J. Hlaváčová (Ed.): ITAT 2017 Proceedings, pp. 167–173
CEUR Workshop Proceedings Vol. 1885, ISSN 1613-0073, © 2017 P. Vlašánek, I. Perfilieva

Inpainting Using F-Transform for Cartoon-Like Images

Pavel Vlašánek, Irina Perfilieva

Institute for Research and Applications of Fuzzy Modeling,
University of Ostrava, 30. dubna 22, Ostrava, Czech Republic
pavel.vlasanek@osu.cz

Abstract: We propose to modify image inpainting technique based on F-transform for application dedicated to cartoon images. The images have typical features which are taken into consideration. These features make original algorithm ineffective, because of its isotropic nature. Proposed modification changes it to an anisotropic.

1 Introduction

Image restoration, in meaning of object removal or damage recovery, so called image inpainting, is challenging task in image processing. Let us consider input image I which contains unwanted pixels considered as a damage. In the process of image inpainting, the damaged area should be erased and replaced by some proper part of I. The selection of the proper part is crucial. One option is to choose square shaped patch and replace the damaged area by its copy. In that case, we are talking about *patch-based image inpainting*[1, 6, 7, 8]. In this paper, as well as in many others, we use principle of the techniques taking colors of the separated pixels in the close neighborhood of the damaged area to the consideration [2, 3, 4, 5].

Structure of the paper is as follows. Section 2 gives preliminaries including information about F-transform and details about its two types. Section 3 describes basics of the specific type of images used in this paper and Section 4 gives information about mathematical morphology. Detailed description of proposed technique is in Section 5 and conclusion is given in Section 6.

2 Preliminaries

Let us fix the following notation to use throughout the paper. Image I is a 2D vector function such as $I : [0,M] \times [0,N] \to [0,255]^3$, where $[0,255]^3$ stands for pixel intensities in three color channels. We denote $[0,M] = \{0,1,2,\ldots,M\}$, $[0,N] = \{0,1,2,\ldots,N\}$ and $[0,255] = \{0,1,2,\ldots,255\}$. Therefore, $M+1$ is the image width and $N+1$ is the image height. Image I is assumed to be partially defined: it is defined (known) on the area Φ and undefined (unknown, damaged) on the area Ω. The border between these areas is denoted by $\delta\Omega$ and assumed to be unknown. It is assumed that $\Phi \cap \Omega = \emptyset$ and $\Phi \cup \Omega \cup \delta\Omega = [0,M] \times [0,N]$. Mask S is a binary image where white pixels denote unknown area $\Omega + \delta\Omega$. The mask is created by user with respect to areas intended for deletion. The notation is illustrated in Fig. 1.

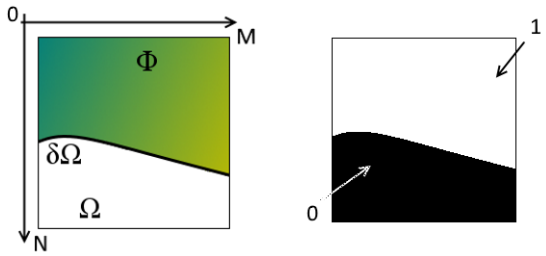


Figure 1: a) Two areas where image I is defined (Φ) and undefined (Ω); b) mask S.

We are focused on image restoration. By this we mean that pixels from $\Omega \cup \delta\Omega$ should be replaced by pixels from Φ. The resulting image should make an impression that damage is not present.

