Object Detection and Pose Estimation from Natural Features Using Consumer RGB-D Sensors: Applications in Augmented Reality

João Paulo Lima*
PhD Candidate
Voxar Labs, CIn-UFPE, Brazil

Veronica Teichrieb*
Thesis Supervisor
Voxar Labs, CIn-UFPE, Brazil

Hideaki Uchiyama†
Research Collaborator
INRIA Rennes

Eric Marchand†
Research Collaborator
INRIA Rennes

ABSTRACT

The work proposed in this document aims to investigate the use of consumer RGB-D sensors for object detection and pose estimation from natural features, with the purpose of using such techniques for developing augmented reality applications. Two methods based on depth-assisted rectification are proposed, which transform features extracted from the color image to a canonical view using depth data in order to obtain a representation invariant to rotation, scale and perspective distortions. While one method is suitable for textured objects, either planar or non-planar, the other method focuses on texture-less planar objects. Qualitative and quantitative evaluations of the proposed methods are performed, comparing it to existing methods for object detection and pose estimation.

Keywords: Augmented reality, natural features, computer vision, RGB-D.

Index Terms: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Depth Cues, Range Data, Tracking; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, Augmented, and Virtual Realities

1 INTRODUCTION

In the last years, AR applications have been benefited with the advent of low cost RGB-D consumer devices. These devices are commonly used in human body detection and tracking for user interaction purposes. RGB-D sensors are able to provide in real-time, besides a color image (RGB channels) of the scene, another image in which each pixel value corresponds to the distance from the scene objects to the camera. Such image is named depth image (D channel). The use of RGB-D consumer devices for object detection and pose estimation has grown significantly over the last years [6][11]. The color and depth images from RGB-D cameras can be employed to obtain 3D models of the objects to be detected and also provide useful information at runtime for accomplishing better results when compared to techniques that use only RGB data.

The main question related to the topics approached in this paper is: "How to improve object detection and pose estimation from natural features for AR using consumer RGB-D sensors?". In order to address this problem, existing object tracking and detection methods based on natural features should be investigated in order to identify how depth information can be exploited to obtain better results than when only RGB data is used. A special attention should also be devoted to methods that already use RGB-D information for object tracking and detection.

*email: {jpsml, vt}@cin.ufpe.br
† email: {Hideaki.Uchiyama, Eric.Marchand}@inria.fr

In this context, the work presented in this document aims to develop novel object detection and pose estimation methods for AR using natural features and consumer RGB-D sensors. The developed solutions are evaluated taking into account performance, robustness and accuracy metrics.

The specific goals to be achieved in this work are:

• Study the natural feature tracking and detection field, with emphasis on object tracking and detection methods, including techniques that make use of RGB-D data, for identifying points of improvement in the state of the art;

• Define and develop object detection and pose estimation methods that use consumer RGB-D sensors for solving some of the identified points of improvement;

• Perform case studies and evaluations of AR applications that make use of the developed methods, in order to verify how the methods contribute to improving user experience.

This paper is organized as follows. Section 2 presents one of the methods developed in this work, which makes use of depth information for rectifying patches around interest points in the color image. Section 3 presents the other method developed in this work, which rectifies contours extracted from the color image using depth data. Section 4 discusses the preliminary results obtained with the techniques described in Sections 2 and 3. The results obtained are compared with other existing object detection and pose estimation methods. Section 5 presents final considerations and future work for the remainder of the PhD course.

2 DEPTH-ASSISTED RECTIFICATION OF PATCHES

This section presents a method developed in this work named Depth-Assisted Rectification of Patches (DARP), which exploits depth information available in RGB-D consumer devices to improve keypoint matching of perspectively distorted images. This is achieved by generating a projective rectification of a patch around the keypoint, which is normalized with respect to perspective distortions and scale.

