Face Detection for Augmented Reality Application Using Boosting-based Techniques

Youssef Hbali[1], Lahoucine Ballihi[2], Mohammed Sadgal[1], El Fazziki Abdelaziz[1]

[1]Cadi Ayyad University. B.P. 2390, Avenue Prince My Abdellah, Marrakech, Morocco
[2]LRIT-CNRST URAC 29, Mohammed V University In Rabat, Faculty of Sciences Rabat, Morocco

***Abstract* — Augmented reality has gained an increasing research interest over the few last years. Customers requirements have become more intense and more demanding, the need of the different industries to re-adapt their products and enhance them by recent advances in the computer vision and more intelligence has become a necessary. In this work we present a marker-less augmented reality application that can be used and expanded in the e-commerce industry. We take benefit of the well known boosting techniques to train and evaluate different face detectors using the multi-block local binary features. The work purpose is to select the more relevant training parameters in order to maximize the classification accuracy. Using the resulted face detector, the position of the face will serve as a marker in the proposed augmented reality.**

***Keywords* — Local Binary Pattern, Boosting, Supervised Learning, Face Detection, Augmented Reality.**

I. Introduction

Object detection and tracking is an important task for computer vision applications. With the growth of computers' power and the proliferation of high quality and low cost video cameras, industries like games, medical, media, automotive and education have gave more importance to this field of com- puter vision and started to take advantage of this technology by enhancing their products with more intelligence.

Many applications have shown an important impact on the daily human life, for example, to reduce the number of accidents on roads, vehicle manufactures have developed a clever system that detects and warns the vehicle drivers in the event of tiredness [1] [2] [3]. A small camera is placed in the cab of the vehicle, the images taken by the camera are used to measure the position of the head and rotation, and to check if eyes are closed or opened. By introducing the Kinect sensor to the game industry, Microsoft was able to position the controller-free gaming device as en entirely new way to experience entertainment in the living room. With Kinect,games-players no longer need to memorize different commands for a hand-held control, they are the controllers themselves [4] [5]. For a visual tracking algorithm to be useful in real-world scenarios, it should be designed to handle and overcome cases where the target's appearance changes from frame-to-frame. Significant and rapid appearance variation due to noise, occlusions, background clutter, pose, scale and illumination changes are the major challenge situations that a detector needs to overcome. Many novel methods have been proposed to resolve each of these variations [6] [7]. The accuracy of a trained face detector is heavily related to the data and algorithm used for the training. In this paper we highlight the importance of the choice of the training parameters values and show how this choice impact the accuracy of the resulted detector.

A. Face detection problem

Human emotions like sadness, happiness and anger are often expressed throw the face, these facial expressions make the human face a very dynamic body part. This high degree of variation combined with pose, scale and illumination changes makes of face detection a difficult problem. Over the two past decades, face detection problem has been an attractive research area for the computer vision community. Real time face detection was made possible since the publication of the seminal approach of Viola and Jones [8], in which they used a cascade of increasing complexity classifiers to detect up- right faces. The face detector accuracy depends not only on the features used for the face representation, but also on the training data and parameters.

B. Objective and Contribution outline

Training an accurate boosting model requires a data-set with a high degree of variation and a fine tuning of the training parameters. In this work we revisit the face detection problem to find the best training parameters that lead to an accurate face detector, the impact of each training parameter is examined by training classifiers with different parameter values. And to overcome the drawback of using markers in augmented reality applications, we integrate our face detector in a 3D augmented reality where the position of the face is used as a marker-less object for placing 3D models.

The contributions of this paper are :

1. Fine-tuning a boosting based detector by varying the training parameters values.
2. Integration the face detector model and a 3D blender model in the Ogre 3D framework to overcome the draw- back of markers based augmented reality applications.

1- fine-tune a boosting based detector by varying training parameters values and 2- Integrating a 3D blender model in a real time augmented reality.

