Segmentation of Black and White Cartoons

Daniel Sýkora*
Czech Technical University

Jan Buriánek†
Digital Media Production

Jiří Žára‡
Czech Technical University

Abstract

We introduce novel semi-automatic, fast and accurate segmentation technique that allow us to simplify color transfer to the old black and white cartoons produced by classical paper or foil technology, where foreground parts are represented by homogeneous regions with constant grey-scale intensity surrounded by bold dark contours. We assume that original analogue material has been converted to the sequence of digital grey-scale images with PAL resolution suitable for TV broadcasting.

Keywords: image processing, image segmentation, edge detection, region growing, skeletonisation

1 Introduction

If we consider that aged cartoons are stored on black and white material due to former TV broadcast facilities, we have chance to enrich them by new color information. If we apply sensitively proper bright or dark colors into the yet designed artificial black and white world, we are able to increase specific artistic impression which is well perceived especially by children's mind. In short, we have big chance to add a new artistic value.

First we have to premise that probably lots of work has been done on semi-automatic inking of aged black and white movies including the most popular brute force approach. Unfortunately research on this field is usually secret due to commercial aspects of TV broadcasting industry. Nevertheless we describe in short two inking techniques that have been previously published.

First generic method for transferring color into grey-scale images is also known as *luminance keys* [Gonzalez and Wintz 1987]. Smoothly selected luminance interval is converted using user defined look-up table to specified hue, saturation and brightness. It is possible to use simultaneously a few luminance keys to cover different luminance intervals in original grey-scale image. However the condition of disjunction of these intervals introduces really hard constraint especially when we want to apply different colors on regions with the same intensity level.

Second approach described in [Welsh et al. 2002] take advantage of *textural information*. Color transfer between already inked color example and grey-scale target image is based on local luminance distribution matching in $l\alpha\beta$ color space. This technique is inspired by framework of image analogies [Hertzmann et al. 2001]. Subset of representative pixels in color image is selected by jitter sampling or manually using rectangular swatches. This technique is surprisingly successful in natural scenes. However cartoon images with homogeneous regions have usually not enough textural information. Luminance distribution matching is reduced to the simple intensity median matching which has similar disadvantages as inking technique based on *luminance keys*.

Our novel approach is based on *image segmentation*. First we observe that usual cartoon consists of artificial scenes originally created using composition of planar layers. We also exploit the main visual feature of cartoons i.e. that foreground parts are usually bounded by bold dark contours. Thanks to our novel segmentation technique, we are able to obtain original layers and apply ink on them separately. After this procedure we restore the original image composite. We take into account also human driven interaction which guarantees the final output quality, hence we have to ensure the interactive performance of used algorithms to avoid users from derange idle time during semi-automatic phase.

This paper is organized as follows. However the main motivation of this work was to simplify the color transfer to the grey-scale cartoon images, we discuss only the proposed segmentation algorithm considering the fact that inking itself is out of the scope of this paper. Since proposed algorithm consists of two independent steps: initial segmentation and region growing, we present each step in separate section (*Section 2* and *Section 3* respectively), where also previous work related to the current type of problems is discussed. Then in *Section 4* we summarize the results obtained by our implementation.

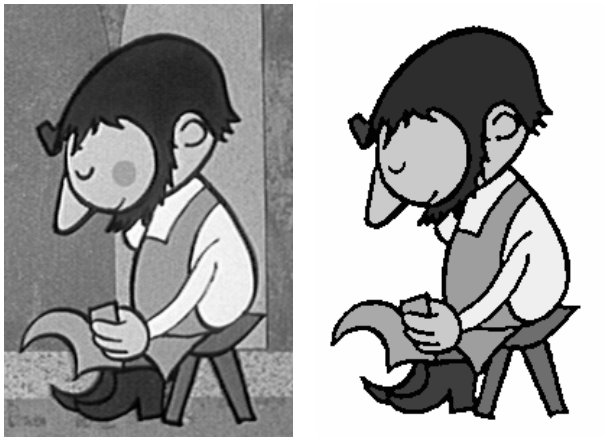

Figure 1: Original grey-scale image (left) and its ideal contour based segmentation (right).

2 Segmentation

Figure 1 demonstrates results we expect as an output of segmentation process. We need to localize and identify each important region in the input grey-scale image which is surrounded by bold dark contour (black on Figure 1). Some of these regions are classified as *background* (white) and another as *foreground* (grey).

Furthermore we assume that background regions are static during whole image sequence, thus they can be manually reconstructed from selected set of frames using one big background layer. Regions on this layer are all at once inked by experienced artist using a

*e-mail: sykorad@fel.cvut.cz
†e-mail: burianek@dmp.cz
‡e-mail: zara@fel.cvut.cz

standard image manipulation software. Due to these circumstances we force our segmentation process to be sensitive only on dynamic foreground layers represented by homogeneous regions with bold dark contours, which have to be inked frame-by-frame.

2.1 Previous work

We first analyze the main disadvantages of widely used segmentation techniques to show the reason why we finally developed a novel contour based segmentation algorithm.

The main problem of classical grey-scale image segmentation techniques based on *multi-level thresholding* using intensity or homogeneity domain [Cheng and Sun 2000] or watersheds algorithm [Vincent and Soille 1991] is connected with oversegmentation (see Figure 2). Lot of work has been done on *region merging* which performs suppression of this artifact (see e.g. [Meyer and Beucher 1990; Koepfler et al. 1994; Haris et al. 1998]]). However most of the successful methods are based on color information or take advantage of user intervention. Unsupervised grey-scale region merging algorithms are usually difficult to implement and introduce unacceptable time complexity.

Another approach to image segmentation widely used in computer vision is based on *edge detection* (see [Ziou and Tabbone 1997] and [Heath et al. 1996] for survey). If we retrieve the exact location of edges, we are able to simply divide the image into several regions considering the fact that edges are their topological boundaries.

