Image Morphing Based on Morphological Interpolation Combined with Linear Filtering

Marcin Iwanowski

Institute of Control and Industrial Electronics
Warsaw University of Technology
ul.Koszykowa 75
00-662 Warszawa POLAND
tel. +48 22 660 54 33
e-mail: iwanowski@isep.pw.edu.pl

ABSTRACT

This paper describes a novel approach to color image morphing which is based on the combination of morphological image processing tools and linear filtering. In the proposed method the morphing engine is provided by the morphological interpolation by means of the morphological median. By the successive generation of morphological medians using an algorithm proposed in the paper, the sequence transforming one input image into another one is produced. The algorithm makes use of the similarity measure between two successive frames. Two versions of the algorithm are proposed - in the first one the required parameter is the number of frames of the final sequence, in the second one the maximal acceptable error between two consecutive frames. Three linear tools are proposed to improve the visual quality of the morphing sequence: the temporal linear filtering, the spatial linear filtering and the auxiliary cross-dissolving. Contrary to the traditional approaches to image morphing, the proposed method doesn't require any control points. The human operator is obliged to introduce only a few input parameters. Two examples showing the results are also presented in the paper.

Keywords: Image morphing, color image processing, mathematical morphology, morphological interpolation

1. INTRODUCTION

Although various image processing concepts are in common use in the computer graphics [Gomes97], an application of mathematical morphology [Serra83,88] to this domain is a recent filed of research. It has proved its usefulness in various areas of image processing: image filtering, segmentation, granulometry, etc. One of the most popular fields of its applications is **image interpolation** – a process of automated creation of a sequence transforming one given image (the initial image) into the second (final) one. This kind of operation is present in computer graphics under the name of **morphing**. The application of mathematical morphology into image interpolation is a recent field of exploration [Iwano00]. The results of the very first research works have been published in 1996 and 1998 [Beuch98, Meyer96, Serra98]. They concerned the interpolation of binary, mosaic and graytone images. In [Beuch98, Serra98] an important operation was introduced: the **morphological median**. This operation - completely different from the median filter widely used in image processing - results in a new image, created on the basis of given two input images, located halfway between them. The iterative generation of median images allows producing the interpolation sequence that transforms the initial image into the final one – a ***morphing sequence***. Important is that, contrary to the well-known cross-dissolving [Gomes97, Wolbe99], the morphological approach transforms the shapes of objects on the images. Cross-dissolving produces merely a kind of blending of input images. The method proposed in [Beuch98] has been extended into color images in [Iwano99]. Two version of interpolation was proposed: the fully automatic and half-automatic interpolation. In this paper we extend the fully automatic method. It allows deforming the images in a automatic manner, without setting up the control points. The original method has a disadvantage: in some cases, the visual quality of the sequence is not high, especially when the input images include the areas of different colors on the initial and final images. Frames of the morphing sequence contain often inclusions of highly contrasted pixels with the hue and luminance values different from their background. Due to that fact the visual quality of the morphing sequence is not high enough to use it in an efficient way. In this paper the quality improvement is proposed. In order to improve the smoothness of the sequence we use the linear image processing techniques. The combination of the morphological interpolation and the quality improvement by linear spatial and temporal filtering allows performing the image morphing, which produces interesting and good-looking results.

The paper presents also a new method for the production of the morphologically interpolated sequence. The algorithm is based on the similarity measure between frames. The traditional method [Beuch98,Iwano99] is a 'blind' one – interpolated frames are produced without comparing the already interpolated ones. The new algorithm does it and is more flexible – it allows producing the interpolation sequence using different criteria, two of which are presented in the paper: the number of frames and maximum error criteria.

The paper is divided into 6 sections. Section 2 describes the morphological median of color images. New method of the interpolation sequence formation is introduced in section 3. Section 4 describes the method of the linear quality improvements. Section 5 contains the results and examples, and finally section 6 summarizes and concludes the paper.

