증강현실 응용을 위한 자연 물체 인식
Natural Object Recognition for Augmented Reality Applications

안잔 쿠마르 폴*, 모하마드 카이룰 이슬람*, 민 재 홍*, 김 영 범*, 백 중 환**

Anjan Kumar Paul*, Mohammad Khairul Islam**, Jae-Hong Min***, Young-Bum Kim****, Joong-Hwan Baek*****

요 약

무마커 증강현실 시스템은 실내나 옥외 환경에서 자연 물체를 인식하고 매칭하는 기능이 필수적이다. 본 논문에서는 비주얼 서술자와 코드북을 사용하여 특징을 추출하고 자연 물체를 인식하는 기법을 제안한다. 증강현실 응용은 동작 속도와 실시간 성능에 민감하기 때문에, 본 연구에서는 멀티 클래스의 자연 물체 인식에 초점을 두었으며 분류와 특징 추출 시간을 줄이는 것을 포함한다. 훈련과 테스트 과정에서 자연 물체로부터 특징을 추출하기 위해 SIFT와 SURF을 각각 사용하고 그들의 성능을 비교한다. 또한, 클러스터링 알고리즘을 이용하여 다차원의 특징 벡터들로부터 비주얼 코드북을 생성하고 나이브 베이즈 분류기를 이용해 물체를 인식한다.

ABSTRACT

Markerless augmented reality system must have the capability to recognize and match natural objects both in indoor and outdoor environment. In this paper, a novel approach is proposed for extracting features and recognizing natural objects using visual descriptors and codebooks. Since the augmented reality applications are sensitive to speed of operation and real time performance, our work mainly focused on recognition of multi-class natural objects and reduce the computing time for classification and feature extraction. SIFT(scale invariant feature transforms) and SURF(speeded up robust feature) are used to extract features from natural objects during training and testing, and their performance is compared. Then we form visual codebook from the high dimensional feature vectors using clustering algorithm and recognize the objects using naïve Bayes classifier.

Keywords : Augment Reality, Object Recognition, Multiple Classification, SIFT, SURF, Bayes classifier

I. Introduction

Augmented reality has its important applications in different industries like production, manufacturing, servicing etc. The user can see the real world around him, with computer graphics superimposed or composited with the real world in augmented reality. It has different applications in automatic navigations, libraries, museum etc. [1]. It takes a real object or space as the foundation and superimposes augmented information on it, so that people can get information from the augmented data [2]. Suppose if the MRI image data can be superimposed onto a patient's body, then it can assist a surgeon to pinpoint a tumor that is to be removed. Augmented reality also might add audio commentary, location data, historical context, or other forms of content that can make a user's experience of a thing or a place more meaningful. Augmented reality has been put to use in a number of fields, like medical imaging, where doctors can access data about patients; aviation, where tools show pilots important data about the landscape they are viewing; and in museums, where artifacts can be tagged with information such as the artifact's historical context [1][2]. Augmented reality applications mostly based upon markers. The marker is attached to different objects and

* 한국항공대학교 ** 한국항공대학교 항공 전자 및 정보통신 공학부
투고 일자 : 2010. 4. 6 수정완료일자 : 2010. 4. 26
게재확정일자 : 2010. 4. 29
* This research is supported by the Gyeonggi Regional Research Center (GRRC) support program supervised by Gyeonggi Province, and partially by KAGERIIC.