II. Related Work

Classifiers are built by taking a set of labeled examples and using them to come up with a rule that will assign a label to any new example. In the general problem, we have a training data set (,); each of the xi consists of measurements of the properties of different types of object, and yi are labels giving the type of the object that generated the example. In this paper we will use different learning-based techniques like decision tree learning [9] that is one of the most widely used and prac- tical methods for inductive inference. The boosting [10] is one of the most popular learning techniques, widely used for object detection, it consists of combining many weak learners to form a strong classifier. For this study will experiment classification using Gentle AdaBoost, Real AdaBoost and LogitAdaboost learning techniques, the algorithms train models sequentially, with a new model trained at each round. At

DOI: 10.9781/ijimai.2016.424

the end of each round, miss-classified examples are identified and have their emphasis increased in a new training set which is then fed back into the start of the next round, and a new model is trained. Viola and Jones [8] introduced the approach of cascade of boosted classifiers. [11] proposed random forests, which is a collection of random trees (RT). Random trees are structurally identical to classical decision trees but are trained differently. During training not an exhaustive search of the possible test candidates is considered but only a randomized subset in order to allow for creating several different and independent random trees.

III. Image Representation

A. The original LBP

The local binary pattern (LBP) [12] [13] is defined as a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood. The original LBP operator labels the pixels of an image by thresholding the *3- by-3* neighborhood of each pixel with the center pixel value and considering the result as a binary number. The decimal result is the sum of, the thresholds multiplied by their weights values, as it can be seen in Fig. 1.

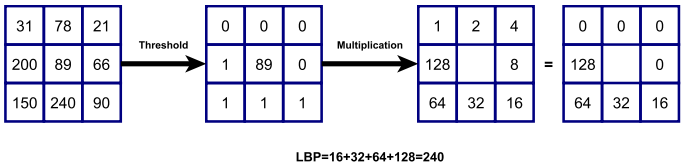

Fig. 1. The original local binary pattern calculation, the central pixel is compared with its neighbors, the thresholded values are then multiplied by a power of 2^i, where i is the pixel index, the sum of all values results to the LBP code.

In other words given a pixel position (x_c, y_c), LBP is defined as an ordered set of binary comparisons of pixels intensities between the central pixel and its surrounding pixels.

The resulting label value of the 8-*bit* word can be expressed as follows :

$$LBP(x_c, y_c) = \sum_{n=0}^{7} t(l_n - l_c)2^n \tag{1}$$

where l_c corresponds to the gray value of the central pixel, l_c the gray value of the neighbor pixel n, and function $t(k)$ is defined as following :

$$t(k) = \begin{cases} 1, & \text{for } k \geq 0 \quad (2) \\ 0, & \text{for } k < 0 \quad (3) \end{cases}$$

According to (2), the LBP code is invariant to monotonic gray- scale transformations, thus the LBP representation may be less sensitive to illumination changes.

The 256-bin histogram of the labels computed over an image can be used as texture descriptor. Each bin of histogram (LBP code) can be regarded as micro-texton and the histogram characterizes occurrence statistics of simple texture primitive. The histogram of the labeled image $f_l(x, y)$ can be defined as:

$$H_i = \sum_{x,y} I(f_l(x, y) = i), i = 0, .., L - 1 \tag{4}$$

where L is the number of different labels produced by the LBP operator and $l(A)$ is 1 if A true and 0 otherwise.

B. Multi-block LBP

The LBP operator has been extended to consider difference between blocks, the MB-LBP [14] operator is defined by comparing the central rectangles average intensity g_c with those of its neighborhood rectangles g_0, ..., g_8. In this way, it can give us a binary sequence. An output value of the MBLBP operator can be obtained as follows:

$$MBLBP = \sum_{i=1}^{8} t(g_i - g_c)2^i \tag{5}$$

where g_c is the average intensity of the center rectangle, $g_i (i = 0, .., 8)$ are those of its neighborhood rectangles. Fig. 2 demonstrates how the MB-LBP features are calculated.

IV. Fine-tuning Training Parameters

A. Data set preparation

To train the classifiers we use a subset of the FERET [15] face database.

Fig. 3. Samples of the cropped training faces from the Feret dataset.

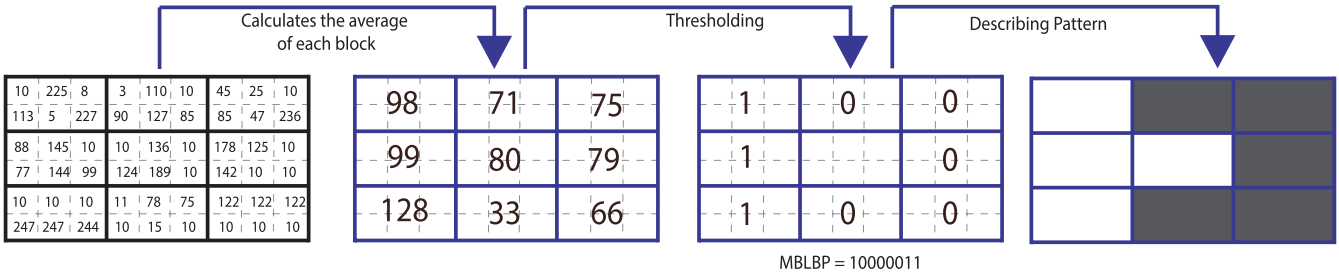

Fig. 2. The multi-block variant of the local binary pattern.

