Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 58 (2015) 698 – 705

Procedia Computer Science

Second International Symposium on Computer Vision and the Internet (VisionNet'15)

Detection of Features to Track Objects and Segmentation using GrabCut for Application in Marker-less Augmented Reality

Pulkit Khandelwal[a,*], Dr. Swarnalatha P[b], Neha Bisht[a], Dr. S Prabu[b]

[a] School of Electronics Engineering, VIT University, Vellore, Tamil Nadu, 632014, India
[b] School of Computing Science and Engineering, VIT University, Vellore, Tamil Nadu, 632014, India

Abstract

Augmented Reality applications have hovered itself over various platforms such as desktop and most recently to handheld devices such as mobile phones and tablets. Augmented Reality (AR) systems have mostly been limited to Head Worn Displays with start-ups such as Magic Leap and Occulus Rift making tremendous advancement in such AR and VR research applications facing a stiff competition with Software giant Microsoft which has recently introduced Holo Lens. AR refers to the augmentation or the conglomeration of virtual objects in the real world scenario which has a distinct but close resemblance to Virtual Reality (VR) systems which are computer simulated environments which render physical presence in imaginary world. Developers and hackers round the globe have directed their research interests in the development of AR and VR based applications especially in the domain of advertisement and gaming. Many open source libraries, SDKs and proprietary software are available worldwide for developers to make such systems. This paper describes an algorithm for an AR prototype which uses a marker less approach to track and segment out real world objects and then overlay the same on another real world scene. The algorithm was tested on Desktop. The results are comparable with other existing algorithms and outperform some of them in terms of robustness, speed, and accuracy, precision and timing analysis.

Keywords: Augmented Reality; Virtual Reality; Histogram of Oriented Gradients; GrabCut; Scale Invariant Feature Transform; Support Vector Machine; Interest Points Detectors; Image Descriptors; Segmentation; Human Body

1. Introduction

The paper discuss about the introduction in Section-I and literature survey with related papers in Section-II as follows: Technology have always dumbfounded humans, most recently with the conversion of plausible ideas into reality having shown remarkable achievement in the field of Virtual and Augmented Reality. This is the intersection of the fantasies of Human Beings with that of real world. AR takes computer or digitally synthesized objects or real inputs from users in the form of gestures, voice commands and eye gaze and produces overlays

* Corresponding author. Tel.: +91-784-585-0403.
E-mail address: kpulkit95@gmail.com

Peer-review under responsibility of organizing committee of the Second International Symposium on Computer Vision and the Internet (VisionNet'15)
doi:10.1016/j.procs.2015.08.090

which can be seen visually on a real world scenario which can be seen by a user. It adds an additional layer of information mostly in the form of computer aided graphics to incorporate extra details in the physical (real) world around us.

Marker based systems employ the use of predefined images usually black and white squares which can be easily recognized. Thus, they can be thought of as reference points. The new graphic image is then overlaid at the position of the determined marker by using pose estimation.

Applications requiring object tracking predominantly uses the marker less approach which is based on detecting interest points in the form of features and then assigning descriptors. These extracted characteristics are then used to augment the virtual graphic with the real world.

Developers have been using Marker based AR systems only to find its weaknesses such as difficulties in placing markers everywhere or in certain specific locations. Markers are black and white in cooour, square in shape mostly; and their design might also vary depending upon the application. Added to this marker based AR systems don't work efficiently if they are partially overlapped. They also tend to reduce the interactivity of users with the system. Although they have an advantage of showing desirable outputs even with cheap detection algorithms with not so good lighting conditions. In contrast, Marker less AR systems has not only proven to be better than marker based systems but also have clearly outnumbered the number of divergent applications which can be realized in the field of Augmented Reality. Marker-less system involves tracking and registration techniques which might be a little more complex to handle.

