

Resolution Independent 2D Cartoon Video Conversion

MSF. Fayaza

Faculty of Information Technology, University of Moratuwa, Sri Lanka
Email: msf.fayaza89@gmail.com

MAM. Aazeer, MFHM. Adheeb, KF.Muhiminah, S.C.Premarathna

Faculty of Information Technology, University of Moratuwa, Sri Lanka
Email: mamreezaa@gmail.com, adheeb91@gmail.com, fathima.mkhalid@gmail.com, samindap@uom.lk

ABSTRACT

This paper describes a novel system for vectorizing 2D raster cartoon. The output videos are the resolution independent, smaller in file size. As a first step, input video is segmented to scene thereafter all processes are done for each scene separately. Every scene contains foreground and background objects so in each and every scene foreground background classification is performed. Background details can be occluded by foreground objects but when foreground objects move to their previous position such occluded details are exposed in one of the next frames so using that frame can fill the occluded area and can generate static background. Classified foreground objects are identified and the motion of the foreground objects is tracked for this simple user assistance is required from those motion details of foreground object's animation is generated. Static background and foreground objects are segmented using K-means clustering and each and every cluster is vectorized using potrace. Using vectorized background and foreground object animation path vector video is regenerated.

Keywords - Vectorizing, Foreground extraction and identification, Background extraction, Background filling, Motion tracking, Scene Segmentation, Clustering

Date of Submission: March 14, 2016

Date of Acceptance: April 20, 2016

I. INTRODUCTION

Cartoons are the result of artistic works which have a long history starting from paper drawings. Today it is preserved as digital videos as a result of technology enhancement. But such videos are mostly in raster format. Raster format videos are resolution dependent and comparatively need high storage and to download high resolution raster video need high bandwidth. Normally raster videos can be seen clearly in specific resolution and when one tries to increase resolution it gets blurred and pixelated. But vector videos are resolution independent and can be seen clearly at any resolution. So this paper presents a novel approach for vectorizing 2D cartoon videos. As a result of this approach the following advantages can be gained:

- Resolution independent videos so can be clearly seen at any resolution also can be displayed in different display capabilities.
- Comparatively need small storage because vectors usually represent by points, lines, curves and polygons.
- Can be edited easily because background, foreground and animation are separated.
- Output video can be transmitted using comparatively small bandwidth.

Today there is increasing demand for vector based contents on internet. So there are several researches on different approaches for vectorizing raster images and

there are several commercial tools for vectorizing raster images ex: Standford VectorMagic. But there is a number of various approaches for vectorizing videos and quality enhancement of video in literature. The approaches were taken in different points of view.

In this paper we provide a system to convert 2D raster to vector cartoon. System needs simple user assistance to track the multiple object motion to create the animation. Also system provides a new way of vectorizing using potrace. Potrace mainly provides the black and white output but this paper addresses this problem and it provides a novel approach to get color vector images.

The rest of the paper is organized as follows. Section II involves the detailed description about the related research work. Section III presents the Proposed Approach. Section IV presents Design and Implementation details of our system. Section V presents evaluation of results. The final section contains conclusion of the study.

II. RELATED RESEARCH WORK

There have been various proposed solutions to resolution quality enhancement of video in literature. Though they are not much closer to our topic. The approaches were taken in different points of view. Under those approaches only some of them were presented related to our topic.

The approach discussed in is the most closely related work to ours [1]. Here they developed a methodology to

vectorize a small cartoon video sequence. The approach described in [20] and [21] discusses techniques to vectorize raster images. The approach in [6] describes the use of SIFT features to track objects in a video sequence. [2, 4, 5, 12, 17] discusses various techniques of shot boundary detection and background extraction.

Vectorizing Cartoon Animations [1] mainly focuses on vectorizing of single video sequence rather than the whole cartoon sequence. They mainly concern about cartoon with static background and decorative lines. And they used ‘trapped-ball’ method [1] to segment frame into regions. Also in their approach simple user assistance was required to extract the background. Unlike the approach in [1], our approach does not aim to produce fully vectorized video but rather at reproducing the animation with vectorized images to output frames with vector images. Therefore we need to track the motion and transformation of the objects of the foreground in order to be able to reproduce the animation

Video segmentation is the basic step in any video processing or analyzing applications. Shot boundary detection facilitates to segment the video in to subdivisions (shots). There are many shot boundary detection algorithms proposed in literature. One of the basic and straightforward ways is pixel difference/ pixel comparison [4, 5, and 18] to determine the shot. Second approach is Block based approach [5] but it is slow because of complexity of statistical formula. Third approach is Histogram comparison [4, 5, and 12]. The logic behind this approach is, if the background and objects (moving and static) are unchanged in successive frame their will have small difference in histogram [4].