The database contains 14051 face images of over 1000 subjects and has variations in expression, lighting, pose and acquisition time. Only frontal views of the different faces have markups of eyes and mouth position, these markups, see Fig.4 are used to crop the face from the image. Thus the training data set is reduced to 5324.

Fig. 4: Feret labeled sample.

For the testing purpose we use the BioID Face database [16]. The dataset consists of 1521 gray level images with a resolution of 384286 pixel. Each one shows the frontal view of a face of one out of 23 different test persons. There are 20 manually placed points on each image. The markup scheme is shown on Figure 5.

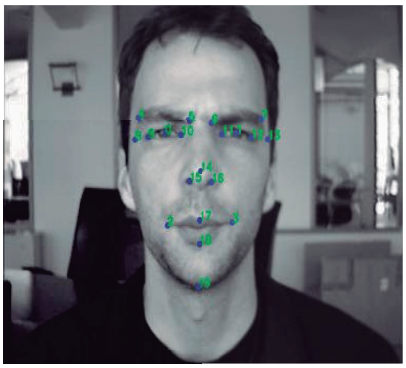

Fig. 5. BioID labeled face image sample.

The top left corner point of the ground truth face (the region of interest) is obtained by subtracting 10 pixel from the left temple x coordinate, index 8 on the figure 5, and by adding 15 pixel to the highest eye brow point, index 7 in this image example. The right bottom corner point of the region of interest is obtained adding 10 to the right temple x coordinate and taking the tip of chin y coordinate, index 19 on the image , as the y coordinate.

TABLE I
List of Labels Indexes and Their Description

Label index	Description
0	Right eye pupil
1	Left eye pupil
2	Right mouth corner
3	Left mouth corner
4	Outer end of right eyebrow
5	Inner end of right eyebrow
6	Inner end of left eyebrow
7	Outer end of left eyebrow
8	Right temple
9	Outer corner of right eye
10	Inner corner of right eye
11	Inner corner of left eye
12	Outer corner of left eye
13	Left temple
14	Tip of nose
15	Right nostril
16	Left nostril
17	Centre point on outer edge of upper lip
18	Centre point on outer edge of lower lip
19	Tip of chin

B. Experiments & Results

The choice of the training parameter values has an important impact on the trained classifiers accuracy. For the Multi-Blocklocal binary pattern features we choose a range of parameter values to apply and we plot a roc curve of each classifier to highlight the influence of each parameter.

1) Minimum hit rate parameter:

The first set of experiments consists of varying the diffrent values used for the parameter minHitRate, the minimum desired hit rate for each stage of the classifier. The Overall hit rate can be estimated as: $(MinHitRate^{number-Of\ -stages})$. Table II lists some values used for the training.

TABLE II. Variation of the Minimum Hit Rate
Parameter to Train MB-LBP Based Classifiers

Classifier	MinHitRate
C_1	0.5
C_2	0.6
C_3	0.7
C_4	0.8
C_5	0.9

2) Maximum false alarm parameter:

In the second set of experiments, we vary the values of the parameter **MaxFalseAlarm**, the maximum desired false alarm rate for each stage classifier. The Overall false alarm rate is estimated as: $(MaxFalseAlarm^{number-Of-stages})$. Table III lists the parameter values used for each feature. Table III lists the different values used for this parameter.

TABLE III. Variation of the Maximum
False Alarm Parameter to Train MB-LBP Classifiers

Classifier	MaxFalseAlarm
C_1	0.3
C_2	0.4
C_3	0.5
C_4	0.6
C_5	0.7

3) Max depth parameter:

In the third set of experiments, we vary the values of the parameter **maxDepth**, the maximum depth in each single weak classifier. Table IV lists the different values used for this parameter.

