

Comics Stylization from Photographs

Catherine Sauvaget & Vincent Boyer

L.I.A.S.D. - Université Paris 8
2 rue de la liberté
93526 Saint-Denis Cedex - FRANCE
{cath, boyer}@ai.univ-paris8.fr

Abstract. We propose a 2D method based on a comics art style analysis to generate stylized comics pictures from photographs. We adapt the line drawing and the colorization according to the depth of each object to avoid depthless problems. Our model extracts image structures from the photograph, generates automatically a depth map from them and performs a rendering to give a comics style to the image. Different comics styles have been realized and are presented. Results prove that this model is suitable to create comics pictures from photographs.

1 Introduction

We present a 2D-based method to transform a photograph into a comics image. Our aim is to avoid the depthless color problems created during the image color segmentation process. This occurs when objects with the same color but with different depths are represented in the final picture with the same color. Based on an analysis of comics graphic art, we put forward a model which extracts images structures, generates a depth map from them and performs a rendering to give a comics style to the image.

”Comics” is generally defined as a graphic medium that consists of one or more images conveying a sequential narrative. In this paper, we consider only image generation so the speech balloon and its associated text are not considered. Comics creation consists of these successive steps : elaborating a sketch, drawing with pencil, then inking and colorizing. To achieve a comics may be the work of one individual or of a team (plotter, penciller, inker, colorist...).

In computer graphics this graphic medium is classified as Non-Photorealistic computer graphics. Existing works [1] have proposed a solution to create cartoon shading for a 3D scene. It is based on particular lighting equations and 3D informations are needed: normals, light position, vertices position. So these methods can not be applied to a 2D photograph. Image processing techniques are often used to produce artistic styles on an image [2], [3], [4]. Previous works have tried to provide image stylization for efficient visual communication and produce images with a style close to comics. DeCarlo et al. [5] describe a computational approach to abstract photographs and to clarify the meaningful structure in an image. Figure 1 presents a photograph and the result produced.

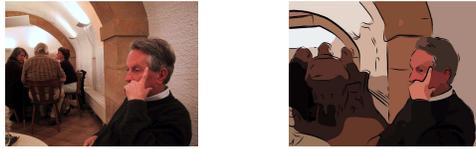


Fig. 1. Stylization and Abstraction of a photograph, DeCarlo et al.



Fig. 2. Abstraction example, Winnemller et al. and our result.

Remark that the distance seems to be overwritten and it is difficult to precise the distance of the group in the produced image.

Wang et al. [6] add a temporal coherence to produce video stylization but the problem remains. Winnemller et al. [7] have proposed an automatic real-time framework that abstracts imagery by modeling color opposition contrast and visual salience in terms of luminance.

Figure 2 (except the picture on the right) presents a result of this method. Note that the green bush on the left seems to be closer in the resultant images than in the original photograph and on the right, the orientation of a brown region is depreciated. Moreover, in this example we remark that the depthless color problem appears even when the image is blurred.

The same problem exists for image editing software (Gimp, Adobe Photoshop) in which cartoon effects are based on edge detection and pseudouniform colorization of the areas.

The main drawback of these works is the appearance of depthless areas: objects which actually have different depths in the photograph appear on the image with the same one. As the colorization of the image is realized with no depth information and with the method used to fill in each region (i.e. one color is used per region), the depth disappears. Moreover these works provide images close to a single comics style and it is impossible to produce different ones. In the following we present our 2D model based on a comics graphic art analysis to transform a photograph into comics-stylized pictures. Using comics creators techniques, depth can be reproduced in our images. Moreover different comics styles have been created and are detailed in the result section.

2 Comics analysis

There are many comics types and it is impossible to give an accurate definition. We prefer to base this analysis on the different styles used in comics [8]. We first present different styles, then we focus on the colors in comics. According to the tools used for strokes and colors, different styles have been identified:

- straight stroke: this is the most frequent cartoon style without relief or black flats (Edmon Baudoin, Hergé, Tito, Bourgeon, Tabary...).
- stroke with black flats: the character is emphasized and the flats permit dramatization of the picture without half tints (Blutch, Brescia...).
- realistic cartoon: straight strokes sometimes are blended with many black flats. The purpose is to produce relief and to give a realistic atmosphere with contrasted lighting (a dramatization) (Adrian Tomine, Gillon...).
- wash: half-tones for black and white applied by touch with different intensities or by scale. It is used to create a cold atmosphere (Pascal Rabate...).
- other styles exist like modeling by stroke (François Schuiten, Bilal, Manara, Moebius...) or an gray-tone stroke (Moebius...).