To fill the occluded area of background there are several approaches proposed in [1] they construct an initial background based on median pixel values in appropriate frames and at the same time, they estimate the background probability of each pixel. In [4] they fill the occluded area by visually plausible background that mimics the appearance of the source region or neighboring pixels. Background pixels that are occluded by the moving foreground if they are exposed in at least one other frame. In [2], using motion vectors of each moving object they calculate how fast the corresponding region can recover and using that frame they extracted the occluded details [2].

To track object there are several approaches the object tracking method used in [6] can track objects invariant of homographic transformation, lighting, viewpoint change and also a certain degree of change in illumination. There are many methods proposed to describe regions that are invariant under homographs. Approaches in [8, 9, 10] have discussed feature descriptors that are invariant to transformation. Transformation invariant features in [6] are based on interest points which are selected according to criteria.

To victories the objects in the background and foreground regions in [1] they have used the approach in [21] using gradient meshes. Unlike in conventional vector graphics which is described in lines, curves and polygons that are then colored within the boundaries, in [21] they have provided a tool called “optimized gradient mesh” which enables to represent objects as a single multicolored object which can be easily manipulated to scale the image.

Another approach to vectorize full color raster images is discussed in [20]. They decompose the image to triangles and piecewise smooth Loop subdivision surfaces to achieve number of vector image editing operations. This approach has produced high quality vectorized results with low color discontinuities compared to approaches in [22].

Such high level of detail is not needed to vectorize cartoon images, as it has lower color variations compared to real world images.

III. PROPOSED APPROACH

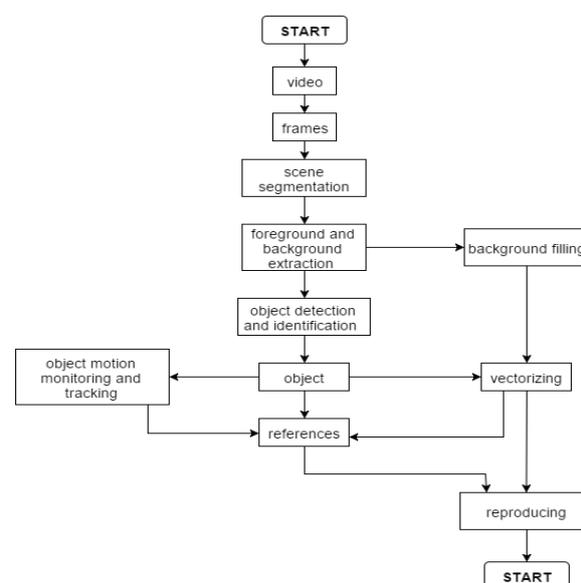


Fig.1.Top level architecture of the system

In this approach if user provides the 2D raster cartoon video to system it's provide output as vector video. Fig.1 shows the top level architecture of the system. Normally input video of the system is resolution dependent 2D cartoon videos with low quality to provide the vector video input video goes through following sequence of steps

- Videos usually consist with multiple scenes. Scene is a segment of video with the same background throughout that segment. As first step video segment into scenes and rest of the process are continue for each scene.
- To extract foreground and background of a scene background, foreground classification is performed.
- Some background details are occluded by foreground objects so as next step occluded

details of background is extracted form one of next frame and using that details background filled and static background generated.

- Foreground objects are identified
- Using simple user assistance motion of foreground objects tracked
- Background and foreground are segmented using K-means clustering
- Each cluster is vectorized using potrace.
- Using vector background, foreground and motion details vectorized animation is output.

IV. DESIGN AND IMPLEMENTATION

This section describes the detail description of system design and how it implemented. There are several approaches for each step. The procedures and techniques are explained below.

A. Scene segmentation

In a video basically scene changes happen in two ways those are abrupt and gradual. Gradual transition is occurring throughout many frames. Abrupt transition is occurring in a single frame, a scene is stopping at particular frame and in the next frame starting a new scene. The system supports for abrupt transitions. To segment the scenes 32 bins rgb color histograms comparison is used. Histograms are compared using the Chi-Square matrices; if the difference between the following frames exceeds the threshold shot is detected.