2.1 F^0-Transform

Below, we recall the definition of a fuzzy partition [10]. Fuzzy sets $A_0,\ldots,A_m$ identified with their membership functions (basic functions) $A_0,\ldots,A_m : [0,M] \to [0,1]$, establish a *fuzzy partition* of $[0,M]$ with nodes $0 = x_0 < x_1 < \cdots < x_m = M$ if the following conditions are fulfilled:

1) $A_k : [0,M] \to [0,1]$, $A_k(x_k) = 1$;
2) $A_k(x) = 0$ if $x \notin (x_{k-1},x_{k+1})$, $k = 0,\ldots,m$;
3) $A_k(x)$ is continuous;
4) $A_k(x)$ strictly increase on $[x_{k-1},x_k]$, $k = 1,\ldots,m$; and strictly decrease on $[x_k,x_{k+1}]$, $k = 1,\ldots,m$;
5) $\sum_{k=0}^{m} A_k(x) = 1$, $x \in [0,M]$.

We say that the fuzzy partition given by $A_0,\ldots,A_m$, is an *h-uniform fuzzy partition* if nodes $x_k = hk$, $k = 0,\ldots,m$, are equidistant, $h = M/m$ and two additional properties are met:

6) $A_k(x_k - x) = A_k(x_k + x)$, $x \in [0,h]$, $k = 0,\ldots,m$;
7) $A_k(x) = A_{k-1}(x-h)$, $k = 1,\ldots,m$, $x \in [x_{k-1},x_{k+1}]$.

Parameter h will be referred to as a radius.

Assume that fuzzy sets $A_0,\dots,A_m$ establish a fuzzy partition of $[0,M]$. The following vector of real numbers $\mathbf{F}_m[I]=(F_0,\dots,F_m)$ is the *(direct) discrete F-transform* of I w.r.t. $A_0,\dots,A_m$ where the $k-$th component F_k is defined by

$$F_k^0=\frac{\sum_{x=0}^{M}A_k(x)I(x)}{\sum_{x=0}^{M}A_k(x)},\ k=0,\dots,m. \tag{1}$$

Let us introduce F-transform of a 2D grayscale image I that is considered as a function $I:[0,M]\times[0,N]\to[0,255]$.

Let $A_0,\dots,A_m$ and $B_0,\dots,B_n$ be basic functions, $A_0,\dots,A_m:[0,M]\to[0,1]$ be fuzzy partition of $[0,M]$ and $B_0,\dots,B_n:[0,N]\to[0,1]$ be fuzzy partition of $[0,N]$.

We say that the $m\times n$-matrix of real numbers $[F_{kl}^0]$ is called *the (discrete) F-transform* of I with respect to $\{A_0,\dots,A_m\}$ and $\{B_0,\dots,B_n\}$ if for all $k=0,\dots,m,\ l=0,\dots,n$,

$$F_{kl}^0=\frac{\sum_{y=0}^{N}\sum_{x=0}^{M}I(x,y)A_k(x)B_l(y)}{\sum_{y=0}^{N}\sum_{x=0}^{M}A_k(x)B_l(y)}. \tag{2}$$

The coefficients F_{kl}^0 are called *components of the F-transform*.

2.2 F^1-Transform

In this section, we recall the (direct) F^1-transform as it has been presented in [11]. Let $\{A_k\times B_l\mid k=0,\dots,m,l=0,\dots,n\}$ be a fuzzy partition of $[0,M]\times[0,N]$. $L_2^1(A_k)\subseteq L_2(A_k)$ ($L_2^1(B_l)\subseteq L_2(B_l)$)[1] be a linear span of the set consisting of two orthogonal polynomials

$$P_k^0(x)=1,\quad P_k^1(x)=x-x_k,$$
$$(Q_l^0(y)=1,\quad Q_l^1(y)=y-y_l),$$

where 1 is a denotation of the respective constant function.

Analogously, let $L_2^1(A_k\times B_l)\subseteq L_2(A_k\times B_l)$ be a linear span of the set consisting of three orthogonal polynomials

$$S_{kl}^{00}(x,y)=1,\quad S_{kl}^{10}(x,y)=x-x_k,\quad S_{kl}^{01}(x,y)=y-y_l.$$

Let $I\in L_2([0,M]\times[0,N])$, and F_{kl}^1 be the orthogonal projection of $I|_{[x_{k-1},x_{k+1}]\times[y_{l-1},y_{l+1}]}$ on subspace $L_2^1(A_k\times B_l)$, $k=0,\dots,m,\ l=0,\dots,n$.