by detecting the markers, computer finds out position of the object and generates the virtual object on it [3]. The main drawback of the marker based system is the size and appearance of the marker. Most of the cases the markers need to have certain dimensions and appearance [4]. Sometimes if there are multiple objects are existed, markers may be detected wrong with other similar kind of objects for outdoor augmented reality it is not convenient to attach markers to all objects like roads, buildings. So marker based system is not feasible for outdoor environment. Robust augmented reality system must have the capability to recognize and match natural objects both in indoor and outdoor environment [5]. The recognition and classification task of objects in different viewpoints are also very important because object may be rotated or scaled up and scaled down. Exact classification and recognition of objects help to superimpose augmented information to the objects properly. We emphasis our work for recognition of objects so that exact information can be augmented with them. To implement this kind of system, we adapted the bag of words concept. We use SIFT [10] and SURF [11] to extract the features from the object images Since the number of SIFT and SURF features extracted from the images is large and they have high dimensions, it is hard to use these features directly for recognition. So we use k-means clustering to form the cluster center using the extracted features. Then we use naïve Bayes classifier for learning and recognitions.

II. Object Classification System

Many researchers have used bag of words concept in object [6][7] and scene [8][9] recognition and also classification purpose. We adapted the concept of bag of words for object classifications and recognitions.

The block diagram in the Fig. 1 shows the steps involved in the total system. In the offline stage we extract the SIFT or SURF keypoints and describe them with 128 or 64 dimensional vectors. Then we use the k-means clustering algorithm over the training image descriptors to find out the cluster centers. We use mostly the total number of centers as 300, which is the size of codebook. We also tried with varying the codebook size. After that we use this codebook with naïve Bayes classifier. Then for testing we use sample images from test set. We extract the key points or feature points from the object images; these are considered as candidates for basic elements, "words". Each key point is described by certain dimensional vectors, which are the feature descriptors. A good descriptor must have the ability to handle intensity, rotation, scale and affine variations. The features should be very distinct, scale and rotation invariant, but not affected by viewpoint change. They should not take much time to extract the features. We use SIFT algorithm proposed by David Lowe [10] for extracting the features from images of objects. This is high dimensional feature vectors. We also use SURF[11] for reducing the dimensions of descriptors.

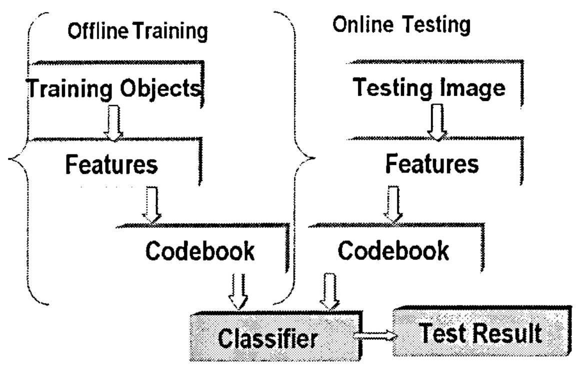


Fig. 1. System Block Diagram

III. Feature Detection and Representation

A. Scale Invariant Feature Transform (SIFT)

SIFT [10] is very popular for feature detection and description. SIFT is the combination of several steps. Scale-space extrema detection, keypoints localization, orientation assignment and keypoint descriptions are the major steps involved in SIFT processing. In scale space extrema detection step Gaussian filters at different scales are used to convolve the input image. Then the difference of successive Gaussian-blurred images is computed. Keypoints are determined by computing the maxima or minima of the difference of Gaussians (DoG) those occur at multiple scales. Scale space extrema detection produces too many candidate keypoints and some of these are unstable. Keypoint localization procedure detect the information of keypoints those are having low contrast, sensitive to noise and also having poor localization along an edge. SIFT allows to filter out these low contrast keypoints by setting up a threshold value. In our case we set up the threshold value as 0.04 experimentally because it gives about 200-300 keypoints from each sample images, which is reasonably good enough for classification. But it can be changed depend upon the applications. Rotation invariant is achieved by assigning each keypoint to one or more orientations based on their local image gradient directions. The feature descriptor is computed as a set of orientation histogram (4 x 4) pixel neighborhoods. The orientation histograms are relative to the keypoint's orientation. The orientation data comes from the Gaussian image closest in scale to the keypoint's scale. Each histogram contains 8 bins, and each descriptor contains a 4 x 4 array of 16 histograms around the keypoint. This makes to a SIFT feature vector with (4 x 4 x 8) =128 elements and we get each feature point as 128-dimensional vector. Fig. 2 shows the different steps involve in the SIFT operations and Fig. 3 shows extracted SIFT key points on the image.