User level libraries mostly open source, Software Development Kits, Application Programming Interface which are under active development are being utilized by research academics, hackers, developers and programmers round the globe to create applications. These include Open CV developed by Intel, Microsoft's Kinect, OpenGL by SGI, ArUco: is a library developed specifically for AR systems on OpenCV. In addition, many AR solutions are available in the form of SDKs developed Metaio, Vuforia by Qualcomm, Satch by Total Immersion, Obvious Engine by Harmony Park, Layar Player by Layar, Wikitude. Moreover, there is an active developer community which use APIs and SDKs to build applications specifically for handheld systems running on operating systems such as Android and IOS. To name a few:

Qualcomm's FastCV is a computer vision library provides framework to make camera-based apps with functionalities such as gesture recognition, face detection, tracking and recognition, text recognition and tracking and of course augmented reality. ANDAR is a project which enables AR on Android devices developed by ARToolworks. ARToolKit is a software library for building Augmented Reality (AR) applications. ARToolKit uses computer vision algorithms to solve this problem. Loris D'Antoni[1] also talks about an Operating system for AR. Computer Science research domains such as computer vision, digital image processing, pattern recognition, machine learning with attributes from other related fields of study are used to solve such problems and develop AR applications. The framework may involve functionalities such as position/ orientation estimation, camera calibration, detecting markers and features. The framework apart from being real time should be open source and support multiplatform such as SGI IRIX, Linux, Mac OS and Windows OS distributions. It should support graphics for rendering 3D objects, should have source codes in various languages.

This paper deals with the development of a marker less AR application. They use the natural features based approach which can recognize and track real life objects such as buildings, cars, trees, humans etc. The general pipeline for this approach involves the detection of interest points such as corners, lines, edges, blobs, textures and regions along with the employment of feature descriptors such as SURF, SIFT, GLOH, STIP etc. This step is followed by tracking of the desired object using the excavated descriptors. Segmentation of the object is required to overlay it over real time camera acquisition or a recorded video. The system performs without any delay and error and have minimum frame to frame lag. Hence, timing analysis is another parameter to be taken care of.

2. Literature Review

The feature based model was first introduced by[20]. They used an approach to calculate camera pose using model features such as lines, circles, cylinders, and sphere. They were able to develop a robust tracking mechanism along with reduction of the effects of the outlier data. Edmund[2] have developed a real time system which uses the already extracted image features (found using modified version of SURF) to calculate homography which contain the coordinates of the virtual object for augmentation purposes and then find the model-view matrix to speed up the tracking step. Kato[3] describes a system where they have used real scene features to learn and track pose estimation and claims to increase the success of the overall registration accuracy for the AR application. Gerhard Reitmayr[19] used textures to develop

a tracking system which use edge information generated dynamically during run time by the detection of edges. Pressigout[5] came up with a hybrid vision system using both edge detection and texture analysis to develop a robust and accurate pose estimation.

In this paper we propose a novel algorithm for marker less AR. Interest point detection along with feature descriptor have been used to detect objects of interest such as human beings in a real time camera feed or a recorded video. Different scenarios have been considered depending upon the area of coverage of the human body (viz. full body or half body). The object under consideration is tracked and then trained using a SVM classifier. Segmentation of the object is done carefully in each and every frame of the video sequence. The final augmentation (i.e., overlay or rendering) the object takes place with a real time camera feed or with another pre-recorded video sequence. Note that "Tracking" in AR and VR refers to determining the pose, i.e., three-dimensional position and orientation, of the camera. "Tracking" in computer vision means data association, also called matching or correspondence, between consecutive frames in an image sequence. Here, both have been used.

The organization of the paper is as follows: Section 3 describes the methodology used to develop the algorithm and section 4 describe the procedure. Section 5 deals with the discussion of the results in detail. Section 6 indicates the further scope of improvement in the proposed work, tells us about what future add-ons could be thought of, thereby concluding the paper.

3. Methodology

Refer figure 1 for an overview of the algorithm.