TABLE IV. Variation of the Maximum
Depth Parameter to Train MB-LBP Classifiers

Classifier	Max Depth
C_1	1
C_2	2
C_3	3
C_4	4

4) The boosting variant parameter:

In the last set of experiments, we vary the values of the boosting type parameter **Boost Type**, Table V lists the different values used for this parameter.

C. Performance evaluation

The resulting classifiers have been applied on the BioID Face database, we present the detection performance by the Receiver Operating Characteristic curves [17].

TABLE V. The Different Boosting Algorithms Variants Used to Train the MB-LBP Features Based Classifiers

Classifier	Boosting variant
C_1	Gentle Adaboost (GAB)
C_2	Discrete Adaboost (DAB)
C_3	Real Adaboost (RAB)
C_4	Logit Adaboost (LB)

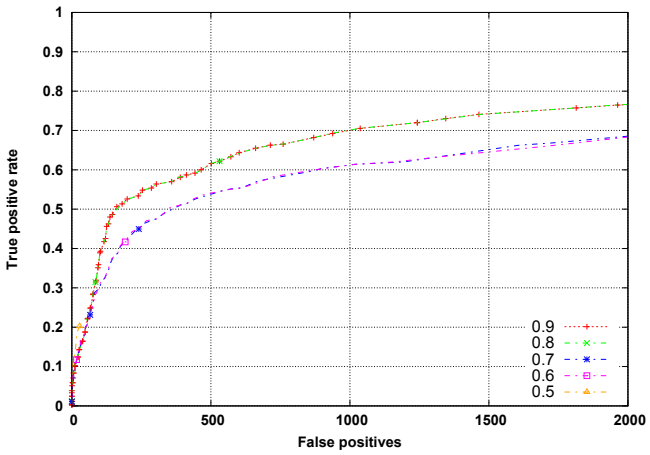

Fig. 6. Variation of the minimum hite rate parameter values for the MB-LBP features based classifiers.

The ROC curves shown in the figures 6, 7, 9 and 8 are obtained by scoring the detected windows on each test image and applying a threshold to decide if the detected window is a face or not. To represent the degree of match between a detection d_i and an annotated region I_j, we employ the commonly used ratio of intersected areas to joined areas:

$$S(d_i, l_i) = \frac{area(l_i) \cap area(d_i)}{area(l_i) \cup area(d_i)} \quad (6)$$

And we use a slightly modified version of the evaluation tool provided by [18], in where we use a rectangular face annotation rather than an elliptical one.

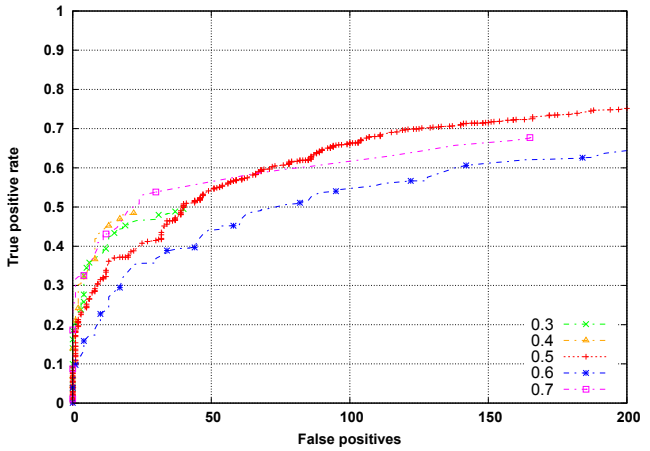

Fig. 7. Variation of the maximum false alarm parameter values for the MB-LBP features based classifiers.

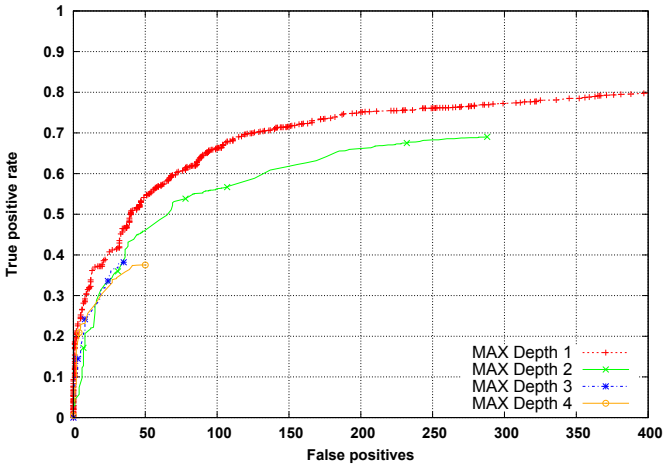

Fig. 8. Variation of the maximum depth parameter (Maximum depth per weak learner) values for the MB-LBP features based classifiers.

