Audio-aware on-the-fly Animation from Single Photo

Konstantin Kryzhanovsky, Aleksey Vil'kin, Ilia Safonov, Zoya Pushchina
Samsung Moscow Research Center, Russia
{k.kryzhanovs, a.vilkin, ilia.safonov, p.zoya} at samsung.com

Abstract

In this paper we propose new approach of automatic generating **real time content adaptive** animation effects from the still images adapted for the low-powerful embedded HW platforms. Displayed animation behaves uniquely each time it's played back, and does not repeat itself during playback duration, creating vivid and lively impression for the viewer. Adaptation of the effect parameters according to background audio greatly increases aesthetic impression of the viewer. Three animation effects such as *Flashing Light*, *Soap Bubbles* and *Sunlight Spot* are described in details. We propose several ways of controlling the effect parameters by music. User opinion survey demonstrates that majority of users are excited by such effects and wants to see them in their devices with multimedia capability.

Keywords: *animation from photo, audio-adaptive effect, multimedia slide-show, attention zones detection.*

1. INTRODUCTION

Creation and sharing of multimedia presentations and slideshows has become a pervasive activity. The development of tools for automated creation of exciting, entertaining and eye-catching photo transitions and animation effects, accompanied by background music and/or voice comments, has become a modern trend [1]. One of the most impressive effects is the animation of still photo, for example, grass swaying in the wind or rain drop ripples in the water, etc.

Special interactive authoring tools, such as Adobe After Effects and Ulead Video Studio, are used to create animation from an image. Development of fast and realistic animation effects is hard task itself; and it is a topical problem of modern computer graphics. For example, paper [2] discusses algorithm for generation of plausible motions animation. In authoring tools the effects are selected and adjusted manually, which may require considerable effort from a user. Resulting animation is saved as video clip, thus requiring noticeable amount of space for storage. During playback, such movie will always be the same, thus leading to repetitiveness feeling of the viewer.

For the multimedia presentations and slideshows it's preferable to generate animated effects on-the-fly with a high frame rate. Very fast and efficient algorithms are necessary to provide required performance. It's especially difficult for low-powerful embedded HW platforms.

Our research was defined as development and implementation of automatically generated animated effects of Full HD images on ARM Cortex A8 and A9 – based embedded platforms, with the CPU frequency 800 - 1000 MHz and without use of GPU – based APIs, such as OpenGL. Only ARM commands were available to use, including SIMD instructions of ARM NEON co-processor. Creation of realistic and complex animated effects in such limited conditions is a challenging task itself, particularly for the well experienced in computer games on powerful PCs and play stations users.

We have developed several algorithms for generation of content-based animation effects from still images, such *as Flashing Light*, *Soap Bubbles*, *Sunlight Spot*, *Magnifier Effect*, *Rainbow*, *Portrait Morphing Transition Effect*, *Snow*, *Rain*, *Fog*, etc. For those effects we propose a new approach of automatic audio-aware animation generation.

In the paper we demonstrate our concept, i.e. adaptation of effect parameters according to background audio, for three effects: *Flashing Light*, *Soap Bubbles* and *Sunlight Spot*. Obviously the concept can be extended to other animated effects.

