Optics Letters

Augmented reality three-dimensional object visualization and recognition with axially distributed sensing

Adam Markman,[1,†] Xin Shen,[1,†] Hong Hua,[2] and Bahram Javidi[1,*]

[1]Electrical and Computer Engineering Department, University of Connecticut, 371 Fairfield Way, Storrs, Connecticut 06269-4157, USA
[2]University of Arizona, College of Optical Sciences, Tucson, Arizona 85721, USA
*Corresponding author: Bahram.Javidi@uconn.edu

Received 23 September 2015; revised 12 November 2015; accepted 21 November 2015; posted 30 November 2015 (Doc. ID 249383); published 8 January 2016

An augmented reality (AR) smartglass display combines real-world scenes with digital information enabling the rapid growth of AR-based applications. We present an augmented reality-based approach for three-dimensional (3D) optical visualization and object recognition using axially distributed sensing (ADS). For object recognition, the 3D scene is reconstructed, and feature extraction is performed by calculating the histogram of oriented gradients (HOG) of a sliding window. A support vector machine (SVM) is then used for classification. Once an object has been identified, the 3D reconstructed scene with the detected object is optically displayed in the smartglasses allowing the user to see the object, remove partial occlusions of the object, and provide critical information about the object such as 3D coordinates, which are not possible with conventional AR devices. To the best of our knowledge, this is the first report on combining axially distributed sensing with 3D object visualization and recognition for applications to augmented reality. The proposed approach can have benefits for many applications, including medical, military, transportation, and manufacturing.

OCIS codes: (100.6890) Three-dimensional image processing; (110.0110) Imaging systems; (100.5070) Phase retrieval.

http://dx.doi.org/10.1364/OL.41.000297

Unlike virtual reality, which completely immerses a user in a virtual world, augmented reality takes a real-world scene and superimposes virtual objects into the scene [1]. There are a myriad of applications for this, including medical [2], commercial [3], and manufacturing [4]. Recently, an emerged form of augmented reality device is smartglasses. These glasses allow a user to view a real-world scene through glasses that also contain a camera and a small digital display. This display allows the virtual information to be combined with the real world. Figure 1 is a schematic illustration of a typical optical see-though head mounted display. An eyepiece magnifies the microdisplay to create a magnified virtual display located at a comfortable viewing distance. A beam splitter is inserted between the eyepiece and viewer's eye to combine the light from the virtual display and the real-world scene. Recently, there has been interest in combining AR with 3D imaging [5,6] to create a true 3D image source in place of a two-dimensional (2D) microdisplay. In [5], a real 3D AR micro integral imaging display system was developed by combining integral imaging with augmented reality. In [6], a micro-integral imaging unit feeds an optically reconstructed 3D scene as the image source to a freeform eyepiece optics, which demonstrates the ability to create a compact, true 3D optical see-through head-mounted display.

In this Letter, we present a method to integrate augmented reality viewing devices such as smartglasses with 3D axially distributed sensing (ADS) [7] to enable a variety of applications, including visualization of occluded objects and 3D object recognition which are not possible with conventional augmented reality devices. Using the 2D camera on a pair of see-through head mounted displays such as smartglasses, 3D ADS is implemented to digitally perform a 3D reconstruction of the scene which may contain an object behind occlusion. The recovered occluded object can be detected, identified, and/or displayed for the viewer with various details such as 3D coordinates superimposed onto the scene. Three-dimensional object recognition is

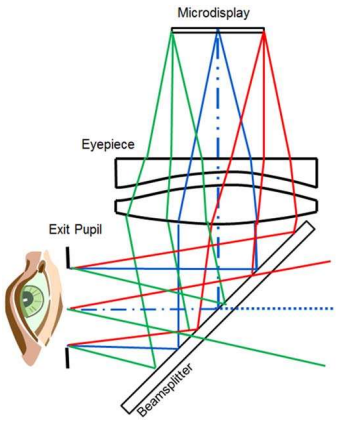


Fig. 1. Typical optical see-through head mounted display.

performed by sliding a window over the reconstructed scene at a particular depth. For each window, the histogram of oriented gradients (HOG) features are extracted and support vector machines (SVM) are then used to detect the object, along with a probability estimate used to determine the optimal window and reconstruction distance or object range. Various types of complex 3D information could be extracted from the scene and displayed in the smartglasses allowing the user to view the 3D scene with the scene or objects unobstructed by obscurations.