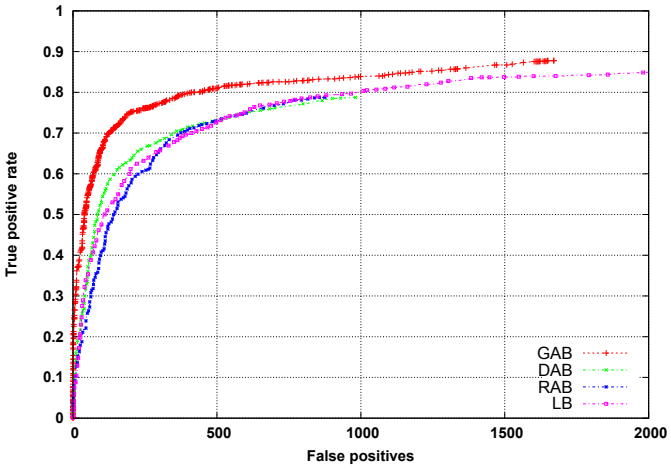

Fig. 9. Different variants of boosting algorithm for the MB-LBP features based classifiers.

D. Results discussion

Figure 6 shows how the values 0:9 and 0:8 of the minimum hit rate parameter give the best result for the Multi-block local binary features. Choosing the value 0:7 for the maximum false alarm parameter gives the best result in the second set of experiments as shown in Figure 7. In Figure 9 we see how applying the Gentle AdaBoost variant gives the more accurate results in the third set of experiments. Finally, the figure 8 shows that a classifier based on weak learner of only one depth performs more accurately than a classifier with deeper weak learners.

These different experiments have shown how a single parameter value can influence the accuracy of a face classifier and for each type of object detection, user might need to use multiple combinations of the different parameters values to find an accurate classifier.

V. Application to Augmented Reality

Augmented Reality (AR) employs computer vision, image processing and computer graphics techniques to merge digital content into the real world. It enables real-time interaction between the user, real objects and virtual objects. AR can, for example, be used to embed 3D graphics into a video in such a way as if the virtual elements were part of the real environment. This technology has known an increasing interest in many fields and it has been explored in the e-commerce

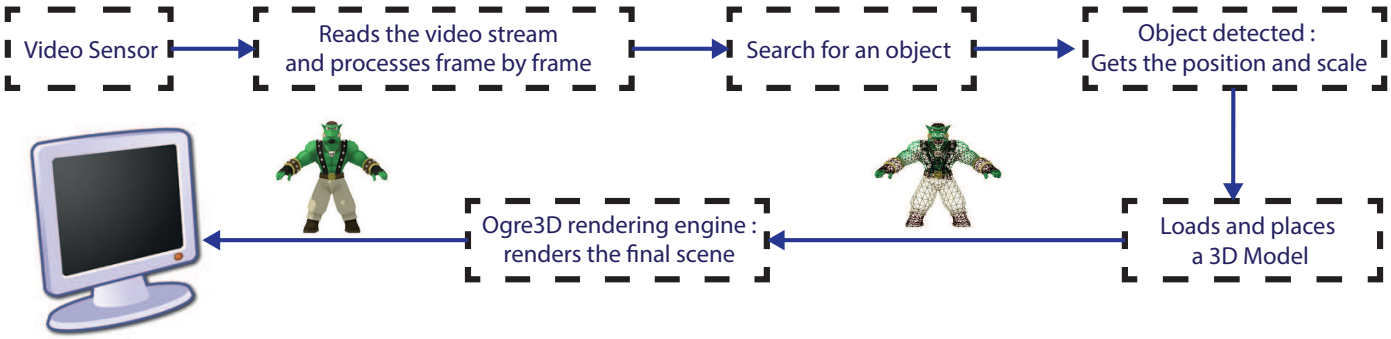

Fig. 10. Augmented reality application workflow.