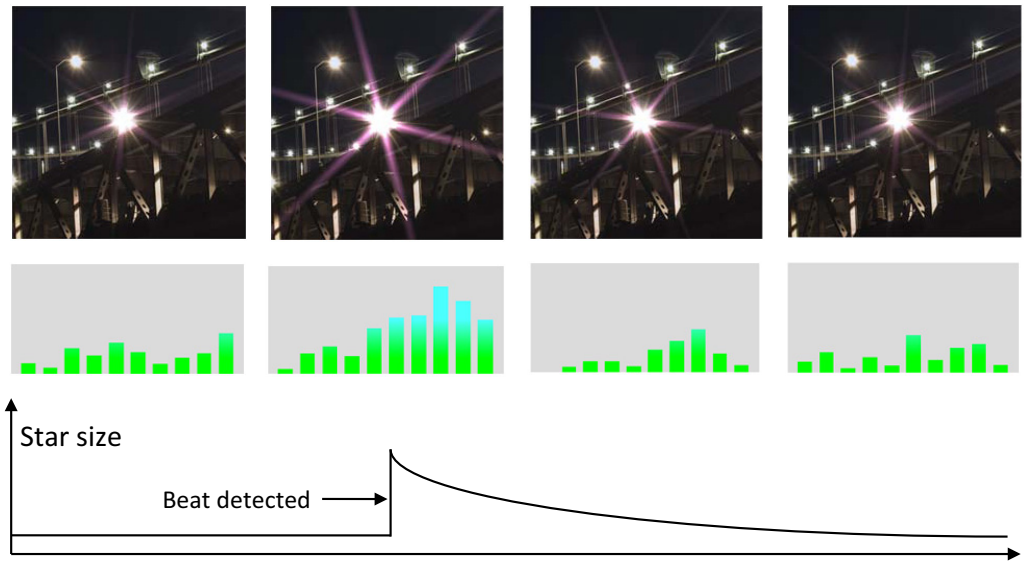


Figure 1. Detected beats affect size of flashing light.

2. RELATED WORKS

Recently, some content-adaptive automatic techniques for generation of animation from static photo were proposed. Paper [3] describes Animated Thumbnail which is a short looped movie demonstrating main objects of the scene in sequence. Animation simulates camera tracking-in, tracking-out and panning between detected visual attention zones and whole scene.

Music plays an important role in multimedia presentations. There are some methods towards to aesthetical audiovisual composition in slideshow. Tiling Slideshow [4] describes two methods for analysis of background audio in order to select timing for photos and frames switching. First one is beats detection. Second one is energy dynamics, calculated using root mean square values of adjacent audio frames.

Also there are other concepts of combining audio and visual information with the automatic generation of multimedia presentations exists. For example, paper [5] suggests approach that focuses on an automatic sound track selection. The process attempts to comprehend what the photos depict and try to choose music accordingly.

3. ANIMATION EFFECTS FROM SINGLE IMAGE

3.1 Flashing Light

The *Flashing Light* effect displays several flashing and rotating colored light stars over the bright spots on the image. In this effect, size, position and color of flashing light stars are defined by detected position, size and color of the bright areas on the source still image.

Algorithm performs the following steps to detect small bright areas on the image:

- calculating the histogram of luma channel of the source image
- calculating segmentation threshold as luma level corresponding to specified fraction of brightest pixels of the image using the luma histogram;
- segmenting source image by thresholding; while thresholding, the majority morphological filter is used to filter out localized bright pixel groups;
- calculation of the following features for each connected region of interest (ROI):
  a. Mean color C_{mean}
  b. Centroid (x_c, y_c)
  c. Image fraction F – fraction of the image area, occupied by ROI;
  d. Roundness – relation of the diameter of the circle with same area as ROI to maximum dimension of the ROI:

$$K_r = \frac{2\sqrt{S/\pi}}{\max(W,H)},$$

  where S is the area of the ROI and W, H are ROI bounding box dimensions;
  e. Quality – integral parameter, characterizing the possibility of to ROI to be a light source and calculated as following:

$$Q_L = w_{Y\max} \cdot Y_{\max} + w_{Ymean} \cdot Y_{mean} + w_R \cdot K_r + w_F \cdot K_F;$$

where $Y_{\max}$ - maximum luma of the ROI,

Y_{mean} - mean luma of the ROI,

K_F - coefficient of ROI size,

$$K_F = \begin{cases} F/F_0, & if\ F \le F_0 \\ F_0/F, & if\ F > F_0 \end{cases}$$, where F_0 - image fraction normalizing coefficient for an optimal lightspot size;

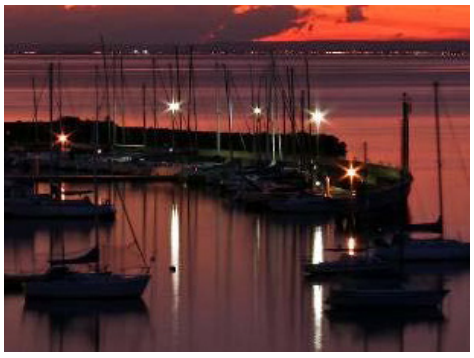
$w_{Y\max}$, w_{Ymean}, w_R, w_F - weighting coefficients.