Axially distributed sensing (ADS) [7] is a passive sensing 3D imaging technique that captures a scene by moving a camera at different depths along its optical axis which is perpendicular to a 3D scene. Each captured 2D image, known as an elemental image (EI), is then used to reconstruct the 3D scene. Figure 2 depicts an example of the ADS pickup and reconstruction stages. A camera captures 2D images of a 3D scene at multiple distances along its optical axis with a step of ΔZ, as shown in Fig. 2(a). For the reconstruction process, as shown in Fig. 2(b), a reference camera position is set at the first capture position (C_0). With the pinhole model, the distance between the pinhole and captured elemental image is g. Pixels on each captured EI can be mapped into multiple planes in the 3D space to reconstruct the 3D scene. Mathematically, 3D reconstruction is [7]

$$I_{Zo}(x,y) = \frac{1}{K}\sum_{n=0}^{K-1} I_n\left(\frac{x}{M_n}, \frac{y}{M_n}\right), \quad \text{where } M_n = \frac{Z_n}{Z_o}, \tag{1}$$

where K is the total number of the elemental images obtained by the ADS pickup process, I_n is the nth elemental image, (x, y) is the pixel index, M_n is the relative magnification of the nth image with respect to the closest image, Z_o is the initial distance of the camera from the scene, and Z_n is the distance of the camera when it is at position n. To detect an object in a 3D reconstructed scene, the following approach was implemented. A sliding window over the scene was used, and feature extraction for that window was performed by computing the histogram of oriented gradients (HOG) [8]. Support vector machines (SVM) [9] then used the HOG features for object classification. Probability estimates from the SVM classifier were then used to determine the optimal sliding window for the detected object.

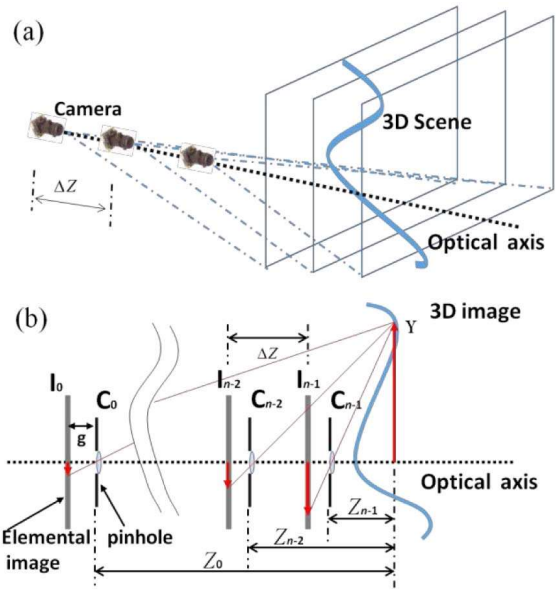


Fig. 2. 3D axially distributed sensing (a) pickup and (b) reconstruction process.

A histogram of oriented gradients observes the distribution of local intensity gradients or edge detection. To implement HOG, the gradients are computed on the 2D grayscale image I using a 1D kernel in the x and y directions as $I_x = I \otimes [-1 \quad 0 \quad 1]$ and $I_y = I \otimes [-1 \quad 0 \quad 1]^T$, respectively, where $\otimes$ denotes convolution and I_x, and I_y denotes the gradient of the image in the x direction and y direction, respectively. These convolutions are used to detect edges in the x and y directions. The magnitude, M, and gradient direction, α, are then computed for each pixel as $M = \sqrt{(I_x)^2 + (I_y)^2}$ and $\alpha = \arctan(I_y/I_x)$.