In DARP, keypoints are extracted using FAST-9 [14] and their normal vectors on the scene surface are estimated using the depth image with the average 3D gradient method [9]. Then, using depth and normal information, patches around the keypoints are rectified to a canonical view in order to remove perspective and scale distortions. Given $\boldsymbol{n} = (n_x, n_y, n_z)^T$ as the unit normal vector in camera coordinates at $\boldsymbol{M_{cam}}$, which is the corresponding 3D point of a keypoint, two unit vectors $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ that define a plane with normal $\boldsymbol{n}$ can be obtained by:

$$\boldsymbol{n_1} = \frac{1}{\|(n_z, 0, -n_x)^T\|} \cdot (n_z, 0, -n_x)^T, \boldsymbol{n_2} = \boldsymbol{n} \times \boldsymbol{n_1}.$$

This is valid because it is assumed that n_x and n_z are not equal to zero at the same time, since in this case the normal would be perpendicular to the viewing direction and the patch would be not visible.

From $\boldsymbol{M_{cam}}$, $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$, it is possible to find the corners $\boldsymbol{M_1}$, …, $\boldsymbol{M_4}$ of the patch in the camera coordinate system. The patch size in camera coordinates should be fixed in order to allow scale

invariance. The corners $\boldsymbol{m_1}$, …, $\boldsymbol{m_4}$ of the patch to be rectified in image coordinates are the projection of the 3D points $\boldsymbol{M_1}$, …, $\boldsymbol{M_4}$. Then, $\boldsymbol{m_i} = K\boldsymbol{M_i}$, where K is the intrinsic parameters matrix. If the patch size in image coordinates is too small, the rectified patch will suffer degradation in image resolution, harming its description. This size is influenced by the location of the 3D point $\boldsymbol{M_{cam}}$ (e.g., if $\boldsymbol{M_{cam}}$ is too far from the camera, the patch size will be small). It is also directly proportional to the patch size in camera coordinates, which is determined by a constant factor k applied to $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ as follows: $\boldsymbol{n_1}' = k \cdot \boldsymbol{n_1}$ and $\boldsymbol{n_2}' = k \cdot \boldsymbol{n_2}$. The factor k should be large enough to allow good scale invariance while being small enough to give distinctiveness to the patch. In the performed experiments, it was computed by $k = \lfloor s/2 \rfloor$, where s is the size of the rectified patch (set to 31 in the experiments).

The corners $\boldsymbol{M_1}$, …, $\boldsymbol{M_4}$ of the patch are given by:

$$\boldsymbol{M_1} = \boldsymbol{M_{cam}} + \boldsymbol{n_1}' + \boldsymbol{n_2}', \boldsymbol{M_2} = \boldsymbol{M_{cam}} + \boldsymbol{n_1}' - \boldsymbol{n_2}',$$

$$\boldsymbol{M_3} = \boldsymbol{M_{cam}} - \boldsymbol{n_1}' - \boldsymbol{n_2}', \boldsymbol{M_4} = \boldsymbol{M_{cam}} - \boldsymbol{n_1}' + \boldsymbol{n_2}'.$$

The corresponding corners $\boldsymbol{m_1}'$, …, $\boldsymbol{m_4}'$ of the patch in the canonical view are:

$$\boldsymbol{m_1}' = (s-1, 0)^T, \boldsymbol{m_2}' = (s-1, s-1)^T,$$

$$\boldsymbol{m_3}' = (0, s-1)^T, \boldsymbol{m_4}' = (0, 0)^T.$$

From $\boldsymbol{m_1}$, …, $\boldsymbol{m_4}$ and $\boldsymbol{m_1}'$, …, $\boldsymbol{m_4}'$, it can be computed an homography H that takes points of the input image to points of the rectified patch.

For rotation invariance, the rectified patch orientation is calculated using the intensity centroid [15]. Finally, a descriptor for the rectified patch is calculated using the assigned orientation with the Rotation-Aware BRIEF (rBRIEF) method [15].

DARP can be used with any local feature detector and descriptor and is suitable for planar and non-planar textured scenes.

3 Depth-Assisted Rectification of Contours

This section presents a method developed in this work named Depth-Assisted Rectification of Contours (DARC) for detection and pose estimation of texture-less planar objects using RGB-D cameras. It consists in matching contours extracted from the current image to previously acquired template contours. In order to achieve invariance to rotation, scale and perspective distortions, a rectified representation of the contours is obtained using the available depth information. DARC requires only a single RGB-D image of the planar objects in order to estimate their pose, opposed to some existing approaches that need to capture a number of views of the target object. It also does not generate warped versions of the templates, which is commonly required by existing object detection techniques.