Weighting coefficients w and optimal lightspot size normalization coefficient F_0 are obtained by minimizing differences between automatic and manual light sources segmentation results.

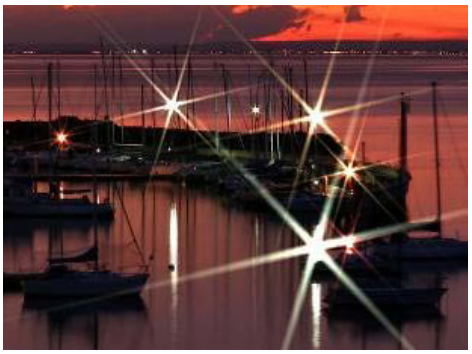
- selection of regions with appropriate features.

  All bright spots, with image fraction falling within appropriate range (F_{min}, F_{max}), and roundness Kr is larger than certain threshold value K_r^0 are considered as potential light sources. Potential light sources are sorted by their quality value Q_L. Specified number of light sources with the largest quality is selected as final positions of "light stars" objects.

Centroids of selected light regions are used as star positions. Star size is determined by dimensions of appropriate light region. Mean color of the region determines the color of the light star. Fig. 2. shows an image with bright spots and corresponding light stars.



a)



b)

Figure 2. a) Bright spots on the image and b) corresponding light stars.

Every light star is composed from bitmap templates of two types, representing star shape elements: halo shape and star ray (or spike) shape. These templates are alpha maps scaled independently. Examples of templates are shown on Fig. 3.

During rendering, the alpha map of complete star of appropriate size is prepared in separate buffer, and then the star is painted with appropriate color with transparency value extracted from star alpha map.

During animation, light star sizes and intensities are changed gradually and randomly to make an expression of flashing lights.

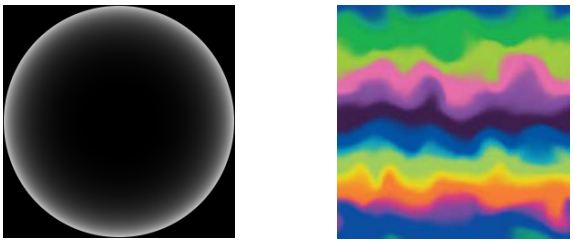
a) b)

Figure 3. Light star shape templates: a) Halo template; b) Ray template.

3.2 Soap Bubbles

The effect displays soap bubbles moving over the image. Each bubble is composed from color map, alpha map and highlight map. The set of highlight maps with the different highlight orientation is precalculated for each bubble. Highlight position depends on lighting direction in corresponding area of the image. Lighting gradient is calculated using downscaled brightness channel of the image.

Fig. 4 shows Soap Bubble components. Color map is modulated with highlight map, selected according average lighting direction around the bubble, and then combined with source image using alpha blending with bubble alpha map.



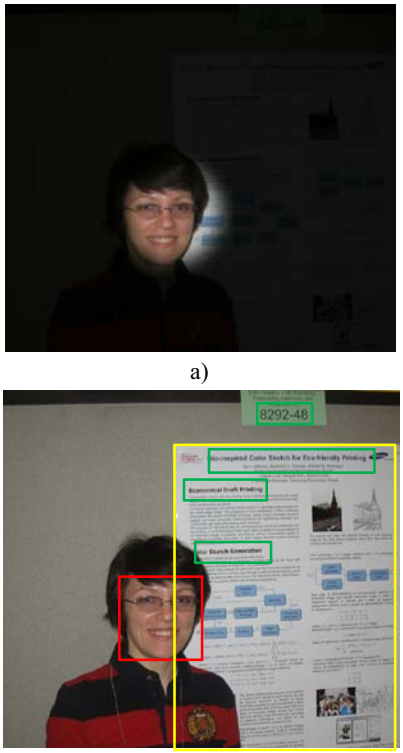
a) b)

Figure 4. Soap Bubble components: a) Alpha map b) Color map.