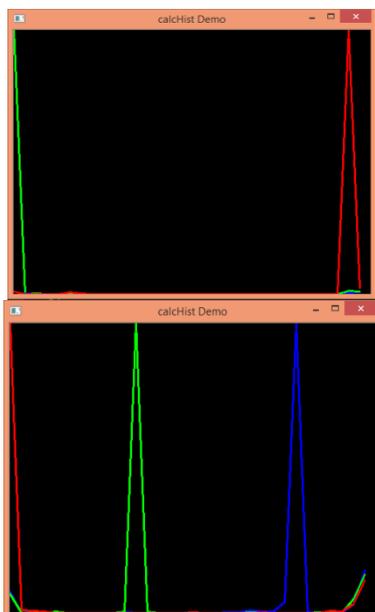


Fig.2. Histogram of two consecutive frames where scene detected

B. Background Extraction

Normally in a video there are two types of backgrounds available those are static background and moving background. In the static background it will not containing any moving objects. In moving background the background objects can have movements. In our system

we are supporting static backgrounds; it will not contain moving objects. If a particular object contains movement it is defined as foreground object. To extract the background as first approach using MOG2 algorithm foreground mask crated. After that in-order to simplify the process foreground mask encapsulated into rectangle box. From the rectangle box occluded area is inedited.

The background pixels are static. Second approach is for a particular scene, pixel based foreground – background classification performs. For this purpose shot detected frame considered as base-frame and N (N=10 in our implementation) consecutive frames used to compare with base-frame and foreground-background pixels classified. After that to simplify the process foreground pixels are encapsulated into rectangle. From that rectangle occlude area of background identified. Background, foreground classification algorithm is given below

```

start
for x=baseFrameNo and x<=baseFrameNo+N{
  for i = 0 and i < rows {
    for j = 0 and j = columns {
      read baseFrame pixel value of base, consecutive frame
      if(baseFrame pixel value- consecutiveFrame pixel
      value)< = thresholdValue;
        pixelValue =0;
      else
        pixelValue = 255;
      }
    }
  }
end
    
```



Fig.3. Base-frame, foreground mask, background, foreground classification

C. Background Filling

If foreground object move from its previous position one of next frame the occlude area is exposed. Motion vector of moving object is used to identify the frame where the occluded area is exposed. For this following equation is used.

$$\sqrt{(x_{current}-x_{previous})^2+(y_{current} - y_{previous})^2} > \text{Rectangle diameter} \quad (1)$$

D. Foreground Extraction

Since static background of a scene already available to extract the foreground absdiff() was used. The absdiff() function will output the computed difference of each and every pixel of the video frame being considered and the background image. The result of absdiff() return the pixel

belongs to foreground that result is threshold and stored. Thereafter threshold value is read using the rgb values and foreground object is identified.

E. Foreground Identification (Object Identification)

Since the same foreground objects will appear in a sequence of frames and in a cartoon, writing the same foreground that appears in all of the frames to an image file is not needed. To avoid redundancy of the same object, we use object identification in order to identify already available image files of a particular foreground appearing in a sequence of frames.

We use the SURF () method to compare two images. In this case we compare the extracted foreground against all the available foreground image files available so far. This algorithm utilizes a Hessian based detector and intensity distribution based description feature vector and leverages several approximations, which allow for fast computation without major sacrifices in accuracy and repeatability.