First, contours are extracted from the query RGB image using the Canny edge detector [5]. Then, for each extracted contour, the 3D points that correspond to the 2D points of the contour and its inner contours are selected. In the remainder of this paper, the set of points that belong to a contour or its inner contours is named *contour group*. Then, for each contour group, the corresponding 3D points $\boldsymbol{M_i}$ of the 2D contour points $\boldsymbol{m_i}$ are used to estimate the normal and orientation of the contour group via Principal Component Analysis (PCA). The centroid $\bar{\boldsymbol{M}}$ of the 3D contour points is calculated, which is invariant to affine transforms. A covariance matrix is computed using $\boldsymbol{M_i}$ and $\bar{\boldsymbol{M}}$, and its eigenvectors $\{\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}\}$ and corresponding eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ are computed and ordered in ascending order. The normal vector to the contour group plane is $\boldsymbol{v_1}$ [3]. If needed, $\boldsymbol{v_1}$ is flipped to point towards the viewing direction. Contour group orientation is given by $\boldsymbol{v_2}$ and $\boldsymbol{v_3}$, which can be seen as the y and x axis, respectively, of a local coordinate system with origin at $\bar{\boldsymbol{M}}$ [3]. There are four possible orientations given by combinations of the x and y axis with different signs. It only makes sense to consider all four orientations if mirrored or transparent objects might be detected. Otherwise, only two orientations are enough, which are given by using both flipped and non-flipped $\boldsymbol{v_2}$ as the y axis and computing the x axis as the cross product of $\boldsymbol{v_2}$ and $\boldsymbol{v_1}$.

In order to allow matching instances of the same contour group observed from different viewpoints, they are normalized to a common representation. Translation invariance is achieved by writing the coordinates of the 3D contour points $\boldsymbol{M_i}$ relative to the centroid $\bar{\boldsymbol{M}}$. Rotation invariance is obtained by aligning $\boldsymbol{v_3}$ and $\boldsymbol{v_2}$ with the x and y global axes, respectively. Since the 3D contour points $\boldsymbol{M_i}$ are in camera coordinates, they are scale invariant. Perspective invariance is obtained by aligning the inverse of the normal vector $\boldsymbol{v_1}$ to the z global axis. This way, the rectified contour points $\boldsymbol{M_i}'$ can be computed as follows:

$$\boldsymbol{M_i}' = [v_3 \quad v_2 \quad v_1]^T (\boldsymbol{M_i} - \bar{\boldsymbol{M}}).$$

The rectified points should lie on the xy plane ($z = 0$). Since two or four orientations given by $\boldsymbol{v_2}$ and $\boldsymbol{v_3}$ are considered, each one is used to generate a different rectification of a contour group. All these rectifications are taken into account in the matching phase. In some cases the estimated orientation is not accurate. However, this is still sufficient for matching and pose estimation purposes.

After being rectified, query contour groups can be matched to a previously rectified template contour group. This is done by comparing each rectified query contour group with the rectified template contour group, considering the different orientations computed. First, a match is rejected if the upright bounding rectangles of the rectified contour groups do not have a similar size. Then, it is calculated a rough pose that maps the 3D unrectified template contour group to the 3D unrectified query contour group. Given the rotation R^t and translation $\boldsymbol{t^t}$ that rectify the template contour group and the rotation R^q and translation $\boldsymbol{t^q}$ that rectify the query contour group, the rough pose is obtained by:

$$\begin{bmatrix} R^q & \boldsymbol{t^q} \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} R^t & \boldsymbol{t^t} \\ 0 & 1 \end{bmatrix}.$$

The 3D unrectified template contour group is transformed using the rough pose $[R|\boldsymbol{t}]$ and then projected onto the query image. After that, the upright bounding rectangle of the projected points is calculated and compared with the upright bounding rectangle of the 2D query contour group. If they are not close to each other or their sizes are not similar, the match is discarded.

The similarity between template contour group projection and 2D query contour group is given by their chamfer distance [1]:

$$\frac{1}{\tau n} \sum_{i=0}^{n} DT^{\tau}(\boldsymbol{m_i^t}),$$

where n is the number of points in the template contour group, $\boldsymbol{m_i^t}$ is the i-th template contour point and DT^{τ} is the query distance transform truncated to a value τ. For each query contour group, the template contour group orientation with smallest chamfer distance is marked as a candidate match.