Canny [1986] derives optimal convolution filter which is able to extract ideal step edge from 1D signal degraded by Gaussian noise using computational approach based on analytical definition of edge as maximum of the first-order derivatives of image function. Another recent popular edge detector is `SUSAN` (Smallest Univalue Segment Assimilating Nucleus) [Smith and Brady 1997]. This detector is not based on a framework of first-order derivatives, it is faster than 2D version of *Canny* detector and produces comparable accurate results.

Figure 2: Segmentation via multi-level thresholding (left) and watersheds algorithm (right).

Unfortunately advanced edge detectors are sensitive on edge strength (see Figure 3). Certain nontrivial non-maxima suppression technique followed by time consuming hysteresis thresholding mechanism has to be performed to extract important edges. Moreover these detectors produce additional information (e.g. edge orientation) which is not essentially valuable for our purpose. We need to estimate only the exact edge location. This spatial location is invariant to absolute intensity level of discreet image function or to the magnitude of its first-order derivatives approximation.

Moreover our case is really different from natural images. Cartoon edges are visible as much as possible. An artist needs to construct a clear conception of scene shape properties in the viewer's mind. Here edges are bold contours represented by two strong boundary gradients. What we are looking for is not actually the edge but the contour. Robust and fast contour detector based on intensity invariant edge detection have to be developed.

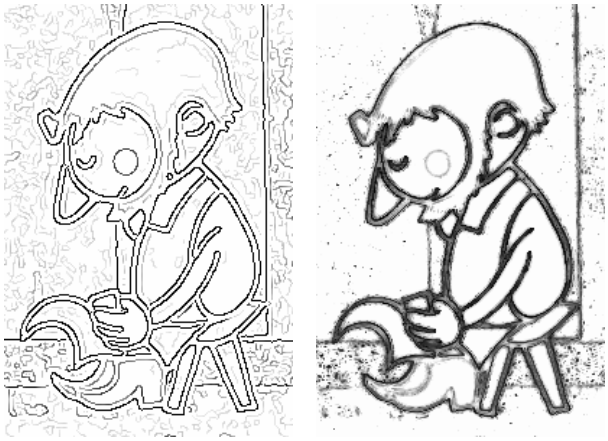

Figure 3: Initial response of *Canny* (left) and `SUSAN` (right) edge detector without thresholding.

Marr and *Hildreth* [1980] proved that human shape understanding is based on the process that is very similar to the convolution with the two-dimensional Laplacian of Gaussian filter ($\mathbf{L} \circ \mathbf{G}$) that approximates second-order derivatives of discreet image function $\mathbf{I}$ and additionally it is not sensitive on the magnitude of its first-order derivatives. This filter should be constructed if we know that:

$$\mathbf{G} = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

and

$$\mathbf{L} = \nabla^2 = \nabla \circ \nabla = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}, \tag{2}$$

where $\mathbf{G}$ is Gaussian and $\mathbf{L}$ is classical Laplacian operator both in two dimensions. We could take the advantage of the convolution linearity $\nabla^2(\mathbf{G} \circ \mathbf{I}) = (\nabla^2 \mathbf{G}) \circ \mathbf{I}$, so $\nabla^2 \mathbf{G}$ could be precalculated by symbolic derivation. Final $\mathbf{L} \circ \mathbf{G}$ convolution filter is expressed using following formula:

$$\mathbf{L} \circ \mathbf{G} = \nabla^2 \mathbf{G} = \frac{1}{\pi\sigma^4} \left(\frac{x^2+y^2}{2\sigma^2} - 1 \right) e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{3}$$

This $\mathbf{L} \circ \mathbf{G}$ filter performs two operations by one-pass filtering: Gaussian removes noise and Laplacian estimates second-order derivative of the noise-free image function. To retrieve the exact edge locations we perform the correct zero crossing test using rotating `2x1` mask.

An important feature of the $\mathbf{L} \circ \mathbf{G}$ filter is that its zero crossings form closed curves, sometimes so called "spaghetti effect". Since the boundary edges are topological boundaries of contour region, we can simply fill this region with constant color using *flood-fill* algorithm to retrieve the shape of searched contour (see Figure 4). This important idea allows us to design required robust and accurate contour detector.

There is one important parameter of the $\mathbf{L} \circ \mathbf{G}$ filter, its standard deviation σ. It enables us to select proper filter scale to fit into our range of interest. If we vary σ we move through image scale-space (see [Witkin 1986]). Edges that are important for us reside only in a small interval of this space. We have to find it experimentally. See Figure 5 for several samples from $\mathbf{L} \circ \mathbf{G}$ scale-space visualized by

zero crossings. There we could observe that the scale of our edges seems to be between $\sigma > 1.0$ and $\sigma < 2.5$.

Unfortunately brute force $\mathbf{L} \circ \mathbf{G}$ filtering of grey-scale image with `PAL` resolution using e.g. $\sigma = 1.60$ on recommended basis `19x19` is time consuming process which exceeds the interactive performance on recent cost effective workstations. However it is possible to speed it up using several decomposition techniques.

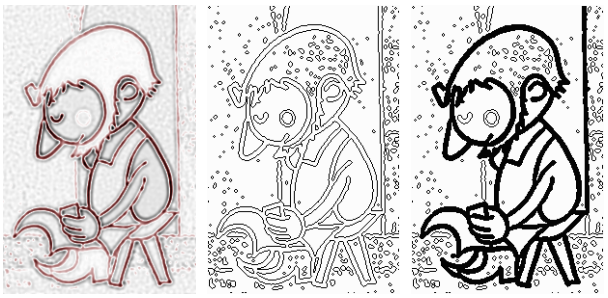

Figure 4: Contour detector in progress: approximation of the second-order derivatives (left), zero crossing (middle), flood-fill (right).