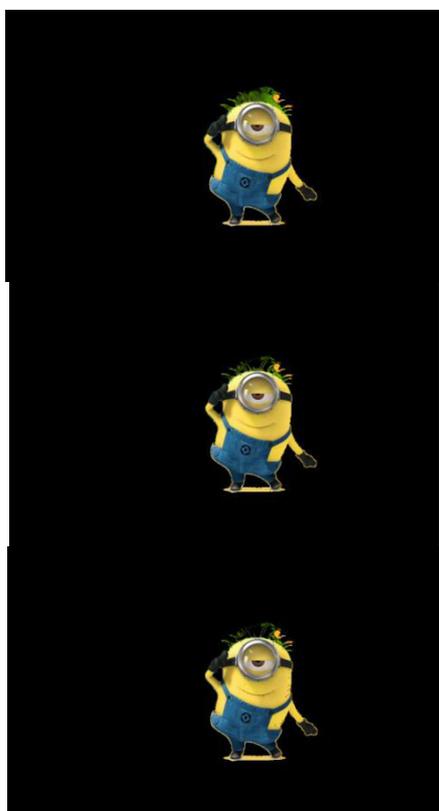


Fig.4. Background subtraction and thresholding using 5, 10,20 respectively

F. Color Extraction for Tracking

To extract the motion of foreground object simple user assistance is needed. The extracted foreground will available to user if user select the foreground object the color value of object is extracted. The extracted color needs to be in the HSV color space. So start by first converting the foreground image to the HSV color space. Thereafter we present the user with these converted images in sequence. When the user click on the image, the HSV color value corresponding to the area clicked by the

user is written to a file along with the references to the foreground object

G. Motion Tracking

Using that color of object the motion of object is tracked. Once each user selection take place the color of object is taken into the system. By using inrange () function of opencv. Thereafter Threshold the HSV image for the range of colors. And after that to remove noise Erode and dilate the image in order to clearly make visible the object. Find contours of object and track the filtered object using moment() function. After the removal of noise track the x, y coordinates of object in each frame. Form that animation points generated.

H. Background, Foreground Segmentation and Vectorization

We have used k-means clustering along with potrace in order to vectorize the image. Potrace is an open source tool that can be used to convert binary raster image files to vector files. A binary image file can be given as an input, and potrace will generate the vectorized image in the required vector representation. Potrace also provides a tool to prepare input images to an acceptable input format or to preprocess the image in order to achieve the best vectorized output. However this cannot convert color images accurately. We have come up with a novel approach to using potrace in order to vectorize color images with high accuracy.

First, the image is clustered by using k-means clustering. We write each cluster of the image to a separate image file in the bmp format. Then we use the 'mkbitmap', another tool provided along with potrace to threshold the image. Thereafter we use potrace to convert the threshold image to the SVG file format.

Since potrace only supports binary images, the output vector images are always black. Therefore we need to modify the colors of all output vector image files. Since we vectorize all clusters of the images separately, we can modify each output SVG file with the color of that particular cluster. We have used php script which will be called in our program to modify the output vector files. Once all image clusters are vectorized, we merge all the SVG file in to one SVG file, which will be used in our program.



Fig.5. Clustered images – 5 clusters and 10 clusters



Fig.6. Clustered images – 20 clusters and 35 clusters

I. Video Reproducing

Using the details available details of background, foreground and motion vector animation is generated. Using the background details master background is generated thereafter on top that foreground object is animated.

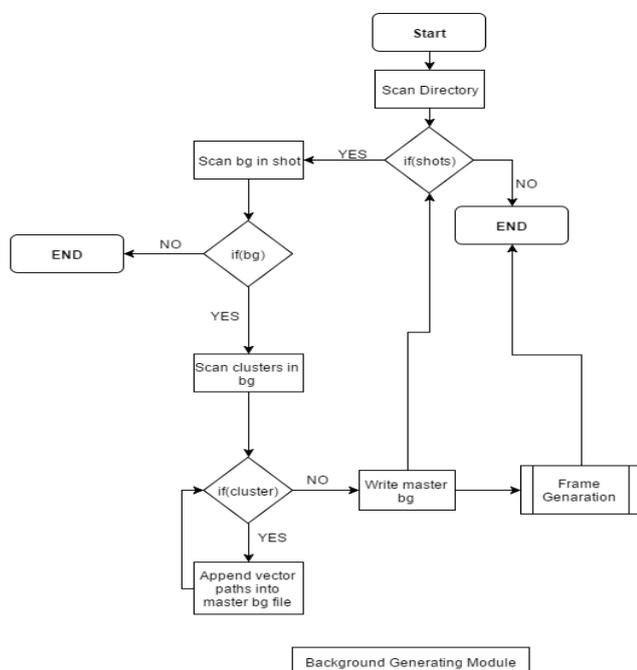


Fig.7. Background Generating Module

V. RESULT AND DISCUSSION

There are many open source and commercial tools available to convert raster images to vector images but there are no many tools to convert raster video to vector video. As a first step of raster video to vector video conversion we started with 2D cartoon videos and we find a novel approach to vectorizing raster videos. Throughout our study we also find a novel approach to vectorizing raster images using Potrace.