Once the gradients have been calculated, cell histograms are computed to generate gradient vectors. The image is divided into small connected regions known as cells. An image is broken into multiple cells which consist of 8 × 8 pixels. A block which can be a 2 × 2 cell is then formed.

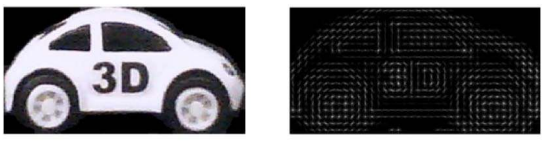
Each cell then accumulates a weighted local 1D histogram based on the gradient directions, α. A weighted vote from each pixel in the cell is placed into bins corresponding to their angles. The final step is to form descriptor blocks by normalizing the histogram gradient for each cell by the "energy" over the cells in a block. For example, to normalize a 2 × 2 cell block, the nine-bin histogram for each cell is used, resulting in 36 features for this block. The block is then normalized, such as using the L_2-norm defined as $v \leftarrow v/\sqrt{\|v\|_2^2 + \varepsilon}$, where v is the block to be normalized, ε is an arbitrary small constant term to ensure that the denominator does not go to zero, and $\|\cdot\|_2$ is the L_2-norm.

The normalization allows the HOG features to be more robust to illumination conditions. The block then moves to the right one cell or down one cell; thus, the blocks overlap. Figure 3 depicts an example of the HOG features computed for a car. Figure 3(a) depicts the original image which is 900(H) × 400(V) pixels, while Fig. 3(b) shows the corresponding HOG features.

Support vector machines [9,10] seek to find the optimal separating hyperplane between the true and false classes were used for classification. Let us define a training vector with n dimensions as $\boldsymbol{x_i} \in R^n$ where $i = 1, \ldots, N$, and N is the total number of data points. We also define an indicator vector $\boldsymbol{y} \in R^N$ such that $\boldsymbol{y_i} \in \{-1, 1\}$ which defines 1 if a data point exists in class 1 and −1, if it belongs in class 2. We wish to find the optimal linear line that separates class 1 from class 2.

For classification, we use SVM [9–11]. We wish to minimize the primal optimization problem:

$$\min_{w,b,\xi} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i \text{s.t. } y_i(\boldsymbol{w}^T\phi(\mathrm{x}_i) + b) \geq 1 - \xi_i, \tag{2}$$
$$\xi_i \geq 0, \qquad i = 1, 2, \ldots, N,$$



(a) (b)

Fig. 3. (a) Image of a 900(H) × 400(V) pixel car. (b) HOG feature.

where $\boldsymbol{w}$ is a vector of coefficients, b is an unknown constant used to determine the offset of the hyperplane, ξ_i are positive slack variables to deal for permitted errors in classification, $\phi(\boldsymbol{x_i})$ is a nonlinear mapping of $\boldsymbol{x_i}$ to a higher-dimensional space, C is a penalty parameters, and N is the number of training cases.

It can be shown that the decision function for a sample $\boldsymbol{x}$ is

$$\text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i K(\boldsymbol{x_i}, \boldsymbol{x}) + b\right), \tag{3}$$

where α_i are the Lagrange multipliers found by optimization, sign is the sign function, $K(\boldsymbol{x_i}, \boldsymbol{x})$ is the kernel function, and b is a known constant. In our experiments, the radial basis function (RBF) kernel was used. This kernel is

$$K(\boldsymbol{x_i}, \boldsymbol{x}) = \phi(\boldsymbol{x_i})^T \phi(\boldsymbol{x}) = \exp(-\gamma \|\boldsymbol{x_i} - \boldsymbol{x}\|^2), \tag{4}$$

where γ is some constant scaling parameter.