King [1982] proved that $\mathbf{L} \circ \mathbf{G}$ could be exactly decomposed into the sum of two separable filters:

$$\nabla^2 \mathbf{G}(x,y) = \mathbf{h}_1(x) \cdot \mathbf{h}_2(y) + \mathbf{h}_2(x) \cdot \mathbf{h}_1(y), \tag{4}$$

where

$$\mathbf{h}_1(\rho) = \frac{1}{\sqrt{2\pi}\sigma^2} \left(1 - \frac{\rho^2}{\sigma^2}\right) e^{-\frac{\rho^2}{2\sigma^2}} \tag{5}$$

and

$$\mathbf{h}_2(\rho) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{\rho^2}{2\sigma^2}}. \tag{6}$$

Additionally, *Chen et al.* [1987] show, that $\mathbf{L} \circ \mathbf{G}$ filtration could be done in two passes using classical Gaussian with the same standard deviation and smaller $\mathbf{L} \circ \mathbf{G}$ with a lower standard deviation exploiting the *Fourier* transformation:

$$\mathscr{G}(u,v) = \exp\left[-\frac{\sigma^2}{2}\left(u^2+v^2\right)\right] \tag{7}$$

and

$$\mathscr{L} \circ \mathscr{G}(u,v) = \left(u^2+v^2\right) \exp\left[-\frac{\sigma^2}{2}\left(u^2+v^2\right)\right], \tag{8}$$

where $\mathscr{G}(u,v)$ and $\mathscr{L} \circ \mathscr{G}(u,v)$ are continuous spatial *Fourier* transformations of $\mathbf{G}(x,y)$ and $\mathbf{L} \circ \mathbf{G}(x,y)$ respectively. We know that two-pass filtering in spatial domain could be done by one-pass filtering in *Fourier* domain using convolution window that was computed as a multiplication of two convolution windows from each spatial pass. This process could be reformulated as follows:

$$\begin{aligned} \mathscr{L} \circ \mathscr{G}(u,v) = \quad & \exp\left[\tfrac{\sigma^2}{2}\left(1 - \tfrac{1}{k_\sigma^2}\right)\left(u^2+v^2\right)\right] \times \\ & \left(u^2+v^2\right) \exp\left[\tfrac{\sigma^2}{2}\left(-\tfrac{1}{k_\sigma^2}\right)\left(u^2+v^2\right)\right], \end{aligned} \tag{9}$$

where k_σ is a *reconstruction constant*. This constant controls the trade off between the standard deviations of the decomposed filters in the spatial domain: $\sigma_G = \sigma\sqrt{1 - 1/k_\sigma^2}$ and $\sigma_L = \sigma/k_\sigma$.

To select proper k_σ we have to take into account that *Sotak* and *Boyer* [1989] suggest step-by-step operator design procedure to estimate the best standard deviation for Gaussian and smaller $\mathbf{L} \circ \mathbf{G}$ from the given brute force σ. They take into account allowable aliasing energy p_a in the spectrum of the truncated digital approximation of the $\mathbf{L} \circ \mathbf{G}$ filter. If we truncate function in spatial domain then we receive periodical repetition in frequency spectrum and vice versa. This aliasing energy can be expressed for the Gaussian and $\mathbf{L} \circ \mathbf{G}$ spectrum as follows:

$$\frac{100 - p_a}{100} = \frac{\sigma_G^2}{\pi} \int_{-\alpha_G}^{\alpha_G} \int_{-\alpha_G}^{\alpha_G} e^{-\sigma_G^2\left(u^2+v^2\right)} \mathrm{d}u\mathrm{d}v \tag{10}$$

and

$$\frac{100 - p_a}{100} = \frac{\sigma_L^6}{2\pi} \int_{-\alpha_L}^{\alpha_L} \int_{-\alpha_L}^{\alpha_L} \frac{\left(u^2+v^2\right)^2}{e^{\sigma_L^2\left(u^2+v^2\right)}} \mathrm{d}u\mathrm{d}v, \tag{11}$$

where α_G and α_L are *aliasing frequencies*. *Sotak* and *Boyer* [1989] computed them by numerical integration for the given percentage of aliasing energy p_a and standard deviations σ_G and σ_L. It is possible to prepare precomputed σ-independent functions $A_G(p_a) = \alpha_G\sigma_G$ and $A_L(p_a) = \alpha_L\sigma_L$. According to [Sotak and Boyer 1989] the best k_σ should be tuned by $k_\sigma = \sigma\pi/(A_L k_d)$, where $k_d = \sigma\pi/\sqrt{A_L^2 + A_G^2}$ is *decimation constant*.

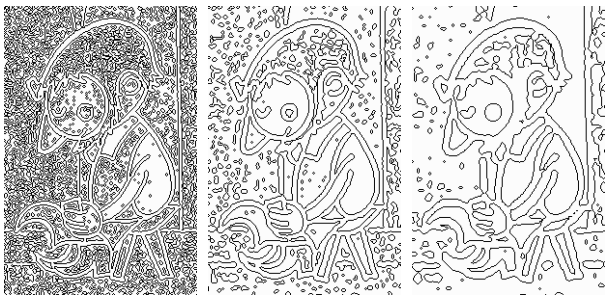

Figure 5: Samples from scale-space of the $\mathbf{L} \circ \mathbf{G}$ filter: (from left to right) $\sigma = 1.0, 2.0, 3.0$.

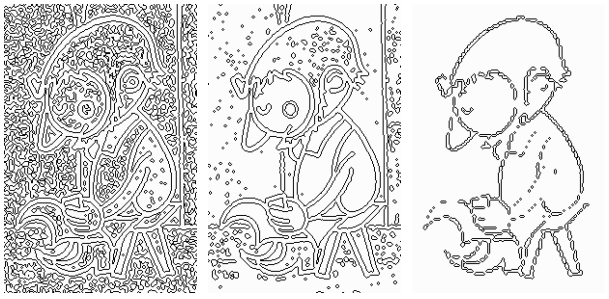

Figure 6: Influence of the allowable aliasing energy: (from left to right) $p_a = 5\%, 10\%, 50\%$.