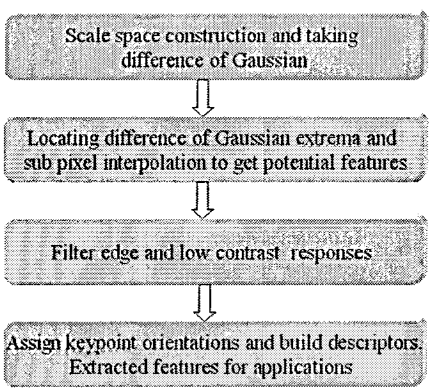


Fig. 2. SIFT operation steps

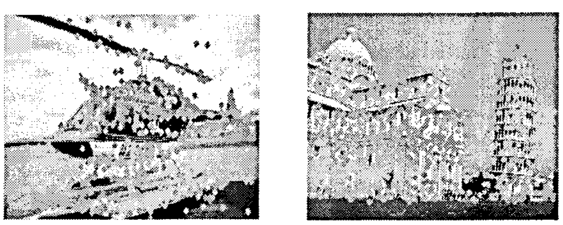


Fig. 3. SIFT features from the sample image

B. Speeded Up Robust Features (SURF)

Bay et al. [11] proposed SURF algorithm for key point detection and matching. Features are located using an approximation to the determinant of the Hessian matrix in SURF algorithm. It is used due to its stability and repeatability, as well as its speed. The Hessian is constructed by an ideal filter. It convolves the input image with the second-order derivatives of a Gaussian of a given scale. SURF uses the integral image concept; the integral image is computed rapidly from an input image and is used to speed up the calculation of any upright rectangular area. Given an input image I and a point (x, y) the integral image I_Σ is calculated by the sum of the values between the point and the origin. Formally this can be defined by the formula in equation (1) and shown in Fig. 4 [11].

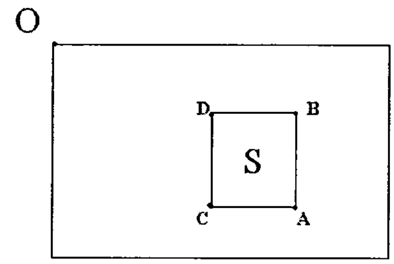


Fig. 4. Integral Image calculation

$$I_{\Sigma}(x,y)=\sum_{i=0}^{i\le x}\sum_{j=0}^{j\le y} I(x,y) \tag{1}$$

The sum of the intensities over any upright rectangular area which is independent of its size can be calculated by only four additions. If we consider a rectangle bounded by vertices A, B, C and D as in Fig.4 then the sum of the pixel intensities is calculated by S = A - B - C + D.

Given a point x=(x, y) in an image I, the Hessian matrix $H(x,\sigma)$ in x at scale σ is defined like this equation (2).

$$H(x,\sigma)=\begin{bmatrix} L_{xx}(x,\sigma) L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) L_{yy}(x,\sigma) \end{bmatrix} \tag{2}$$

Here $L_{xx}(x,\sigma)$ is the convolution of the Gaussian second order derivative $\frac{\delta^2}{\delta x^2} g(\sigma)$ with the image I in point x, and similarly for $L_{xy}(x,\sigma)$ and $L_{yy}(x,\sigma)$. SURF is scale invariant. For achieving scale invariance the filter size is varied in different scales. In SIFT there are the usage of variation of image sizes in different scales. In SURF there is the variation of filter size in different scales. The initial filter size 9 x 9 and here $s=\sigma$, it is gradually increased 9 x 9, 15 x 15, 21 x 21, and 27 x 27. The value of initial scale is $s=1.2$. For localizing the interest point in the image and over scales, the scale of the Gaussians used to derive the box filters. A minimum threshold values of H_o limits the total number of features according to equation (3). The location x_0 of each feature is then refined to sub-pixel accuracy shown in equation (4).

