain (Fig. 15b). In addition, we mimic foreshortening over a smooth shape by recursively subdividing the patches adjacent to the outline (Fig. 15c). This mesh also provides a parameterization which allows us to perform mapping.



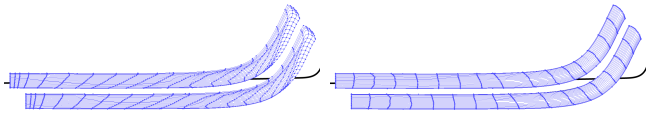
(a) Rectangular parameterization (b) One subdivision (c) Three subdivisions

Figure 15: *Our meshing algorithm maps the shape to a rectangular domain (a) that is then subdivided to create patches (b). We iteratively subdivide patches adjacent to the outline to mimic foreshortening (c).*

A limitation of our algorithm is that it does not support regions with toroidal topology. Regions of very high curvature can also yield meshes that fold over themselves. We only encountered this situation for the front fender of the car in Fig. 1, the handle of the candle and the tailpipe of the motorbike in Fig. 17. In these three cases we manually cut the line drawings into simpler regions to avoid folding.

Estimating normals from silhouettes. An important difficulty encountered by gradient mesh users is to specify colors at mesh vertices to depict plausible shading. To automate this task, we estimate normals over the mesh using the *Lumo* inflation algorithm [Johnston 2002]. The Lumo algorithm assumes that the shape is bounded by smooth silhouettes and generates a normal field corresponding to an inflation of the shape oriented towards the viewer.

Cones and cylinders. Our default meshing and inflation technique assumes isotropic shapes and as such does not capture well the curvature and shading of tubes and cone-like shapes (Fig. 12 step 3, Fig. 16a). As described in Sec. 5, the user can mark segments of a boundary to indicate the two ends of such tubular shapes. We interpret the marked segments as non-smooth silhouettes and exclude them from the shape inflation algorithm, similar in spirit to the *diffusion barriers* of Bezerra et al. [2010]. We also use these indications to orient the mesh subdivision inbetween these two segments, resulting in a more plausible tubular appearance (Fig. 16b).



(a) Default mesh assuming smooth silhouette (b) Tubular mesh after marking the extremities as non-smooth silhouettes

Figure 16: *Our default subdivision technique does not capture well the curvature of tubes and cylinders (a). Users can mark the tube extremities to obtain an improved subdivision aligned with the tubular structure (b). See the user indications in Fig. 12, step 3.*

Highlight placement To place highlights on gradient meshes, we compute the cosine of the angle between the normal and the viewing direction for each vertex. We then position highlights on the vertices for which the cosine forms a local maxima and is above a threshold (.95 in our implementation).

Number of primitives. Our shade trees automate the creation of vector primitives that would take significant effort to draw manually. As an example, the simple tree of gold in Fig. 11 generates three paths (sky and ground reflections and highlight) with their corresponding deformation mesh, one gradient mesh for the base node, and three palettes (base shading and environment reflection node). Fig. 14 shows the Illustrator layers that encode these primitives. More complex illustrations contain hundreds of primitives. The car in Fig. 1c that was composed of 42 paths as input and resulted in 11 palettes, 112 paths, gradient and deformation meshes for reflections and highlights, and 33 gradient meshes for base shading.

A note on file size. Using the Illustrator SDK provides numerous advantages, but results in very large vector files when they contain gradient meshes. From reverse engineering, we estimate that a single gradient mesh vertex requires 300 bytes in this proprietary format. An optimized format (2D coordinates, color and control points), could probably be stored at 1/10th the cost.

7 Results and Informal Evaluation

Our tool allowed us to create many high-quality illustrations with various materials in a matter of minutes (Fig. 17, 20). The materials for the stapler in Fig. 17 and car in Fig. 12 (step 1,2) were created from line-drawing outlines in only 6 and 3 minutes respectively by users with no artistic training.