3.1. Pre-processing

Each frame of the video sequence is grabbed and converted to into its Grayscale equivalent. Any given frame can be referred as an image (and is often used interchangeably in this paper). Filters such as Gaussian function are used to remove any unnecessary noise present in the image.

3.2. Interest Points detection and use of Descriptors

There is a necessity of encoding distinctive local structure in an image which can be used to determine other high level contextual information needed to develop an algorithm. Feature detectors and descriptors come in handy in applications such as object recognition, 3D reconstruction, stereoscopic vision, motion tracking, robot vision, image stitching .Interest points or feature points are determined at various scales and orientation. A good feature point should be pose invariant and should give acceptable results for scale, orientation and position. It should have high detecting ability and low false positive rate and should give a good detection even when the viewing conditions are changed. A local feature detector should be preferred. A feature descriptor should be distinctive and insensitive to local deformation. Therefore, each feature model has to consider location, scale and orientation as the key parameters. A good analysis of these descriptors is given by[6]. He compares the different descriptors based on the recall and 1-precision graphs. Feature detectors such as Harris detector used for the detection of corner points, Harris Laplace, Hessian Laplace use local maxima and Laplacian of Gaussian to determine scale space. Descriptors such as SIFT, GLOH, Jet, Image Moments, Shape Contexts have also been depicted. But, SIFT outperforms the other considerably and hence the choice for this paper. Histogram of Oriented Gradients proposed by[7] has been used as a descriptor to detect and learn important features, to recognize and detect a human figure in the given image.

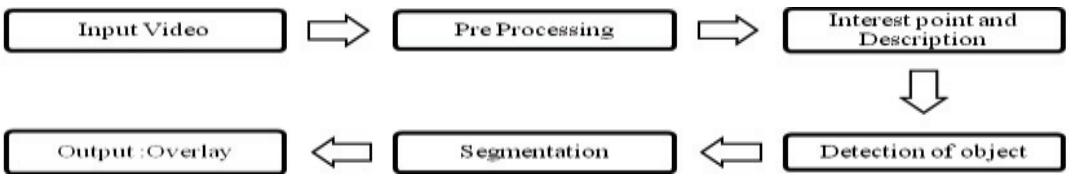

Fig. 1. Algorithm

3.3. Detection of Object

The steps involved in the detection of the object under consideration are as follows: The implementation uses the default values as elucidated in[7].

3.3.1. Gaussian smoothing was applied to an image followed by the application of a one dimensional derivative mask such as [-1 +1], [-1 0 +1], [1, −8, 0, 8, −1]. In this implementation, the mask [-1 0 +1] was used with standard deviation of eight.

3.3.2. A detection window is considered with size of 64 x 128 and is divided into 16 x 16 blocks of four 8 x 8 pixels cells. Here, a block of 8 pixels has been used.

3.3.3. Each pixel then determines the vote for an edge orientation histogram based on the orientation of the gradient element centered on it. The orientation bins are evenly spaced between 0-180 giving 9 bins in total.

3.3.4. The descriptor blocks are normalized using L2-Hys (Lowe-style clipped L2 norm) block normalization and are called Histogram of Oriented Gradients. The detection window has an overlapping grid of these HOG descriptors and has combined feature vector with block spacing stride of 8 pixels (hence 4-fold coverage of each cell).

3.3.5. SIFT and HOG representation acquires edge and gradient structure in the local context and thus it is invariant to the local geometric and photometric transformations.

3.3.6. A linear SVM classifier has been used to classify the human and non-human objects. Note: The 64 x 128 pixel detection window will be divided into 7 blocks across and 15 blocks vertically, for a total of 105 blocks. Each block contains 4 cells with a 9-bin histogram for each cell, for a total of 36 values per block. This brings the final vector size to 7 blocks across x 15 blocks vertically x 4 cells per block x 9-bins per histogram = 3,780 values.