It is now interesting to show which p_a is suitable for our case. If we increase the p_a then we propagate smoothing (σ_G is growing and σ_L is decreasing, see Figure 6). We selected experimentally $p_a = 10\%$ as a compromise between aliasing and smoothing to reach best filtering quality.

Now we could refine the scale-space interval (see Figure 5) by assumption that integer part of our decimation constant should be $\lfloor k_d \rfloor = 1$. Otherwise if $\lfloor k_d \rfloor < 1$ then our smaller $\mathbf{L} \circ \mathbf{G}$ decomposition is undefined and we should use the brute force $\mathbf{L} \circ \mathbf{G}$ filter with the lowest reasonable support `7x7` or better enter the sub-pixel resolution by an image upsampling (see Figure 7 on the right side, top). With sub-pixel resolution we are able to disintegrate one-pixel distant zero crossings that represent boundaries of a very tight region. On the other side if $\lfloor k_d \rfloor > 1$ then $\mathbf{L} \circ \mathbf{G}$ is sensitive on edges in decimated image with half resolution. This is useful if we detect

edges on zoomed image, where contours are nearly twice as thicker as in standard zoom, hence the $\mathbf{L} \circ \mathbf{G}$ detector using $\lfloor k_d \rfloor > 1$ filter estimates unwanted zero crossings inside bold contours. These "phantom" edges are defined as local minimum extreme in first-order derivatives of image function [Clark 1989] (see Figure 7 on the left side, bottom).

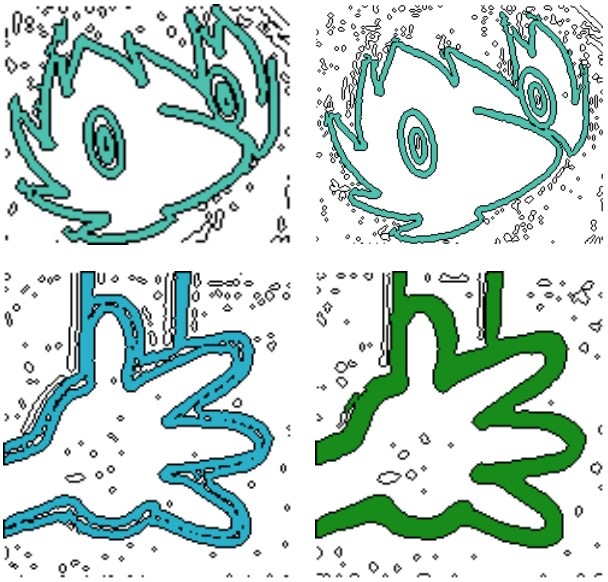

Figure 7: $\mathbf{L} \circ \mathbf{G}$ filtering with different k_d: $[k_d = 1,\ \sigma = 1.23]$ vs. $[k_d < 1,\ \sigma = 1.23]$ sub-pixel accuracy (top), $[k_d = 1,\ \sigma = 1.23]$ vs. $[k_d > 1,\ \sigma = 2.00]$ decimated resolution (bottom).

If we follow the instructions in [Sotak and Boyer 1989] we found that this constraints yield us to eight different $\mathbf{L} \circ \mathbf{G}$ decompositions as we can see in Table 1. For a given σ interval (first column) we decompose brute force $\mathbf{L} \circ \mathbf{G}$ convolution with support $\lfloor 8\sqrt{2}\sigma \rfloor$ (second column) into the two-pass $\mathbf{G}$ and $\mathbf{L} \circ \mathbf{G}$ filtering using smaller separated (*horizontal* + *vertical*) one-dimensional supports. We use full resolution or the decimated image ($\lfloor k_d \rfloor = 2$) to perform final convolution. To preserve the same output resolution, *Sotak* and *Boyer* [1989] suggest to use upsampling by bilinear interpolation. They also recommend to apply the *DC-padding* instead of the *zero-padding* to avoid an erosion of the zero crossings near the image boundaries.

σ	$\mathbf{L} \circ \mathbf{G}$	$\lfloor k_d \rfloor$	$(\mathbf{L} \circ \mathbf{G}) \circ \mathbf{G}$
$\langle 0.80, 0.95 \rangle$	9x9	1	(7+7) ∘ (3+3)
$\langle 0.95, 1.20 \rangle$	11x11	1	(7+7) ∘ (5+5)
$\langle 1.20, 1.50 \rangle$	13x13	1	(7+7) ∘ (7+7)
$\langle 1.50, 1.60 \rangle$	17x17	1	(7+7) ∘ (9+9)
$\langle 1.60, 1.70 \rangle$	19x19	2	(7+7) ∘ (5+5)
$\langle 1.70, 1.90 \rangle$	23x23	2	(7+7) ∘ (7+7)
$\langle 1.90, 2.15 \rangle$	25x25	2	(7+7) ∘ (9+9)
$\langle 2.15, 2.40 \rangle$	27x27	2	(7+7) ∘ (11+11)

Table 1: Useful decompositions of brute force $\mathbf{L} \circ \mathbf{G}$.

While the analysis of *Sotak* and *Boyer* [1989] does not cover filtering on sub-pixel resolution the Table 1 presents decompositions for $\lfloor k_d \rfloor = 1$ and $\lfloor k_d \rfloor = 2$ only. If we want to work with a sub-pixel accuracy we should analyze degradations caused by a bilinear extrapolation [Steger 1998]. For an example of possible distortion caused by the sub-pixel extrapolation see Figure 8, where complicated hair shape caused separation of two zero crossings with the same spatial location. If we focus only on topological properties of zero crossings, we experimentally proved that the best behavior of $\mathbf{L} \circ \mathbf{G}$ filtering on a sub-pixel resolution was reached when $\sigma \in (1.20, 1.60)$ using same filter decomposition as for $\lfloor k_d \rfloor = 1$.

