Learning and Recognition of 3D Visual Objects in Real-Time

Shihab Hamid[1,3] and Bernhard Hengst[1,2,3]

[1] School of Computer Science and Engineering, UNSW, Sydney, Australia
[2] ARC Centre of Excellence for Autonomous Systems
[3] NICTA, Sydney, Australia

Abstract. Quickly learning and recognising familiar objects seems almost automatic for humans, yet it remains a challenge for machines. This paper describes an integrated object recognition system including several novel algorithmic contributions using a SIFT feature appearance-based approach to rapidly learn incremental 3D representations of objects as aspect-graphs. A fast recognition scheme applying geometric and temporal constraints localizes and identifies the pose of 3D objects in a video sequence. The system is robust to significant variation in scale, orientation, illumination, partial deformation, occlusion, focal blur and clutter and recognises objects at near real-time video rates.

1 Introduction

The problem of object recognition is a long standing challenge. Changes in scale, illumination, orientation and occlusions can significantly alter the appearance of objects in a scene. Humans easily deal with these subtle variations but machines notice these changes as significant alterations to the matrix of pixels representing the object.

There are two broad approaches to 3D object representation: object-based - 3D geometric modeling, and view-based - representing objects using multiple 2D views. In this paper we report on the development and evaluation of a system that can rapidly learn a robust representation for initially unknown 3D objects and recognise them at interactive frame rates. We have chosen a view-based approach and identified the Scale Invariant Feature Transform (SIFT) [1] as a robust local feature detector, which is used as the computer vision primitive in the object recognition system. In the learning phase a video camera is used to record footage of an isolated object. Objects are learnt as a graph of clustered 'characteristic' views, known as an aspect-graph.

Our contribution is an integrated vision system capable of performing generic 3D object recognition in near real-time. We have developed a fast graphics processing unit (GPU) implementation of SIFT for both building a view-based object representation and later for rapidly recognising learnt objects in images. The system uses an approximate kd-tree search technique, Best-Bin-First (BBF) [2], to significantly speed up feature matching. Additional innovations include: a method to aggregate similar SIFT descriptors based on both a Euclidean distance

A. Nicholson and X. Li (Eds.): AI 2009, LNAI 5866, pp. 150–159, 2009.

Fig. 1. Ten sample objects (left). Part of the 'Mighty-Duck' circular aspect-graph (right).

threshold and a reduced nearest-neighbour ratio; a view clustering algorithm capable of forming rich object representations as aspect-graphs of temporally and geometrically adjacent characteristic views; and an improved view clustering technique using bounding boxes during the learning phase.

The system was evaluated using one to ten objects, each generating about 35 views. Figure 1 shows the ten objects used in our experiments and some of the adjacent views of the generated aspect-graph for the Mighty-Duck object. Recognition speed measurements demonstrate excellent scaling. Our evaluation found the system robust to object occlusion, background clutter, illumination changes, object rotation, scale changes, focal blur and partial deformation.

2 Background and Related Work

Many approaches to 3D object recognition can be found in research literature. A recent survey [3] has concluded that there are as many approaches as there are applications. We only discuss a small subset relevant to our approach.

Local image features usually comprise an interest-point detector and a feature descriptor. One such highly discriminatory feature is the Scale-Invariant Feature Transform (SIFT) [4]. SIFT features are widely adopted because of their ability to robustly detect and invariantly define local patches of an image. Their limitations are that they cannot adequately describe plain untextured objects, their speed of extraction is slow, and their high dimensionality makes the matching process slow. Nevertheless, SIFT is a successful multi-scale detector and invariant descriptor combination. It has been used in object recognition with severe occlusions, detecting multiple deformable objects, panorama stitching and 3D reconstruction.

View-clustering is the process of aggregating similar views into clusters representing the *characteristic views* of the object. Lowe [5] performs view clustering

using SIFT features and unlike 2D object recognition, features vote for their object view as well as their neighbour views. There are several limitations to this solution. New features are continually added, potentially creating a scalability problem. The unsupervised clustering cannot recover the orientation of the object. The approach assumes that input object images are a random sequence of images, disregarding any possible temporal link between consecutive frames.