$$H(\mathrm{x}) = H + \frac{\delta H^T}{\delta \mathrm{x}}\mathrm{x} + \frac{1}{2}\mathrm{x}^T \frac{\delta^2 H}{\delta \mathrm{x}^2}\mathrm{x} \tag{3}$$

$$\hat{x} = x_0 - (\frac{\delta^2 H}{\delta x^2})^{-1} \frac{\delta H}{\delta x} \tag{4}$$

Where x = $(x, y, s)^T$ are scale-space coordinates, H = |det(H)| is the magnitude of the Hessian determinant. The derivatives of H are computed around x_0 via finite differences. To achieve the rotation invariance the dominant orientation of the image around each feature using the high pass coefficients of a Haar filter in both the x and y directions inside a circle of radius 6*s* is detected. Here *s* is scale at which the feature was extracted. The size of the Haar filter kernel is scaled to be 4*s* × 4*s*, and the sampling locations are also scaled by *s*, which is easily done using the integral image. The resulting 2D vectors are weighted by a Gaussian with δ = 2*s* and then sorted by orientation. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of size 60 degree. The two summed responses then yield a local orientation vector. The longest such vector over all windows defines the orientation of the interest point. In Fig. 5 [11] shows dominant orientation is red color arrow.

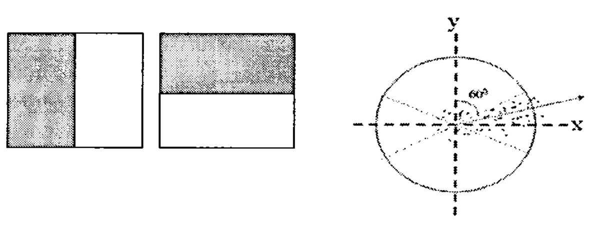


Fig. 5. Haar wavelet filters to compute the responses in x and y direction

Descriptor computation is done by estimating a square region of size *20s* is defined centered on the feature point and oriented along the dominant orientation. This region is split into 4 × 4 square sub-regions. In each sub-region, regularly spaced sample point share taken and over these point, the Haar wavelets responses in x and y directions are calculated. The size of the filter is *2s*, they are called dx and dy, these responses are weighted with a Gaussian($\sigma = 3.3s$) centered at the feature point. This is done to make these robust towards geometric deformations. Also by doing this it can avoid localization errors. The responses over each sub-region are summed obtaining a vector over each region. These vectors and the sum of the absolute value of the responses over each sub-region give us the total entry of the descriptor. So each sub-region will contribute to the descriptor with 4 values. The structure of the descriptor is shown in equation (5).

$$D = \left(\sum dx_i, \sum dy_i, \sum |dx_i|, \sum |dy_i|\right) \tag{5}$$

Where, $i = 1, \ldots\ldots, 16$ the descriptor is turned into a unit vector to achieve invariance to contrast. After applying the SURF algorithm over the training samples we have the extracted keypoints like in the Fig. 6.

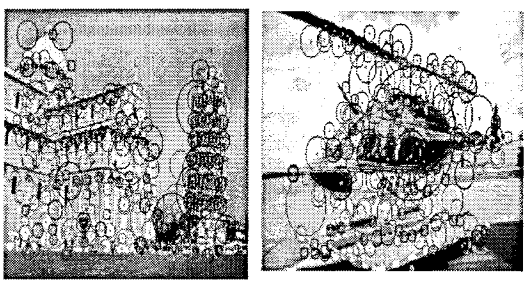


Fig. 6. SURF keypoints from sample images

C. Codebook Generation

From the extraction of feature points using the SURF and SIFT we are able to find out a huge amount of feature points. It is hard to direct use these feature descriptors which have 128 dimensions for SIFT and 64 dimensions for SURF. *k*-means clustering is applied over all the vectors. Codewords are then defined as the centers of the learned clusters. Each keypoint in an image is mapped to a certain codeword through the clustering process and the image can be represented by the histogram of the codewords. To classify a new image, we apply naïve Bayes classifier and take the largest a posteriori score as the prediction. The number of the clusters is the codebook size is analogous to the dictionary. Fig. 7 shows the codewords histogram representation of an image.