Although the sub-pixel accuracy should be helpful we cannot accept its computational and memory complexity. We have to emphasize that the proposed contour detection technique should be easily extended to the sub-pixel accuracy without any additional adjustments but with expectation of 4x slower performance and additional storage space with the same proportions.

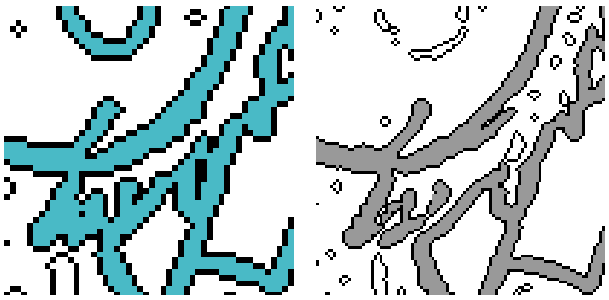

Figure 8: Zero crossings in original resolution (left) and in sub-pixel extrapolation (right).

2.2 Contour detector

To identify pixels, where we could start with flood-filling we introduce a novel adaptive thresholding mechanism. First let us make the observation that this start point can be located only at a pixel which receives negative value after convolution with $\mathbf{L} \circ \mathbf{G}$ filter. See Figure 9 on the left side, where white pixels have positive and black negative value.

We compute a global intensity minimum of these $\mathbf{L} \circ \mathbf{G}$-negative pixels p_i using values from corresponding original grey-scale image I:

$$I_{min} = \min_{i:\mathbf{L} \circ \mathbf{G}(I(p_i)) < 0} \{I(p_i)\}. \tag{12}$$

Pixels with this minimal intensity are the good start points for our iterative algorithm. We start flood-fill algorithm on pixels $I(p_i) = I_{min}$ and compute intensity median $\tilde{I}$ of pixels from the area being filled. Now unfilled pixels with properties $I(p_i) < k\tilde{I}$ and $\mathbf{L} \circ \mathbf{G}(I(p_i)) < 0$ become start points for the next flood-filling step, where k is convergence constant which was experimentally adjusted to 0.5. We repeat this operation while median of new filled area is lower or the same as its value from the previous step.

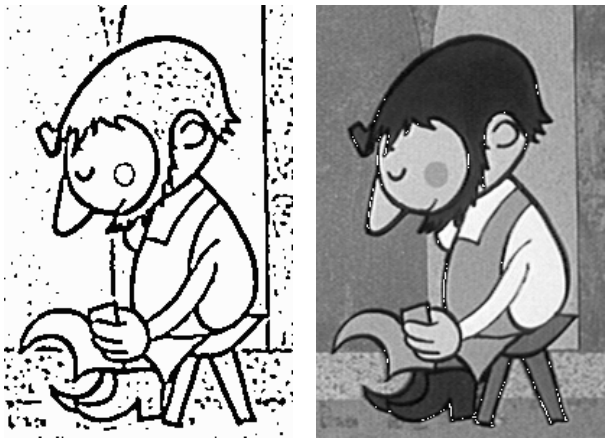

Figure 9: Pixels p_i: with $\mathbf{L} \circ \mathbf{G}(I(p_i)) < 0$ (left) and $I(p_i) = I_{min}$ (right).

This adaptive contour finding algorithm converges very fast. On the average it stops after the second pass when new median $\tilde{I}_{k+1}$ is

not bigger than $\tilde{I}_k$. This phenomenon illustrates Figure 10 on the left, where almost all contours were filled in initial I_{min} step and only the eye was filled in second $\tilde{I}$ step when the algorithm also terminated. By using this automatically predicted filling threshold $\tilde{I}$, we are able to retrieve all important contours without user intervention. Median $\tilde{I}$ stays usually constant during whole sequence but sometimes it differ due to luminance fluctuation, thus its automatic temporal tuning is necessary.

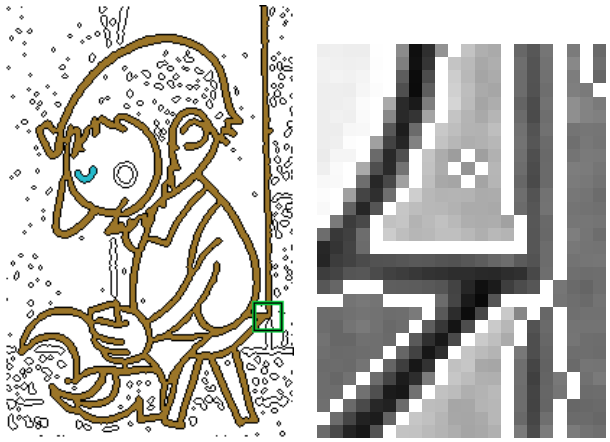

Figure 10: Two steps of contour filling algorithm (left) and T-junction problem (right).

The main problem of the presented contour filling algorithm are T-junctions of $\mathbf{L} \circ \mathbf{G}$ zero crossings (see Figure 10 on the right). They connect foreground and background contours topologically to a single solid contour. If we fill foreground contour we also fill the background one.

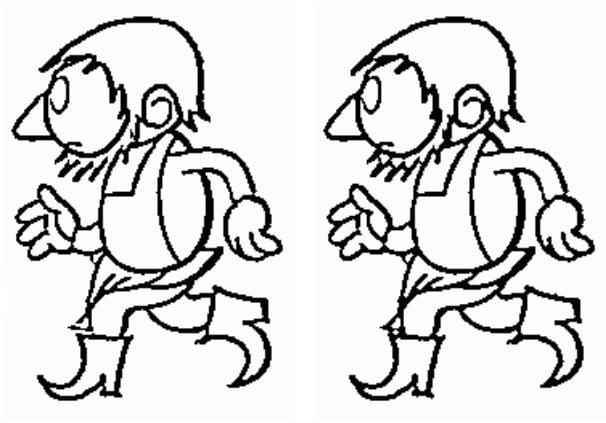

Figure 11: Contour extraction via image subtraction (left) and morphological erosion (right). The second method preserves contour connectivity.