In contrast to view-clustering, view-interpolation explicitly attempts to interpolate the geometric changes caused by changes in the viewing direction. Revaud, et al. [6] apply linear combination theory to the framework of local invariant features. Their model is constructed by forming a homography of features from two nearby object views. On a modest PC, recognition of objects in 800×600 images can be achieved in under half a second. In comparison, the system described here can recognise objects at 12 frames per second using 640×480 images.

The characteristic views of a 3D object can be related using an *aspect-graph*. An aspect-graph is a collection of nodes representing views, connected by edges representing small object rotations. Aspect-graphs can be used to cluster silhouette views of 3D models based on their shape similarity [7]. This method cannot be directly applied to general 3D object recognition as real objects display much more information than just their boundary contours. Possibly closest to our approach is Noor et al.'s [8] aspect-graph for 3D object recognition based on feature matching and temporal adjacency between views.

3 System Description/Implementation

The goal is to rapidly learn the representation of 3D objects with a few training examples, and rapidly recognise the objects and their orientation from unseen test images. The assumptions are that: objects have some texture; images are available for each object without occlusion or background clutter in the training phase; and the camera and objects move slowly relative to each other to produce crisp images. We limit the scope to ten real objects as shown in Figure 1 and view objects around a single plane.

The learning process in the vision system allows for the incremental formation of object representations. The learning algorithm is iterated over all frames in the video stream so that a multi-view object representation can be formed for a single object. This process is repeated for each object to be learnt. Figure 2 depicts the stages of the learning algorithm applied to each training frame.

Capture Frame. A rotating platform captures views of the object at 15 frames per second, producing an image every half-degree of rotation.

Segment Object. Determines a bounding box around the object within the entire image frame by colour separating the distinct fluoro background. This step ensures that the system learns the object within the image frame and not the entire frame itself.

Extract Features. Applies SIFT to identify and describe local key-points in the object image. A SIFT feature is composed of a key-point (x, y, σ, θ) and a 128-dimensional descriptor.

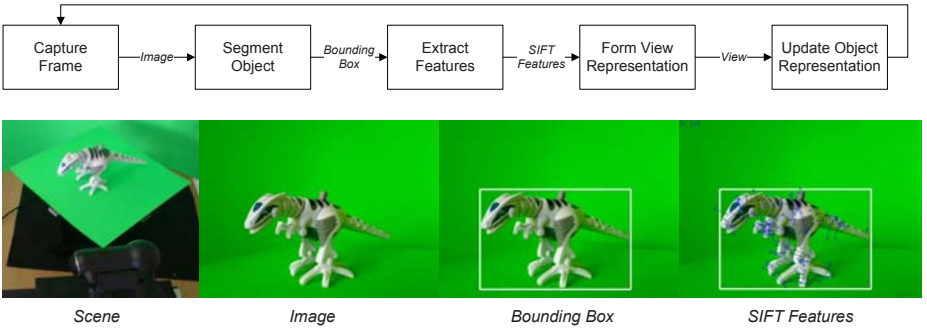

Fig. 2. The stages of the object learning algorithm illustrated with the 'dino' object

Form View Representation. Stores a collection of the extracted features. The (x, y) co-ordinates of the features are recorded relative to the segmentation bounding box.

Update Object Representation. Stores the view and associated features in the object database. If the newly formed view significantly matches an existing learnt view, it will be discarded.

The recognition process involves the steps depicted in Figure 3:

Capture Frame. Captures an image from the camera or video.

Extract Features. Finds SIFT features in the entire frame.