2. COLOR MORPHOLOGICAL MEDIAN

2.1 Definition of morphological median

The morphological median has been introduced for binary [Serra98], mosaic and graytone [Beuch98] images. It is based on the morphological operations of erosion and dilation [Serra88]. ***Erosion*** is defined as a minimum operator, which assigns to every image pixel a minimum value from among their neighbors. The neighborhood is defined in mathematical morphology using a ***structuring element***. In the case considered in this paper an elementary structuring element is used. It contains the closest pixel's neighborhood. The erosion of the input image F is defined by:

$$G=\varepsilon(F)\Leftrightarrow \forall p\in \mathrm{P}: G(p)=\min_{q\in N(p)}\{F(p+q)\}-h \qquad (1)$$

where G represents the output image, ε is the erosion operator, P is an image domain, *N(p)* represents the closest neighborhood of pixel p, and h is the height of the structuring element. The operation of ***dilation*** is based on the maximum value among the neighboring pixels and is written as:

$$G=\delta(F)\Leftrightarrow \forall p\in \mathrm{P}: G(p)=\max_{q\in N(p)}\{F(p+q)\}+h \qquad (2)$$

where δ represents the operator of dilation (the rest of symbols has the same meaning as described above).

The erosion and dilation of given size n of image F are defined as, respectively:

$$\varepsilon^{(n)}(F)=\underbrace{\varepsilon(\varepsilon....\varepsilon}_{n-times}(F)..)\ ;\quad \delta^{(n)}(F)=\underbrace{\delta(\delta....\delta}_{n-times}(F)..)$$

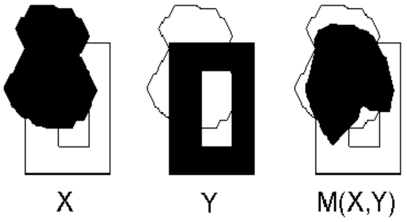
The ***median image*** [Beuch98] is defined as:

$$M(F,G)=\sup_{\forall\lambda}\left\{\inf\left[\delta^{(\lambda)}(\inf(F,G)),\varepsilon^{(\lambda)}(\sup(F,G))\right]\right\} \qquad (3)$$

where $\lambda=1,2,\ldots$ are increasing integer values. 'sup' and 'inf' symbols represent a supremum and infinimum images which are defined as, respectively:

$$G=\sup(F_1,F_2)\Leftrightarrow \forall p\in \mathrm{P}\ \ G(p)=\max\{F_1(p),F_2(p)\} \qquad (4)$$

$$G=\inf(F_1,F_2)\Leftrightarrow \forall p\in \mathrm{P}\ \ G(p)=\min\{F_1(p),F_2(p)\} \qquad (5)$$



Morphological median of binary images.
Figure 1

In case of the binary images the height of the structuring element have obviously to be $h=0$. In the binary case, however, there exists an additional condition - the intersection of the input binary images must be non-empty [Serra98,Iwano00]. The example of binary median is presented on Fig.1.

When computing the morphological median of the graytone images, the height of the structuring element $h>0$ [Beuch98,Iwano00]. Formally saying it corresponds to the operation performed with the cylindrical structuring element. On the other hand, the in case of multivalued images the condition of a non-empty intersection doesn't exist. In the color case the height is represented by a vector $h=[h_1\ h_2\ h_3]$ such that $h_1, h_2, h_3 > 0$ are – in the RGB color space - the incremental values of red, green and blue component respectively.

2.2 Median of color images

In case of graytone images the computation of (3) is obvious - the 'min' and 'max' operators in (1),(2),(4) and (5) are being computed on the scalar values. In the case of color images functions 'min' in (1) and (5) as well as 'max' and (2) and (4) must be calculated on the values from color (vector) space. In [Iwano99] a solution based on lexicographic ordering and comparative color space was proposed. In every Cartesian color space one always has to compare triples of numbers (three-element vectors) when comparing pixels. One of the most popular ways of comparing is a lexicographic ordering. It has been applied to color morphology in [Talbo98]. It is based on the successive comparisons of vectors' components beginning with components with the lowest indexes. This approach has however one important disadvantage - the *a priori* ordering of the importance of the vector components. In the case of the RGB color space, the ***r***-component is considered as more important than the ***g***-component, while the ***b***-component is the less important one. But there is no reason to apply such an order of preference. To solve this problem a new color space – the ***comparative space*** [Iwano99] - is introduced exclusively for vector comparison using the lexicographic ordering. The initial RGB color space is converted to the comparative one by using a 3x3 conversion matrix.

