

Sketching Cartoons by Example

D. Sýkora^{1,†}, J. Buriánek², and J. Žára¹

¹ Czech Technical University in Prague

² Digital Media Production

Abstract

We introduce a novel example-based framework for reusing traditional cartoon drawings and animations. In contrast to previous approaches our aim is to design new characters and poses by combining fragments of the original artwork. Using standard image manipulation tool this task is tedious and time consuming. To reduce the amount of manual intervention we combine unsupervised image segmentation, fragment extraction and high-quality vectorization. The user can simply select an interesting part in the original image and then adjust it in a new composition using a few control scribbles. Thanks to ease of manipulation proposed sketch-based interface is suitable both for experienced artists and unskilled users (e.g. children) who wish to create new stories in the style of masters. Practical results confirm that using our framework high-quality cartoon drawings can be produced within much shorter time frames as compared with standard approaches.

Categories and Subject Descriptors (according to ACM CCS): I.2.6 [Artificial Intelligence]: LearningAnalogies I.3.4 [Computer Graphics]: Graphics UtilitiesGraphics editors I.4.6 [Image Processing and Computer Vision]: SegmentationEdge and feature detection J.5 [Computer Applications]: Arts and HumanitiesFine arts

1. Introduction

A long history of cartoon animation provides a respectable amount of artistically advanced works [Len99]. These traditional cartoon drawings and animations contain visual style which is unique and typical for their authors. Generations of children and adults love these classical works and wish to watch new stories in the same style. Unfortunately when a classical cartoonist is deceased it is usually very tedious and time consuming to mimic his or her style in order to create new poses and characters undistinguishable from the original. It would be wonderful to be able to teach a machine how to draw in a traditional style given by an existing example, and then have the machine utilize this "learned" style to create new poses and characters.

Example-based or data-driven approaches become popular quite recently in motion re-targeting [BLCD02] and 3D modelling [FKS*04]. A lot of work has been also done on transferring particular artistic styles to images. Hertzmann

et al. [HJO*01] transfer style pixel-by-pixel by matching local image statistics between example and target image. Freeman *et al.* [FTP99] and later Hertzmann *et al.* [HOCS02] use similar approach for translating line drawings into different styles. Drori *et al.* [DCOY03] decompose the input image into a couple of small fragments, and then stitch them together to form a new image that follows several constraints. Jodoin *et al.* [JEGPO02] presented hatching by example which combines ideas of pixel-based texture synthesis in order to stylize user-defined curves. However such techniques transfer only local information which is insufficient while global features are usually much more important for viewer to recognize a typical drawing style.

Several authors attempt to overcome this limitation by asking the artist to prepare a set of stand-alone fragments that can be later reused in a variety of different ways. Buck *et al.* [BFJ*00] and later Chen *et al.* [CLR*04] used this technique for example-based synthesis of faces in different hand-drawn artistic styles. This approach is also common in computer assisted cartoon animation. A skilled artist first prepare a set of simple fragments from which more complex scenarios are composed [FBC*95].

† sykorad@fel.cvut.cz

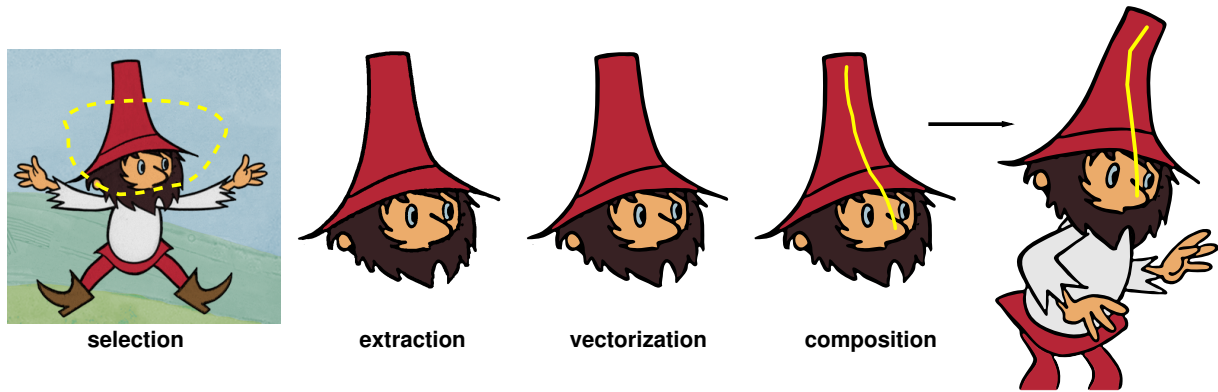


Figure 1: Framework overview – the user first selects a desired fragment in the original image (left), then the system automatically extracts it and perform vectorization (middle), finally the fragment is arranged in a new position using two composition scribbles (right).