We say that matrix $\mathbf{F^1_{mn}}[I]=(F_{kl}^1),\ k=0,\dots,m,\ l=0,\dots,n$, is the F^1-transform of I with respect to $\{A_k\times B_l\mid k=0,\dots,m,l=0,\dots,n\}$, and F_{kl}^1 is the corresponding F^1-transform component.

[1] $L_2(A_k)$ is a Hilbert space of square-integrable functions $f:[x_{k-1},x_{k+1}]\to\mathbb{R}$ with the *weighted inner product* $\langle f,g\rangle_k$ given by

$$\langle f,g\rangle_k=\int_{x_{k-1}}^{x_{k+1}}f(x)g(x)A_k(x)dx, \tag{3}$$

where the *weight function* is equal to A_k.

The F^1-transform components of I are linear polynomials in the form

$$F_{kl}^1(x,y)=c_{kl}^{00}+c_{kl}^{10}(x-x_k)+c_{kl}^{01}(y-y_l), \tag{4}$$

where the coefficients are given by

$$\begin{aligned}c_{kl}^{00}&=\frac{\sum_{y=0}^{N}\sum_{x=0}^{M}I(x,y)A_k(x)B_l(y)}{\sum_{y=0}^{N}\sum_{x=0}^{M}A_k(x)B_l(y)},\\ c_{kl}^{10}&=\frac{\sum_{y=0}^{N}\sum_{x=0}^{M}I(x,y)(x-x_k)A_k(x)B_l(y)}{\sum_{y=0}^{N}\sum_{x=0}^{M}(x-x_k)^2A_k(x)B_l(y)},\\ c_{kl}^{01}&=\frac{\sum_{y=0}^{N}\sum_{x=0}^{M}I(x,y)(y-y_l)A_k(x)B_l(y)}{\sum_{y=0}^{N}\sum_{x=0}^{M}(y-y_l)^2A_k(x)B_l(y)}.\end{aligned} \tag{5}$$

2.3 F-Transform Image Inpainting

In [12], the technique of F-transforms was proposed for the inpainting. It uses two steps: *direct and inverse* of the 0th-degree F-transform. The direct step is described in the previous section whereas the inverse is as follows

$$O(x,y)=\sum_{k=0}^{m}\sum_{l=0}^{n}F_{kl}^0A_k(x)B_l(y), \tag{6}$$

where O is the output (reconstructed) image. In fact, the algorithm computes the F-transform components of the input image I and spreads the components afterwards to the size of I. For details see [12].

Let us recall basics of the technique and illustrate its update for the cartoon images. The original technique works with the assumption that damaged pixels of I should not be included in a component value. For that purpose, the binary mask S is used in the computation:

$$F_{kl}^0=\frac{\sum_{y=0}^{N}\sum_{x=0}^{M}I(x,y)S(x,y)A_k(x)B_l(y)}{\sum_{y=0}^{N}\sum_{x=0}^{M}S(x,y)A_k(x)B_l(y)}.$$

This approach works well for photos as was shown in [13, 15, 12, 9]. For cartoon images, the quality of reconstruction is not sufficient because of the isotropic nature of the algorithm. The problem is that edges are not taken into consideration during the computation.

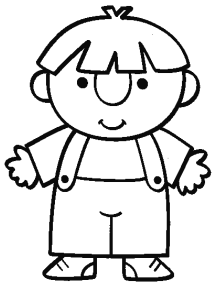
3 Cartoon Images

In this paper, we suggest inpainting technique aimed to be applied to images with two specific features:

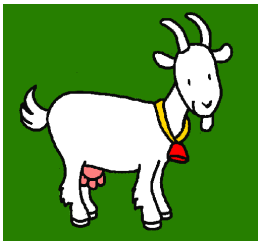
- limited color palette,
- strong and thick uni-color edges.