Determine View Matches. Matches extracted features to features in the object database. Matches are grouped by views of the object that they are likely to represent. The implementation parameterises affine, perspective and similarity transforms, mapping key-points in a model-view to a test-view. Geometric verification is applied to each of the view-groups to identify the

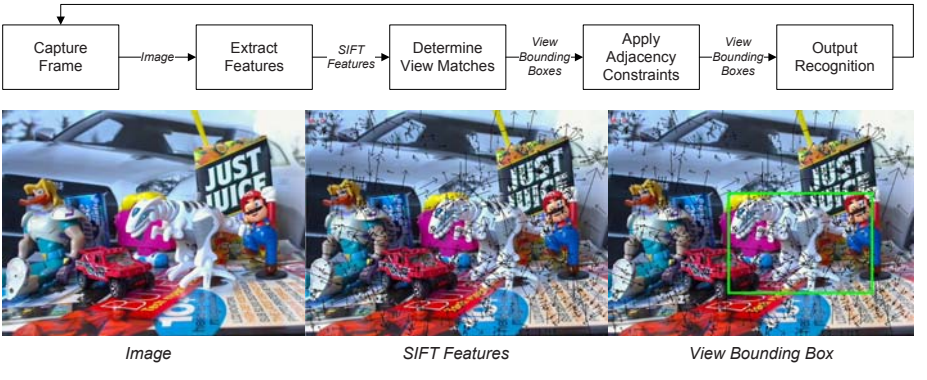

Fig. 3. The stages of the object recognition algorithm, illustrated with the 'dino' object

transformation that maps key-points in the stored view to key-points in the test frame. The transformation-model is used to place a bounding-box around the view of the object in the test-frame.

Apply Adjacency Constraints. Determines whether a change in the view of an object in the test frame is permissible using the aspect-graph. If the detected views over time do not correspond to a traversal of the aspect-graph, the views are rejected.

Output Recognition. The final results of the recognition correspond to bounding-boxes around the successfully detected objects.

We have developed a GPU implementation of SIFT that is 13 times faster (24 frames per second) and 94% accurate compared to the OpenCV implementation. It is almost twice as fast as the leading publicly available GPU implementation [9] when run on the same hardware. A full account of the GPU implementation of SIFT is beyond the scope of this paper.

A SIFT feature is said to *correspond to* or *match* another feature if the two descriptors are similar. We use the nearest-neighbour *ratio* strategy [1] that rejects matches if the ratio of the distance between a feature's nearest-neighbour and second nearest-neighbour is high. We also use a Euclidean distance metric to stop distant features from 'matching' during the initial construction of the feature database. To avoid this method from rejecting valid matches to similar SIFT features in different views we store unique view-keypoint pairs in our object model database.

Finding the two nearest neighbours of the test descriptor in the collection of descriptors stored in the database is $O(n)$ and hence prohibitively expensive. We have implemented an approximate kd-tree, Best-Bin-First (BBF) search. The accuracy varies with the number of bins searched, typically 200 [1,4]. We experimentally found a hyperbolic relationship between the accuracy of the search and the maximum number of bins searched, m (Figure 4(a)). By comparing the accuracy results to the speed results (Figure 5 (b)), we trade-off a 200% speed improvement with a 5% loss in accuracy by reducing m from 200 to 50. This

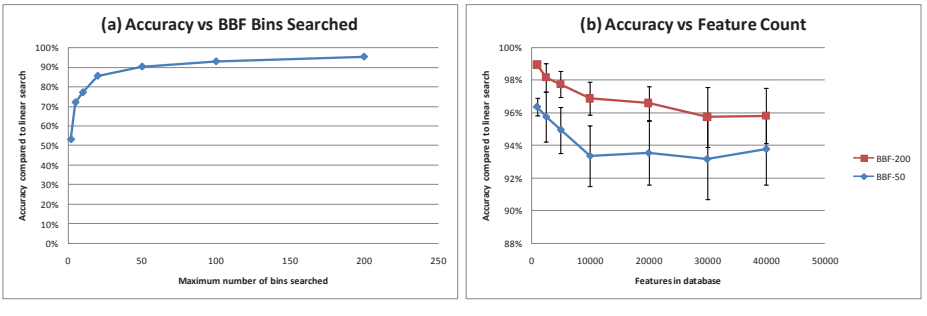

Fig. 4. Left: The accuracy of BBF increases with the number of bins searched. Right: the reduction in accuracy with increased feature numbers.

