Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 24 (2016) 1468 – 1474

Procedia Technology

International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015)

ARTAR-Artistic Augmented Reality

Nithin G., Reshmi S. Bhooshan

College of Engineering Trivandrum, Kerala- 695016, India

Abstract

Augmented reality is a technique which adds computer generated virtual objects into the real world scene. ARTAR proposes a method to enhance the experience of paintings or artistic works by adding an extra level of perception through the inclusion of sound, music, and animations. It contains two layers of perception, the physical appearance of the paintings perceived by naked eye and an augmented layer containing animations and sounds which can be perceived by a mobile device. When an artwork is scanned using a predesigned mobile application certain image reference portions get animated along with some music and sound. In this work, a reference area is found in the input video frame using SURF detector and BRISK descriptor and the virtual object is placed in the particular position. Experimental result shows that SURF-BRISK combination provides better result when compared with other detector descriptor combinations in case of ARTAR.

Peer-review under responsibility of the organizing committee of ICETEST – 2015

Keywords: Augmented reality;

1. Introduction

Augmented Reality(AR) is a field of computer research which aims at supplementing reality by mixing computer-generated data and real world environments. The basic AR system is shown in fig.1, where an input video is converted into frames and the presence of a reference object is checked in each frames. When the reference object is found, a virtual object is placed with reference to it. AR has been demonstrated widely on mobile phones, with different applications such as games, navigation and references [8]. In recent years AR research has focused on various sub areas of interest. Highly developed techniques in Graphics and Virtual Reality allow for increasingly realistic AR visuals. Elaborate tracking systems, input devices and computer vision techniques improve the registration of images and user interaction in AR setups. However, most commonly used AR systems still require sophisticated hardware tracking solutions, or at least fiducial markers [9]. ARToolKit [4] is a software library for building marker based Augmented Reality (AR) applications. Squared black and white markers with an embedded code is used in ARToolKit.

* Corresponding author.
E-mail address: 2nithin.g@gmail.com (Nithin G.), reshmibhooshan@gmail.com (Reshmi S. Bhooshan)

Peer-review under responsibility of the organizing committee of ICETEST – 2015
doi:10.1016/j.protcy.2016.05.183

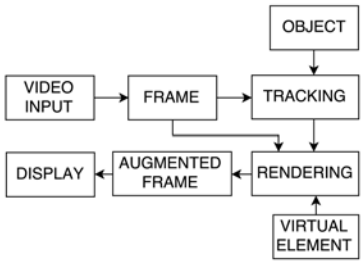

Fig. 1. Basic Augmented Reality system

Inorder to overcome the limitations of marker based AR, markerless AR was proposed. To find camera pose, natural features such as colour, shape, texture etc are used. The main objective is to track an object in each frame of video stream. Several methods of interest point detection are available for this purpose. Harris [3] is the most popular corner detector used in computer vision. SIFT (Scale Invariant Feature Transform) algorithm proposed by Lowe [6] is efficient to find interest points using Difference of Gaussian(DoG) of the image, and SIFT descriptor is based on histogram of gradients. SURF (Speed Up Robust Features) algorithm [2] is an approximation of SIFT in which DoG is approximated to box filters. Recently, several binary keypoint descriptors are proposed for efficient local feature matching in real-time applications.

The ARTAR is centered around the several layers of reality that are hidden and presented to the spectators as response to their interaction with the painting or artwork. Such interaction happens on the spectators' mobile devices, which act as windows into the multiple dimensions of the artwork. The system consists of three main elements: paintings, augmented elements, and interaction. In ARTAR, two types of augmented elements, animations and soundscapes, are embedded into the paintings. Animations provide a dynamic visual dimension that gets activated at specific positions of the painting and reveals hidden visual aspects of the artwork. Soundscapes accompany each painting, reproducing the sounds of the depicted ecosystem.

When a painting is viewed through the predesigned mobile application spectators feel as if the painting gets some life. This sort of a system can be used at art exhibitions. In ARTAR instead of a separate marker being used for image detection, parts of paintings themselves act as markers. When a painting is viewed through the mobile device, These image portions will be detected and are replaced by the frames of predesigned animations. Different keypoint detection and description techniques like SIFT, SURF, ORB, FAST, FREAK, BRISK etc are used to identify the area of interest. The keypoints in the reference image will be initially calculated and these points will be matched with the key points of each frame from camera. If a matching portion is obtained, that particular area is replaced with the desired animation. Also soundscapes can be associated with these areas. The performance of different combinations of detectors and descriptors is analysed and the best combination is used to frame the mobile application.