These features are usually included in simple cartoon images as can be seen in Fig. 2.

For testing purposes, we created a set of artificial images with the same features. The set is in Fig. 3.



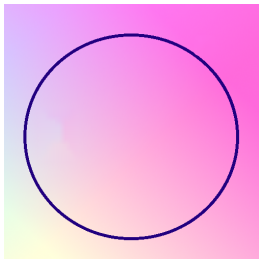
(a) Boy



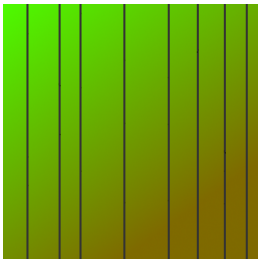
(b) Goat

(c) Homer

Figure 2: Test set of cartoon images.



(a) Circle



(b) Lines

(c) Shape

Figure 3: Test set of artificial cartoon images.

4 Mathematical Morphology

Application of mathematical morphology [14] is an important step in the proposed method. Let us give a short description of this technique.

In mathematical morphology, a structuring element is selected and applied to the input image. For our method, a binary image is used. We recall three main operations: *erosion*, *dilation* and *closing*.

4.1 Erosion

The erosion is defined as follows

$$I \ominus T = \{z \in [0,M] \times [0,N] | T_z \subseteq I\},$$

where T is a structuring element and z is a translation vector. Operator of binary erosion is in fact a test whether image I contains areas like T.

4.2 Dilation

The dilation is defined as follows

$$I \oplus T = \bigcup_{t \in T} I_t,$$

where T is a structuring element and I_t is the translation of I by t.

4.3 Closing

Operator of closing is the erosion of dilation defined as follows

$$I \bullet T = (I \oplus T) \ominus T.$$

The effect of binary closing is in filling small holes and imperfections in the image I.

5 Novel Inpainting Technique

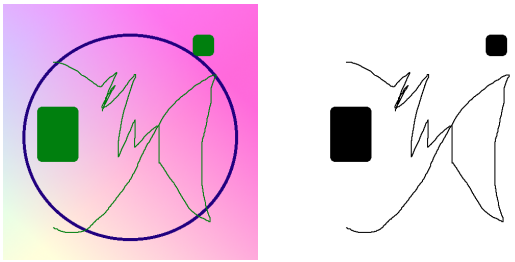
If image I contains only few colors and thick edges, reconstruction using original algorithm based on (6) is affected by visible artifacts. Illustration is in Fig. 12. A new proposed algorithm is based on the assumption that similar areas should be reconstructed independently. Main idea is to separate these areas and reconstruct each of them with respect to a particular color. In this paper, we propose to separate edges (pixels with high gradient) from the rest of the image, reconstruct their damaged parts and continue with the other areas afterwards.

For this purpose, another binary image V is taken into consideration. The image V is created automatically during the reconstruction process and it influences the computation of the F^0-transform components as it is shown below

$$F^0_{kl} = \frac{\sum_{y=0}^{N} \sum_{x=0}^{M} I(x,y) S(x,y) V(x,y) A_k(x) B_l(y)}{\sum_{y=0}^{N} \sum_{x=0}^{M} S(x,y) V(x,y) A_k(x) B_l(y)}.$$

We can say that image V and mask S overlaps image I. Mask S coincides with the characteristic function of area Φ. Image V designates by 1 the so called *valid pixels*. The latter are used in the reconstruction process. Therefore, the edges are reconstructed from pixels of the known part of edges only and similarly, for pixels from the other non-edge areas. This feature changes isotropic nature of the original inpainting algorithm to anisotropic because pixel colors are not necessarily distributed to the all neighbourhood.

Bellow, the proposed algorithm is illustrated on the input from Fig. 4.