In [11], a method to generate probability estimates for SVM classification results was derived. This is conveniently incorporated into the LIBSVM [10] software which was used for SVM classification in our experiments. The Epson Moverio BT-200 smartglasses were used to perform the ADS experiment. The native smartglasses camera has a resolution of 640(H) × 480(V) pixels creating a relatively low resolution image. A higher resolution mini camera was used as a replacement to generate high-resolution images. The camera used was a 1/3″ CMOS color camera which has a pixel resolution of 1600(H) × 1200(V) and sensor size of 4.48 × 3.36 mm. Moreover, the focal length used was 8 mm, and the physical dimension of the camera was 36 × 36 × 20 mm. A camera holder for the mini camera was created using a 3D printer and placed on the smartglasses, as shown in Fig. 4. The scene used was a police car occluded by pine needles as shown in Fig. 4. The AR glasses with the 3D printed camera holder and camera were placed on a translation stage. The number of elemental images captured was 25, and reconstruction was performed offline. Figure 5(a) depicts one of the captured elemental images. ADS reconstruction was then performed. Figure 5(b) depicts a 3D reconstruction of the scene at $z = 980$ mm. Note that in the 3D reconstruction, the letters "POLICE" are visible, whereas they are occluded in Fig. 5(a).

Once reconstruction was performed for a distance d, the following process was used to locate the object of interest: a rectangular window of width w and height h was slid across the scene. For each window, the HOG feature was computed and classified using SVM, along with the probability estimate. The SVM model, which was built using 10 true class and 10 false class images, was split in half for testing and training. In addition, ten-fold cross validation was used to find the best model parameters. It was found that an RBF kernel with a standard deviation of 0.7 was sufficient. Once the car was detected in the window, the coordinates of the top left pixel and bottom right pixel of the window are stored. The probability

Fig. 4. ADS experimental setup with AR glasses and camera.

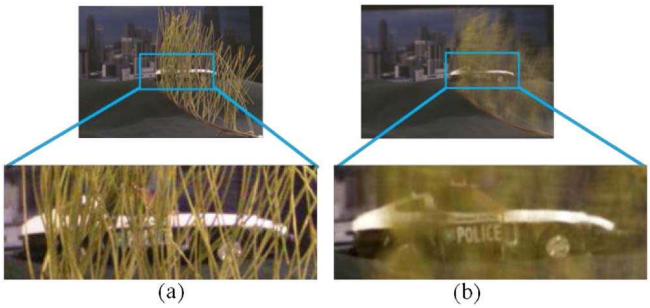


Fig. 5. Comparison of (a) 2D elemental image and (b) 3D reconstructed image at a distance of 980 mm obtained by ADS.

estimate for that window was also recorded. After the window has searched the scene, the optimal window is found which corresponds to the window that resulted in the maximum probability estimate from the SVM model. We note that in this experiment the conventional 2D imaging failed to detect the occluded car; however, the occluded car was successfully detected using the proposed 3D approach.

The classification scheme was also performed over different image scales to find a window that closely matches the object. One approach is to upscale or downscale the image generating an image pyramid as shown in Fig. 6. Note that scaling is done by using the bicubic interpolation [12], and no image blurring is performed. At each scale, a window is slid across the scene, and the optimal window is found if the object is in the scene. The location of the window in the scene, the probability estimate, and the image scale value are then recorded. Observing the image over all scales, the maximum probability estimate is used to indicate the ideal scale for the image and generating a box that closely matches the object.