The conversion is based on the visual importance of color channels for the human perception. The color components are not of the same importance for the human vision. The transformation to the comparative space sorts out the color components and/or combines them by introducing their linear combinations so that the consecutive comparisons follow the visual importance of components.

Transformation matrix M of initial color space into comparative multiplies the vector of color components:

$$[v_1 \quad v_2 \quad v_3]^T = M \cdot [r \quad g \quad b]^T$$

where $[v_1 \quad v_2 \quad v_3]^T$ is a vector in the comparative vector space and $[r \quad g \quad b]^T$ is a vector in the RGB color space. Matrix M is in this paper computed according to ***lgr***-ordering [Iwano99]:

$$M = \begin{bmatrix} 0.3 & 0.6 & 0.1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

In other words, the order of comparisons is following: the luminance value, the g-value, and finally the r-component.

The approach presented above allows comparing the vectors of RGB color space while taking into account the importance of colors for human vision. Due to that fact it allows also performing the morphological operations on color images. Consequently it enables the calculation the morphological median of color images.

2.3 The algorithm

Equation (3) is applied to construct the iterative algorithm of median image calculations [Iwano99]. Starting from a pair *(F,G)* of initial images, we introduce the three auxiliary images *Z, W, M,* which are initially equal to:

$$Z_0 = \inf(F,G)\,;\, W_0 = \sup(F,G)\,;\, M_0 = \inf(F,G)$$

The indexes represent the number of iteration. Iterated values in the *i*-th iteration are computed using the following rules:

$$Z_i = \delta(Z_{i-1})\,;\, W_i = \varepsilon(W_{i-1})\,;\, M_i = \sup[\inf(Z_i, W_i), M_{i-1}]$$

Iterations are performed until idempotence of *M*, which means that when the image *M* stops to change the task is accomplished and finally:

$$M(F,G) = M_i$$

where *M(F,G)* is the morphological median of images *F* and *G*, and *i* is the lowest iteration number such that $M_i = M_{i+1}$.

The above algorithm is convergent, in a sense that the idempotence is always reached. It is guaranteed by the equation (3) and the discrete nature of digital images. The only theoretical danger for the convergence are the oscillations such that $M_i = M_{i+2} \neq M_{i+1} = M_{i+3}$. Since however that $M_{i+1} \geq M_i$, such a situation cannot happen.

Examples of color median images obtained using this algorithm are presented on Fig.2 and Fig.3. Fig.2 shows a morphological median of two images with a similar color palette containing mostly green and yellow hue values. Fig.3 contains the median of two images with different color palettes with dominant hues: red-yellow and green-blue. It's clearly visible that the median image contains the objects obtained by the shape-deformation of the objects on both input images.

3. MORPHING SEQUENCE

In the previous section the method of a single interpolated image generation was presented. This section describes how to produce the complete sequence. Traditional method of the production of the interpolation sequence [Beuch98] is based on the successive production of new medians between pairs of already generated ones. Such an approach has, however, one disadvantage. It doesn't consider the content of the image - so it is a kind of 'blind' operation. Moreover it doesn't permit obtaining the sequence of any given length. We propose here a new method, which generates iteratively the frames of the interpolation sequence – one frame per iteration. New frames are inserted in different positions in the sequence. The algorithm decides, depending on the difference between two neighboring images, whether it is necessary to generate a new median or not. This decision is based on the difference between images and is taken after calculating the similarity measure between two frames.

In the method from [Beuch98] the task was to find an equidistant distribution of interpolation levels without taking into consideration the content of the images. In our case instead of the distribution of levels, one optimizes the distribution of measures between every pair of consecutive frames.