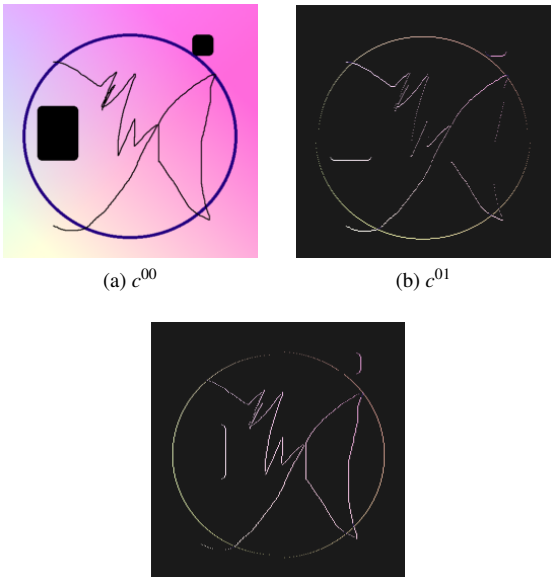


(a) I (b) S

Figure 4: Input image and mask for algorithm description.

1) Compute the F^1-transform.

Comment: At this step, we compute coefficients c^{00}, c^{10}, c^{01} of I F^1-transform components in accordance with (5). The output for the input in Fig. 4 is in Fig. 5.



(a) c^{00} (b) c^{01}

(c) c^{10}

Figure 5: Coefficients of F^1-transform. Contrast was enhanced for better visibility.

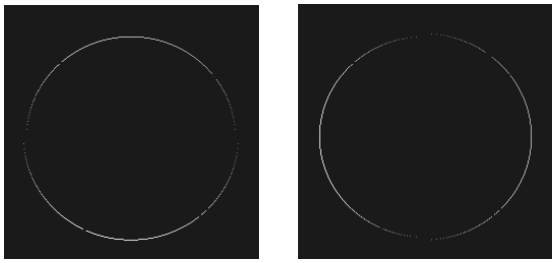
2) Upscale c^{10} and c^{01} to the size of image I and convert them to gray-scale.

3) Update c^{01} and c^{10} by subtracting mask S from them.

Comment: Performing this update we eliminate false edges. Illustration is in Fig. 6.

4) Make shifted copies of c^{01} and c^{10}.

Comment: Edges are detected in the places with the highest gradient. Because of our assumption about the thick edges in I, we copy and shift c^{01} to the left and c^{10} to the up. Doing this, we restrict horizontal and vertical edges.



(a) c^{01} (b) c^{c10}

Figure 6: Updated coefficients c^{01} and c^{10}. Contrast was enhanced for better visibility.

5) Create new image V as the union of c^{01}, c^{10} and their shifted copies. Threshold V to obtain the binary image.

Comment: After this step, we obtain binary image V. Its white pixels represent edges whereas black pixels represent areas without significant gradient. Illustration is in Fig. 7.

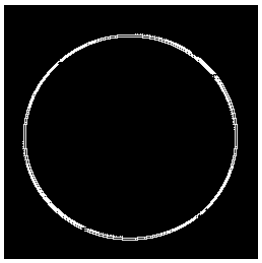


Figure 7: Image V composed from c^{01}, c^{10} and their shifted copies.

6) Apply morphological closing to V.

Comment: The purpose of this step is to fill in all imperfections of V. In step 3, we subtracted the mask and that created holes in the detected edge area. By closing, we fix these holes and prolong (connect) appropriate parts of image edges. Illustration is in Fig. 8.