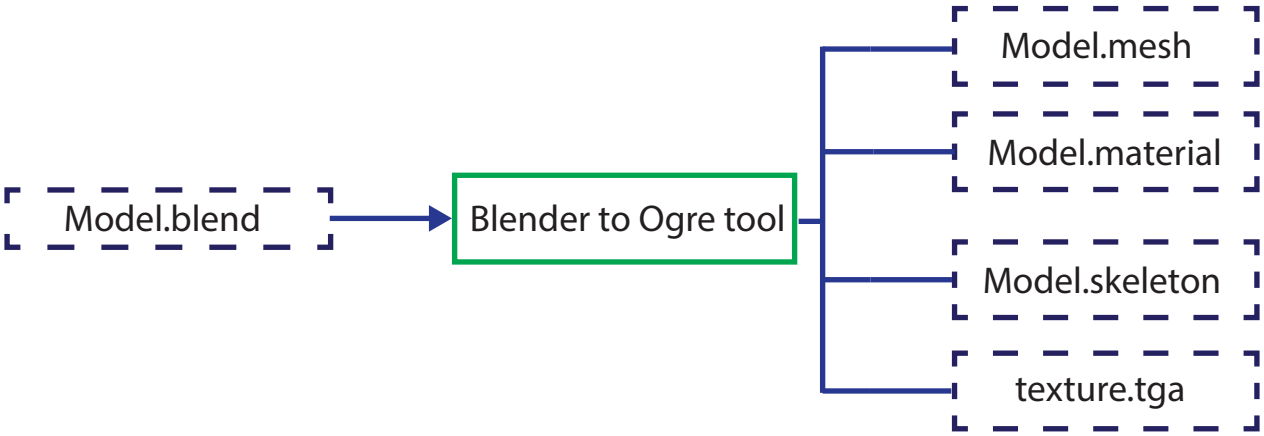

Fig. 11. Converting the blender 3D model to Ogre 3D format.