We asked several trained artists to use our prototype implementation including six professional designers – with at least 5 years experience – and one fourth-year design student. We provided line drawings for two scenes, shown in Fig. 18, one containing a jelly cake, cutlery and a glass and the other a vacuum cleaner. In the instructions, we provided images of stylized materials which they

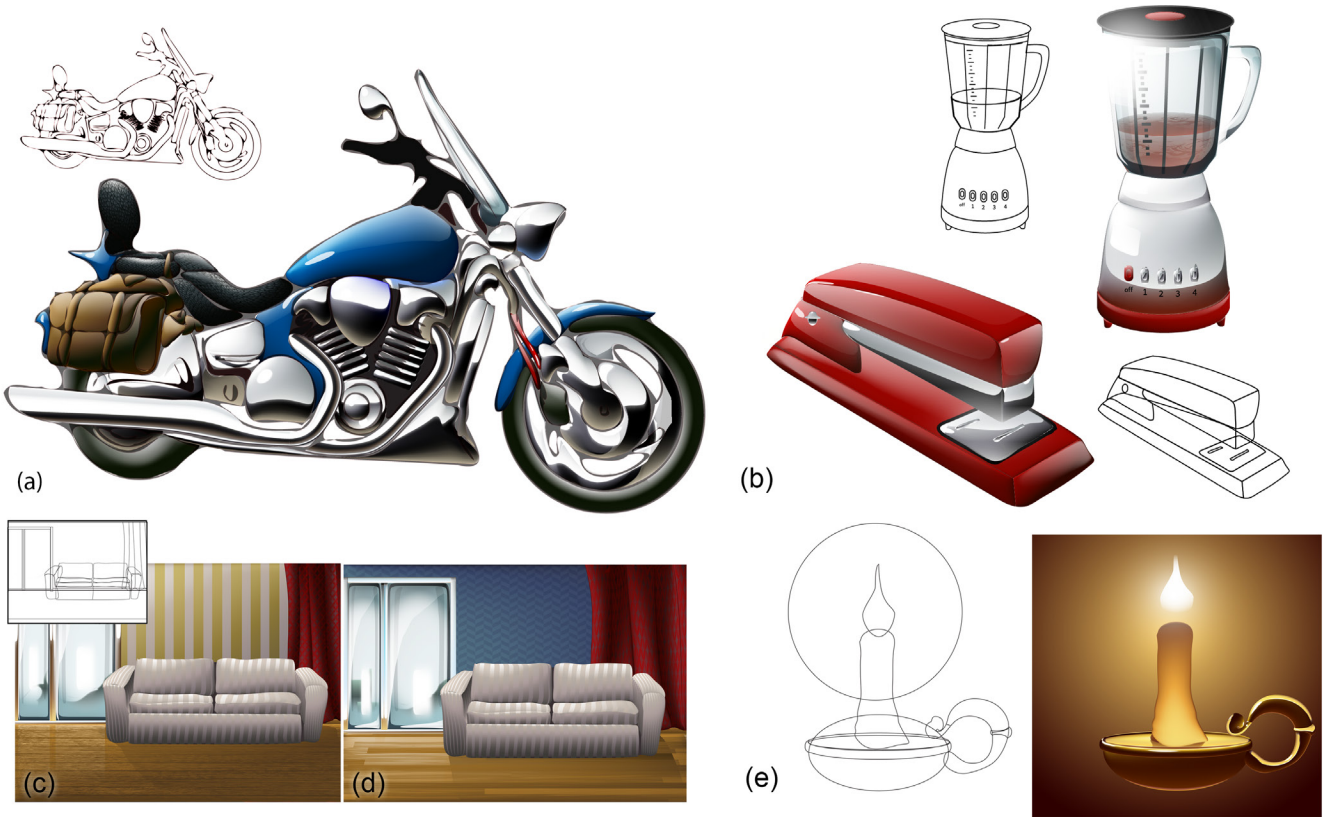


Figure 17: *Different drawings colored with our tool. (a) Depiction obtained from a loosely hand-drawn vector motorbike (inset shows the original lines by MeNext, openclipart.org©). (b) Drawings colored with our tool by a user with no artistic training. Input line art from penclipart.org©(blender by MeNext). (c),(d) Examples of visually complex materials applied to a simple line drawing (inset). An artist needed less than 15 minutes to go from the image in (c) to the result shown in (d). (e) Candle with gold and translucent wax. The flame was added with standard vector graphics tools. (a), (b) colored versions are © the authors, and (c), (d) and (e) are © J. Lopez-Moreno.*