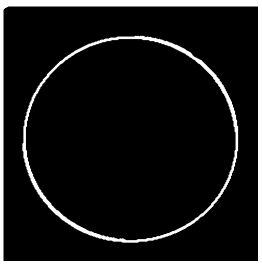
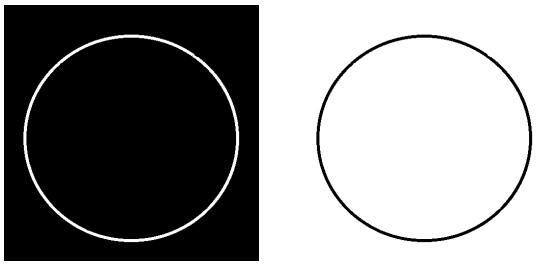


Figure 8: Morphological closing applied on Fig. 7.

7) Use white pixels of V to find edge area of I and by histogram analysis determine a dominant color of it. Further on, the color is called *edge color*.

8) Based on the *edge color* divide I to V_g and V_c and subtract mask from both. Turn V_g and V_c to binary images and apply morphological closing.

Image V_g represents edges of the I whereas V_c the rest. Image V_g contains holes because of the mask subtraction. By closing, we fill the holes. Illustration of this step is in Fig. 9.



(a) V_g (b) V_c

Figure 9: Binary division of image I to the edge area V_g and the rest V_c followed by a morphological closing.

9) Find intersections of V_g and the mask S.

The intersections determine places on the edge area which are damaged. Let us name this intersection S_g. Illustration is in Fig. 10.

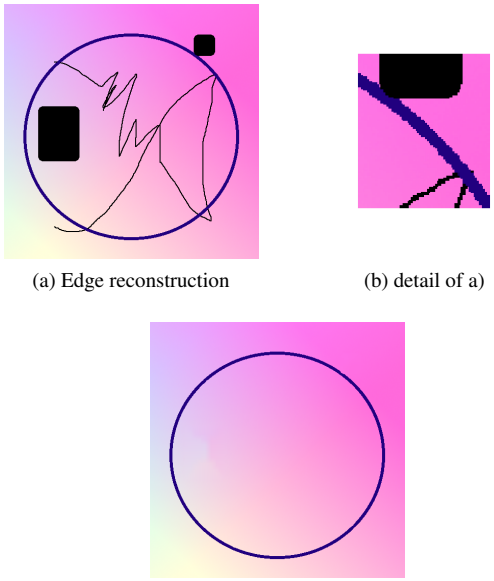
Figure 10: Mask of damaged part of the edges.

10) Use S_g as a mask and V_g as an valid pixels set for edge reconstruction. Use $S - S_g$ as a mask and V_c as a valid pixels set for reconstruction of the rest.

Because we separated edges from the rest, we can reconstruct these two parts independently. Illustration is in Fig. 11.

5.1 Examples and Comparison

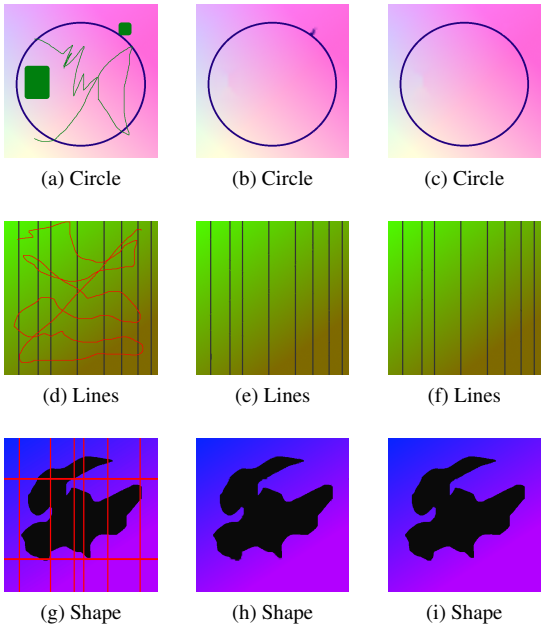
Let us illustrate the proposed inpainting algorithm side by side with original technique based on F-transform. The



(a) Edge reconstruction (b) detail of a)

(c) Reconstruction of the rest

Figure 11: a) Reconstruction of the edge; b) detail; c) reconstruction of the rest of the image I.