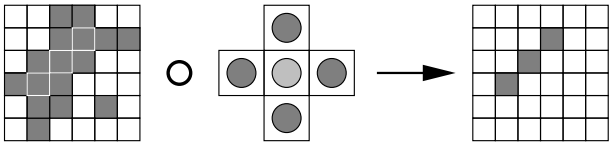

Figure 12: Example of image erosion using 4-connectivity morphological structure element.

To avoid these artifacts we apply an easy contour authentication test. We compute the minimal intensity value I_{min} from a small neighborhood of the contour pixel (e.g. window 5x5) and compare it with the median value $\tilde{I}_n$ from the last iteration of our algorithm. If $\tilde{I}_n < I_{min}$, we can conclude that the proper pixel does not come from a foreground contour.

Finally, residual zero crossings have to be removed. We can not perform simple image subtraction because thus would destroy an important topological characteristics. This is due to the fact that some zero crossings appear on two neighbor pixels and if we remove them we clearly break the contour connectivity (see Figure 11).

To solve this problem we use morphological *erosion* operator. This operator was derived using the theory of *mathematical morphology* (for further study on this field see [Haralick and Shapiro 1992] or [Serra 1993]).

Erosion of a binary image is very similar to the convolution process on the grey-scale domain. We use moving window known as *structure element* (see Figure 12 in the middle), which reduces thickness of contours and wipe off the residual one pixel wide zero crossings. Additionally, it preserves 4-connectivity of resulting image due to its shape properties.

The origin of structure element is placed on each pixel inside the binary image and the following relation is performed: if all pixels covered by the structure element are classified as edges (this means contour or zero crossing in our case) then we place edge pixel on origin position into the output image, otherwise we place background pixel (see Figure 12 for example).

After erosion we have the binary image, where black pixels together with an image border represent contours and white pixels regions. It is now easy to assign unique index for each closed region. We can achieve this by selective flood-filling algorithm which starts filling on every white pixel using unique region index. We could also do it even faster if we use the scan-line based 4-connectivity two-pass region marking algorithm [Rosenfeld and Kak 1982].

To divide foreground regions from background layer, we simply use the region size thresholding. For most scenes it is suitable to set size threshold to the 15% of total image size. Regions with size below this threshold are classified as foreground, the others as background. However using size threshold we are not able to exclude small regions that look like foreground parts but they are actually background holes. These regions have to be classified as parts of background by human intervention.

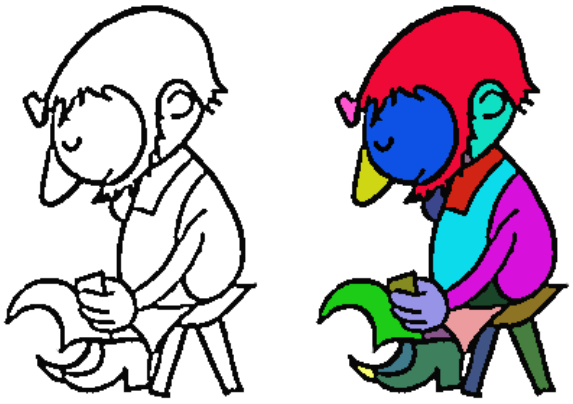

Figure 13: Towards final segmentation.

The final index map of contour based segmentation is shown in Figure 13 (right picture). This figure presents the last step of proposed contour based segmentation algorithm. We summarize three important features: it is nearly full automatic, it produces accurate results and it is fast (full frame segmentation of image in PAL resolution takes in average 0.5 second on the 750MHz CPU).

3 Region growing

We additionally need to retrieve real region boundaries. These boundaries should be defined as medial axes of contours also known as *skeleton*. It helps us to estimate how deeply into the contour anti-aliasing we have to apply color. On Figure 14 we can exactly see what does it mean. The left top picture is inked using original contour boundaries and the right top picture using the contour skeleton.

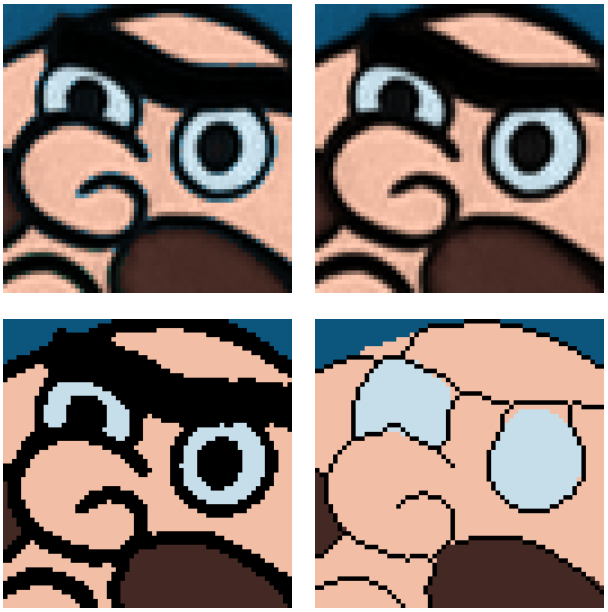

Figure 14: Color flooding driven by real contours (left) and by contour skeleton (right).

Since locations of zero crossings are placed near the middle of the edge downhill and the contour boundaries are not ideal step edges, we sometimes omit visually important pixels during color luminance modulation. This modulation is driven by spatial location of regions produced by initial contour based segmentation. Problematic pixels reside topologically inside the contour region, hence their intensities are modulated using incorrect black color. Contour skeleton allows us to enlarge the initial region shapes beyond the contour gaps and provides us to reach correct color modulation inside contour anti-aliasing.