Another issue is finding the ideal reconstruction distance. Thus, the object recognition scheme is then repeated for every desired reconstruction distance. Probability estimates for the optimal window is then compared among the optimal window for other reconstruction distances. The window that produces the highest probability estimate is then assumed to be the ideal reconstruction distances. For example, using the scene in our

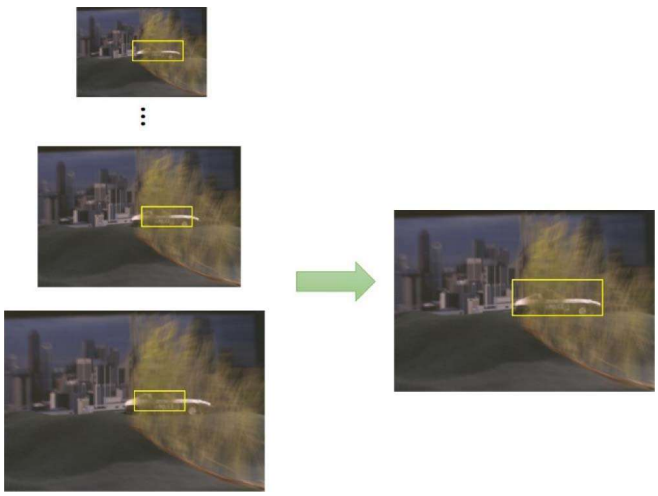


Fig. 6. Ideal size of the window is found by resizing the image followed by scaling the window appropriately once the ideal window size has been found.

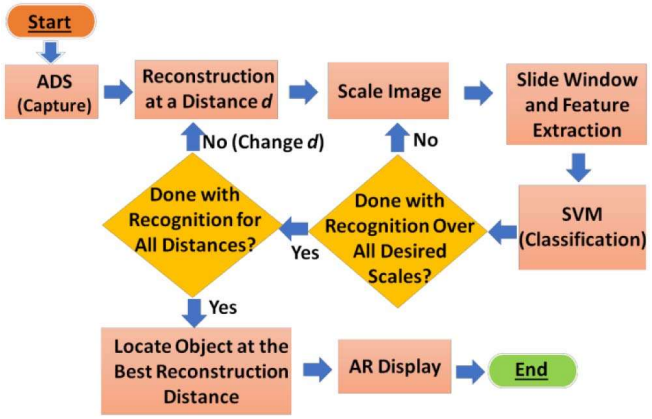


Fig. 7. Object recognition with 3D ADS for augmented reality.

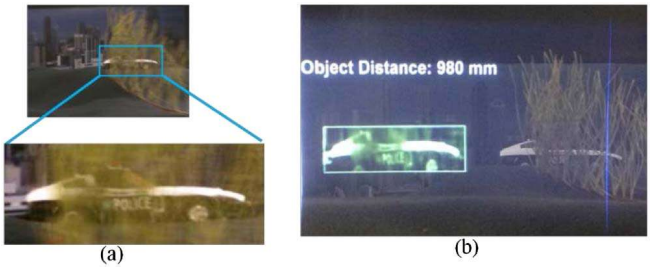


Fig. 8. 3D augmented reality experimental results with unknown sensor positions. (a) 3D reconstructed image at 980 mm by ADS with unknown sensor positions. (b) ADS 3D object recognition combined with augmented reality.

experiments, it is difficult to determine where the best object reconstruction distance is for distances of 960–990 mm. By using the window associated with the highest probability estimate from the SVM, it was found that the optimal window was at z = 980 mm with a probability estimate of 0.1924. An overview of the classification scheme is shown in Fig. 7.

In practice, it may not be possible to know the exact camera positions. Thus, we performed 3D computational reconstruction with unknown sensor positions [13]. The 3D reconstructed image at 980 mm using ADS with unknown sensor positions is shown in Fig. 8(a). To evaluate the performance of the 3D reconstruction with unknown sensor positions, the root mean squared error (RMSE) [14] was computed between the 3D reconstruction of the car shown in Figs. 5(b) and 8(a). The reconstructed images were normalized between [0,1] prior to computing the RMSE ($\sqrt{\sum_{k=1}^{M}\sum_{j=1}^{N}(\hat{x}_{k,j}-x_{k,j})^2/(MN)}$) which was 0.0169. The 3D reconstruction with object recognition was then combined with augmented reality. Figure 8(b) depicts the scene combined with augmented reality information. The user can identify the car with occlusion removal, along with approximately how far it is from the user.