If there is a candidate match for a given query contour group, then a refined pose of the contour group is estimated from the previously computed rough pose using the Levenberg-Marquardt algorithm. The query distance transform is used to compute the residuals. Finally, the chamfer distance between the template contour group and query contour group is calculated using the refined pose. If it is below a threshold, then the match is considered as correct.

In the current implementation, a single contour group is used for defining the pose of a given object. If the object contains several disjoint contour groups, one of these has to be selected for being used as template.

4 Preliminary Results

This section describes some results obtained with the DARP and DARC methods. The techniques were also evaluated regarding performance and pose estimation quality. All the experiments were performed with 640x480 images using a laptop with Intel Core i7 720QM @ 1.60GHz processor and 6GB RAM.

4.1 DARP Results

In order to evaluate DARP, some image sequences from the publicly available University of Washington's RGB-D Object Dataset [10] were used. The results obtained with DARP were compared with ORB [15], since the current implementation of DARP performs keypoint detection, orientation assignment and patch description in a way similar to ORB. Two images of the same object with different poses were matched using both techniques.

Initially the tests were done with planar objects, as can be seen in Figure 1. It is shown in the top images the results obtained with ORB and in the bottom images the results obtained with DARP. It can be seen in the left images the matches between the two instances of the object. The projection of a 3D point cloud of the object using the pose calculated from the matches is shown in the right images. It can be noted that the DARP method provides better results than ORB when the object has an oblique pose with respect to the viewing direction. The matches obtained with ORB led to a wrong 3D pose, while it was possible to estimate a reasonable pose using DARP, as evidenced by the projection of the 3D model. After, some tests were done with non-planar objects, as shown in Figure 2. DARP also obtained better results than ORB in the oblique pose scenario, since the 3D pose computed from the matches (right images) was closer to the correct pose when the DARP method was used.

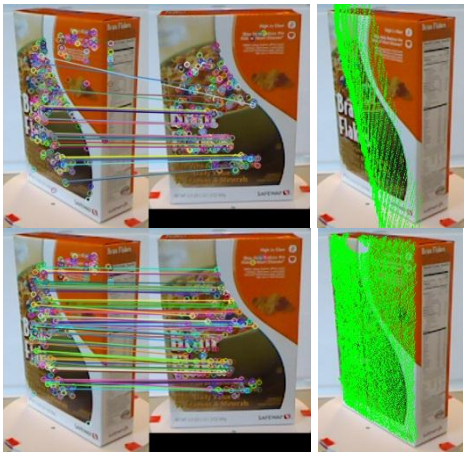


Figure 1: Planar object matching using ORB (top) and DARP (bottom): matches (left) and pose estimation (right).

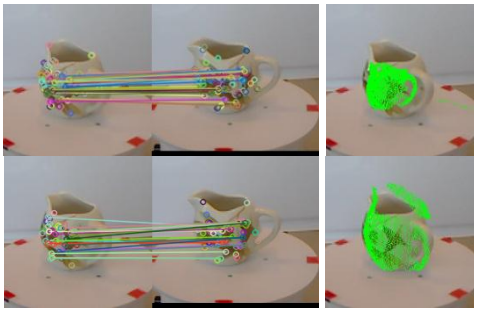


Figure 2: Non-planar object matching using ORB (top) and DARP (bottom): matches (left) and pose estimation (right).

4.2 DARC Results

In order to evaluate DARC, some image sequences were captured using the Microsoft Kinect for Xbox 360 RGB-D camera and synthetic RGB-D images were also generated.

Figure 3 shows some results obtained with DARC for detection and pose estimation of different planar objects. A video with the results can be found at `http://goo.gl/PmXzQ`. It can be seen that DARC can deal with significant changes in rotation and scale as well as with perspective distortions. The contour group used as template is the octagon of the stop sign.