The similarity measure is equal to an error *e*, which is computed as the mean square error (MSE) between the luminance values of pixels belonging to consecutive frames *P* and *Q*:

$$e(P,Q) = \frac{1}{x_{\max} \cdot y_{\max}} \cdot \sum_{i=0}^{x_{\max}-1} \sum_{i=0}^{y_{\max}-1} \left[lum(P(i,j)) - lum(Q(i,j))\right]^2 \tag{6}$$

where function *lum* represents a luminance value of its argument, and $x_{\max}, y_{\max}$ are the image sizes.

The algorithm makes use of two vectors. The first one is a vector of sequence frames $S = \lfloor S_0, S_1, \ldots \rfloor$, the second one is a vector of similarity measures between every pair of consecutive frames ***e***. Both vectors have the same number of elements. The input images are *X* and *Y*. The temporary variable *i* (counter) is equal to the number of already produced frames.

The algorithm of morphing sequence production:

```
• Let: i = 1 (start the counter)
• Let: $S_0 = X$ , $S_1 = Y$
• Compute the error between the input
  images: $e_0 = e(S_0, S_1)$
• Let: imax = 0 (index of the highest
  error value)
• While (not(stop-condition)) do:
     1. Insert new, empty frame
        between frames imax and imax+1
        (elements of vectors S and e
        with indexes between 0 and
        imax remains unchanged;
        elements with indexes from
        imax + 1 to i get new indexes
        from imax + 2 to i + 1
        respectively, index of a new
        element is imax+1)
     2. Calculate new median:
        $S_{i\max+1} = M(S_{i\max}, S_{i\max+2})$
     3. Let: $e_{i\max} = e(S_{i\max}, S_{i\max+1})$
     4. Let: $e_{i\max+1} = e(S_{i\max+1}, S_{i\max+2})$
     5. Let: i= i + 1
     6. Find imax such that:
        $e_{i\max} = \max\{e_0, e_1, \ldots, e_{i-1}\}$
• Stop the algorithm
```

The final sequence of images $S = \lfloor S_0, S_2, \ldots, S_{i-1} \rfloor$ is the morphologically interpolated sequence, such that $S_0 = X$, $S_{i-1} = Y$; frames S_1 , S_2 ,… , S_{i-2} represent the interpolated images.

The use of the structuring element of a given height *h>0* during the calculations of single medians, guarantees the convergence of the above algorithm. It means that when computing $S_k = M(S_{k-1}, X)$, after certain number of iterations $S_k = S_{k-1} = X$. The same result would be obtained if, instead of the initial image *X*, the final one – *Y*, would be used.

Two versions of the algorithm are proposed. The difference between them lies in the stop-condition. First one generates the sequence with ***given number of frames*** $n \geq 2$. The algorithm's *stop-condition* in this case is: ($i \geq n-1$).

In the second version is based on ***the highest acceptable error*** e_{MAX} between two consecutive frames. The number of frames of the final sequence can vary depending on the complexity and difference between input images. The iterations are performed until the error (6) becomes smaller than e_{MAX} . In this case the *stop-condition* is: ($e_{i\max} < e_{MAX}$). The final value of *i* indicates the number of frames of the interpolation sequence.

Depending on the version of the algorithm applied either the required number of frames *n*, or the value of e_{MAX} should be given.

4. QUALITY IMPROVEMENTS

The biggest problem that occurs, while observing the frames of the morphologically interpolated sequence, are the inclusions of highly contrasted groups of pixels. They make the sequence look unnatural and disturb its visual quality by decreasing the spatial smoothness of the sequence frames. The next problem lies in the temporal smoothness of a sequence. It happens that the transitions between the frames of the sequence sometimes seem abrupt. The solution of both problems is based on the linear filtering. It allows improving both the spatial and the temporal smoothness of a sequence. We propose three quality-improving tools that can be applied either separately or jointly depending on the particular demand. First two of them are the linear filters. The reason for introducing the linear filtering is not to remove the inclusions, but to soften theirs color value and to obtain smoother transitions between sequence frames. The third tool is an auxiliary cross-dissolving operation that improves the temporal smoothness of the sequence.