The organisation of the paper is as follows. Section 2 explains the proposed method ARTAR. Experimental results along with the analysis done on different detector descriptor combinations, to find which combination provides better result, is given in section 3 and section 4 concludes the work.

2. ARTAR

The proposed ARTAR algorithm for adding animations and sounds into paintings and other artworks is shown in fig.2. Input video is converted into frames and is analysed to see whether the reference image is present in it or not. Animation is designed corresponding to a reference portion of the image and then the feature points and their descriptors are found in the reference image and the frames from camera. Then these feature descriptors will be matched using similarity measures to find the matching feature points. When feature matching is obtained, homography matrix is calculated to accurately place the desired video frame over the camera frame.

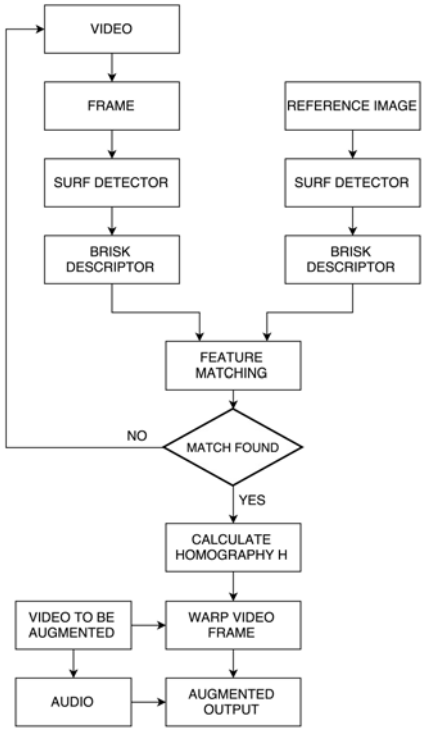

Fig. 2. Flowchart of the proposed ARTAR method

2.1. SURF keypoint detection

Keypoints are spatial locations, or interesting points in the image that can be detected even when changes like rotation, shrinking or distortion occur in the image. Different keypoint detection techniques like SIFT, FAST, SURF, ORB, BRISK etc are availabe. In this work, we have used SURF feature detector for keypoint detection, since it provides better result in case of speed and accuracy when compared with other detection algorithms.

In SURF algorithm keypoint detection works using Fast-Hessian detector which is based on Hessian matrix.

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma)L_{xy}(x,\sigma) \\ L_{\mathrm{xy}}(x,\sigma)L_{yy}(x,\sigma) \end{bmatrix} \tag{1}$$

$L_{xx}(x,\sigma)$ is the convolution of the Gaussian second order derivative with the image I in point x, and similarly for $L_{\mathrm{xy}}(x,\sigma)$ and $L_{yy}(x,\sigma)$. SIFT uses Difference of Gaussians (DoG) to approximate Laplacian of Gaussian (LoG) whereas SURF uses box filter to approximate second order Gaussian derivative. Convolution of image with box filters can be easily computed by using integral images. After computing the integral image, it is straight forward to calculate the sum of the intensities of pixels over any upright, rectangular area. Determinant of the Hessian is used to obtain the location and scale of interest points which can be found using the second order derivatives, D_{xx} , D_{yy} , and D_{xy} . Determinant of Hessian matrix is found by choosing adequate weight for box filter.

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{2}$$

The feature points are found out by applying non-maximum suppression in the neighbourhood of each pixel. Fig.3 shows the SURF feature points extracted from a reference image. The 482 keypoints present in the image are represented using coloured circles.

Fig. 3. SURF feature points extracted.

2.2. BRISK descriptor

Achieving real time processing is the major constraint in case of ARTAR system. Non binary descriptors like SIFT or SURF requires a lot of computational power, hence becomes less effective for real time applications. Binary descriptors like BRISK, FREAK [1] or ORB [7] can be used for these applications. Among this BRISK provides better result when combined with SURF detector.

BRISK [5] is a binary descriptor which can be obtained from weighted gaussian averaged pattern of points in the vicinity of the keypoint. A value of 1 or 0 is assigned to each point in the pattern by comparing the values of specific pairs of Gaussian windows, depending on which window in the pair is greater. This gives out 512 bit binary descriptor BRISK, which can be used for keypoint matching using Hamming distance.

2.3. Feature points matching

After the extraction of feature points and descriptors from reference image and camera frame, a similarity check must be carried out to find whether the reference image is present in the camera frame. Since the BRISK descriptors obtained are binary in nature, hamming distance can be used to find similarity. The method of the K-nearest neighbour (K-NN) matching is used to find the best k matching feature points in camera frame corresponding to the reference image. After the matches are found, Lowe's ratio test is carried out to eliminate error matching. Fig.4 shows the keypoint matching between the reference image and camera frame. Reference image is at the left side and video frame at right. The matching keypoints are connected using lines.