In our framework we assume that original fragments are no longer available. We have only bitmap image of the final composition. For this case *Juan and Bodenheimer* [dJB04] suggest a method through which they extract dynamic fragments from the entire sequence and reuse them in different frame orderings. This approach has only limited application.

To create truly new poses and characters, it is necessary to extract and seamlessly compose fragments of the original artwork. This is a challenging task that usually cannot be resolved without additional manual intervention. *Barrett and Cheney* [BC02] use an object-based image editing to reduce the amount of manual intervention. However, their method suffers from visible artifacts while all editing operations are performed in the image domain without additional post-processing. *Pérez et al.* [PGB03] and later *Agarwala et al.* [ADA*04] introduced gradient domain image editing for seamless fragment composition. However these approaches produce compelling results only when fragments do not contain distinct boundaries.

The aim of our work is to alleviate shortcomings of previous approaches. We improve an existing image segmentation technique in order to lower the burden connected with fragment selection and extraction. For ease of composition we purpose an intuitive sketch-based interface suitable both for skilled artists and unexperienced users. The final visual quality is preserved thanks to our novel raster to vector conversion scheme which outperforms standard vectorization tools.

Saund et al. [SFLM03] proposed system called *ScanScribe* that is similar to ours but has slightly different motivation. It can simplify the selection, grouping and manipulation of free-form sketches and handwritten notes. In contrast to our approach *ScanScribe* works only in the image domain and assumes homogeneous background. Also the manipulation is limited only to several standard transformations.

2. Framework

In this section we describe our novel framework. First we present a short overview and later, in successive sections, we describe each step in more details including implementation and optimization issues.

2.1. Overview

The input to our system is a set of classical cartoon images originally created using a cel or paper based technique. In this case each frame is created as a planar composition of two layers: background and foreground. Typically, the foreground layer consists of several homogeneous regions while the background layer usually contains more complicated textural information. Our aim is to reuse shapes and regions stored in the foreground layer.

The first step in our framework (see Figure 1) is an unsupervised image segmentation that allows us to partition the input image into a set of regions. Each region is then classified as to whether it belongs to the background or to the foreground layer. An interactive phase follows. In this step the user simply selects a subset of regions called fragment. Afterwards, the system extracts the fragment together with corresponding outlines and perform vectorization. Finally the user arranges it in a new position by sketching two composition scribbles that make possible to define a combination of rigid transformation and free-form deformation.

2.2. Segmentation

In order to separate the original image into a set of regions we adopt an unsupervised image segmentation scheme first presented in [SBv03] and later refined in [SBv05]. This technique has been designed specifically for cartoon images. It uses a robust outline detector which utilizes negative response of Laplacian-of-Gaussian ($L \circ G$) filter [SB89] and

adaptive flood-filling scheme to locate and extract outlines. Standard region labelling algorithm [RK82] is then used to produce the final segmentation. In this section we present several improvements that allow us to use the original segmentation technique as a preprocess for high-quality vectorization phase.

The first important factor that significantly affect the final visual quality is sub-pixel precision. We achieve it by computing $L \circ G$ -negative response in fourfold higher resolution using method of [HM86] (see Figure 2). In this case unsupervised σ -fitting [SBv05] also helps us to eliminate small spurious regions inside outlines.



Figure 2: Super-resolved cartoon segmentation in progress: a) original image, b) $L \circ G$ -negative response, c) extracted outlines, d) final segmentation.