4.1 Temporal filtering

The temporal filtering allows reducing the contrast of the image pixels. The temporal filter of size 1 is given by:

$$S_t'(x,y) = \begin{cases} S_t(x,y) & if \quad t = 0 \\ \dfrac{S_{t-1}(x,y) + \lambda \cdot S_t(x,y) + S_{t+1}(x,y)}{\lambda + 2} & if \quad 0 < t < n-1 \\ S_t(x,y) & if \quad t = n-1 \end{cases} \tag{7}$$

where $S_i(x,y)$ is a pixel on the *i*-th frame of the initial sequence, $S_i'(x,y)$ is pixel on the filtered frame, *n* is a total number of frames of the morphologically interpolated sequence.

It is a weighted mean value - the weights of the pixels from the previous and the next frame are set to 1. The weight of the current pixel λ can be chosen manually depending on the visual quality of the sequence. Since the first and the last sequence frame are equal to two initial frames, they are not filtered.

The parameter λ controls the influence of the current frame on the filtered one. It can be equal to any positive integer. The growth of this parameter reduces however the effect of filtering.

The application of temporal filtering softens the color values of inclusions without introducing the spatial blurring. It allows reducing the contrast of the inclusions. It results in more natural visual effect and smoothes out the temporal transitions between frame pixels.

The temporal filtering offers the best results for the inclusions present at the current frame being at the same time absent on the preceding. In such a case an effect of blending is visible and contrast reduction is remarkable. On the other hand when small inclusions are not growing on several consecutive frames, they start to be too contrasted again after few successive frames. Such kind of inclusions cannot be filtered using the temporal filtering.

The example of temporal filtering is shown on Fig.4b. It contains the result of the temporal linear filtering (with λ*=1*) of image on Fig.4a.

4.2 Spatial filtering

Another kind of linear filtering proposed is the spatial filtering. It reduces the visual impact of the inclusions by introducing the spatial blurring.

The linear spatial filtering is expressed by using the following equation:

$$S_t'(x,y) = \frac{\sum_{i=-s}^{s}\sum_{j=-s}^{s} S_t(x+i, y+j) \cdot m(i+s+1, j+s+1)}{\sum_{i=1}^{2s+1}\sum_{j=1}^{2s+1} m(i,j)} \tag{8}$$

where $S_i(x,y)$ is a pixel on the *i*-th frame of the initial sequence, $S_i'(x,y)$ is a pixel on the filtered one, *s* is a size of a linear filter, and *m* is a mask of this filter represented by a matrix of size *2s+1*. Similarly to the case of the temporal filtering, in the current one we also filter only the morphologically interpolated frames i.e. frames with the indexes: $0 < t < n-1$. The first and the last frame of the sequence are simply copied from the initial sequence to the filtered one.

Various masks can be applied to perform the spatial filtering. The following three masks of size *s=1* have been considered:

$$m_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad m_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad m_3 = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{9}$$

Mask m_1 represents the mean filter *3x3*, mask m_2 – its weaker version and mask m_3 – the Gaussian filter.

This kind of filtering is necessary when the case, mentioned at the end of the preceding section, happens. It occurs when the contrast of the inclusions is too high and, at the same time, the inclusions have similar shape on several consecutive frames. In such a case the effect of locally high contrast cannot be removed by temporal filtering and the spatial one must be applied.

The choice of filter depends on the visual quality of the final sequence. If one prefers to get a sequence more “mysterious” one may use stronger filters like e.g. m_1. If one wants to have sharper interpolated images weaker filter like e.g. m_2 is more suitable.

The example of spatial filtering is presented on fig.4c. The filtering (8) has been performed using filter m_2 from (9). After both kinds of filtering the artifacts caused by the morphological median (clearly visible on Fig.4a), has been either removed (by temporal filtering) or softened (by the spatial one).