Fig. 4. Keypoints in the reference frame and the matching keypoints in the video frame are connected using coloured lines.

2.4. Image registration

After the correct matching points are obtained, the mapping relations between the reference image and the position of the reference image in the camera frame is to be found. Homography matrix, H can be used to obtain the positions of reference image points in the camera frame. Hence we find the H matrix from the matching keypoints using which

the required video frame to be augmented can be placed over the reference image location. Let p_{cam} denote the points in the camera frame corresponding to the points p_{ref} in the reference image, then we can represent the relation as,

$$p_{cam} = Hp_{ref} \tag{3}$$

where $p_{cam} = \begin{bmatrix} x_{cam} \\ y_{cam} \\ 1 \end{bmatrix}$ and $p_{ref} = \begin{bmatrix} x_{ref} \\ y_{ref} \\ 1 \end{bmatrix}$

3. EXPERIMENTAL RESULTS

Experiments are carried out in Intel core i5-3337U 1.8GHz processor with 4 GB RAM using OpenCV, which is a cross platform library. By utilising the results obtained, an android mobile application is created for ARTAR. SIFT, SURF, ORB, FREAK and BRISK are the different detectors and descriptors considered for finding the best possible combination of detector and descriptor applicable for the particular application with reference to processing speed and accuracy.

3.1. Processing speed

The time taken by different detectors to find the feature points in a reference image is calculated and the results are given in fig.5. From this we can see that SIFT takes huge amount of time to compute the feature points in the reference image. Hence it cannot be used in the proposed system, because ARTAR requires realtime processing. SURF, BRISK, and ORB provide satisfactory results and can be used in this case. ORB is the fastest in the analysed detectors and gives enough number of key points.

In this work, 500 SURF feature points are found and the time taken by different descriptors to find the descriptor vector corresponding to these feature points is analysed. The result obtained from different descriptors is shown in fig.6. The time taken to compute the descriptor vector was not very good in case of SURF. At the same time binary descriptors like BRISK, FREAK, and ORB provided better performance. Due to larger computational time SURF detector cannot be used in the proposed work.

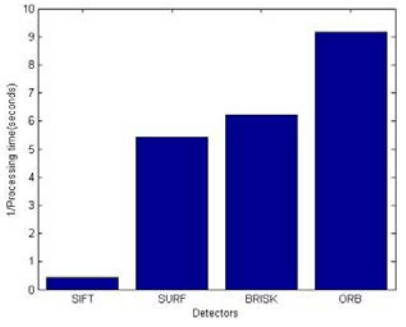

Fig. 5. Graph representing the 1/processing time of different detectors.

3.2. Accuracy

The performance evaluation of different detector-descriptor combinations is done utilising precision and recall. 10 videos are created with the reference image in view, form different angles and distances, where each video contains 1000 frames of size 480x360. Different detector-descriptor combinations are applied on these videos to find whether the reference image is present or not. The matching of reference image with different video frames is analysed manually. We compute recall vs 1-precision of detector-descriptor combinations for each set of frames.

$$recall = \frac{N_c}{N_c + N_n} \tag{4}$$

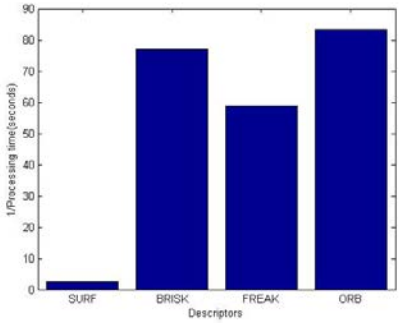

Fig. 6. Graph representing the 1/processing time of descriptor vectors corresponding to 500 keypoints.

$$1 - precision = \frac{N_f}{N_c + N_f} \tag{5}$$

Where, N_c represents the number of frames in which the reference image is correctly matched, N_n is the number of frames in which the reference image is not detected even when it is present and N_f is the number of frames in which the reference image is matched wrongly. The resulting precision-recall curve is shown in fig.7. From the graph, we can see that SURF-SURF detector-descriptor combination provides highly accurate results. SURF-BRISK combination provides better result when compared with ORB-ORB combination. But the rest of the combinations fail to provide satisfactory results.