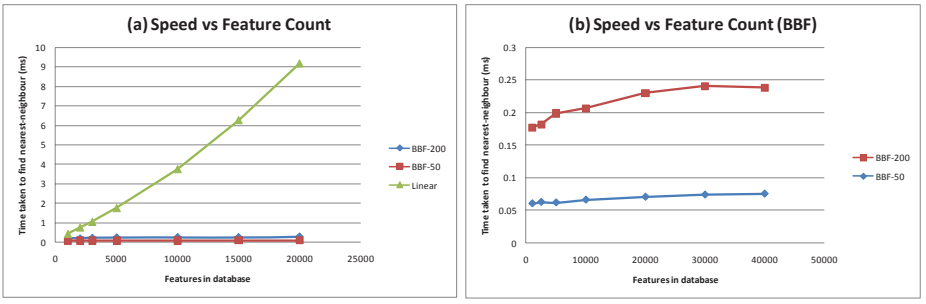

Fig. 5. Left: BBF search speed is almost constant whereas the full nearest-neighbour search is proportional to the number of features. Right: BBF searching 50 bins is significantly faster than searching 200 bins. Error bars are not shown as the standard errors are an order of magnitude smaller than the timing value.

approach is approximately 90% as accurate as linear searching but over 100 times faster for datasets as large as 20,000 features (Figure 5 (a)).

The major difference between our view-clusering approach from [5] is the use of RANSAC [10] to simultaneously develop a geometric model and reject outliers. We cluster views based on how closely they are related by a geometric transform and record view clusters that are *geometrically adjacent* - related by a geometric transform, and *temporally adjacent* - consecutive views in the training video-clip. A view is considered *geometrically identical* to another if it is possible to build a similarity transform to map corresponding key-points using an error tolerance of $\epsilon_{identical}$ pixels. If the error tolerance in pixels is greater than $\epsilon_{identical}$ but less than $\epsilon_{adjacent}$ then the view is part of a geometrically adjacent cluster.

Two views can have many features in common yet not be similar, for example when an object has the same motif on two sides. To avoid clustering these views we use a bounding box matching technique. The error in the view matching bounding box relative to the segmentation bounding box can be measured by considering the distance between each corner of the match-box to the corresponding corner of the segmentation-box. We require each match-box corner to lie within β% of the corresponding segmentation-box corner. This ensures that the tolerance for bounding box matching is proportional to the size of the bounding box. If a view does not sufficiently match the segmentation bounding box it is not clustered and is stored as a new characteristic view.

As recognition is performed on a continuous stream of frames, it is possible to assume that the location and pose of an object through time will remain temporally consistent and there are no rapid changes in the location or pose of the object through time. Restrictions can be imposed to eliminate random view matches which are not temporally consistent. Rather than develop explicit temporal features, temporal consistency is enforced by ensuring consecutive frames from a test video sequence of an object are consistent with the aspect-graph.

4 Evaluation

While there are many image databases available, such as the COIL-100 image library [11], they generally do not expose any variability in the object images. All images are taken by a uniform camera under uniform lighting with no occlusions or clutter. Instead, we examine the ability of our system to recognise objects in live data from a web-cam. Our system performs with 100% accuracy with unseen video under training conditions and this allows us to measure the accuracy of the system under variations such as clutter and occlusion, etc.

Our first experiment involves learning a representation for each of the sample objects. An object from the sample set is first selected and placed on the centre of the rotating platform. The camera is positioned such that the centre of the rotating platform is 20cm from the plane of the camera. A fluorescent light source shines light toward the object from the left of the camera. The 'slow' speed on the rotating platform is used to rotate the object in an automated manner and a video clip of the viewing circle of the object is recorded. The recording commences when the front of the object is facing the camera and ceases after the object has completed a full 360° revolution. Each clip is approximately 60-65 seconds and the entire set of object clips amounts to 5.13GB of loss-less video. We apply the aspect-graph clustering algorithm to each of the training video clips.