4.3 Cross-dissolving

In order to make the transitions between morphologically interpolated frames smoother, the additional cross-dissolving is applied. This operation can be written using the following equations:

$$\begin{aligned} &S'_{\alpha t}(x,y) = S_t(x,y), \\ &S'_{\alpha t+\beta}(x,y) = (1-\tfrac{\beta}{\alpha})\cdot S_t(x,y) + \tfrac{\beta}{\alpha}\cdot S_{t+1}(x,y), \qquad (10) \\ &S'_{\alpha(t+1)}(x,y) = S_{t+1}(x,y) \\ &0 < \beta < \alpha;\ 0 \le t < n-1 \end{aligned}$$

where α is given number of new frames included between every pair of consecutive sequence frames, *S’* is a frame of cross-dissolved sequence, *S* is a frame of the initial one.

After the cross-dissolving operation the length of a sequence *S’* grows $\alpha+1$ times by inserting α cross-dissolved frames between every pair of consecutive frames of the initial – morphologically interpolated and linearly filtered – sequence *S*. Cross dissolving is especially useful in the transformation of the initial, given image to the closest morphologically interpolated one. It allows human perception to adapt to the interpolated image. An example of the cross-dissolving is presented on Fig.5a-5g. Fig.5a contains the input, given image. Fig.5g – the first frame of the morphologically interpolated sequence. Fig.5b-5f show the cross-dissolved frames (α=5).

5. RESULTS

Two morphing sequences are presented as an example. Both have been obtained using the proposed method. All three proposed improvements have been applied: temporal filtering, spatial filtering and cross-dissolving. The first sequence contains the transformation of an image of flowers into another one presenting a bottle of beer. Both images have similar color palettes. Fig.6 presents the frames of a sequence, which were produced using the morphological interpolation and the linear filtering. The morphological median has been calculated using the dilations (1) and erosions (2) using a structuring element of height *h=[1,1,1]*. The sequence was generated using the second rule. The given maximal error was equal to $e_{MAX} = 1000$. It resulted in 8 morphologically interpolated frames. Each interpolated frame was filtered using the temporal filter (7) with λ=1 and spatial filter (8) with mask m_2 from (9). The error values and order of production of new medians is shown in Table 1. To create the final sequence the cross-dissolved frames was inserted between every pair of consecutive frames according to (10). The number of cross-dissolved frames was α=10.

The second example shows the transformation of the initial image presenting a tree in summer into the final image with a forest in autumn. The color palettes in both input images are different. Fig.7 presents the frames of a sequence, which were produced using the morphological interpolation and the linear filtering. The morphological median has been using a structuring element of height *h=[1,1,1]*. The given length of a sequence was: *n=13*. Each frame was filtered using the temporal filter (7) with λ=1 and spatial filter (8) with mask m_3 from (9). Fig. 4 shows the result of cross-dissolving produced after the construction of morphological interpolation followed by temporal and spatial filtering. In this case the number of cross-dissolved frames inserted between every pair of consecutive ones was α=5.

Frame	**Prod. order**	**Error**
S_0	Initial	$e(S_0, S_1)=998$
S_1	7	$e(S_1, S_2)=182$
S_2	6	$e(S_2, S_3)=262$
S_3	4	$e(S_3, S_4)=425$
S_4	2	$e(S_4, S_5)=760$
S_5	1	$e(S_5, S_6)=820$
S_6	3	$e(S_6, S_7)=601$
S_7	5	$e(S_7, S_8)=455$
S_8	8	$e(S_8, S_9)=883$
S_9	Final	

First example - list of errors
Table 1

6. CONCLUSIONS

The method of automatic morphing has been presented. It combines the morphological interpolation and linear filtering. The method is able to produce the interpolation sequence using exclusively the image processing tools. Starting from two initial images the morphing sequence transforming the initial image into the final one is created. The transformation is produced without applying the control points. The only human assistance is required in order to indicate the number of morphologically interpolated frames and - if needed - the parameters for the linear filtering tools (number of cross-dissolved frames as well as the type of temporal and spatial linear filter).