3.1 Previous work

Skeleton is another entity described in the terminology of the mathematical morphology [Serra 1993]. It could be defined informally as the union of circle centers which are located inside contour and herewith tangent the contour boundaries. Contour skeleton is theoretically obtained by sequential *thinning* process using so called *hit-or-miss* transformation. This transformation is very similar to the erosion operator however it additionally preserves skeleton topology features. This iterative peeling is usually a very slow process and so several performance enhancing techniques has been developed.

Suzuki proposed the sequential thinning algorithm based on the distance field transformation [Suzuki and Abe 1986] (see also [Zhou and Toga 1999]). *Kégl* developed a robust piecewise linear skeletonisation algorithm based on principal curves (smooth curves which pass through the middle of a d-dimensional probability distribution) polished using several curvature penalty and vertex degradation rules [Kégl and Kryźak 2002].

We solve this problem using a slightly different approach. We need not an exact skeleton geometry but only enlargement of already segmented regions towards the contour region to hide visible gaps between real region boundary and contour anti-aliasing (see Figure 14). This should be done for every contour pixel by simple retrieval of the nearest region using growing circle mask which produce very similar results to that using segmentation based on skeleton boundaries. But as we can see on Figure 15 right picture in the middle, this approach is not as robust as we want to be. It fails in cases when the image contains tight `V`-shaped inlets which should be classified as contour even if we select large σ for $\mathbf{L} \circ \mathbf{G}$ filtering.

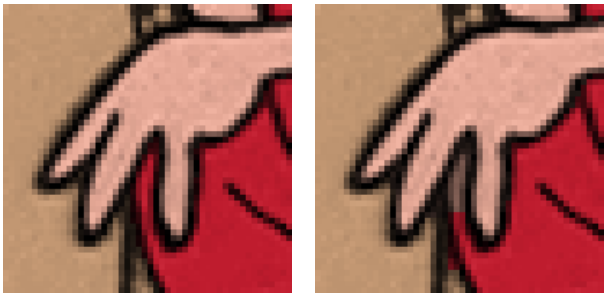

Figure 15: Region enlargement driven by skeleton (left) and by intensity gradient (right).

3.2 Gradient seek

To make enlargement more robust we propose a novel region growing technique that is not based on looking for a minimal distance (which is definitely equivalent to skeletonisation), but on the region highest intensity using image intensity gradient. This is inspired by the fact that intensity near the contour boundary usually upraises from dark contour pixels to brighter value similar to the region intensity median. If we follow this gradient slope we indeed retrieve the nearest region. On the other side it is clear that if we simply retrieve only the nearest highest intensity we do not complete proposed task. We have to follow image gradient using 4-connectivity or 8-connectivity curve to make sure that we do not cross over deep intensity valley.

For every contour pixel from the region marker array `M` we call a simple function that retrieves the marker of the first pixel from the nearest regular region that has to be connected with initial pixel via a 4-connectivity gradient curve. To solve this task we exploit priority queue of the already visited pixels in the original grey-scale image `I`. See following function source code:

```
MARKER *gradient_seek(int x, int y)
{
 PQueue *pixel=new PQueue(I);

 while (M[x][y]->id==CONTOUR) {

       pixel->Push(x+1,y);
       pixel->Push(x,y+1);
       pixel->Push(x-1,y);
       pixel->Push(x,y-1);

       pixel->Pop(&x,&y); }

 delete pixel; return &M[x][y];
}
```

Method `Push()` of `PQueue` class inserts the new pixel into the priority queue while preserving a nondecreasing order of pixel intensities and at the same time it checks if the included pixel is not already in the queue. The second method `Pop()` takes out a pixel from the top of the priority queue and returns its coordinates. See Figure 16 for gradient seeking algorithm in progress. Especially

steps 5 and 6 are interesting. There, seeking process get stuck to the deep intensity valley which should be here a simulation of the negative noise intensity peak or any other type of local distortion.

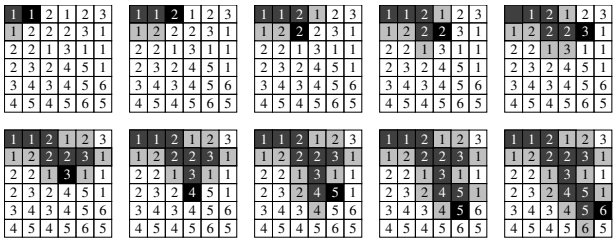

Figure 16: Gradient seek: visited pixels (dark grey), never visited pixels (white), pixels in priority queue (grey), pixel on the top of the priority queue (black).

While presented algorithm is based on a robust *backtracking* technique which is able to make the round of local intensity peaks or valleys, it may introduce significant slowdown for a large data sets. On the other hand if we consider that contours usually take place in approximately 5% of image pixels and the longest path toward a regular region is on average shorter than 10 pixels, we conclude that this algorithm demands negligible computational overhead in front of presented sophisticated skeletonisation methods. While it also uses original grey-scale image it is much more precise in sense of the real contour shape in contrast to simple skeletonisation algorithm based only on information taken from the binary image (see Figure 17 to compare results).

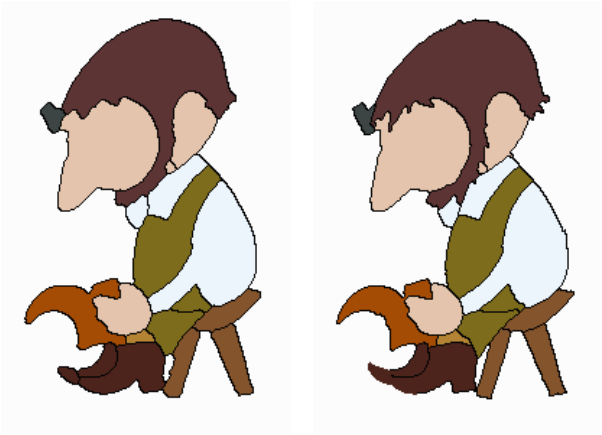

Figure 17: Segmentation based on contour skeleton (left) and on gradient seek (right). The second method preserves important details.