When used, colors are very important in comics pictures because they create depth. To produce depth the main principle of the comics creators is to decrease the color saturation according to the depth of the objects in the scene. Moreover, a background color can be used to illustrate objects located at a long distance. They are filled in with a low saturation (gray or a special desaturated color) while the foreground objects are filled in with colors with high saturation. The top of figure 3 presents two original pictures. On the left, we measure the color saturation of the building on the left. It decreases (from 25 to 7) according to the depth. On the right, lights and lighting effects on the corridor are different on the ground according to the depth: colors are not saturated. At the bottom, we modify the original picture (hue and saturation) and the depth disappears (are the cypress farther away than the hill on the left?). Remark also on the right, that when the same color is used, the mountain seems closer.

Colors can also be used to give an atmosphere (see figure 4). For example, dramatic atmospheres can be produced with browns and dark blues, mysterious atmospheres are generally composed of dark blues and greens, violent atmospheres use oranges and reds, threatening atmospheres are realized mainly with ochres and browns.

3 Our model: CSM

CSM, Comics Stylization Model, performs different comics stylized images from a photograph.

Our model is composed of three successive parts. The basic workflow of our framework is shown in figure 5. We first extract image structures from the photograph. Then we generate a depth map using them. Finally a stylization process is realized using previous results and a style given by the user. In the following we present each part and detail each component.



Fig. 3. Top : How do the illustrators produce depth:(left) ”Je suis Légion” by Fabien Nury & John Cassaday, (right) ”Le chant des Stryges” by Cobeyran & Guérineau. Bottom: Original, constant saturation, constant tint (from ”Quintett” by Cyril Bonin).



Fig. 4. Examples of atmosphere: (left) ”Escale dans les étoiles” by Jackk Vance, illustrator Alain Brion (2003), (right) ”Black OP 4” by Desberg and Labiano (2008).

3.1 Image structure and analysis

As previously described, comics images are often characterized by a pen and ink drawing of contours and a colorization by one color (one tint) per region. So, we propose to detect contours with an edge detection method and the regions in the photograph using a segmentation method and a mean shift filtering method.

Edge detection Several works have been proposed in edge detection. We use the method proposed by Meer et al. [9]. This approach consists of reformulating the framework in a linear vector space where the role of the different subspaces is emphasized. Artifacts are then explained quantitatively and an independent measure of confidence in the presence of the employed edge model is provided. It is well adapted in detecting weak edges.

We construct a list of contours \mathcal{L}_C composed of n contours in which each contour C_i ($i \in [1; n]$) is characterized by a starting point and an end point. Contours have no loops (i.e when a loop or when two possible paths are detected, two contours are created). It is used for plane detection and not for stylization. A length threshold (named LT in figure 5) can be used to remove non-significant contours then \mathcal{L}_C is updated.

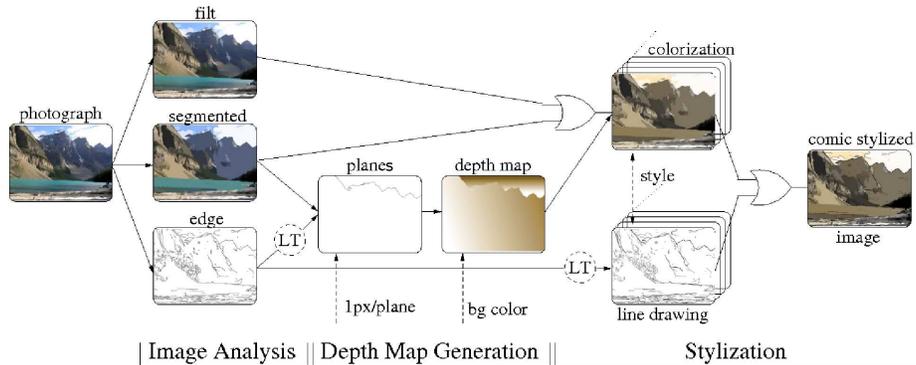


Fig. 5. Framework overview: the three successive parts of CSM where dash lines represent parameters given by the user.