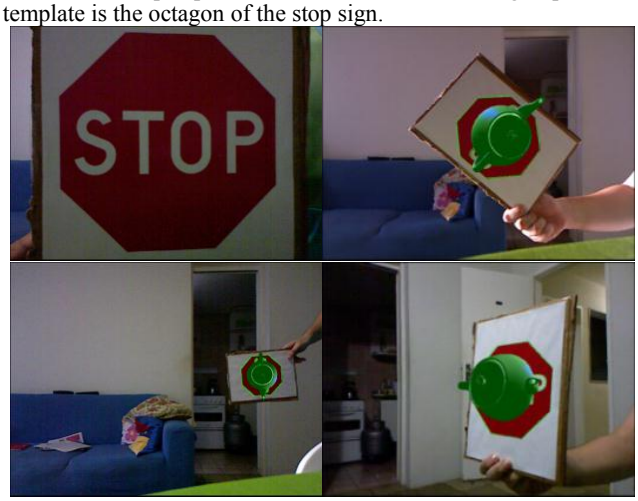
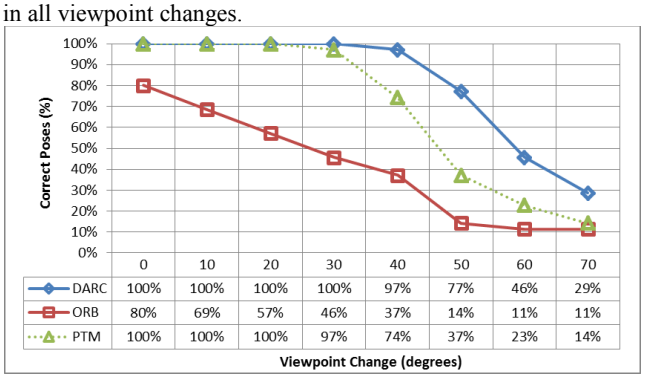


Figure 3: Augmentation of planar objects using DARC.

DARC was compared to some existing techniques regarding pose estimation quality and performance. The methods selected for the evaluation were ORB [15], which is a texture based technique, and PTM [7], which exploits edge information. Pose estimation quality was evaluated with a database of 280 synthetic RGB-D images of a stop sign under different viewpoints on a cluttered background. The poses were under a degree change range of $[0°, 70°]$ with a $10°$ step and a scale range of $[1.0, 2.0]$ with a 0.2 step. The percentage of correct poses estimated by each method was calculated. The pose was considered as correct only if the root-mean-square (RMS) reprojection error was below 3 pixels. Figure 4 shows that DARC outperformed ORB and PTM in all viewpoint changes.



	0	10	20	30	40	50	60	70
DARC	100%	100%	100%	100%	97%	77%	46%	29%
ORB	80%	69%	57%	46%	37%	14%	11%	11%
PTM	100%	100%	100%	97%	74%	37%	23%	14%

Figure 4: Percentage of correct poses of different methods with respect to viewpoint change.

5 Final Considerations

It was shown that the use of RGB-D sensors allows improving object tracking and detection from natural features. The DARP method has been proposed, which exploits depth information to improve keypoint matching. This is done by rectifying the patches using the 3D information in order to remove perspective effects.

The depth information is also used to obtain a scale invariant representation of the patches. It was shown that DARP can be used together with existing keypoint matching methods in order to help them to handle situations such as oblique poses with respect to the viewing direction. It supports both planar and non-planar objects and is able to run at interactive rates. The DARC technique has also been presented, which performs detection and pose estimation of texture-less planar objects by making use of depth information available in RGB-D consumer devices. In order to achieve this, contours extracted from a query image are rectified for removing distortions caused by rotation, scale and perspective transforms. The normalized representation is matched to templates acquired a priori using chamfer distance and a rough pose is calculated. This pose is later refined using a Levenberg-Marquardt optimization. DARC showed to be robust to in-plane and out-of-plane rotations, scale and perspective deformations, providing a pose with reasonable accuracy for AR applications. The DARC technique works in real-time, being able to run at ~20 fps while detecting a single template.

Regarding DARP, a quantitative evaluation of pose estimation quality will be performed. Future work will also focus in evaluating how normal estimation can be speeded up, maybe using faster approaches such as the one described in [6]. A refinement step for patch pose estimation using a template tracking method such as [2] will be considered. Another issue that should be investigated is that when the object suffers from severe perspective or scale distortion, the rectified patch loses resolution, which impacts on its description. One alternative to be studied for solving this would be to generate distorted versions of the reference images prior to keypoint matching [4]. Then, the available depth and normal information could be used to select a set of most probable matching keypoints for each patch. Finally, tests with other detectors and descriptors will be done.