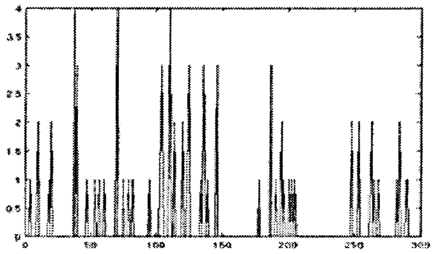


Fig. 7. Codewords histogram representation of an image

k-means clustering is a clustering method where n observations are partitioned into k clusters in which each observation belongs to the cluster with the nearest mean. Given a set of observations (x_1, x_2, ⋯, x_n), where each observation is a d-dimensional real vector, then k-means clustering algorithm partitions the n observations into k sets ($k < n$) S = {S_1, S_2, ⋯, S_k}, so as to minimize the within-cluster sum of squares shown in equation (6).

$$\underset{s}{\arg\min} \sum_{i=1}^{k} \sum_{x_j \in s_i} \left\| X_j - \mu_i \right\|^2 \tag{6}$$

Where, μ_i is the mean of S_i.

D. Learning and Recognition based on the bag of words model (Naïve Bayes)

Naïve Bayes classifier is simple and fast. So it can be used for learning and recognition of objects. Each class of object has its own distribution over the codebook that is different from others. The classifier learns different distributions for different classes from a given set of training examples. To classify a new image, we apply Bayes's rule and take the largest a posteriori score as the prediction.

Bayesian Methods: It uses prior probability of each class given no information about an item. Classification produces a posterior probability distribution over the possible classes given a description of an item. The basic assumption of naïve Bayes is like that, suppose we represent a total class as C, then training set consists of instances of different classes described c_j as conjunctions of attributes values. Now for classifying a new instance based on a tuple of attribute values into one of the classes $c_j \in C$, we need to assign the most probable class using Bayes theorem. These are shown is equation (7) to (9).

$$C_{MAP} = \underset{c_j \in C}{\arg\max}\, P(c_j \mid X_1, X_2, X_3, \ldots, X_n) \tag{7}$$

$$= \underset{c_j \in C}{\arg\max} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)} \tag{8}$$

$$= \underset{c_j \in C}{\arg\max}\, P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j) \tag{9}$$

Parameters estimation:

$P(c_j)$ can be estimated from the frequency of classes in the training examples. $P(x_1, x_2, \cdots, x_n|c_j)$ could only be estimated if a very large number of training examples are available. Independence assumption: attribute values are conditionally independent given the target value naïve Bayes. Equation (10) and (11) show the parameter estimation for naïve Bayes.

$$P(x_1, x_2, \ldots, x_n \mid c_j) = \prod_i P(x_i \mid c_j) \tag{10}$$

$$C_{NB} = \arg\max P(c_j) \prod_i P(x_i \mid c_j) \tag{11}$$

Underflow Prevention: Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow. Since, log(xy) = log(x) + log(y) it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities like the equation (12).

$$C_{NB} = \underset{c_j \in C}{\arg\max} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j) \tag{12}$$

If we represent w_n as each patch in an image. Then it can be $w_n=[0, 0, \cdots, 1, \cdots, 0, 0]^T$, thus w is a collection of all N patches in an image, w = [w_1, w_2, ⋯, w_N]. Object class determination is done based on the following equation (13).

$$C^* = \arg\max p(c \mid w) \propto p(c) p(w \mid c) = p(c) \prod_{n=1}^{N} p(w_n \mid c) \tag{13}$$

Here, C^* is object class decision and $p(c)$ is prior probability of the object class. $p(w \mid c)$ is image likelihood given the class.