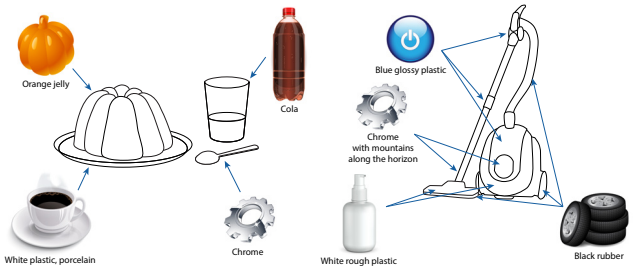


Figure 18: *Scenes to be colored by expert users in our validation study. Left: Jelly Cake and cutlery scene. Complex translucent and transparent materials dominate in this task (glass, jelly, cola). Right: Vacuum cleaner scene, to be depicted both with and without our tool. The set of materials was chosen for its diversity, the original sample illustrations are from shutterstock.com© and the vaccum cleaner line drawing is from Machovka, openclipart.org©.*

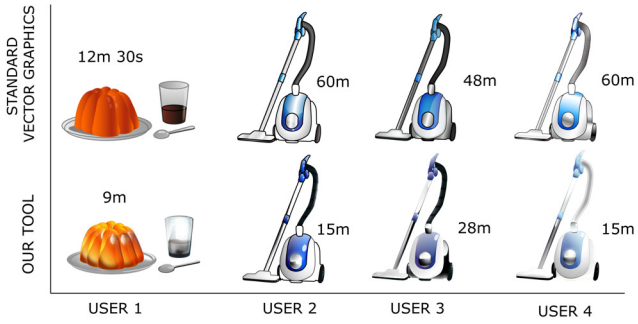


Figure 19: *Images created using both our plugin and standard vector graphics tools and guidelines. The corresponding times for completion are shown with each illustration.*

should try to reproduce. Users were also instructed to perform manipulations such as moving the light or changing the default shape of the highlight.

Three of the users created materials for one or both of the scenes using only our tool, shown in Fig. 20. Their illustrations show that all users were capable of using our solution to create the results as instructed; all took less than 15 minutes to complete the task.

Four users created materials for one of the scenes, both using our plugin and manually using standard Illustrator functionality (Fig. 19). We show the times required to create the illustrations next to each figure. User 1 took approximately the same time for both, yet the visual quality with our plugin is much richer. In the results of users 2, 3 and 4, the visual quality using our plugin is comparable to – or better than – the manual solution; creating the results with our method was 1.7, 3.6 and 4 times faster respectively.

Feedback from these users suggests that achieving material depic-

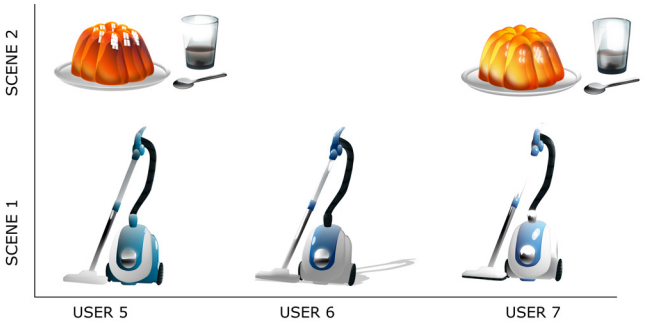


Figure 20: *Three vacuum cleaner and two jelly cake images created using our plugin only, all took less than 15min to create.*

tion quality similar to our results, using only standard vector editing functionality is very difficult and requires great expertise. The standard process first requires the observation of the desired materials, or previous training on depicting them, and then rendering the materials using a combination of vector primitives. Our interviews also revealed that the time devoted to this process could be in the order of hours depending on the complexity of material effects and corresponding guidelines required to create them.

Overall, all users liked our tool, and they all praised its potential and expressivity. One user claimed that a major benefit of our tool compared to the standard vector graphics workflow was the predictability of the result in terms of the appearance achieved.

8 Conclusion

In this paper, we first analyze and classify guidelines for depicting stylized materials, both using traditional artistic media and vector graphics software. Our guideline classification allows us to introduce Vector Shade Trees, by defining a small but powerful set of basic nodes which can be combined to express several different materials. We provide an interface to parameters of our tree, which allows the manipulation of high-level properties of the resulting materials.

Acknowledgments. We would like to thank the artists who participated in our study: J. Fernandez, P. Barrachina, A. Erdocian, G. Benito, I. Gil and R. Bonne and the EUTIZ industrial design school of Zaragoza. We also thank Pascal Barla for his valuable feedback. This work was funded in part by Adobe and Autodesk (research and software donations), by the INRIA-UC Berkeley associate team CRISP and by ANR-12-JS02-003-01 DRAO.

References

ANJYO, K., AND HIRAMITSU, K. 2003. Stylized highlights for cartoon rendering and animation. *IEEE Computer Graphics and Applications 23*, 4 (July), 54–61.