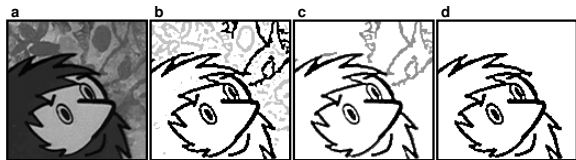


Figure 3: Outline authentication test in progress: a) original image, b) the case when a simple outline detector failed, c) $L \circ G$ -negative mask with inverted pixel priorities, d) extracted outlines after flood-filling with priority queue.

The another issue is connected with outline authentication test that is necessary when a couple of foreground outlines coalesce with $L \circ G$ -negative areas in the background layer (see Figure 3b). In [SBv03] we use only simple median thresholding scheme that unfortunately tends to peel regular outlines. To alleviate this artifact we introduce a more precise test. First the original image is sharpened to enhance the difference between dark outlines and brighter $L \circ G$ -negative areas in the background layer. We do this by adding full $L \circ G$ response to the original image. Afterwards we repeat last flood-filling step from outline detection phase but instead of stack we use priority queue to store and expand

seed pixels. Priority for each pixel is given by the inverted intensity in the sharpened image (see Figure 3c). Using this approach we fetch dark pixels first and then gradually expand towards brighter values. The expansion is terminated in pixels that have priority lower than median of priorities at zero-crossings of previously detected outlines. All pixels ever stored in priority queue are marked as foreground outlines (see Figure 3d).

2.3. Classification

When the input frame is partitioned into a set of regions we use area size thresholding as in [SBv05] to roughly classify whether a given region belongs to the foreground or to the background layer. Smaller fragments which belong to the background layer (holes in the foreground layer) can be further classified using two different approaches.

In the first approach we assume that the original animation sequence is available. In this case we can track camera motion through the time and stitch fragments of background as in [SBv05]. Afterwards it is possible to compute normalized sum of absolute differences between the pixels of region and corresponding area in the reconstructed background. When this sum falls under a specified limit then such a region can be classified as background.

When only a static image is available or when several parts of the background are not visible during the animation, the only feasible feature which can help us to distinguish the background from the foreground is region homogeneity. We can estimate it using technique described in [CS00]. Two or more significant peaks in homogeneity histogram indicate that a given region is not homogenous and thus may belong to the background layer. However when the occluded part of the background is also homogeneous this simple approach fails and further manual classification during the fragment selection is needed.

Finally when the classification is done it is necessary to assign visually dominant color to each region in the foreground layer. For this task it is possible to simply use mean or median color. However colors of smaller regions can be significantly biased by the noise and outline anti-aliasing and thus the same colors in different regions may appear inconsistent. To address this issue we use global mean-shift color clustering [CM02] to obtain visually consistent color assignment for all regions.

2.4. Fragment selection

After the preprocessing the user is allowed to select an interesting part in the pre-segmented image. Because we exactly know which pixel belongs to which region and vice versa, it is possible to use various selection tools, e.g. to simply click on each desired region or to draw selection scribbles over them. In our experiments we have found that the most useful

selection tool is free-form curve that perform selection on intersected regions and on regions that are inside the selection area (when the curve is closed). The similar approach has been used in *ScanScribe* [SFLM03], however in our framework such a curve can be really sloppy (see Figure 1) while we automatically exclude regions in the background layer.

2.5. Fragment extraction

When the user selects desired regions we are still not able to extract entire fragment while remaining regions share the same outlines. What we want is to decide which part of the original outline semantically belongs to the selected region. In general this problem can be ambiguous (see e.g. [PGR99]) and only additional constraints such as convexity may help us to solve it.

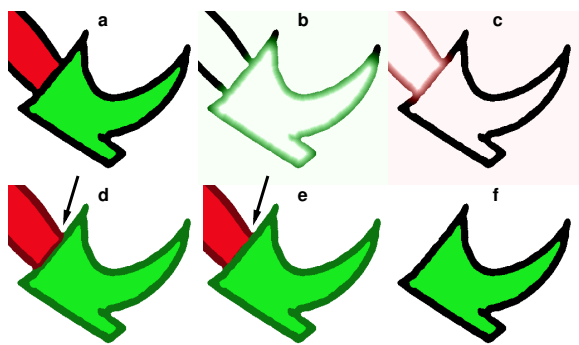


Figure 4: Fragment extraction: a) the green shoe is extracted from the red leg, b) distance field for shoe and c) for leg, d) the partition where the distance to the shoe and to the leg is same (medial axis), e) refined partition, f) the final extraction.