Segmentation Segmentation can be considered a general technique to extract significant image features and different approaches exist. We use the mean-shift algorithm [10], a non-parametric procedure for estimating density gradients. The main idea is that the vector of differences between the local mean and the center of a window, named the mean-shift vector, is proportional to the gradient of the probability density at a point. Regions are created in the joint domain by grouping together neighboring pixels which are closer to a convergence point. The regions containing less than M pixels are eliminated and the process is repeated as long as the number of regions given by the user is not reached. Based on this method we segment our photograph with an interactive tool.

Filt The mean-shift algorithm [10] can also be used to produce a discontinuity preserving smoothing technique. It is a simplification of the segmentation method where the kernel window moves in the direction of maximum increase in the joint density (i.e. there are no created or eliminated regions). It adaptatively reduces the amount of smoothing near abrupt changes in the local structure (i.e. edges). The produced image is more detailed than the segmentation one. The number of regions is not given by the user. Features with a high color contrast are preserved independently of the number of pixels in the region, and regions containing pixels with a low color contrast are almost completely smoothed.

3.2 Depth Map Generation

Depth-map generation is a crucial step of our workflow. It is used during the stylization process to avoid the depthless color problem.

A lot of work has been realized in generating a depth map from an image. Methods based on vanishing-point detection using edge detection are not well adapted to a landscape photograph but more to a city photograph with geometry. [11] estimates the depth information by a color-based segmentation, a

specific region detection based on semantic rules and an image classification. The classifier is necessary to obtain a reliable depth map and to tag each part of the picture. An approximated depth map is then generated. Unfortunately, the image classifier is not complete and depth map generation must often be realized interactively. [12] improves this method but some problems remain: when the inclination of the photograph is not horizontal and when an object like a grid is present, superpixels are created (these details will not be considered here).

We propose an approach based on a border plane detection and a color shading per plane. The depth map is generated by the segmentation and the edge-detection images.

Border-plane detection To detect border planes, we use the edge-detection image and the segmented one. Our border-plane detection is partially interactive: the user selects a pixel P in \mathcal{L}_C for each border plane on the edge detection image. Then our algorithm computes the most probable path for each border plane as following:

1. Find the contour C_i in \mathcal{L}_C containing P .
2. Starting from P follow C_i in the two directions and search KP_i (KP_i^1 or KP_i^2 depending on the direction). KP_i can be the first crossing point, the extremity or a point of the border plane list (in that case, KP_i is not evaluated).
3. Find the next point P' :
 - if KP_i is a crossing, choose the next point P' of the crossing which does not belong to (KP_i^1, KP_i^2) and which has the nearest characteristics to KP_i in the segmented image;
 - if KP_i is an extremity, search its nearest point P'' of C_k ($k \neq i$) in the segmented image which has characteristics close to KP_i (a threshold value based on a color difference is used to determinate if the characteristic of two points are close). If the distance between KP_i and P'' is greater than the distance between KP_i and the border of the image or if no P'' exists then goto the last step (condition 1). Else search the key points KP_k (i.e KP_k^1, KP_k^2) for P'' in C_k . We find P' , a point of C_k and a point of C_i which minimize their euclidean distance and which belong to KP_k and KP_i respectively. A new contour is created between these two points.
4. We construct the border-plane list by adding the path between P and P' and we iterate this process (P becomes P') until condition 1 is verified.
5. Last step: the path is completed by adding a link between KP_i and the border of the image.

Applying color shadings We have to fill in the different planes with a color shading. The user chooses a plane and its color shading direction adding two points and two color values. The values can be grey levels or a color to make an atmosphere. A linear interpolation is done between these two points.

In our model, each value of the depth map is in $[0;1]$ (one component is used for a grey-level depth map, otherwise three components are used) where 1 refers to the camera viewer and 0 to infinity.

3.3 Stylization

Stylization is the last step of our workflow. It uses the depth map previously generated and results produced during the image analysis and the structure steps. **CSM** realizes a colorization and a line-drawing image that can be mixed to produce the comic-stylized image.

Colorization To realize the colorization, the user chooses the "filt image" or the segmented one. Then the depth map created below is applied automatically. To generate the colorization image we represent the colors in the HSV model.

Let $P(P_h, P_s, P_v)$ be a pixel of the colorization image, $I(I_h, I_s, I_v)$ the pixel of the segmented or "filt image", depending on the user choice and $DM(DM_h, DM_s, DM_v)$ the pixel of the Depth Map with the same coordinate of P . We compute the saturation of each pixel P of the colorization image as $P_s = DM_v \times I_s$ or $P_s = DM_v \times MaxSat(DepthMap)$ depending on the style chosen by the user. In the second case, the maximal saturation of the depth map is used and affects the global saturation of the colorization image. In fact if a dark depth map is created, the colorization image will have low saturation values. For the other components: $P_h = I_h$ if a grey-level depth map is used, otherwise $P_h = DM_h$ and $P_v = I_v$.