IV. EXPERIMENTAL RESULTS

We use Intel® Core™ 2 CPU 6320 1.86 GHz processors with 2GB ram and Visual C++ compiler for testing our system. We have considered different classes of object images for the experiment; at first we use stop sign, hour glass, helicopter and tower Pisa. We consider those images because these represents images for outdoor and indoor natural objects. We took total 326 images for four class; 64 for stop sign, 84 for hour glass, 88 for helicopter and 90 for tower Pisa from Caltech-101 and Caltech-256 dataset. Among those 326 total images, half of these images are used for training and rest half are used for testing. We train the samples from 4 classes together. Our code book size is 300 for both the SIFT and SURF descriptors. Since about 34,000 features from 163 training images are extracted, an average number of features per image is about 208. So we have selected codebook size into 300. We train our sample images at offline. For testing we use the code book generated from the trained samples. Naïve Bayes classifier decides the class of the test image. Fig. 8 and Fig. 9 show sample images and correctly classified images. The classification rate shows in the Table 1 and 2 using SIFT and SURF descriptors. Using SIFT the average classification rate is 76% and using SURF it is 84.6%.

We have tested our images with five classes adding Eiffel tower class. The result is shown in the Table 3 for SURF based system. The classification rate reduces to 74.5% because one more class is added and tower Pisa and Eiffel tower are similar. We also have tested with another 5 classes; roads, airplanes, people, bridge and ferry (Fig. 10). Classification rate is 74.5% for SURF. The result is shown in Table 4.

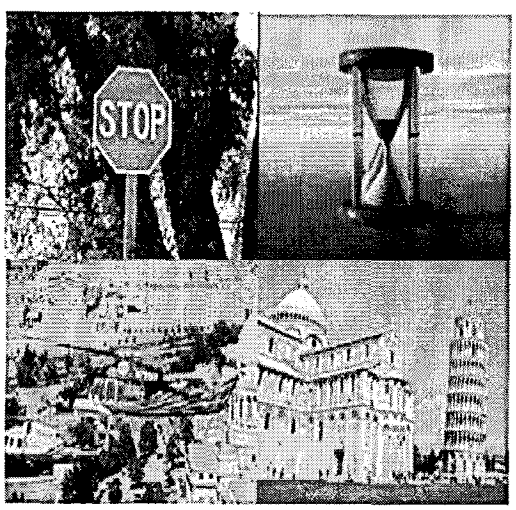


Fig. 8. Sample images for our experiments: stop sign, hour glass, helicopter and tower Pisa

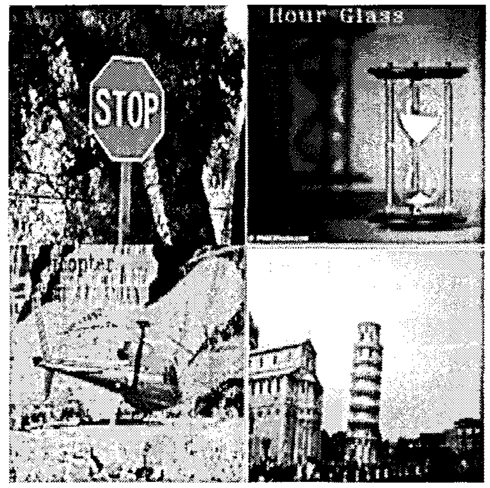


Fig. 9. Correctly classified images

Table 1. Classification result when using SIFT

Object Class	Stop Sign	Hour glass	Helicopter	Tower Pisa	Classification Rate
Stop Sign (32)	27	3	2	0	84.3%
Hour glass(42)	3	31	8	0	73.8%
Helicopter(44)	0	13	31	0	70.4%
Tower Pisa (45)	0	4	6	35	77.7%
Average Rate	124 correct out of 163 : 76%				