To vector color images we cluster the images based on colors using K-means clustering. We tested 100 images starting form 5 clusters to 35 clusters. We find that number of cluster for good quality output varies depends on the images. But most of images get aspect output in 22 clusters. And also for an image vectorizing it takes less than one minute.



Fig.8. SVG of image clusters and merged SVG fil



Fig.9. Raster, vector frame

VI. CONCLUSION

We have been successful in vectorizing raster images by being able to find a novel approach to vectorizing color images using k-means clustering and potrace. This approach is efficient and can produce good quality images based on the number of clusters.

We have introduced a novel approach to vectorizing 2D cartoon videos by vectorizing the foreground and background separately and by using these vectorized components to re produce the video. So far, though we have not been entirely successful for all cartoon video, we have been able to demonstrate our approach on a set of videos.

REFERENCES

- [1]. Song-Hai Zhang, Tao Chen, Yi-Fei Zhang, Shi-Min Hu, Member, IEEE, and Ralph R. Martin, Vectorizing Cartoon Animations
- [2]. Srenivas Varadarajan*, Lina J. Karam*, and Dinei Florencio, Background recovery from video sequences using motion parameters
- [3] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2
- [4] Irena Koprinska, Sergio Carrato, Temporal video segmentation: A survey
- [5] Comparison of video shot boundary detection techniques John S. Boreczky, Lawrence A. Rowe

- [6] D.G. Lowe, "Object Recognition from Local Scale-Invariant Features," Proc. Int'l Conf. Computer Vision, pp. 1150-1157, 1999
- [7] J. Xiao and M. Shah, "Motion Layer Extraction in the Presence of Occlusion Using Graph Cuts," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27
- [8] A. Baumberg. Reliable Feature Matching Across Widely Separated Views. In Proceedings of the International Conference on Computer Vision and Pattern Recognition
- [9] D. Lowe. Object Recognition from Local Scale-Invariant Features. In Proceedings of the International Conference on Computer Vision, pages 1150–1157, Corfu, Greece, September 1999
- [10] C. Schmid and R. Mohr. Local Grayvalue Invariants for Image Retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5)
- [11] Aseem Agarwala, Aaron Hertzmann, David H. Salesin, Steven M. Seitz, Keyframe-Based Tracking for Rotoscoping and Animation
- [12] L. Congcong, Y. T. Wu, Y. Shiaw-Shian, and T. Chen, "Motion-focusing key frame extraction and video summarization for lane surveillance system," *ICIP 2009*.
- [13] L. J. Lai, Y. Yi, Key frame extraction based on visual attention model, *Journal of Visual Communication and Image Representation* 23(1)
- [14] N. Ejaz, T. Bin Tariq, and S. W. Baik, "Adaptive key frame extraction for video summarization using an aggregation mechanism," *Journal of Visual Communication and Image Representation*, vol. 23
- [15] Ravi Kansagara, DarshakThakore, Mahasweta Joshi, "A Study on Video Summarization Techniques," *International Journal of Emerging Technology and Advanced Engineering*, (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4(9))
- [16] S. Ariffa Begum, Dr. A. Askarunisa, "Performance Analysis of Various Key Frame Extraction Methods for Surveillance Applications," *International Journal of Emerging Technology and Advanced Engineering* Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, vol. 4)
- [17] S. K. Kuanar, R. Panda, and A. S. Chowdhur, "Video key frame extraction through dynamic Delaunay clustering with a structural constraint," *Journal of Visual Communication and Image Representation*, vol. 24
- [18] Sujatha, C.; Mudenagudi, U., "A Study on Keyframe Extraction Methods for Video Summary," *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*, vol., no., pp.73, 77, 7-9
- [19] Soma Datta, Debotosh Bhattacharjee, Prमित Ghosh, "Path Detection of a Moving Object", *International Journal of Recent Trends in Engineering*, Vol 2, No. 3
- [20] Zicheng Liao, Hugues Hoppe, David Forsyth, and Yizhou Yu, A Subdivision-Based Representation for Vector Image Editing
- [21] Jian Sun Lin Liang Fang Wen Heung-Yeung Shum, Image Vectorization using Optimized Gradient Meshes
- [22] T. Xia, B. Liao and Y. Yu,"Patch-Based Image Vectorization with Automatic Curvilinear Feature Alignment," *ACM Trans. Graph.* vol. 28