(a) Circle (b) Circle (c) Circle

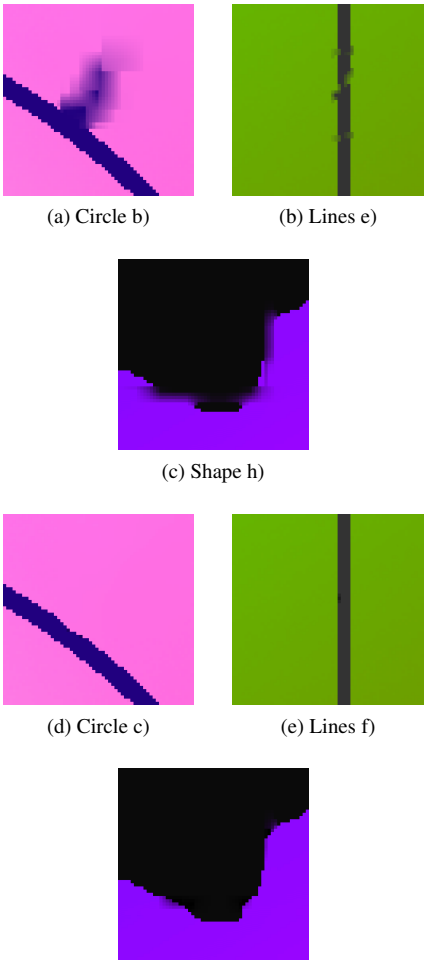
(d) Lines (e) Lines (f) Lines

(g) Shape (h) Shape (i) Shape

Figure 12: Application of our algorithm to damaged images from Fig. 3. Images a), d), g) are the damaged ones, images b), e) and h) were reconstructed using the original technique and images c), f) and i) were reconstructed using the proposed one.

images from Fig. 3 was damaged and reconstructed afterwards. Results are in Fig. 12.

Let us magnify the details to demonstrate a difference in higher resolution. In Fig. 13, the comparison is given. The original technique blurs the lines, do not follow edges



(a) Circle b) (b) Lines e)

(c) Shape h)

(d) Circle c) (e) Lines f)

(f) Shape i)

Figure 13: Details of Fig. 12.

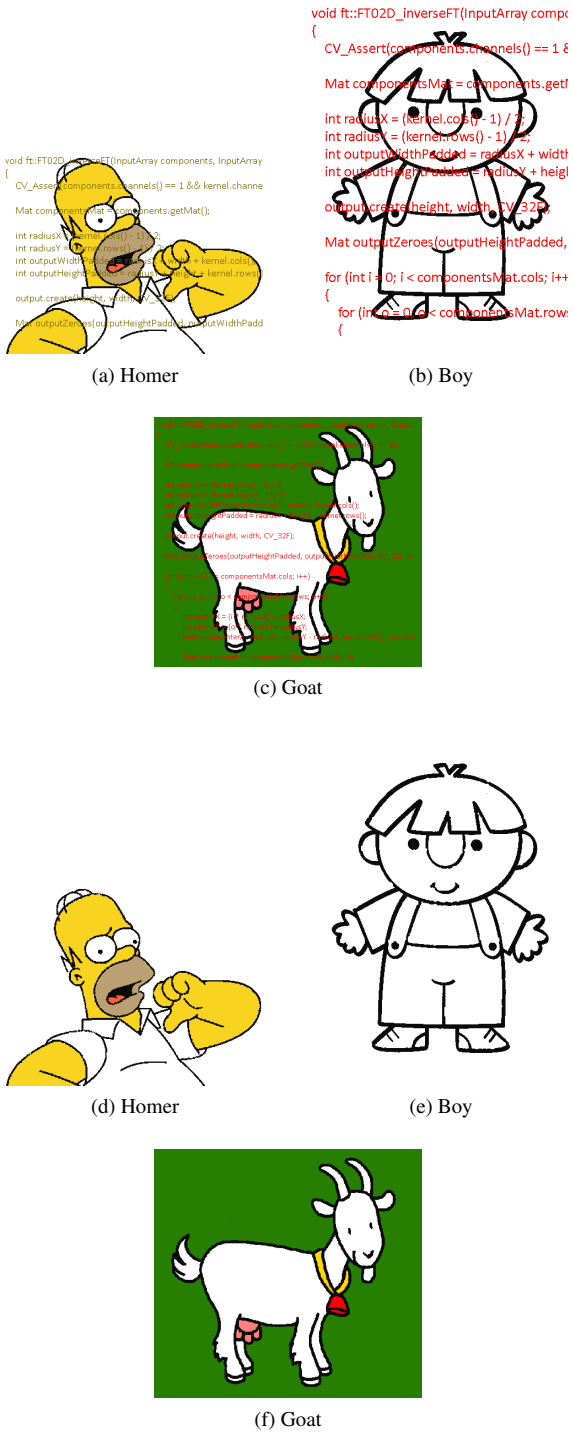
and mix colors together. Reason is the isotropic nature of the original formula. Thus for cartoon images, we propose to use different approach described in this paper.