In conclusion, we have presented an approach to integrate augmented reality with a 3D imaging technique known as axially distributed sensing (ADS). This can be useful for a variety of complex applications such as visualization and object recognition, including in the presence of partial occlusion in the scene. In our experiments, ADS was used to create a 3D reconstruction of a scene containing an object (car) behind occlusion. At a given reconstruction distance, the histogram of oriented gradients (HOG) feature was computed for the image region inside a sliding window. Using the HOG features, a support vector machine (SVM) was then used to classify the window and determine if the object was present in the scene. Moreover, the probability estimates obtained from the SVM were used not only to find the best window for the target, but also the optimal reconstruction distance. Once the object has been identified, it was placed in a smartglasses display that overlooks the scene with the occluded object. Thus, a user can visually see the object with occlusion removal, along with the approximate distance that the object is from the user. While we have used ADS, a variety of other 3D imaging approaches may be used [6,15–19]. Likewise, other object recognition algorithms may be employed for detecting and recognizing the object.

Funding. National Science Foundation (NSF) (NSF/IIS-1422179, NSF/IIS-1422653).

Acknowledgment. Hong Hua has a disclosed financial interest in Magic Leap Inc. which has been properly disclosed to the University of Arizona and reviewed by the Institutional Review Committee in accordance with its conflict of interest policies.

†These authors contributed equally to this work.

REFERENCES

1. R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, IEEE Comput. Graph. Appl. **21**, 34 (2001).
2. X. Kang, M. Azizian, E. Wilson, K. Wu, A. Martin, T. Kane, C. Peters, K. Cleary, and R. Shekha, Surg. Endosc. **28**, 2227 (2014).
3. H. Mukawa, K. Akutsu, I. Matsumura, S. Nakano, T. Yoshida, M. Kuwahara, and K. Aiki, J. Soc. Inf. Disp. **17**, 185 (2009).
4. F. Doil, W. Schreiber, T. Alt, and C. Patron, in *ACM Proceedings of the Workshop on Virtual Environments* (2003), pp. 71–76.
5. J. Wang, X. Xiao, H. Hua, and B. Javidi, J. Disp. Technol. **11**, 889 (2014).
6. H. Hua and B. Javidi, Opt. Express **22**, 13484 (2014).
7. R. Schulein, M. DaneshPanah, and B. Javidi, Opt. Lett. **34**, 2012 (2009).
8. N. Dalal and B. Triggs, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), Vol. **1**, pp. 886–893.
9. B. Schölkopf and A. Smola, *Learning with Kernels-Support Vector Machines, Regularization, Optimization and Beyond* (MIT, 2002).
10. C. Chang and C. Lin, ACM Trans. Intell. Syst. Technol. **2**, 1 (2011).
11. T. F. Wu, C. J. Lin, and R. C. Weng, J. Mach. Learn. Res. **5**, 975 (2004).
12. R. Gonzalez, *Digital Image Processing* (Pearson, 2009).
13. X. Xiao and B. Javidi, Opt. Lett. **36**, 1086 (2011).
14. R. Schalkhoff, *Pattern Recognition, Statistical, Structural and Neural Approaches* (Wiley, 1992).
15. B. Javidi, F. Okano, and J. Y. Son, *Three Dimensional Imaging, Visualization, and Display* (Springer, 2009).
16. F. Okano, J. Arai, and M. Kawakita, Opt. Lett. **32**, 364 (2007).
17. X. Xiao, B. Javidi, M. Martinez-Corral, and A. Stern, Appl. Opt. **52**, 546 (2013).
18. M. J. Cho, M. Daneshpanah, I. Moon, and B. Javidi, Proc. IEEE **99**, 556 (2011).
19. Y. Frauel, T. Naughton, O. Matoba, E. Tahajuerce, and B. Javidi, Proc. IEEE **94**, 636 (2006).