Fig. 2. HOG detectors cue mainly on silhouette contours (especially the head, shoulders and feet). (a) The average gradient image over the training examples; (b) A test image; (c, d) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights. Image Courtesy: Navneet Dalal[23]

3.4. Segmentation

The algorithm behind the segmentation of the object under consideration is as follows:

The interactive segmentation of an object in an image proposed by[22] has been exploited in this paper. Interactive segmentation techniques include Intelligent Scissors, Bayes Matting, Magic Wand, Knockout 2, Level Sets and GraphCuts [8]. Grab Cut is developed on Graph Cut. In this paper, the Graph Cut approach has been hacked to give an automated segmentation result. The results though are not that accurate, give a good segmentation of the foreground from the background.

Algorithm: The OpenCV implementation of grabcut is follows: A user defines a rectangle with four coordinates. Everything inside this rectangle is assumed to be foreground and everything outside corresponds to background. An initial labelling (hard labelling) is done depending upon the given input. A classic pixel based energy function can be formulated either using a Markov Random Field or Gaussian Mixture model to model the foreground and the background. Grabcut is based on GMM. The following gives the necessary equations for the GMM and MRF model required in the segmentation problem as specified in[7].

$$E(f) = \Sigma E_r(i,j) + E_b(i,j) \quad (1)$$

Where $E_r(i,j)$, is the region term and $E_b(i,j)$, is the boundary term in an image I with (i,j) being the pixel location used in binary MRF energy optimization. GMM learns and creates new pixel distribution. The unknown pixels are labelled as either probable foreground or probable background. A graph is realized using this pixel distribution with pixels being the nodes. Two nodes are added, known as the Source Node and Sink Node. It is important to note that every foreground pixel is connected to the Source Node and every Background Pixel is connected to the Sink Node. Weights are assigned to the edges between the connecting pixels and it depends upon the probability of a pixel being either foreground or background. Edges are weighted depending upon the similarity between the two given pixels. If the difference in the colour is large then a low weight is assigned. The mincut algorithm is used to segment the graph. It cuts the graph with a minimum cost function and the cost function is the sum of all the weights of the edges. All the pixels connected to the Source Node become foreground and the pixels connected to the Sink Node become the background. The process continues until convergence.

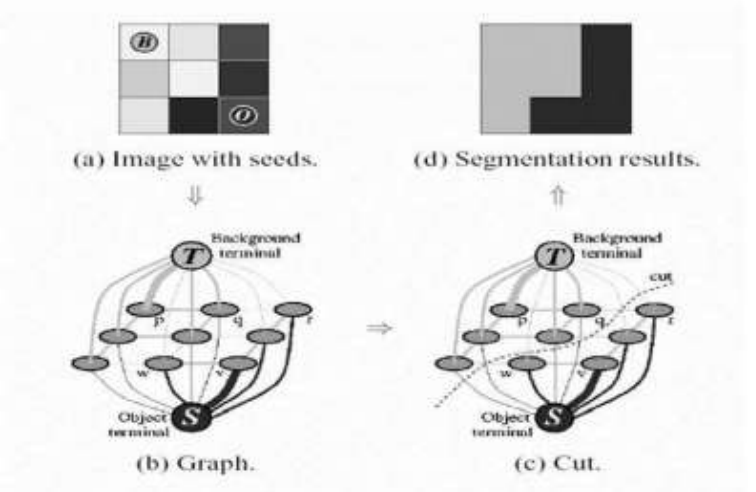

Fig. 3. The graph cut algorithm which segments out the foreground and background. Note the Source and the Sink Nodes in the graph. Image Courtesy: Boykov Y[8].

4. Procedure

The algorithm has been developed in C++ using OpenCV libraries and other necessary packages. Currently, it has a desktop version running on Ubuntu 14.04.

STEP 1: Grab all the frames of the given video (refer this as Video 1) and iterate through each of them.