During animation, soap bubbles are moved smoothly over the image from bottom to top or vice versa while oscillating slightly in horizontal direction to make an impression of real soap bubbles floating in the air.

3.3 Sunlight Spot

The effect displays bright spot moving over the image. Prior starting effect, the image is dimmed according to its initial average brightness. Fig. 5 shows an image with sunlight spot effect. The spotlight trajectory and size are defined by attention zones on the photo.



a) b)

Figure 5. a) Frame of *Sunlight Spot* effect; b) Selected attention zones.

Similar to many existing publication we find human faces and salient regions using pre-attentive vision model. Basing on these regions we form attention zones. In addition we consider text inscriptions as attention zones too. For example it can be the name of hotel or town on the background of which the photo was made. Or, in case of the newspaper, it will be headlines.

Well-known OpenCV software library contains implementation of face detection for front and profile faces. In general the technique that is based on state-of-the-art Viola-Jones face detector [6] provides good results. However it gets a lot of false positives. The number of false positives can be decreased with additional skin tone segmentation and processing of downsampled image [7]. We have ported OpenCV 2.3 to our embedded platform. Time of face detection for 480x320 images is about 0.8 s.

So far the universal model of human vision does not exist, but pre-attentive vision model based on feature integration theory is well-known. Since in this case, the observer is on attentive stage while viewing photo, a model of human pre-attentive vision is not strictly required. However existing approaches for the detection of regions of interest are based on saliency map and they often provide reasonable outcomes, whereas the use of attentive vision model requires too much prior information about the scene and it is not generally applicable. Classical saliency map building algorithms like [9] have a very high computational complexity. That is why researchers recently devote a lot of efforts to develop fast saliency map creation techniques. Paper [10] compares several modern algorithms for salient regions detection. We implemented on our embedded platform *Histogram- based Contrast* (HC) method. The time of salient regions detection for 480x320 images is about 0.1 s.

While developing the algorithm for detection of areas with text, we take into account the fact that text components are ordered the same way and are similar in texture features, color. Firstly, we apply LoG edge detector and restore missing parts using combination of morphological operations. After edge detection we end up with many circuits, which can be ordered as connected tree of objects and voids

Then we filter resulting connected components based on the analysis of the texture features. We use features from [8], as well as analysis of geometric dimensions and relations. We merge closely located connected components, arranged the same order and similar in color, texture features, in groups. Then we classify resulting groups. We form final zones with the text on the basis of groups that are classified as text. Time of text regions detection for 480x320 images is about 0.5 s.

Fig. 5 shows detected attention zones. Red rectangle depicts face detected; green rectangles denote text regions; yellow is bounding box of the most salient area according to HC method.

4. ADAPTATION TO AUDIO

What animation parameters may depend on characteristics of background audio signal? Firstly it is size and intensity of animated objects, also speed of their movement and rotation can be adjusted. In addition, we investigated the question: How can we change color of animate objects, depending on music? Famous Russian composer and pianist Alexander Scriabin about 100 years ago proposed a theory of connection between music and color. Colors corresponding to notes are shown on fig. 6. This theory connects major and minor tonality of the same name.

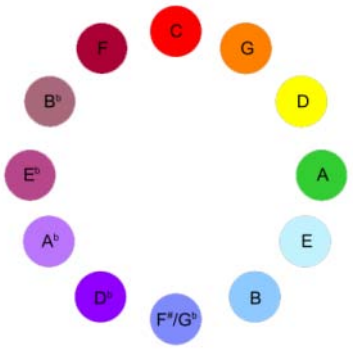


Figure 6. Accords with the circle of fifths corresponding to Scriabin's theory.

On our platform we work with stereo audio signal on frequency 44 kHz. We consider 4 approaches to connect animation of 3 effects mentioned above with background audio. In all approaches we analyze the average of two signal channels in frequency domain. The spectrum is built 10 times per second for 4096 samples. Spectrum is divided into several bands as in conventional graphic equalizer. The number of bands depends on approach selected.