We are able to produce a binarization with threshold to obtain black and white colorization, grey levels or short straight lines (see figure 6).

Note that **CSM** is able to mix multiple colorization images (an atmosphere, a binarization...) using classical functions (difference, addition, threshold value) to select pixels from these and produce the colorization image used to generate the comics-stylized image.

Line Drawing To realize the line drawing, the edge-detection image corresponding to a sketch is used. A length threshold can be applied to select significant contours. The line drawing image can be a simple copy of the edge-detection image or one of the three styles available:

- thick lines: to draw the contours as distorted and thick lines, we divide the contours into 5 equal parts. We increment the thickness of: the extremal parts by one pixel; the medium parts by two pixels; the center part by three pixels. Increasing the thickness of the contours produces aliasing. To avoid this artifact, we apply a 3x3 median kernel filter. Finally to obtain contours, we apply a threshold (to convert the gray scale previously obtained to black or white).
- short breaking lines: the user specifies a length and an orientation. We divide each contour and realize a rotation at the center of each breaking line according to these parameters.

- Spline lines: for each contour, we create a spline:
 1. We determine a list of control points: we decompose the contour into successive straight lines. The control points are placed at the junction of two consecutive straight lines.
 2. We update this list by adding and/or deleting some control points:
 - due to the discretization and the edge-detection approximation, it is possible that for a given straight line in the photograph, we detect more than one straight line in the previous step. To solve this problem, we compute the vectors between two consecutive control points and if two consecutive vectors are similar (i.e the dot product is close to 1), the control point belonging the two vectors is removed.
 - when an acute or a reflex angle is detected between two straight lines (i.e the dot product of two consecutive vectors is close to 0), we want to create a curve close to the straight lines (we do not want to approximate a right or a reflex angle by a “circle arc”). So we propose creating a cusp and control point duplicating (i.e a control point with the same coordinates is added to the list).
 3. The new contour is created with a Catmull-Rom spline because it guaranties that each control point is hit exactly and the tangents of the generated curve are continuous over multiple segments.

Comics-stylized image Comics-stylized images are produced by mixing the colorization and the line-drawing images. We are able to produce the styles presented above and others (see section 2):

- straight strokes are realized with a colorization generally without atmosphere and contours obtained by the edge detection or thick contours.
- strokes with black flats look like the previous ones but we mix the black flats obtained by binarization during the colorization process.
- realistic cartoons are obtained as strokes with black flats but atmosphere is always included.
- wash is realized using the “filt image”, an atmosphere and one of the line-drawing styles.

4 Results

This section presents images produced with **CSM**. The CPU time needed to produce the comics-stylized image depends mainly on the image size (less than one second for a 800×600 image on a Pentium 3.6GHz with 1Go of memory). All depth maps have been generated automatically (i.e the user never modifies the border planes).

The top of figure 6 illustrates the depthless problem. Remark that for the leaves of the tree on the right, it is very difficult to see what is in the foreground and in the background when the depth map is not used (segmented). On the contrary, it is very easy to see it when using a depth map. At the top

right an atmosphere is added. The last two use Catmull-Rom contours. At the bottom, from left to right, we present different line-drawing styles: thick line contours, with short straight-line colorization, short breaking-line contours and short straight-line colorization with black flats.

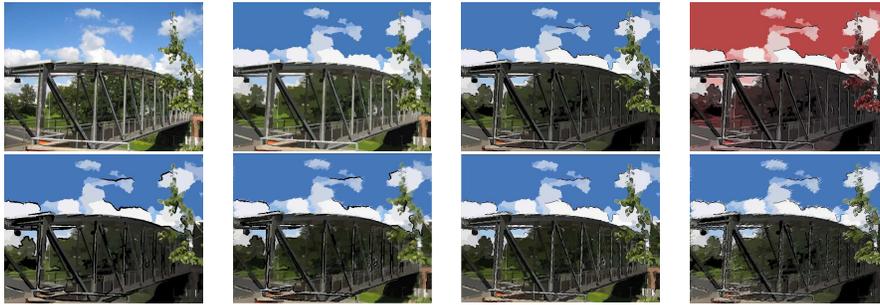


Fig. 6. Results on a bridge. Top: original; segmented; with depth map; with atmosphere in the depth map. Bottom: line drawing and post-treatment colorization examples.