In our framework we assume that outlines are not jaggy and have locally constant thickness. This assumption lead us to estimate medial axis between selected and remaining regions. Topologically correct solution can be obtained using combination of Euclidian and geodesic distance [BK01]. However, in most cases simple Euclidian distance provides a good approximation.

We compute two distance fields [Bor86]: one for selected and one for remaining regions (see Figure 4b,c). Then both distances are compared in each outline pixel to decide whether it belongs to the desired fragment or to the remaining part of the image. Pixels with the same distance in both fields form medial axis (see Figure 4d). Using distances assigned to those pixels we compute median distance that is treated as a one half of the overall outline thickness. Then we refine the partition by adding pixels that have distances from the selected regions smaller than the overall outline thickness (see Figure 4e). Finally we can easily extract an entire fragment together with corresponding outlines (see Figure 4f).

2.6. Vectorization

When the desired fragment is extracted we apply raster to vector conversion using standard contour tracing algorithm with piecewise *Bézier* cubic fitting [WH04]. The main advantage of our approach is that we fit *Bézier* curves in pre-processed image at fourfold higher resolution (see Figure 2d and 4f) where regions have constant color and outlines are represented by super-resolved $L \circ G$ -negative areas. In Figure 5 it is possible to compare how this kind of preprocessing outperforms standard vectorization tools such as *VectorEye* [VLDP03], *AutoTrace* [WH04] and *KVec* [Kuh03].

2.7. Composition

During the composition phase the aim is to specify new position, orientation and deformation of selected fragments. Using standard vector manipulation tools this task is nonintuitive and tedious while it is necessary to combine a number of basic transformations like translation, rotation, scale, bend, shear, etc.

This problem has been extensively studied in the literature (for nice overview see e.g. [MJB02]). In our framework we use approach similar to popular technique called *warp* [CJTF98] that has been successful used in *Teddy* [IMT99]. It exploits two smooth curves to define free-form deformation. We call them *composition scribbles*. The first scribble is drawn in the source image and the second in the target composition (see Figure 1).

The idea of *warp* is to assume that both composition scribbles are drawn in a constant speed and have the same parametric length. In this case it is easy to estimate point correspondences between scribbles using simple arc length parameterization. The source fragment is deformed according to changes in positions and tangent orientations of corresponding points (see Figure 6). However the problem is that such a simple approach produces non-uniform scale distortion that is sometimes useful but in our application it does not fit the user's intuition while source fragments have usually different scale and it is necessary to adjust them to meet the scale of target composition (see Figure 7).

To address this issue we introduce composition tool called *uniform warp* that utilizes actual length ratios of source and target scribbles to obtain proper scale normalization. In this method the source fragment is first scaled together with its composition scribble to meet the actual length of the target scribble and then the standard *warp* is applied to perform free-form deformation.

In our experiments we found that such a simple adjustment can significantly reduce the amount of manual intervention needed during the composition phase. However we do not limit the user to use only this composition tool. Our system also offers standard *warp* and other commonly available tools such as translation, rotation, scale, flipping and mirroring.

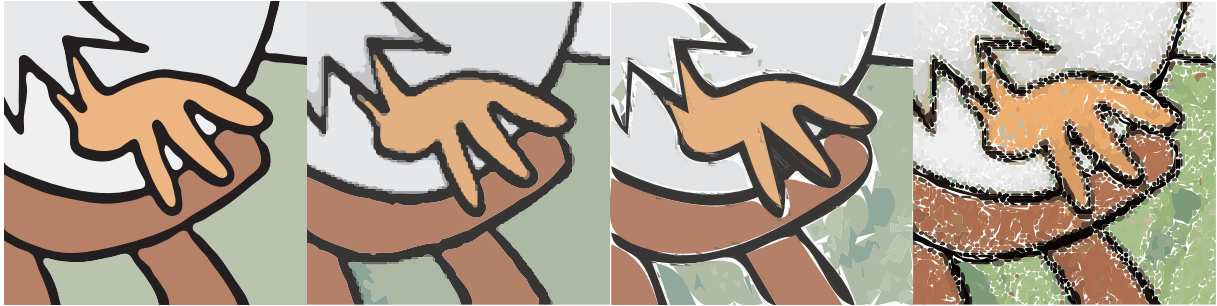


Figure 5: Compare the accuracy of vectorization (from left to right): our approach, Vector Eye, AutoTrace and KVec.