For fast Fourier transform computing with fixed point arithmetic we use *kiss_fft* library. It is open source library distributed under BSD license. This library does not use platform-specific commands and is easily ported to ARM. On our platform processing time for one buffer is about 0.004 s.

Our first approach of visualizing music by colors was inspired by Luke Nimitz demonstration of "Frequency spectrograph – Primary Harmonic Music Visualizer". It is similar to Scriabin idea. It can be considered as a specific visualization of the graphic equalizer. In this demonstration music octaves are associated with HSL color wheel as shown in fig. 7 using statement:

$$Angle = 2\pi \log_2(\frac{f}{c}),$$

where f is frequency, c is origin on frequency axis. Angle defines hue of current frequency.

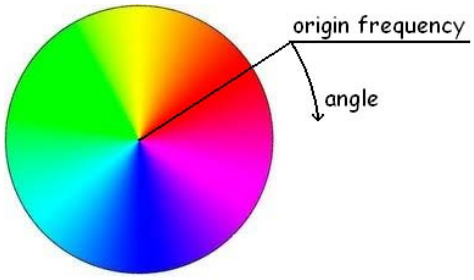


Figure 7. Color circle corresponding to each octave.

Depending on value of current note we define brightness of selected hue and draw it on color circle. We use three different approaches to display color on the color wheel: paint sectors, paint along radius or use different geometric primitives inscribed into the circle.

In *Soap Bubbles* effect, depending on generated color circle, we determine color of bubble texture. On fig. 8 there is an example of soap bubbles with color distribution depending on music. In *Sunlight Spot* effect generated color circle determines distribution of colors on highlighted spot (fig. 9).

Figure 8. Generated color distribution of soap bubbles depending on music.

Figure 9. Generated color distribution of sunlight spot depending on music.

In second approach we detect beats or rhythm of the music. We tried several techniques for beats detection in time and frequency domains [11, 12, 13, 14]. We faced constraints due to real-time performance limitation and we were dissatisfied with the outcomes for some music genres. Finally we assume that the beat is present if there are significant changes of values in several bands. This method meets performance requirements with acceptable quality of beats finding. Fig. 1 illustrates how detected beats affect size of flashing light. If the beat is detected we instantly maximize size and brightness of lights and then they gradually return to their normal state until next beat happens. Also it is possible to change flashing lights when beat happens (turn on and off light sources). In the *Soap Bubbles* effect we maximize saturation of the soap bubble color when the beat takes place. We also change the direction of moving soap bubbles as beat happened. In *Sunlight Spot* effect if the beat is detected we maximize brightness and size of spot and then they gradually returned to their normal state.

In third approach we analyze presence of low, middle and high frequencies in audio signal. This principle is used in color music installations. In *Soap Bubbles* effect we assign frequency range for each soap bubble and define its saturation according value of corresponding frequency range. In *Flashing Light* effect we assign each light star to its own frequency range and define its size and brightness depending on value of the frequency range. On fig. 10 you can see how presence of low, middle and high frequencies affect on flashing lights.

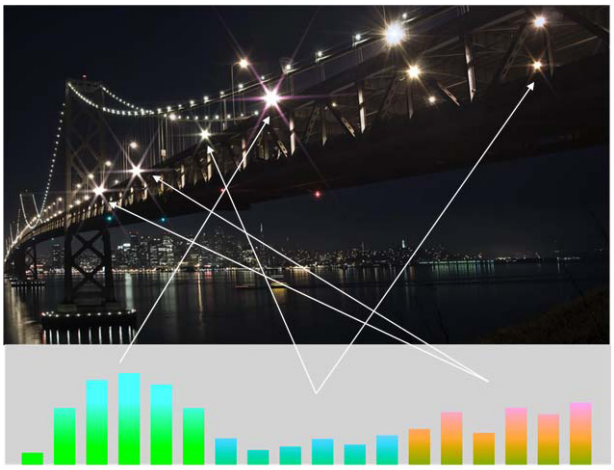


Figure 10. Low, middle and high frequencies affect on brightness and saturation of corresponding flashing lights.