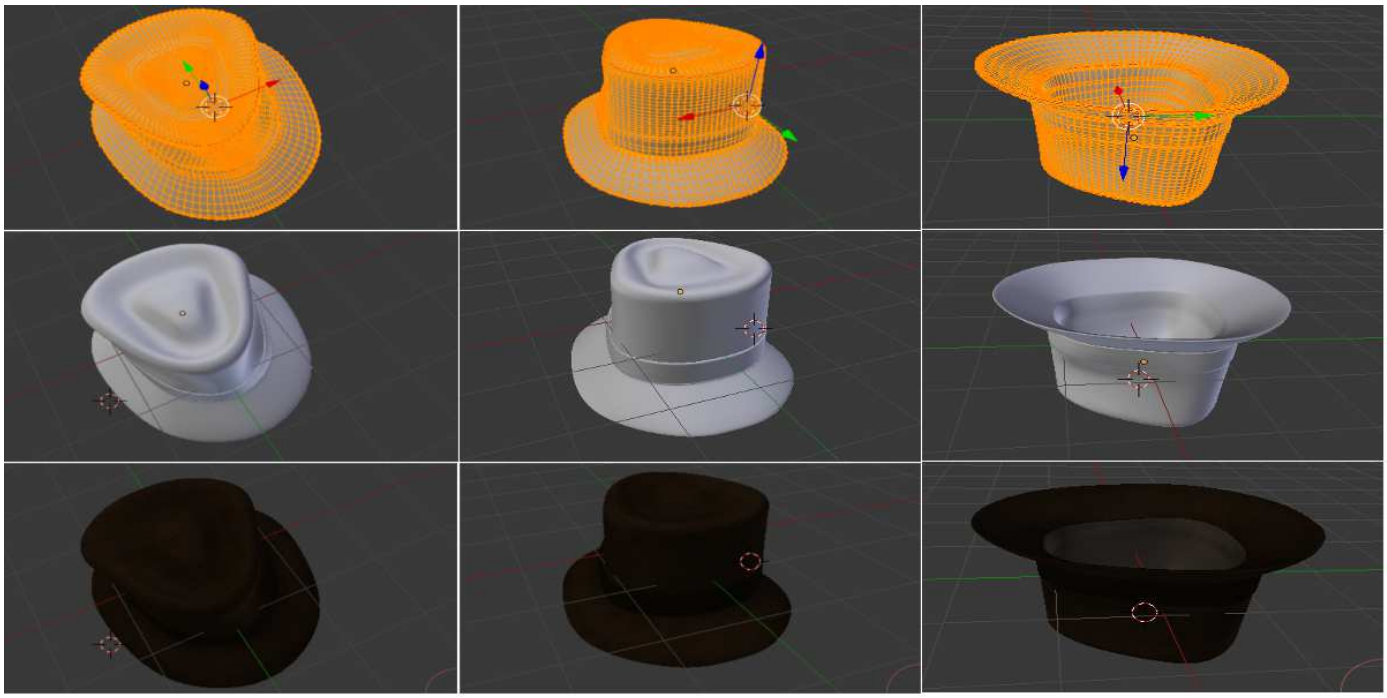

Fig. 12: The Blender 3D hat model.

applications [19] where clients try clothes online, discover and test itmes they are interested in without having to move to a store. It was also applied in the e-learning [20] where the classic teaching way can be mixed with the augmented reality to make students having fun while learning new things in a new manner.

For augmented reality, marker-less model-based tracking approaches [21] appear to be the most promising among the standard vision techniques currently applied in AR applications. While marker-based approaches such as ARToolkit [22] or commercial tracking systems such as ART provide a robust and stable solution for controlled environments, it is not feasible to equip a larger outdoor space with fiducial markers. Hence, any such system has to rely on models of natural features such as architectural lines or feature points extracted from reference images. The proposed augmented reality of this work

makes use of the powerful open source real time 3D rendering engine OGRE [23] and the 3D modeling tool Blender. OGRE enables a programmer to deal with the threedimensional graphical presentation of a particular application in a very object oriented manner and that is exactly what explain the name OGRE, Object-Oriented Graphics Rendering Engine. It acts as wrapper to the rendering subsystem (OpenGL or DirectX), allowing us to focus on the application rather than the rendering details.

Fig.10 shows the workflow of an OGRE based application, in which a standard web camera is use to capture the video stream, the trained face detector is then loaded using the open computer vision library [24] and used to detect faces present on each frame of the video. finally, ogre is used to place a 3D object over the face region and the video stream is rendered to the user.

A. Modeling a 3D Hat

The 3D model that we use for our application is a 3D hat, see Fig.12. To model the hat we use the powerful modeling tool blender. The Ogre rendering engine uses meshes and skeletons for movable objects, in order to use the modeled hat, we need to convert blender format to resources that are managed by Ogre to render a 3D model. Figure 11 shows the generated resources from converting the blender 3D model to Ogre format.

B. Adding the 3D model to an Ogre scene

Every 3D rendering library uses a scene graph to organize its renderable items. This scene graph typically is optimized for fast search- ing and querying, providing the user with the ability to find items in the vicinity of other items, and allowing the library to find, sort, and cull polygons as needed in order to provide the most efficient rendering possible. For the proposed augmented reality application, the background of the scene will be the stream captured by the camera. Then for adding the 3D model to our scene, we process the camera stream frame by frame to detect the face position and to place the 3d model. Using the trained face detector, we process frame by frame to detect the face position. Fig. 13 shows the resulted scene, where the user is wearing a virtual 3D hat.

Fig. 13. A real scene augmeted by a 3D hat model.

VI. Conclusion

In this paper, we presented an augmented reality application using the 3D rendering engine Ogre. We experimented the different boosting techniques using the local binary pattern features by applying multiple values for the classifier training task. The chosen approach of using boosting techniques with a real time based augmented reality system has shown a satisfying results and has avoided the end user the burden of wearing any form of markers to interact with the computer.

The perspectives of this work are to use deep learning for hand gesture recognition and to apply the different gestures for an augmented reality applications in the domain of education. The institutions from rural zones suffers from the lack of funding for buying laboratory materials, virtual experiments can come for help to reduce the gap between theory and practice for subjects like physics and chemistry.

References

[1] A. Reichert, "Method and device for recognizing driver fatigue using a torque sensor system," Feb. 12 2013, uS Patent 8,374,750.

[2] J. Jo, S. J. Lee, K. R. Park, I.-J. Kim, and J. Kim, "Detecting driver drowsiness using feature-level fusion and user-specific classification," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1139–1152, 2014.

[3] C. Sun, J. H. Li, Y. Song, and L. Jin, "Real-time driver fatigue detection based on eye state recognition," *Applied Mechanics and Materials*, vol. 457, pp. 944–952, 2014.