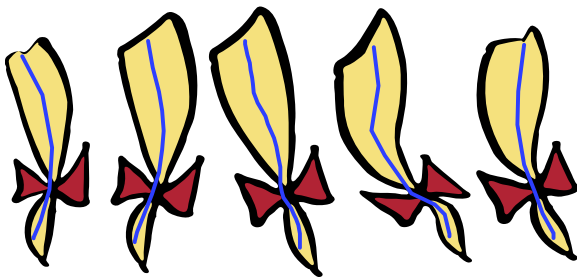


Figure 6: Free-form fragment manipulation using standard warp (the original fragment is in the middle).

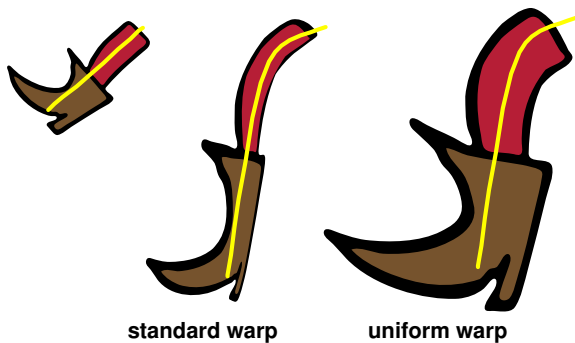


Figure 7: During the composition the user intuition is to scale fragment uniformly. Instead of standard warp we use uniform warp to produce expected results.

2.7.1. Implementation issues

The original *warp* operates with smooth curves. In our framework we use mouse or tablet to acquire composition scribbles and thus it is necessary to implement the concept of *warp* in a discrete domain. To accomplish this we first approximate a set of drawn pixels using piecewise linear curve (line strip). Then we compute the length of each scribble by summing lengths of individual line segments and an uniform arc length parameterization allows us to estimate new

positions of source control points on the target strip (see Figure 8). When the source strip does not fit the target one well we perform subdivision in a problematic area and recompute new locations of corresponding points. Finally we apply feature-based image metamorphosis [BN92] that produces similar results as piecewise linear planar warp [MJBFO2].

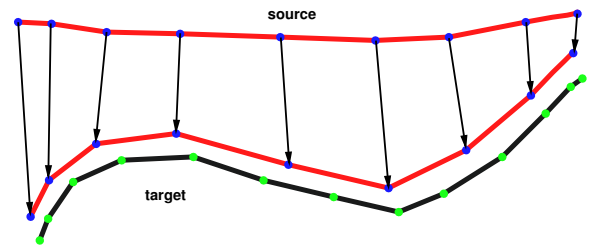


Figure 8: Linear mapping of composition scribbles: first the source strip is subdivided to match curvature of the target strip and then linear mapping is used to map source control points on the target strip.

3. Results

In this section we present several results obtained on real cartoon images scanned in the resolution of 720x576 from the original celluloid of the traditional Czech cartoon *O loupežníku Rumcajsovi* by Radek Pilař. Five new characters have been composed in his style (see Figure 10). Using our framework each character takes only tens of seconds to complete. Experienced user needs several minutes to obtain comparable results using standard image manipulation tool.

The common workflow was to first draw composition scribbles in such a way that they simultaneously intersect desired regions. Whenever this was not possible additional selection scribbles were added to cover remaining regions. The selection and composition process is also depicted in Figure 10. During the composition the user was allowed to define fragment layering while it was much better to first sketch new position of character's body and then add head, legs and hands.



Figure 9: Several examples of drawing styles not suitable for our framework. Problems may arise when the input image contains lots of unstructured outlines, small and inhomogeneous regions, soft shadows and color gradients, dark coalescent outlines in the background, etc.