Another approach is not to divide spectrum to low, middle and high frequencies but rather to assing it to different tones inside octaves. So, we work with equalizer containing large amount of bands, where each octave have enough corresponding bands. We accumulate values of each equalizer band to buffer cell, where corresponding cell number is calculated using the following statement:

$$num = \frac{(\log_2(\frac{f}{c}) \times 360) \mod 360}{\frac{360}{length}} + 1,$$

where f is frequency, c is origin on frequency axis, *length* is number of cells.

Each cell controls behavior of selected objects. In *Soap Bubbles* effect we assign each soap bubble to corresponding cell and define its saturation depending on the value of the cell. In *Flashing Light* effect we assign each light to corresponding cell and define its size and brightness depending on the value of the cell.

Obviously, other approaches to adapt behavior of animation to the background audio are also possible. In particular, it is visible to analyze the left and right audio channels separately and apply the different behavior to the left and right sides of the screen, respectively. Other effects friendlier for music adaption can be created.

5. RESULTS AND DISCUSSION

The major issue is how can we implement the functions in modern multimedia devices for real-time animation? The algorithms were optimized for ARM Cortex A8 and A9 – based platforms with CPU frequency 800 - 1000 MHz. Limited computational resources of the target platform combined with absence of graphics hardware acceleration is serious challenge for implementation of visually rich animation effects. Therefore comprehensive optimization is required to obtain smooth framerates. Total performance win is 8.4 times in comparison to initial implementation. The most valuable optimization approaches are listed in table 1.

Table 2 contains performance data for described effects. Such figures provide smooth and visually pleasant animation.

TABLE 1 OPTIMIZATION APPROACHES

Approach	*Speeding-up, times*
Fixed-point arithmetic	4.5
SIMD CPU instructions (NEON)	3
Effective cache usage	1.5
Re-implementing of key glibc functions	1.25

TABLE 2 PERFORMANCE OF PROPOSED EFFECTS FOR HD PHOTO.

Effect	*Initialization time, s*	***FPS***
Flashing Light	0.15	20
Soap Bubbles	0.08	45
Sunlight Spot	1.4	50

As objective evaluation of the proposed audiovisual presentation is difficult, we evaluate the advantage of our technique through subjective user opinion survey. *Flashing Light*, *Soap Bubbles* and *Sunlight Spot* effects with octave based audio adaptation were used for demonstration. Two questions were asked for three audio-visual effects:

- Are you excited by the effect?
- Would you like to see that effect in your multimedia device?

23 observers participated in the survey. Diagram on fig. 11 reflects survey results. In general, absolute majority of the interviewees rates effects positively. Only two people said that they do not like not only demonstrated effects, but any multimedia effects. Some observers stated: it's entertaining, but I cannot say

"I'm excited", because such expression would be too strong. Several participants of the survey said that they do not like photos or background music used for demonstration. It is also worth to notice that 8 of the respondents were women and, on average, they rated the effects much higher than men.

So we can claim that the outcomes of subjective evaluation demonstrate the satisfaction of the observers with this new type of audiovisual presentation, because audio-aware animation behaves uniquely each time it is played back, and does not repeat itself during playback duration, thus creating vivid and lively impression for the observer. A lot of observers were excited by the effects; and they want to see such features in their devices with multimedia capabilities.

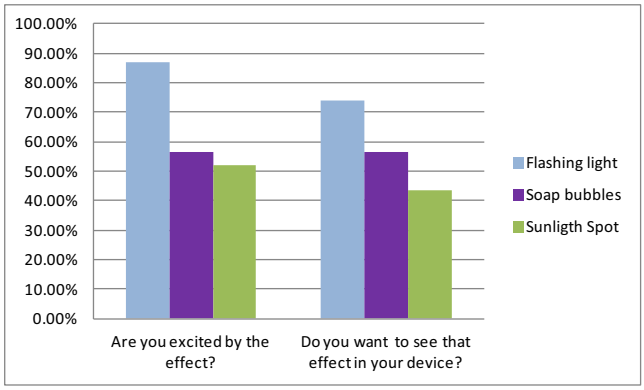


Figure 11. Survey results.