In Fig. 14, the novel inpainting technique is illustrated on the set of cartoon images.

6 Conclusion

We propose a novel inpainting technique aimed to specific type of images. Original inpainting technique based on F-transform was applied on photos and introduced in [12].

The main idea of the novel algorithm is in division of input image and independent processing of its parts. In this introduction paper, we suggest to divide the image to two parts: *edges* and *rest*. The edges are separated using coefficients of F^1-transform. Its damaged (missing) parts are connected together using mathematical morphology. Based on that, missing parts of the edges are identified and reconstructed using inpainting technique with updated formulas. The same is applied to the rest of the image.



(a) Homer (b) Boy

(c) Goat

(d) Homer (e) Boy

(f) Goat

Figure 14: The cartoon images from testing set in Fig. 2 damaged and reconstructed.

We illustrated our technique on the two sets of images and compared with original one.

Acknowledgment

This work was supported by the project LQ1602 IT4Innovations excellence in science.

References

[1] M. Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226. ACM, 2001.

[2] C. Ballester, V. Caselles, J. Verdera, M. Bertalmio, and G. Sapiro. A variational model for filling-in gray level and color images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 10–16. IEEE, 2001.

[3] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–355. IEEE, 2001.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[5] T. F. Chan and J. Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001.

[6] J. S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 361–368. ACM Press/Addison-Wesley Publishing Co., 1997.

[7] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[8] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[9] V. Pavel and P. Irina. Interpolation techniques versus F-transform in application to image reconstruction. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 533–539. IEEE, 2014.

[10] I. Perfilieva. Fuzzy transforms: Theory and applications. *Fuzzy sets and systems*, 157(8):993–1023, 2006.

[11] I. Perfilieva, P. Hodáková, and P. Hurtík. Differentiation by the F-transform and application to edge detection. *Fuzzy Sets and Systems*, 2014.

[12] I. Perfilieva and P. Vlašánek. Image reconstruction by means of F-transform. *Knowledge-Based Systems*, 70:55–63, 2014.

[13] I. Perfilieva, P. Vlašánek, and M. Wrublová. Fuzzy transform for image reconstruction. In *Uncertainty Modeling in Knowledge Engineering and Decision Making*, Singapore, 2012. World Scientific.

[14] J. Serra. *Image analysis and mathematical morphology, v. 1*. Academic press, 1982.

[15] Vlašánek and I. Perfilieva. Image reconstruction with usage of the F-transform. In *International Joint Conference CISIS'12-ICEUTE'12-SOCO'12 Special Sessions*, pages 507 – 514, Berlin, 2013. Springer.