Speed of Learning and Recognition. The system is able to construct a representation for an object at 7-9 frames per second. This speed is suitable for learning from live video. Object recognition speed is 11-15 fps in a largely featureless background. In highly cluttered, feature-rich scenes, the recognition speed reduces to 7-10 fps due to the computational cost of feature extraction and matching. This speed is suitable for performing recognition on a live video stream.

Model Size. The view clustering algorithm is able to compress the viewing circle of the sample objects into 20-57 characteristic views (Figure 6). On average, each view consists of 120 features. The more textured and detailed the object, the more features are required to represent it.

Accuracy and Robustness. The recognition system is able to accurately classify, localise and identify the pose of the objects in video streams under considerable variation (Figure 7). The accuracy is 100% for the training clips and diminishes as more variations are introduced. As clutter is added to the background, the accuracy reduces by various amounts for the different objects. Objects that generate more SIFT features, especially features within the body of the object as opposed to the boundary, fare better under heavy clutter.

Most objects can withstand a scale change between −10cm to +15cm. As the resolution of the object decreases with scale change, the accuracy of object recognition rapidly diminishes. Scales changes could be explicitly accommodated in the aspect graph by modelling zooming as well as rotation transitions.

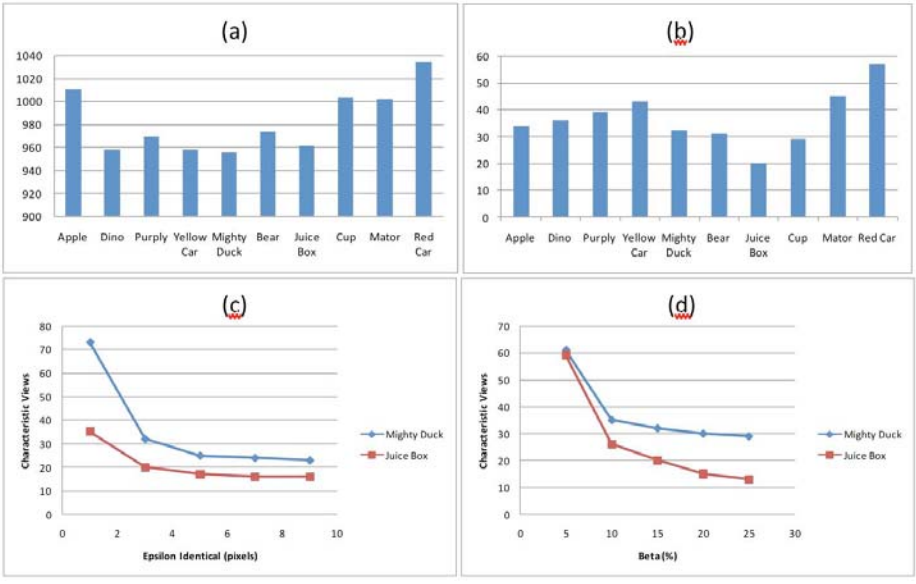

Fig. 6. Learning ten 3D objects. (a) the number of frames in the video sequence, (b) the number of view clusters formed, (c) varying epsilon, (d) varying β.

Fig. 7. (a) recognition of significantly occluded objects, (b) object with background clutter, (c) recognition of deformable object, (d-e) examples of scale change, (f) lighting variations, (g) misaligned bounding box with scarcity of cup-handle features, (h-i) blur

Lighting variation can severely affect recognition if the object exhibits few features or a majority of the features are formed as a result of shadows. Objects that are highly textured due to printed or drawn patterns fare better than those that are sculpted.