Table 2. Classification result when using SURF

Object Class	Stop Sign	Hour glass	Helicopter	Tower Pisa	Classification Rate
Stop Sign (32)	24	7	0	1	75%
Hour glass(42)	4	37	0	1	88.1%
Helicopter(44)	1	2	40	1	90.9%
Tower Pisa (45)	1	7	0	37	82.2%
Average Rate	138 correct out of 163: 84.6%				

Table 3. Classification result when adding Eiffel tower

	Stop Sign	Hour glass	Helicopter	Tower Pisa	Eiffel tower	Classification Rate
Stop Sign (32)	25	3	0	0	4	78%
Hour glass(42)	0	34	0	1	7	81%
Helicopter(44)	0	0	37	1	6	84%
Tower Pisa (45)	0	1	2	30	12	67%
Eiffel tower(41)	0	5	8	2	26	63%
Overall rate	152 out of 204 : 74.5%					

Table 4. Classification result of another 5 classes

	Airplanes	Ferry	Roads	People	Bridge	Classification rate
Airplanes(52)	42	5	5	0	0	80.7%
Ferry (33)	5	24	0	1	3	73%
Roads (48)	1	6	35	5	1	73%
People (52)	1	1	3	43	4	82.6%
Bridge (40)	5	2	3	1	30	75%
Overall rate			174 from 225 : 77%			

After testing with different data set we finally test our system on real video. We have collected some images around our campus varying the scale, viewpoint as well as orientation for training. Here 5 different objects are tested: notice board, garden tree, footpath, telephone booth, university direction map. Then we train our classifier with those images and test with the video clips. We can recognize all the objects in the video. Some images those are correctly recognized from the video are shown in Fig. 11.

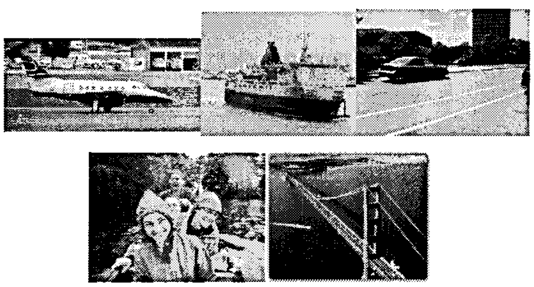


Fig. 10. Sample images of another 5 classes

Fig: 11 Tested images for video clips: notice board, garden tree, footpath, telephone booth, university direction map

Processing times for different steps for image testing are shown in Table 5. Notice that most of processing time takes for feature extraction. We can improve the timing by optimize the code and use dedicated hardware for feature extractions as it is the main part of timing consideration.

Table 5. Processing times for image testing

	Feature extraction (ms)	Codebook generation(ms)	Classification (ms)	Total test time(ms)
SIFT	1778	85	4	1867
SURF	1244	57	3	1304

V. Conclusion And Future Works

We have presented a method that recognizes different classes of objects from indoor and outdoor environment for markerless augmented reality applications. It is not feasible to use markers for outside applications. Natural object features is very important and handling large natural feature set is quite difficult. We test our system with some existing data set for checking its ability of recognition of natural objects. SIFT and SURF are utilized for the features, and bag of words algorithm is applied for object recognition. Our experiment shows that SURF is superior to SIFT. Also SURF is much faster than SIFT as we expected. Our experiment mostly focus on natural object recognition for augmented reality and it is an improvement over marker based augmented reality system. In the future we will integrate our system with mobile phones so that it can be used in outdoor augmented reality applications. Examples of the applications on mobile phone would be automatic navigations, guidance at museums, building recognitions, etc.