STEP 2: Apply the methodology given in 3.1, 3.2 and 3.3 to detect and track the subject viz., the human body, on each of the frame of video 1.

STEP 3: Segment out the human body from STEP 2 using the method explained in 3.4. This again has to be done for each frame by iterating through all of them. In Video 1, the tracked human figure is surrounded by a rectangle. In OpenCV, the coordinates of this rectangle are specified by stating one coordinate and length and breadth of the rectangle. This rectangle is considered as the input for the GrabCut algorithm which then specifies the labelling for the foreground and background models. The number of iterations in the application of GrabCut could be increased to give a much better segmented output.

STEP 4: Save the frames so obtained after the segmentation in STEP 3 as a video (refer this as Video 2). The video is saved with a black mask i.e. the foreground in this video is the segmented object obtained from STEP 3 and the background is black.

STEP 5: Get a live camera feed or another pre-recorded video (refer either of them as Video 3). Iterate through each of the frame in this video and simultaneously in the video 2. Remove the mask from each of the frame in Video 3. Overlay Video 2 on Video 3 by copying frames onto each other. Alpha blending is required at this step to give an effect of holographs. The alpha channel is changed according to the level of transparency required.

5. Results and Discussion

The proposed methodology has been tested on many videos. The duration of each video along with its specifications such as resolution, frame rate has also been noted. The frame rate is kept constant at 24 fps. The image or the frame dimension is 640 x 480 pixels. Use the package ffmpeg[23] to encode and decode multimedia files. Here, ffmpeg is used to adjust the frame sizes and frame rate of the images and videos. Figures 4 through 7 describe the various stages and scenarios considered in this experiment. The timing analysis has been done. The time duration has been recorded for the input of the frame, the time taken for the execution of the algorithm, the time taken to save the final augmented video as a video file.

The duration of the sample video from which the object is to be detected is 18s. Now, as per new algorithm, when the program is executed, the results obtained are discussed as follows; Time taken to:

1]Open the sample video: 235 ms (235 ms average across 1 run)

2]Read each frame: 0 ms (0 ms average across 566 runs)

3]Writing each frame or saving the frame to get the final video: 1ms (2ms average across 183 runs)

4]Execute the algorithm for detection, tracking and segmenting the object under consideration: 682 ms (600 ms average across 183 runs).

where, run refers to the number of iterations of each command or the number of iterations in each loop. The final augmented video is saved with an appropriate file format. Also, note that the audio is intact along with the video. This can be achieved through the usage of Gstreamer[24]. A package is used to encode audio along with the video

Fig. 4. (a) Image Background; (b) Required Object to be segmented; (c) Overlay of the segmented object

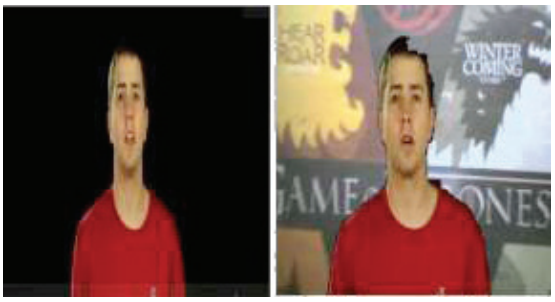

Fig. 5. (a) Image of a person; (b) Overlay of Segmented Object

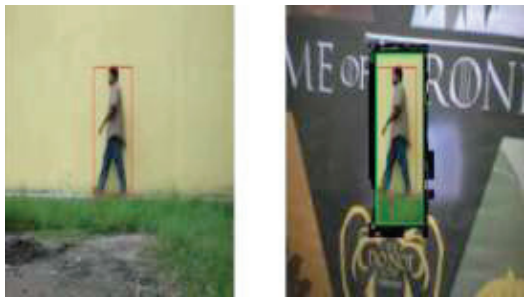

Fig. 6 . (a) Image of a frame; (b) Overlay of the video

Figure 4(a) represents an image background. 4(b) is an image from which the required object is to be detected, tracked and segmented. Here, the object is the three apples. 4(c) represents the overlay of the segmented object over the image background. Figure 5(a) represents an image of a person in a video. This is a video frame. 5(b) The segmented object i.e. the person is overlaid on another video running in the background. Here, shown as a frame.