The proposed method can be compared with another automatic method: the ‘pure’ cross-dissolving. Contrary however to the last one it contains shape metamorphosis of the objects on the image. In comparison on the other hand with the classic morphing techniques [Wolbe99], it doesn’t require an introduction of the control points, which is time-consuming and manual process. But, in fact, it isn’t a real competitor of classic morphing techniques. Both methods are complementary and their usage depends on the images one is dealing with. The proposed method can be employed to transform the images without sensitive areas, transformation of which have to be controlled precisely, like eyes, mouth, nose etc. in the human face morphing. The method can be successfully applied to the transformation of images where the correspondence of areas on both images is not crucial. It is useful, in particular, when at least one of two images contains a lot of small details (like flowers or trees on examples shown in the paper), which should be deformed - not blended (blending could be obtained using the simple cross-dissolving). The deformation of such images using traditional mesh-warping would require a huge amount of control points which have to be indicated manually by the operator. The proposed method deforms them without introducing the control points.

The results, obtained using the proposed method, are interesting also from the artistic point of view – the transformation of the shape looks “mysterious”. The method can be successfully applied to the production of special visual effects in TV, film and multimedia industry.

7. REFERENCES

[Beuch98] Beucher S.: Interpolation of sets, of partitions and of functions In: H.Heimans and J.Roedink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing.* Kluwer, 1998

[Gomes97] Gomes J., Velho L.: *Image Processing for Computer Graphics,* Springer-Verlag, 1997

[Iwano99] Iwanowski M., Serra J.: Morphological Interpolation and Color Images *Proc. of 10th International Conference on Image Analysis and Processing* Sept. 27-29, 1999 Venice, Italy; IEEE Computer Society

[Iwano00] Iwanowski M.: Application of mathematical morphology to interpolation of digital images, Ph.D. thesis *Warsaw University of Technology, School of Mines of Paris,* Warsaw-Fontainebleau 2000

[Meyer96] Meyer F.: Morphological interpolation method for mosaic images, In P.Maragos, R.W.Schafer, M.A.Butt *Mathematical morphology and its application to image and signal processing,* Kluwer, 1996.

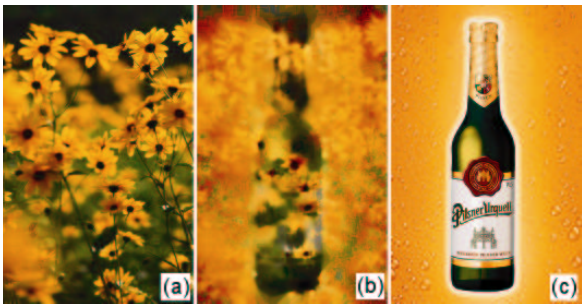
[Serra83] Serra J.: *Image Analysis and Mathematical Morphology vol. 1.* Academic Press, 1983

[Serra88] Serra J.: *Image Analysis and Mathematical Morphology vol. 2.* Academic Press, 1988

[Serra98] Serra J.: Hausdorff distance and interpolations In: H.Heimans and J.Roedink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing.* Kluwer, 1998

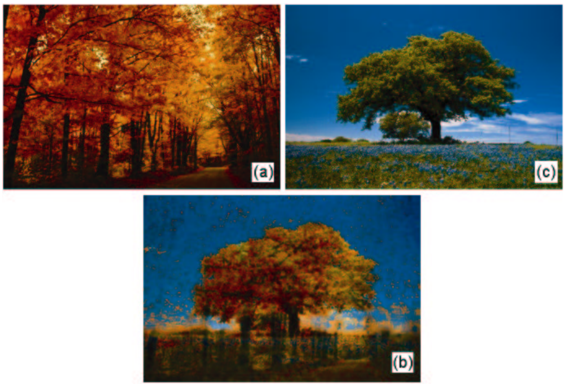
[Talbo98] Talbot H., Evans C., Jones R.: Complete ordering and multivariate morphology In: H.Heimans and J.Roedink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing.* Kluwer, 1998

[Wolbe99] Wolberg G.: *Digital Image Warping,* IEEE Computer Society Press, Los Alamos CA, 1999



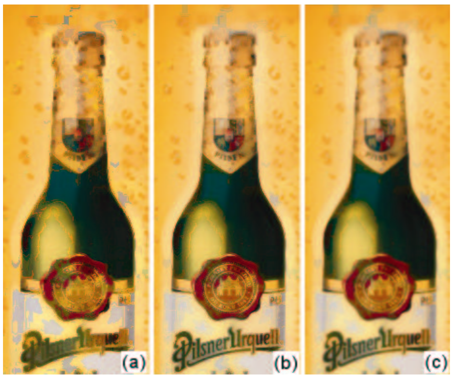
Two initial images (a) and (b) containing colors of similar hue values (green and yellow) and their morphological median (c).