The recognition system is very robust to partial occlusions. Only 4-5 feature matches are required for accurate recognition. Feature-rich objects can be accurately classified even when the object is more than 80% occluded. The system is

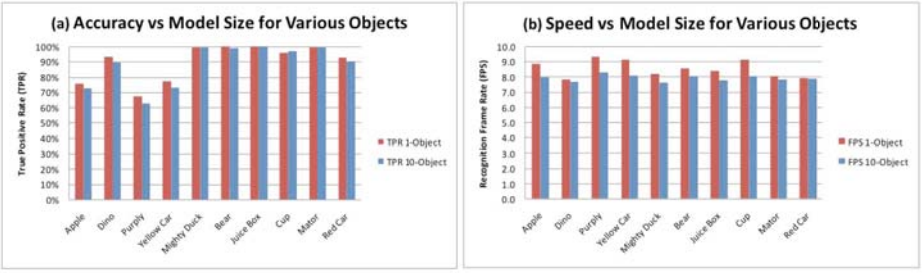

Fig. 8. Accuracy and speed results for the recognition of various objects under severe clutter using a single object model and a 10-object model

able to recognise objects exhibiting minor deformations or can explicitly model the deformations during the learning phase.

The recognition system is able to cope with severe focal blur but is unsuitable to recognise objects in frames which exhibit significant motion blur.

The accuracy of the standard system, without using temporal properties, produces an extremely low false positive rate ($FPR < 1\%$). Temporal constraints can be used to reduce the FPR to near-zero. The system is able to perform more accurate object recognition using objects that are feature-rich. Objects that are highly textured exhibit true positive rate (TPR) accuracy rates above 90% under severe clutter; whereas plain objects exhibit TPR accuracy rates above 67%. The TPR rates can be brought above 80%, for all objects, by exploiting the temporal properties of video and aggregating SIFT features from contiguous frames at the expense of increasing the near-zero FPR to 4%. The recognition system is able to recognise objects with non-reflective surfaces and non-sculpted textures under a considerable amount of environmental variation. Objects that are self-similar can cause the system to confuse similar views during the pose identification process.

Scalability. The recognition system is capable of loading 10 object models into memory and performing recognition at above 7 fps in heavy clutter. Increasing the model size from 1 object to 10 objects does not significantly impact the accuracy of the recognition process (see Figure 8) suggesting that the system should be able to scale to more objects. The system is capable of simultaneous multiple object recognition without showing significant degradation in speed or accuracy, but cannot currently recognize multiple identical objects.

5 Conclusion

The 3D object learning and recognition system described entails several innovations to achieve both speed and high accuracy under varying conditions of occlusion, clutter, lighting, and scale. They include using both geometric and temporal consistency checks from the aspect-graph, comparing estimated bounding boxes between training and test images, and a tuned BBF kd-tree nearest

neighbor search. Future work could see the the system benefit from GPU feature matching, active vision, and feature tracking. The system is able to identify object orientation which could be exploited in the robotic manipulation of objects, visual SLAM, and intelligent surveillance applications.

Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
2. Beis, J.S., Lowe, D.G.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: CVPR, pp. 1000–1006. IEEE Computer Society, Los Alamitos (1997)
3. Lepetit, V., Fua, P.: Monocular model-based 3d tracking of rigid objects. Found. Trends. Comput. Graph. Vis. 1(1), 1–89 (2005)
4. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
5. Lowe, D.G.: Local feature view clustering for 3D object recognition. In: CVPR, pp. 682–688. IEEE Computer Society, Los Alamitos (2001)
6. Revaud, J., Lavoué, G., Ariki, Y., Baskurt, A.: Fast and cheap object recognition by linear combination of views. In: ACM International Conference on Image and Video Retrieval (CIVR) (July 2007)
7. Cyr, C.M., Kimia, B.B.: 3D object recognition using shape similarity-based aspect graph. In: ICCV, pp. 254–261 (2001)
8. Noor, H., Mirza, S.H., Sheikh, Y., Jain, A., Shah, M.: Model generation for video-based object recognition. In: Proceedings of the 14th ACM International Conference on Multimedia, Santa Barbara, CA, USA, October 23-27, pp. 715–718. ACM, New York (2006)
9. Wu, C.: SiftGPU - a GPU implementation of David Lowe's scale invariant feature transform, SIFT (2007), `http://cs.unc.edu/~ccwu/siftgpu/` (accessed: December 18, 2007)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
11. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (COIL-100). Columbia University (1996) (accessed: May 1, 2008)