With respect to DARC, it will be evaluated the possibility of extending the technique for working with non-planar objects. It could also be noted that computational performance drops linearly with the number of detected templates. As future work, DARC scalability should be improved. Optimizations are planned in order to allow better frame rates while estimating the pose of a higher number of contour groups. Special attention should be devoted to the pose refinement step, which showed to be a bottleneck and its performance depends on the number of detected templates. The use of a template tracking method such as [2] should be considered. The current approach is also not robust to partial occlusions, since it uses only a single contour group to estimate object pose. If any contour that belongs to the contour group is occluded, detection tends to fail. Occlusion handling is a direct consequence of estimating the pose of more contour groups, which is a future work mentioned in the previous paragraph. If the object is composed by several contour groups and the pose of the visible contour groups can be calculated, the object pose can then be inferred even if some of its contour groups are occluded. In addition, a verification method using neighboring contours such as the one described in [8] could also be used. Contour detection showed to be not robust to illumination changes, noise and blur caused by very fast movements. The use of more robust region detectors such as MSERs [13] will be investigated. Finally, confusions can occur when the template contour groups do not have enough discriminative power. It will be studied if the discriminative power of contour matching can be improved by making use of oriented chamfer matching [16] or directional chamfer matching [12].

An investigation of other ways of improving object detection and pose estimation using RGB-D sensors will also be performed. Then, AR case studies using the developed methods will be done.

REFERENCES

[1] H. Barrow, J. Tenembaum, R. Bolles, and H. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In *IJCAI '77*, pages 659–663, Cambridge, Massachusetts, 1977.

[2] S. Benhimane, and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE/RSJ IROS '04*, pages 943–948, Sendai, Japan, 2004.

[3] J. Berkmann and T. Caelli. Computation of surface geometry and segmentation using covariance techniques. In *IEEE PAMI*, volume 16, issue 11, pages 1114–1116, 1994.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *ECCV '10*, Lecture Notes in Computer Science, volume 6314, pages 778–792, Heraklion, Greece, 2010.

[5] J. Canny. A computational approach to edge detection. In *IEEE PAMI*, volume 8, issue 6, pages 679–698, 1986.

[6] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of textureless objects. In *IEEE PAMI*, volume 34, issue 5, pages 876–888, 2012.

[7] A. Hofhauser, C. Steger, and N. Navab. Edge-based template matching and tracking for perspectively distorted planar objects. In In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Paolo Remagnino, Fatih Porikli, Jörg Peters, James Klosowski, Laura Arns and Yu Ka Chun et al., editors, *ISVC '08*, Lecture Notes in Computer Science, volume 5358, pages 35–44, Las Vegas, Nevada, 2008.

[8] S. Holzer, S. Hinterstoisser, S. Ilic, and N. Navab. Distance transform templates for object detection and pose estimation. In *IEEE CVPR '09*, pages 1177–1184, Miami, Florida, 2009.

[9] S. Holzer, R. Rusu, M. Dixon, S. Gedikli, N. Navab. Real-time surface normal estimation from organized point cloud data using integral images. In *IEEE/RSJ IROS '12*, Vilamoura, Algarve, Portugal, 2012 (to be published).

[10] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *ICRA '11*, pages 1817–1824, Shangai, China, 2011.

[11] W. Lee, N. Park, and W. Woo. Depth-assisted real-time 3D object detection for augmented reality. In *ICAT '11*, pages 126–132, Osaka, Japan, 2011.

[12] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *IEEE CVPR '10*, pages 1696–1703, San Francisco, California, 2010.

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. In *BMVC '02*, pages 384–393, Cardiff, Wales, 2002.

[14] E. Rosten, and T. Drummond. Machine learning for highspeed corner detection. In *ECCV*, pages 430–443, 2006.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *IEEE ICCV '11*, pages 2564–2571, Barcelona, Spain, 2011.

[16] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. In *IEEE PAMI*, volume 30, issue 7, pages 1270–1281, 2008.