Figure 6(a) represents a video frame which consists of a human figure. It is detected, tracked and segmented. 6(b) The segmented object of interest is then overlaid on another video. Figure 8 depicts the properties and usefulness of the alpha channel of an image. It is used to set the level of the transparency of an image. Here, this property is to blend two videos to give an effect of holographic videos. A screenshot of the frame has been shown in this figure b. Figure 8 further elucidates the concepts of alpha channel. (a) alpha = 0.1 (b) alpha = 0.3 and (c) alpha = 0.6.

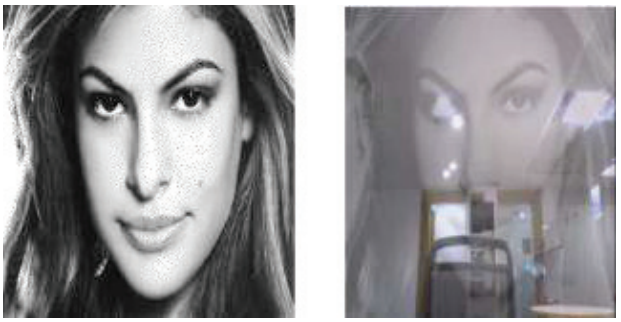

Fig.7. (a) Alpha channel of an Image; (b) Blend two videos to give holographic videos

Fig. 8. (a) Alpha = 0.7; (b) Alpha = 0.3; (c) Alpha = 0.6

6. Conclusion

The paper dealt with the development of an augmented reality system and has led to new research and advancements in this field with applications in gaming, medical sciences, video and sound production. The concepts of Green screen can be mingled to get yet another plethora of applications. The system can detect, track and segment out an object of interest in real time. SIFT and HoG descriptors were used to detect the interest points and extract the features from the video image. This was then further used to recognize real world objects and classify the same using Support Vector Machine. The segmentation was done using GrabCut algorithm. Homography techniques were used to determine the pose matrix to control the rotation, scaling and translation of the virtual object. The proposed algorithm was tested as a Desktop version on Ubuntu 14.04 OS. The augmented reality application developed was based on the marker less approach. The future work includes the execution of the proposed algorithm for handheld devices using the fastCV libraries developed by Qualcomm on Android and IOS devices. This could give more insights for real world deployment and subsequent scope for improvement of the proposed algorithm. The adapted algorithm has been found to work efficiently, accurately and reliably, but it has scope for further improvement by the utilization of better computationally efficient descriptors and interest point

detectors such as STIP. Also, we have limited our scope to 2D applications and efforts should be made so that this work could be used with other graphic utilities to render 3D systems.