Drawing styles similar to Radek Pilař's represent ideal case for our framework. For other styles that do not strictly satisfy the basic assumption on region homogeneity and outline continuity it is necessary to use additional segmentation scribbles or to apply outline joining algorithm [SC94] and independent color clustering for each inhomogeneous region. The problem may also arise when outlines are drawn close together or when the image contains lots of small regions. Resulting vectorization appears visually different as compared to the original drawing. Dark coalescent outlines in the background layer are also problematic. In this case couple of additional vector editing operations are needed to remove them. In Figure 9 we present several examples of challenging styles which are not suitable for our framework.

4. Conclusion and Future work

In this paper we have made the following contributions. A novel example-based framework for computer assisted cartooning has been presented. We show how to speed up the fragment selection phase and how to preserve compelling visual quality of the final composition. Our intuitive sketch-based composition tool successfully extends toolbox of existing vector manipulation tools. Practical experiments performed on real cartoon images confirm that for particular drawing styles the proposed framework allows to synthesize new cartoon drawings within much shorter time frames as compared with standard image manipulation tools.

As a future work we plan to extend our framework for different drawing styles and to use more advanced deformation techniques like [IMH05]. Another natural extension is cartoon animation. In this case the idea is to let the user to sketch several key-frames and then use shape interpolation technique [ACOL00] to obtain in-between animation.

5. Acknowledgements

All images in this paper are published with permission of © Vít Komrží, Universal Production Partners and Lubomír

Celar, *Digital Media Production*. This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under research program No. MSM-6840770014 (Research in the area of information technologies and communications) and under the student research program: FRVŠ-2005-1170.

References

- [ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *ACM SIGGRAPH Conference Proceedings* (2000), pp. 157–164. 6
- [ADA*04] AGARWALA A., DONTCHEVA M., AGRAWALA M., DRUCKER S., COLBURN A., CURLESS B., SALESIN D., COHEN M.: Interactive digital photomontage. In *ACM SIGGRAPH Conference Proceedings* (2004), pp. 294–302. 2
- [BC02] BARRETT W. A., CHENEY A. S.: Object-based image editing. In *ACM SIGGRAPH Conference Proceedings* (2002), pp. 777–784. 2
- [BFJ*00] BUCK I., FINKELSTEIN A., JACOBS C., KLEIN A., SALESIN D., SEIMS J., SZELISKI R., TOYAMA K.: Performance-driven hand-drawn animation. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering* (2000), pp. 101–108. 1
- [BK01] BOYKOV Y., KOLMOGOROV V.: Computing geodesics and minimal surfaces via graph cuts. In *Proceedings of IEEE International Conference on Computer Vision* (2001), vol. 1, pp. 26–33. 4
- [BLCD02] BREGLER C., LOEB L., CHUANG E., DESHPANDE H.: Turning to the masters: Motion capturing cartoons. In *ACM SIGGRAPH Conference Proceedings* (2002), pp. 399–407. 1
- [BN92] BEIER T., NEELY S.: Feature-based image metamorphosis. In *ACM SIGGRAPH Conference Proceedings* (1992), pp. 35–42. 5