REFERENCES

1. Chen, J., Xiao, J., Gao, Y. *iSlideshow: a Content-Aware Slideshow System, ACM Intelligent User Interface conf., 2010.*
2. Sakaino, H., *The photodynamic tool: generation of animation from a single texture image. IEEE ICME, 2005.*
3. Safonov, I, Bucha, V. *Animated thumbnail for still image, GRAPHICON-2010, pp. 79-86, 2010.*
4. Chen, J.C., Chu, W.T., Kuo, J.H., Weng, C.Y., Wu, J.L. *Tiling slideshow. ACM Multimedia 2006, 25-35, 2006.*
5. Dunker, P., Popp, P., Cook, R., *Content-aware auto-soundtracks for personal photo music slideshows. IEEE ICME 2011, 1-5, 2011.*
6. Viola, P., Jones, M., *Rapid object detection using a boosted cascade of simple features, In Proc. of Conference Computer Vision and Pattern Recognition, 2001.*
7. Egorova, M.A., Murynin, A.B., Safonov, I.V., *An Improvement of face detection algorithm for color photos, Pattern Recognition and Image Analysis, vol. 19, No. 4, pp. 634–640, 2009.*
8. Vil'kin, A.M, Safonov, I.V., Egorova, M.A., *Bottom-up Document Segmentation Method Based on Textural Features, Pattern Recognition and Image Analysis, vol. 21, No. 3, pp. 565–568, 2011.*
9. Itti, L., Koch, C., Niebur, E., *A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on Pattern analysis and machine intelligence, Vol. 20, No. 11, pp. 1254-1259, 1998.*
10. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M., *Global Contrast based Salient Region Detection. IEEE CVPR 2011, 409-416, 2011.*
11. Goto, M., *Real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals, Speech Communication, vol. 43, no. 4, pp. 311–329, 2004.*
12. Dixon, S., *MIREX 2006 Audio Beat Tracking Evaluation: BeatRoot, in MIREX at 7th International ISMIR 2006 Conference, 2006.*
13. Scheirer, E.D., *Tempo and beat analysis of acoustic musical signals, J. Acoust. Soc. Amer., vol. 103, no. 1, p. 588-601, Jan. 1998.*
14. McKinney, M.F., Moelants, D., Davies, M.E.P., Klapuri, A., *Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms. Journal of New Music Research **36**(1), 1-16, 2007.*

About the authors

Konstantin A. Kryzhanovsky received his MS degree in cybernetics from Moscow Engineering Physics Institute/University (MEPhI), Russia in 2000. Since 2004 he works as an instructor of faculty of Cybernetics of MEPhI. Since 2011, K.A. Kryzhanovskiy joined Samsung Moscow Research Center, Russia where he is working on computer graphics, image and video processing projects.

Aleksey M. Vil'kin received his MS degree in mathematics from National Research Nuclear University MEPhI in 2011. From 2011 he is PhD student in MEPhI working on pattern recognition, page segmentation problems. In 2011 Aleksey joined Samsung Moscow Research Center, where he is engaged in on computer graphics, image and video processing projects.

Ilia V. Safonov received his MS degree in automatic and electronic engineering from Moscow Engineering Physics Institute/University (MEPhI), Russia in 1994 and his PhD degree in computer science from MEPhI in 1997. Since 1998 he is an assistant professor of faculty of Cybernetics of MEPhI (now National Research Nuclear University) while conducting researches in image segmentation, features extraction and pattern recognition problems. Since 2004, Dr. I. Safonov joined Samsung Moscow Research Center, where he is engaged in image and video processing projects.

Zoya V. Pushchina graduated from Moscow Bauman State Technical University and received her MS degree in CS in 1996. From 2009 she works in Samsung Moscow Research Center. Her role is image and video algorithm optimization for embedded platforms.