4 Results

We experimentally prove usability of proposed segmentation algorithm. It has been implemented inside proprietary semi-automatic PC application that allows us to efficiently apply ink on the aged black and white cartoon *"O loupežníku Rumcajsovi"* directed by unfortunately yet deceased but still immortal guru of Czech cartoon school *Radek Pilař*. This project came out from co-operation of three subjects: *Czech TV* (CTV), *Universal Production Partners* (UPP) and *Digital Media Production* (DMP).

The whole semi-automatic inking process consists of four independent phases: segmentation (explained in this paper), marker prediction, color luminance modulation and final composition of foreground and background layers. For detail description of algorithms being used in this inking pipeline and for more information about this project see [Sýkora 2003].

The average processing speed vary from 20 to 40 seconds per frame. If we do not take into account pre-processing and post-processing phases, one operator is able to apply ink on whole episode (12000 frames) in two weeks of full time inking. The main slowdown caused by contour detector is connected with retouching of tight regions, broken contours and contours inside background area. This correction work takes in average 15% of whole interaction time.

5 Conclusions

Although proposed algorithm has been developed and employed only on mentioned cartoon and all example images in this paper came from this piece of *Pilař*'s work, proposed principles are general enough to be applied on cartoons with similar artistic properties. In other words, in this paper we have solved a general problem: extraction of foreground regions from given sequence of grey-scale images featuring bold dark contours and intensity homogeneous regions.

6 Acknowledgements

This work has been supported by *Universal Production Partners* (UPP), *Digital Media Production* (DMP) and partly by the Ministry of Education, Youth and Sports of the Czech Republic under research program No. Y04/98: 212300014 (Research in the area of information technologies and communications). Images in this paper are published by the courtesy of © UPP.

References

CANNY, J. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 6, 679–698.

CHEN, J. S., HUERTAS, A., AND MEDIONI, G. 1987. Fast convolution with Laplacian-of-Gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 9*, 4 (July), 584–590.

CHENG, H.-D., AND SUN, Y. 2000. A hierarchical approach to color image segmentation using homogenity. *IEEE Transactions on Image Processing 9*, 12 (Dec.), 2071–2082.

CLARK, J. J. 1989. Authenticating edges produced by zero-crossing algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 1, 43–57.

GONZALEZ, R. C., AND WINTZ, P. 1987. *Digital Image Processing*, 2nd ed. Addison-Wesley, Reading, Massachusetts.

HARALICK, R. M., AND SHAPIRO, L. G. 1992. *Computer and Robot Vision*, vol. 1. Addison Wesley, New York, NY, USA.

HARIS, K., ESTRADIADIS, S. N., MAGLAVERAS, N., AND KATSAGGELOS, A. K. 1998. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing 7*, 12, 1684–1699.

HEATH, M. D., SARKAR, S., SANOCKI, T., AND BOWYER, K. W. 1996. Comparison of edge detectors: A methodology and initial study. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 143–148.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *SIGGRAPH 2001 Conference Proceedings*, 327–340.

KÉGL, B., AND KRYŹAK, A. 2002. Piecewise linear skeletonization using principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 1 (Jan.), 59–74.

KING, D. 1982. Implementation of the Marr-Hildreth theory of edge detection. Tech. Rep. ISG-102, The University of Southern California, Oct.

KOEPFLER, G., LOPEZ, C., AND MOREL, J. M. 1994. A multiscale algorithm for image segmentation by variational method. *SIAM Journal on Numerical Analysis 31*, 1 (Feb.), 282–299.

MARR, D., AND HILDRETH, E. C. 1980. Theory of edge detection. In *Proceedings of Royal Society*, vol. B207, 187–217.

MEYER, F., AND BEUCHER, S. 1990. Morphological segmentation. *Journal of Visual Communication and Image Representation 1*, 1 (Sept.), 21–46.

ROSENFELD, A., AND KAK, A. C. 1982. *Digital Picture Processing*, vol. 1. Academic Press, Orlando, USA.

SERRA, J. 1993. *Image Analysis and Mathematical Morphology*, 4th ed., vol. 1. Academic Press, Oval Road, London, UK.

SMITH, S. M., AND BRADY, J. M. 1997. SUSAN – a new approach to low-level image processing. *International Journal of Computer Vision 32*, 1 (May), 45–78.

SOTAK, G. E., AND BOYER, K. L. 1989. 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 (Nov.), 147–189.

STEGER, C. 1998. Evaluation of subpixel line and edge detection precision and accuracy. In *International Archives of Photogrammetry and Remote Sensing*, vol. 32, 256–264.

SUZUKI, S., AND ABE, K. 1986. Sequential thinning of binary pictures using distance transformation. In *Proceedings of the 8th International Conference on Pattern Recognition*, 289–292.

SÝKORA, D. 2003. *Inking Old Black and White Cartoons*. Master's thesis, Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic.

VINCENT, L., AND SOILLE, P. 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 6 (June), 583–598.

WELSH, T., ASHIKHMIN, M., AND MUELLER, K. 2002. Transferring color to greyscale images. In *SIGGRAPH 2002 Conference Proceedings*, 277–280.

WITKIN, A. P. 1986. Scale space filtering. In *From Pixels to Predicates: Recent Advances in Computational and Robot Vision*, Ablex, Norwood, NJ, USA, A. P. Pentland, Ed., 5–19.

ZHOU, Y., AND TOGA, A. W. 1999. Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics 5*, 3 (July/Sept.), 196–209.

ZIOU, D., AND TABBONE, S. 1997. Edge detection techniques - An overview. Tech. Rep. 195, Department of Mathematics and Informatique, Universit de Sherbrooke, Oct.