BARLA, P., THOLLOT, J., AND MARKOSIAN, L. 2006. X-toon: an extended toon shader. In *Proc. Symp. on Non-Photorealistic Animation and Rendering (NPAR)*, ACM, New York, NY, USA, 127–132.

BEZERRA, H., EISEMANN, E., DECARLO, D., AND THOLLOT, J. 2010. Diffusion constraints for vector graphics. In *Proc. Symp. on Non-Photorealistic Animation and Rendering (NPAR)*, ACM Press.

BOUSSEAU, A., CHAPOULIE, E., RAMAMOORTHI, R., AND AGRAWALA, M. 2011. Optimizing environment maps for material depiction. *Computer Graphics Forum (Proc. of EGSR) 30*, 4 (07).

BOYÉ, S., BARLA, P., AND GUENNEBAUD, G. 2012. A vectorial solver for free-form vector gradients. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (Nov.), 173:1–173:9.

COOK, R. L. 1984. Shade trees. *SIGGRAPH '84 18*, 3, 223–231.

CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. 1997. Computer-generated watercolor. *SIGGRAPH '97*, 421–430.

DOYLE, M. 2006. *Color Drawing: Design Drawing Skills and Techniques for Architects, Landscape Architects, and Interior Designers*. Wiley.

EISEMANN, E., WINNEMÖLLER, H., HART, J. C., AND SALESIN, D. 2008. Stylized vector art from 3d models with region support. *Computer Graphics Forum (Proc. of EGSR) 27*, 4 (June).

EISSEN, K., AND STEUR, R. 2008. *Sketching (5th Printing): Drawing Techniques for Product Designers*. Art and Design Series. Bis.

EISSEN, K., AND STEUR, R. 2011. *Sketching: The Basics*. Bis Publishers.

FINCH, M., SNYDER, J., AND HOPPE, H. 2011. Freeform vector graphics with controlled thin-plate splines. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 30*, 6 (Dec.), 166:1–166:10.

GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A non-photorealistic lighting model for automatic technical illustration. *SIGGRAPH '98*, 447–452.

GRABLI, S., TURQUIN, E., DURAND, F., AND SILLION, F. X. 2010. Programmable rendering of line drawing from 3d scenes. *ACM Transactions on Graphics 29*, 2, 18:1–18:20.

HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. *SIGGRAPH '98*, 453–460.

JOHNSTON, S. F. 2002. Lumo: illumination for cel animation. In *Proc. Symp. on Non-Photorealistic Animation and Rendering (NPAR)*.

KIM, Y., YU, J., YU, X., AND LEE, S. 2008. Line-art illustration of dynamic and specular surfaces. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 27*, 5 (Dec.).

LAI, Y.-K., HU, S.-M., AND MARTIN, R. R. 2009. Automatic and topology-preserving gradient mesh generation for image vectorization. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

LIAO, Z., HOPPE, H., FORSYTH, D., AND YU, Y. 2012. A subdivision-based representation for vector image editing. *IEEE Trans. on Visualization and Computer Graphics 18*, 11, 1858–1867.

MARTIN, J. 1989. *Rendering Highlights*. Airbrush Artist's Library. North Light Books.

MARTIN, J. 1989. *Rendering Metals*. Airbrush Artist's Library. North Light Books.

MARTIN, J. 1989. *Rendering Transparency*. Airbrush Artist's Library. North Light Books.

MCGARRY, R., AND MADSEN, G. 1992. *Marker Magic: The Rendering Problem Solver for Designers*. John Wiley & Sons.

ORZAN, A., BOUSSEAU, A., WINNEMÖLLER, H., BARLA, P., THOLLOT, J., AND SALESIN, D. 2008. Diffusion curves: A vector representation for smooth-shaded images. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*.

POWELL, D. 1986. *Design rendering techniques: a guide to drawing and presenting design ideas*. North Light.

ROBERTSON, S. 2003. *How to Draw Cars the Hot Wheels Way*. MBI.

SLOAN, P.-P. J., MARTIN, W., GOOCH, A., AND GOOCH, B. 2001. The lit sphere: a model for capturing npr shading from art. In *Graphics Interface*, Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 143–150.

SUN, J., LIANG, L., WEN, F., AND SHUM, H.-Y. 2007. Image vectorization using optimized gradient meshes. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3 (July).

WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. *SIGGRAPH '94*, 91–100.