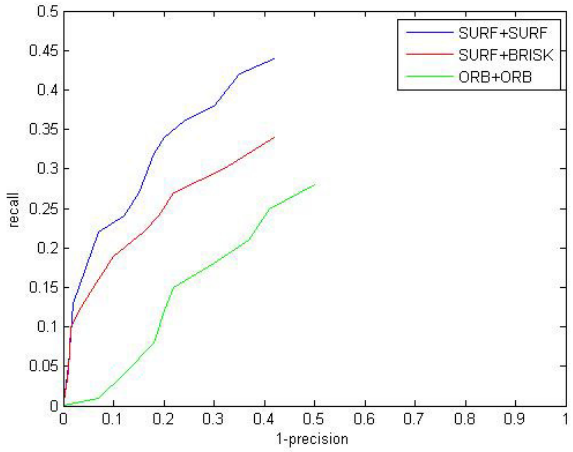

Fig. 7. Precision-recall curves corresponding to the three detector descriptor combinations, SURF-SURF, SURF-BRISK and ORB-ORB.

From the above discussions we can see that SURF is the better detector which will provide good detection speed. But SURF cannot be used as descriptor because of the higher processing time. BRISK provides satisfactory result when combined with SURF detector.

A sample mobile application is created for a drawn image of Charlie Chaplin of size 20cm X 30cm. When the image is scanned using the mobile application we feel the illusion that the eyes of the drawing opens and closes. The final output which can be seen in mobile phone is shown in fig.8. Fig.9 shows the reference image which can be seen by naked eye and the frames which canbe seen through mobile device which provides a blinking effect. Eye portion of the image is identified using SURF detector and BRISK descriptor, and that portion is replaced with the animation frames which gives the feel of blinking eyes.

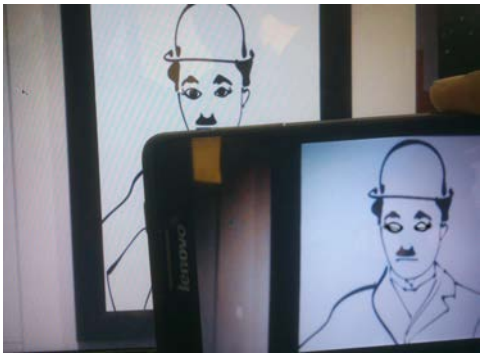

Fig. 8. View of Charlie Chaplin illustration through the mobile application. The image in mobile phone gives the illusion of blinking eyes.

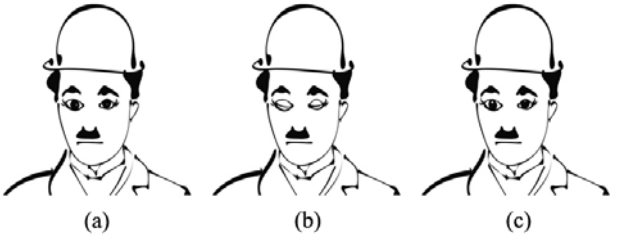

Fig. 9. Different frames used to animate Charlie Chaplin illustration. (a)Original image which can be seen by naked eye. (b)&(c) Frames which can be seen in mobile device that provides the feel of blinking eyes.

4. Conclusion

Artistic augmented reality(ARTAR) is a method to add extra levels of perceptions by including animations, music and sound to paintings. It helps the viewers to get immersed in the feelings associated with the artwork. When a painting is scanned using a mobile device, spectators feel the illusion of the painting getting animated along with music and sound. ARTAR works by recognizing a predefined image portion in the camera frames coming from the mobile device, and replacing them with frames of animation. Also the sound associated with the animation is played. SURF detector along with BRISK descriptor is used to recognize the reference image portion.

In future the stability and accuracy of the system can be improved by analysing the intensity variations of natural light in the real environment and modifying our augmented image portion according to it.

References

[1] Alahi, A., Ortiz, R., Vandergheynst, P., 2012. Freak: Fast retina keypoint. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. Ieee, pp. 510–517.
[2] Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features. In: Computer vision–ECCV 2006. Springer, pp. 404–417.
[3] Harris, C., Stephens, M., 1988. A combined corner and edge detector. In: Alvey vision conference. Vol. 15. Citeseer, p. 50.
[4] Kato, I. P. H., Billinghurst, M., Poupyrev, I., 2000. Artoolkit user manual, version 2.33. Human Interface Technology Lab, University of Washington 2.
[5] Leutenegger, S., Chli, M., Siegwart, R. Y., 2011. Brisk: Binary robust invariant scalable keypoints. In: Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, pp. 2548–2555.
[6] Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision 60 (2), 91–110.
[7] Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. Orb: an efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, pp. 2564–2571.
[8] Van Krevelen, D., Poelman, R., 2010. A survey of augmented reality technologies, applications and limitations. International Journal of Virtual Reality 9 (2), 1.
[9] Wagner, D., Pintaric, T., Ledermann, F., Schmalstieg, D., 2005. Towards massively multi-user augmented reality on handheld devices. Springer.