Figure 2



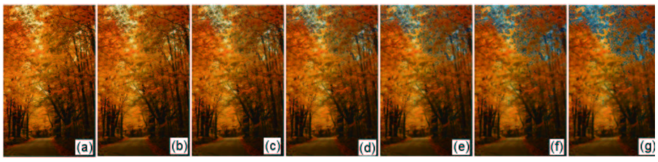
Initial images (a) and (c) containing colors of different hue values and their median image (b).

Figure 3



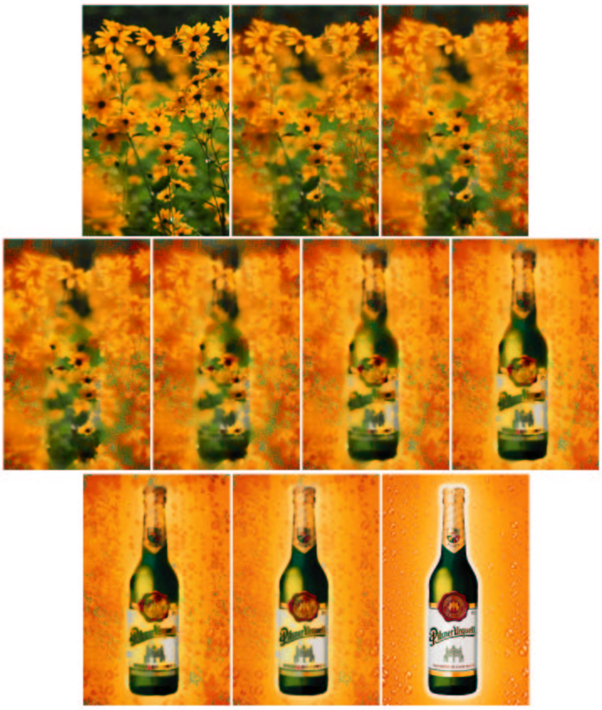
Morphological median – frame of the morphing sequence (a); after temporal linear filtering (b); and after spatial and temporal filtering (c).

Figure 4



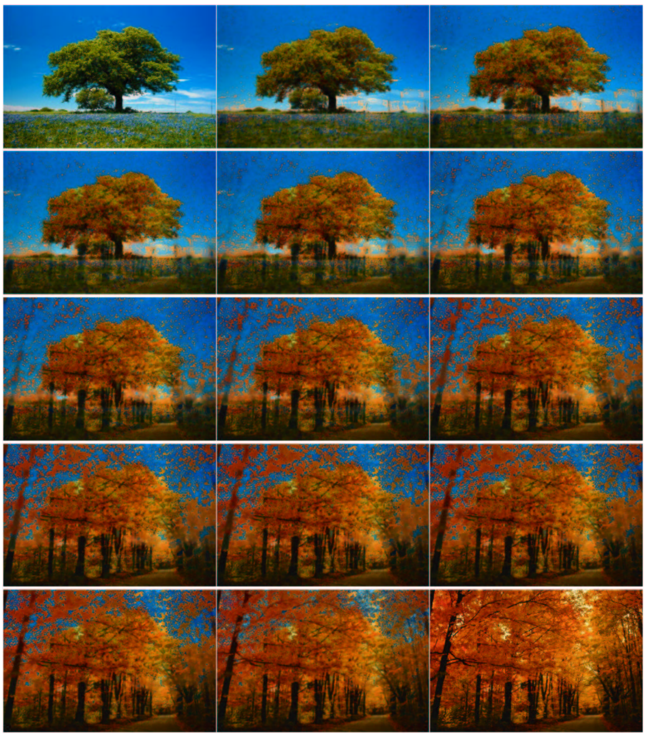
Cross-dissolving

Figure 5



First example

Figure 6



Second example

Figure 7