- [Bor86] BORGEFORS G.: Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing* 34, 3 (1986), 344–371. 4
- [CJTF98] CORRÊA W. T., JENSEN R. J., THAYER C. E., FINKELSTEIN A.: Texture mapping for cel animation. In *ACM SIGGRAPH Conference Proceedings* (1998), pp. 435–446. 4
- [CLR*04] CHEN H., LIU Z., ROSE C., XU Y., SHUM H.-Y., SALESIN D.: Example-based composite sketching of human portraits. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering* (2004), pp. 95–102. 1
- [CM02] COMANICIU D., MEER P.: Mean Shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 5 (2002), 603–619. 3
- [CS00] CHENG H.-D., SUN Y.: A hierarchical approach to color image segmentation using homogeneity. *IEEE Transactions on Image Processing* 9, 12 (2000), 2071–2082. 3
- [DCOY03] DRORI I., COHEN-OR D., YESHURUN H.: Example-based style synthesis. In *Proceedings of Conference on Computer Vision and Pattern Recognition* (2003), pp. 143–150. 1
- [dJB04] DE JUAN C., BODENHEIMER B.: Cartoon textures. In *Proceedings of Eurographics Symposium on Computer Animation* (2004), pp. 267–276. 2
- [FBC*95] FEKETE J.-D., BIZOUARN E., COURNARIE E., GALAS T., TAILLEFER F.: TicTacToon: A paperless system for professional 2D animation. In *ACM SIGGRAPH Conference Proceedings* (1995), pp. 79–90. 1
- [FKS*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. In *ACM SIGGRAPH Conference Proceedings* (2004), pp. 652–663. 1
- [FTP99] FREEMAN W. T., TENENBAUM J. B., PASZTOR E.: *An example-based approach to style translation for line drawings*. Tech. Rep. TR99-11, MERL, 1999. 1
- [HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D.: Image analogies. In *ACM SIGGRAPH Conference Proceedings* (2001), pp. 327–340. 1
- [HM86] HUERTAS A., MEDIONI G.: Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 6 (1986), 651–664. 3
- [HOCS02] HERTZMANN A., OLIVER N., CURLESS B., SEITZ S.: Curve analogies. In *Proceedings of Eurographics Workshop on Rendering* (2002), pp. 233–245. 1
- [IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: As-rigid-as-possible shape manipulation. In *ACM SIGGRAPH Conference Proceedings* (2005). 6
- [IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3D freeform design. In *ACM SIGGRAPH Conference Proceedings* (1999), pp. 409–416. 4
- [JEGP02] JODOIN P.-M., EPSTEIN E., GRANGER-PICHI M., OSTROMOUKHOV V.: Hatching by example: A statistical approach. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering* (2002), pp. 29–36. 1
- [Kuh03] KUHLMANN K.-H.: KVec: Raster graphics to vector graphics converter, 2003. <http://www.kvec.de>. 4
- [Len99] LENBURG J.: *The Encyclopedia of Animated Cartoons (2nd edition)*. Facts on File, New York, NY, 1999. 1
- [MJBFO2] MILLIRON T., JENSEN R. J., BARZEL R., FINKELSTEIN A.: A framework for geometric warps and deformations. *ACM Transactions on Graphics* 21, 1 (2002), 20–51. 4, 5
- [PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. In *ACM SIGGRAPH Conference Proceedings* (2003), pp. 313–318. 2
- [PGR99] PAO H.-K., GEIGER D., RUBIN N.: Measuring convexity for figure/ground separation. In *Proceedings of IEEE International Conference on Computer Vision* (1999), pp. 948–955. 4
- [RK82] ROSENFELD A., KAK A. C.: *Digital Picture Processing*, vol. 1. Academic Press, Orlando, USA, 1982. 3
- [SB89] SOTAK G. E., BOYER K. L.: The Laplacian-of-Gaussian kernel: A formal analysis and design procedure for fast, accurate convolution and full-frame output. *Computer Vision, Graphics, and Image Processing* 48, 2 (1989), 147–189. 2
- [SBv03] SÝKORA D., BURIÁNEK J., ŽÁRA J.: Segmentation of black and white cartoons. In *Proceedings of Spring Conference on Computer Graphics* (2003), pp. 245–254. 2, 3
- [SBv05] SÝKORA D., BURIÁNEK J., ŽÁRA J.: Colorization of black-and-white cartoons. *Image and Vision Computing (to appear)* (2005). 2, 3
- [SC94] SOON S. H., CHOON C. B.: A skeletal line joining algorithm. In *Proceedings of the Computer Graphics International* (1994), pp. 63–73. 6
- [SFLM03] SAUND E., FLEET D., LARNER D., MAHONEY J.: Perceptually-supported image editing of text and graphics. In *Proceedings of ACM Symposium on User Interface Software and Technology* (2003), pp. 183–192. 2, 4
- [VLDP03] VANTIGHEM C., LAURENT N., DECKEUR D., PLANTINET V.: Vector Eye: Raster to SVG converter, 2003. <http://www.siame.com>. 4
- [WH04] WEBER M., HERZOG B.: Autotrace, 2004. <http://autotrace.sourceforge.net>. 4



Figure 10: Five new characters created in the style of Radek Pilař using our framework. Desired fragments are selected using several selection scribbles marked as dashed curves (top). Pairs of composition scribbles marked as solid curves have been used to sketch new compositions (bottom). Note that some composition scribbles serve also as selection scribbles.