[4] I.-C. Chung, C.-Y. Huang, S.-C. Yeh, W.-C. Chiang, and M.-H. Tseng, "Developing kinect games integrated with virtual reality on activities of daily living for children with developmental delay," in *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*. Springer, 2014, pp. 1091–1097.

[5] T. C. Davies, T. Vinumon, L. Taylor, and J. Parsons, "Let's kinect to increase balance and coordination of older people: Pilot testing of a balloon catching game," 2014.

[6] F. Moreno-Noguer, "Deformation and illumination invariant feature point descriptor," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1593–1600.

[7] P. Sharma, R. N. Yadav, and K. V. Arya, "Pose-invariant face recognition using curvelet neural network," 2013.

[8] P. Viola and M. Jones, "Rapid object detecting using boosted cascade of simple features," *in Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, 2001.

[9] F. Baumann, A. Ehlers, K. Vogt, and B. Rosenhahn, "Cascaded random forest for fast object detection," in *Image Analysis*. Springer, 2013, pp. 131–142.

[10] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on- line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[13] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 2037– 2041, 2006.

[14] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block lbp representation. icb," 2007.

[15] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The feret database and evaluation procedure for face-recognition algorithms," *Image and vision computing*, vol. 16, no. 5, pp. 295–306, 1998.

[16] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detec- tion using the hausdorff distance," in *Audio-and video-based biometric person authentication*. Springer, 2001, pp. 90–95.

[17] [M. H. Zweig and G. Campbell, "Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine." *Clinical chemistry*, vol. 39, no. 4, pp. 561–577, 1993.

[18] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," no. UM-CS-2010-009, 2010.

[19] W. Shen, "Augmented reality for e-commerce," *Applied Mechanics and Materials*, vol. 433, pp. 1902–1905, 2014.

[20] H.-K. Jee, S. Lim, J. Youn, and J. Lee, "An augmented reality- based authoring tool for e-learning applications," *Multimedia Tools and Applications*,

vol. 68, no. 2, pp. 225–235, 2014.

[21] G. Reitmayr and T. W. Drummond, "Going out: robust model-based tracking for outdoor augmented reality," in *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*. IEEE, 2006, pp. 109–118.

[22] G. Reitmayr and D. Schmalstieg, "Opentracker-an open software ar- chitecture for reconfigurable tracking based on xml," in *Virtual Reality Conference, IEEE*. IEEE Computer Society, 2001, pp. 285–285.

[23] Ogre Community, "Ogre3D," August 2006.

[24] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

Youssef Hbali is a PhD candidate in the field of computer vision at the Cadi Ayyad University in Marrakech, Morocco. He received his master's degrees in computer science from the Faculty of Sciences Semlalia in 2007. He is working as an ap- plication integrator and data scientist consultant for a large financial company in Paris, France. His current research interests include deep learning, augmented reality, and image processing.

Lahoucine Ballihi received the Master of Advanced Studies (MAS) degree in Computer science and telecommunications from Mohammed V University Rabat, Morocco in 2007 and the Ph.D degree in Computer Science from the University of Lille, France and the Mohammed V University Rabat, Mo- rocco in 2012. He is also a member of the Computer Science and Telecommunications Research Labora- tory of Mohammed V University (LRIT - CNRST URAC 29). He is currently an associate professor in Faculty of Science Rabat (FSR) at Mohammed V University In Rabat, Morocco. His research interests are mainly focused on statistical three-dimensional face analysis, actions recognition, face and gender recognition using RGB-D images.

Mohammed Sadgal is professor of computer sci- ence at Cadi Ayyad University, Morocco, and re- searcher on computer vision with the Vision team at the LISI Laboratory. His research interests include object recognition, image understanding, video anal- ysis, multi-agent architectures for vision systems, 3D modelling, virtual an augmented reality. Before Marrakech, he was in Lyon (France), working as Engineer in different computer Departments. He obtained a PhD in 1989 from Claude Bernard Uni- versity, Lyon, France.

Abdelaziz El Fazziki is a Professor of computer science at Marrakech University, where he has been since 1985. He received an MS from the University of Nancy (France) in 1985. He received his PhD in computer science from the University of Mar- rakech in 2002. His research interests are in soft- ware engineering, focusing on information system development. In the MDA arena, he has worked on agent based systems, and service oriented systems an decisional systems. He is co-author agent based image processing, and is the author of over fifty papers on software engineering.