References

1. Loris D'Antoni , Alan Dunn , Suman Jana , Tadayoshi Kohno , Benjamin Livshits , David Molnar , Alexander Moshchuk , Eyal Ofek, Franziska Roesner, Scott Saponas, Margus Veanes and Helen J. Wang. *Operating System Support for Augmented Reality Applications*; 2013.
2. Edmund Ng Giap Weng, Rehman Ullah Khan, Shahren Ahmad Zaidi Adruce and Oon Yin Bee. *Objects tracking from natural features in mobile augmented reality.* The 9th International Conference on Cognitive Science, Vol. 97, 2013, pp. 753–760.
3. Kato, H. and M. Billinghurst. *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*;Proceedings 2nd IEEE and ACM Internationa Workshop on Augmented Reality, IWAR'99, 1999, pp. 85 – 94.
4. Vacchetti, L., V. Lepetit and P. Fua. *Combining edge and texture information for real-time accurate 3d camera tracking;* Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality,2004,pp.48-57.
5. Pressigout, M. and E. Marchand. *Hybrid tracking algorithms for planar and non-planar structures subject to illumination changes;* IEEE/ACM Int. Symp. on Mixed and Augmented Reality, 2006, pp 52-55.
6. Subhransu Maji .*A Comparison of Feature Descriptors*; University of California at Berkeley, Department of EECS,University of California, Berkeley,2006
7. Carsten Rother and Vladimir Kolmogorov. *GrabCut—Interactive Foreground Extraction using Iterated Graph Cut*; Microsoft Research Cambridge, UK ,Andrew Blake,2012.
8. Boykov Y., Jolly M.P. *Interactive graph cuts for optimal boundary and region segmentation of objects in N-D image*; Proceedings of International Conference on Computer Vision, Vancouver, Canada,vol.I,2001,pp. 105-112.
9. Pyry Matikainen , Martial Hebert and Rahul Sukthankar. *Representing Pairwise Spatial and Temporal Relations for Action Recognition*; The Robotics Institute, Carnegie MellonUniversity, Intel Labs Pittsburgh,2010,pp.1-14.
10. Reitmayr, G. and D. Schmalstieg. *Location based applications for mobile augmented reality*; 4th Australasian User Interface Conference , Adelaide, Australia: Australian Computer Society, Inc.,2010,pp.1-8.
11. Pielot M., Nickel N.H.,, C., Menke C., Samadi S., and Boll S. *Evaluation of Camera Phone Based Interaction to Access Information Related to Posters in Mobile Interaction with the Real world*; 2009, pp.61-72.
12. Boykov Y and Kolmogorov V. *Computing Geodesics and Minimal Surfaces via Graph Cut*; In Proc. IEEE Int. Conf. on Computer Vision,2003.
13. Empster D, Laird, A. L AIRD M., Rubin D. *Maximum likelihood from incomplete data via the EM algorithm*; J. Roy. Stat. Soc. B., 1977, pp.39, 1–38.
14. Kolmogorov V and Zabih R. *What energy functions can be minimized via graph cuts?*; Proc. ECCV. CD–ROM,2002.
15. Paul Viola and Michael Jones, " *Robust Real-time Object Detection*", In International Journal of Computer Vision, 2001.
16. Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., Moore R., Kipman A., and Blake A. *Real-time human pose recognition in parts from a single depth image*; Computer Vision and pattern recognition 2011,pp1.8
17. Harris-Affine, Hessian Affine, Mikolajczyk K. and Schmid C. *Scale and Affine invariant interest point detectors*; International Journal of Computer Vision,Vol.60, No.1,2004,pp.63-86.
18. Mikolajczyk K. and Schmid C. *A performance evaluation of local descriptors*; IEEE Transactions on Pattern Analysis and Machine ntelligence, Vol.27,No.10,2005, pp.1615-1630.
19. Gerhard Reitmayr, Tom W Drummond. *Going out: robust model-based tracking for outdoor augmented reality*; 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2006.
20. Wuest, H., F. Vial, and D. Strieker. *Adaptive line tracking with multiple hypotheses for augmented* reality; IEEE and ACM International Symposium on Mixed and Augmented Reality, 2005.
21. Comport, A.I., E. Marchand, and F. Chaumette. *A real-time tracker for markerless augmented reality*; ISMAR '03, IEEE Computer Society, 2003.
22. Rother, C., Kolmogorov, V., and Blake, A. *GrabCut –interactive foreground extraction using iterated graph cuts*; ACM Transactions on Graphics (Proc. SIGGRAPH2004), 23(3):309–314.
23. Navneet Dalal, Bill Triggs. *Histograms of Oriented Gradients*; CVPR, 2005.
24. FFmpeg (www.ffmpeg.org)
25. GStreamer (www.gstreamer.freedesktop.org)