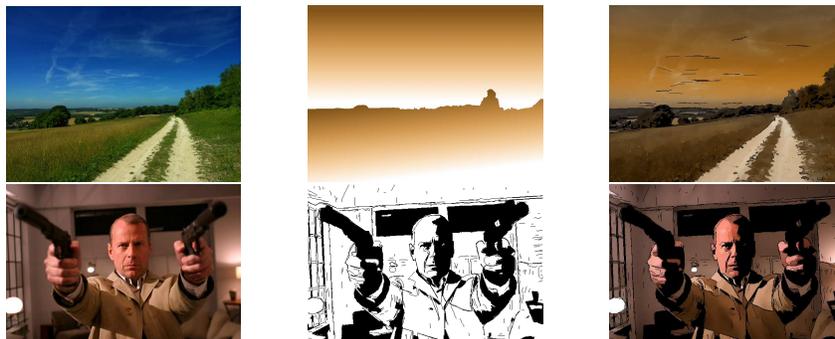


Fig. 7. Results on a landscape (Top): original, depth map, fild with atmosphere. Results on a face (movie "Slevin") (Bottom): original; B & W; strokes with black flats.

As one can see in figure 7 (Top), the depth map produced is quite good. A little problem occurs with the tree on the right. The resultant image has been produced mixing the segmented image with the colored depth map. Note that the saturation increases far into the picture. The contours have been created with Catmull-Rom splines. The bottom shows different comics styles. The black and white picture illustrates the binarization with hand-like drawing contours (spline lines). The last image is a mix between a colorization and the binarization.

Last example, we produce stylized images with thick contours(right) from the photograph in figure 2. Even if the photograph is blurred, we preserve the depth in the picture (see the bush on the left) and the orientation (on the right).

5 Conclusion

We have proposed a model to transform a photograph into comics-stylized images. **CSM** provides images without the depthless color problem. Based on an image analysis and structure, it constructs a depth map used to perform the stylization process. The results prove that the depthless problem disappears when a depth map is integrated. Different styles are proposed and can be chosen interactively by the user.

Future work will be done to increase the number of comics styles available. For example, a blur could be applied to the image according to a particular depth, line drawing could be weighted according to the depth, details can be preserved for a given distance. . . Moreover, we will improve the depth-map generation: a given pixel for each border plane will be automatically detected; an automatic verification will be integrated; and styles will be added to the filling in part of the depth map generation. We will also try to apply this model to a movie in which temporal coherence remains a problem.

References

1. Lake, A., Marshall, C., Harris, M., Blackstein, M.: Stylized rendering techniques for scalable real-time 3d animation. In: NPAR. (2000) 13–20
2. Treavett, S.M.F., Chen, M.: Statistical techniques for the automatic generation of non-photorealistic images. In: Proc. 15th Eurographics UK Conference. (1997)
3. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.: Image analogies. In: SIGGRAPH. (2001) 327–340
4. Hertzmann, A.: Paint by relaxation. In: Computer Graphics International. (2001) 47–54
5. DeCarlo, D., Santella, A.: Stylization and abstraction of photographs. In: SIGGRAPH. (2002) 769–776
6. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. In: SIGGRAPH. (2004) 574–583
7. Winnemöller, H., Olsen, S.C., Gooch, B.: Real-time video abstraction. *ACM Trans. Graph.* **25** (2006) 1221–1226
8. McCloud, S.: *Making Comics: Storytelling Secrets of Comics, Manga and Graphic Novels*. Harper Paperbacks (2006)
9. Meer, P., Georgescu, B.: Edge detection with embedded confidence. *IEEE Trans. Pattern Anal. Mach. Intell.* **23** (2001) 1351–1365
10. Comaniciu, D., Meer, P.: Robust analysis of feature spaces: color image segmentation. In: CVPR. (1997) 750–757
11. Battiato, S., Curti, S., La Cascia, M., Tortora, M., Scordato, E.: Depth map generation by image classification. In: Proceedings of the SPIE, Volume 5302, pp. 95–104 (2004). (2004) 95–104
12. Hoiem, D., Efros, A.A., Hebert, M.: Automatic photo pop-up. *ACM Trans. Graph.* **24** (2005) 577–584