References

[1] R. T. Azuma. A survey of augmented reality. Teleoperators and Virtual Environments, 6(4):355 - 385, Aug. 1997.

[2] In Mixed Reality: Merging Real and Virtual Worlds. Yuichi Ohta and Hideyuki Tamura (ed.), Springer-Verlag, 1999. Chp 21 pp. 379-390. ISBN 3-540-65623-5.

[3] http://www.hitl.washington.edu/artoolkit/

[4] D. Beier, R. Billert, B. Br¨uderlin, D. Stichling, B. Kleinjohann, "Marker-less Vision Based Tracking for Mobile Augmented Reality" Proc. of IEEE Int. Conf.on Mixed and Augmented Reality pp.258 - 259, Oct. 2003.

[5] Q. Wang and S. You ,"Real-Time Image Matching Based on Multiple View Kernel Projections" IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[6] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual

Categorization with Bags of Keypoints." ECCV Workshop on Statistical Learning in Computer Vision, pp.1-22, 2004.

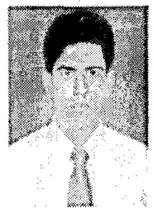
[7] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman "Discovering Objects and Their Location in Images." ICCV, vol.1, pp.370 - 377, 2005.

[8] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky "Learning Hierarchical Models of Scenes, Objects, and Parts",in ICCV, vol. 2, pp.1331 - 1338, 2005.

[9] L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories." Proc. of IEEE Computer Vision and Pattern Recognition. pp. 524 - 531, 2005.

[10] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints." "IJCV 60(2), pp.91-110, 2004.

[11] H. Bay, A. Ess, T.Tuytelaars, and Luc Van Gool: "Speeded-Up Robust Features (SURF)." Computer Vision and Image Understanding CVIU, Vol.110.



Anjan Kumar Paul

2000년 1월 : Khulna University,Bangladesh (BE)
2006년 7월 : Indian Institute of Science, India (M.Tech)
2006년 9월 ~ 현재 : 한국항공대학교정보통신공학과 박사과정

관심분야 : Augmented Reality, 멀티미디어, 컴퓨터비전

Mohammad Khairul Islam

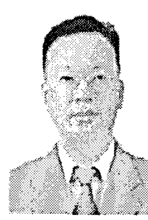
2000년 7월 : Shahjalal University of Science & Technology, Bangladesh(BS)
2007년 8월 : 한국항공대학교 정보통신공학과 (공학석사)
2007년 9월 ~ 현재 : 한국항공대학교 정보통신공학과 박사과정

관심분야 : 멀티미디어, 영상처리, 컴퓨터비전

민 재 홍 (Jae-hong Min)

1997년 2월 : 한국 항공대학교 통신정보공학과 (공학사)
2001년 8월 : 한국 항공대학교 정보통신공학과(석사)
2008년 3월 ~ 현재 : 한국항공대학교 정보통신공학과 박사과정

관심분야 : 객체 기반 영상처리, Augmented Reality, 멀티미디어, 컴퓨터 비전



김 영 범 (Young-bum Kim)

1997년 2월 : 한국 항공대학교 통신정보공학과 (공학사)
2001년 8월 : 한국 항공대학교 정보통신공학과(석사)
2008년 3월 ~ 현재 : 한국항공대학교 정보통신공학과 박사과정

관심분야 : 객체 기반 영상처리, Augmented Reality, 멀티미디어, 컴퓨터 비전

백 중 환 (Joong-hwan Baek)

1981년 2월 : 한국항공대학교 항공통신공학과(공학사)
1987년 7월 : (미)오클라호마 주립대학교 전기 및 컴퓨터공학과(공학석사)
1991년 7월 : (미)오클라호마 주립대학교 전기 및 컴퓨터공학과(공학박사)
1992년 3월 ~ 현재 : 한국항공대학교 항공전자 및 정보통신공학부 교수
관심분야 : 영상처리, 패턴인식, 멀티미디어