Tracking Moving Objects Improves Recognition

Nimit Dhulekar[1], Andrew Felch[2], and Richard Granger[1, 2]

1 Department of Computer Science, Dartmouth College

2 Psychological and Brain Sciences Department, Dartmouth College

Abstract

We describe a new family of algorithms that analyze time-varying scenes, recognizing and tracking learned objects over time. The new methods are intended to address key questions of moving images, including unpredictable moment-to-moment changes in location, size, orientation, lighting, and occlusion. We introduce a novel task in which objects revolve and rotate while suspended from a mobile's arms; the recognition & tracking algorithm incorporates characteristics of a number of prior published methods, combining them in a novel fashion to enable this newly introduced task. Other methods have found that improving recognition will improve tracking; we show that improved tracking improves object recognition.

1. Introduction

Most work on visual recognition has focused on static images, and typical studies of moving image recognition usually depend on the existence of extended datasets of complex movie scenes containing labeled identified objects in the frames of the scene, requiring extensive (and expensive) human labeling.

There are no agreed-upon engineering specifications of the task of recognizing and tracking moving objects in scenes; the task is defined by comparison with human abilities, and the belief in its computability arises solely from the fact that humans (and other animals) readily perform it. However, the mechanisms by which these abilities arise in biological systems are unknown. Our work has been based on deriving a series of perceptual and cognitive algorithms from brain circuit operation [20]. Here we describe a newly derived method for recognizing and tracking objects in time-varying scenes.

We introduce a new dataset in which objects are suspended from a revolving mobile and viewed from a fixed camera. The objects are subject to revolution of the mobile, change in apparent location and size due to mobile revolution, rotation in-place and oscillatory wobble, and time-varying occlusion due to passing behind other objects during mobile revolution.

Training in this task is performed only on a set of static images of the objects, outside the mobile environment. Testing consists of recognizing and tracking each of four objects suspended from the moving mobile.

Although changes to size, location, rotation, lighting and occlusion present difficulties beyond those of static images, they also confer information about successive incremental object changes, thus providing valuable constraints over time about the objects and their various poses and rotations.

2. Detailed method description

For a dataset of objects, initial training is performed on a limited fixed set of static images of each object (four images per object in the experiments reported here). The system is then exposed to videos of the objects suspended from the arms of a mobile, and thus revolving with the mobile as well as freely rotating in place.

Video processing is described here in three stages: individual frames, frame sequences, and final object recognition.

Stage 1: Individual frames. Input to the algorithm are real-time sequences of video frames; currently 10 frames/sec are used for videos of 30-60 sec duration. An edge detection algorithm is run on an initial frame of the video. Bounding boxes are constructed

around the edges of identified blobs in the frame. The steps in the edge detection algorithm are as follows:

i) Contrast gradients are calculated via Sobel filter to create a binary mask containing the segmented object.
ii) Gaps in lines surrounding the Sobel image are dilated using linear structuring elements in vertical and horizontal directions.
iii) Vacant locations in image's interior are filled.
iv) The resulting blob is smoothed two times via erosion with a diamond structuring element. The resulting blob has continuous boundaries.
v) We identify each such blob in the image by traversing connected points on its boundary.
vi) By identifying the boundary of each blob, we can compute bounding boxes around it by taking the top left and bottom right coordinates of each blob.

This results in a set of bounding boxes surrounding blobs that may contain one or more objects. Subsequent search is constrained to use only the enclosed areas.

Stage 2: Sequential frames. Input to this stage consists of the bounding box information computed in stage 1. The objects exhibit instantaneous changes in direction of motion, both revolving on the mobile and rotating in place, thus subject to changing shadows and unpredictably occluding and un-occluding each other over time. Frame-to-frame changes are used to distinguish between individual objects vs. composites that may arise due to shadows or occlusion:

vii) For each frame, compare bounding-box positions with those in succeeding frame. If box is changing size, either:
   a. It may contain multiple objects occluding each other, or
   b. A single object may be rotating, exhibiting unequal widths.
viii) If box retains size but changes position, it may contain one or more objects moving together.
ix) For rules in vii) and viii), blobs are either retained as denoting single objects or separated into multiple proposed objects. The process is repeated for all successive frames.
x) After a single complete pass, a second pass is run to interpolate the positions of objects in frames that previously had not been separated.

The result of this stage is a set of blobs, i.e., distinct regions per frame in which a proposed object may occur. The next stage identifies each blob as one of the objects in the training dataset.

Despite only labeling a few instances of (static) training objects, the processing that occurs on the video results in an extended series of automatically-labeled training data containing different poses, size (distance), lighting, and partial occlusion.

Stage 3: Object recognition. Object recognition using local invariant features is based on the following concept: keypoints extracted from the test image and the reference model are matched against each other. Commonly employed local interest point detectors are the single-scale Harris detector [13] and the multi-scale Lowe's sDoG+Hessian detector [3]. The best performing interest point detectors are the Harris-Affine and the Hessian-Affine [19]. Use of the SDoG+Hessian detector, SIFT descriptors [3, 14] and a probabilistic hypothesis rejection stage is very common, due in part to its near-real-time speed. That system though suffers from large false-positive detection rate as a consequence of its use of just a simple probabilistic hypothesis rejection stage, which is unable to reduce the number of false positives. The L&R method [15-17] reduces false positives via several hypothesis-rejection stages, but is limited in the angles, perspectives, and amount of occlusion that it can handle [18]. Since our dataset consists of objects in varying perspectives, widely different angles and partial to complete occlusion, we developed a modified and extended system.

Using the output of stage 2, blobs within bounding boxes are matched against learned objects. Candidate objects are assigned via a

nomination, runoff, and election procedure, to result in a proposed identification of each blob as one of the objects in the training dataset. The matching method is implemented as a two-pass algorithm applied across the video dataset. In the first pass:

xi) For each blob identified in stage 2, identify the earliest frame in which that blob has less than 10% overlap with any other blob.

xii) Compare the blob against the training set objects using SIFT descriptors with second-nearest-neighbor distance of 0.5 ("low threshold match").

xiii) For each match obtained between a blob and a candidate object (CO) in the training dataset, "nominate" the CO to be considered as a specific candidate for that blob.

xiv) Sum all frame occurrences of nominated COs and record occurring frame numbers.

xv) Return top five nominated COs for each blob.

The result is a set of five nominated COs that may be instances of each blob. The second pass of the object recognition algorithm is performed for each of these nominated COs and returns the top two promoted COs as the "runoff" candidates for that object's identity.

xvi) Compare blob v. COs using SIFT descriptors with higher threshold (second nearest neighbor distance 0.95).

xvii) Determine similarity via Hough transform [see 14] (bin sizes of 30 degrees for orientation axis, factor of 2 for scale axis, and 0.25 times width and height for each position axis).

xviii) For each bin B created in xvii), perform pre-affine elimination tests:

a. eliminate invalid bins (those with <4 votes)

b. eliminate bin B if direct neighbor in Hough space has more votes.

c. If either of the linear correlation coefficients r_{REF} or r_{TEST} of interest points in the bin B are > 0.9, eliminate bin (the points lie in a nearly straight line, and affine transforms can be unstable).

d. If fast probability [21] associated with bin B is < 0.9, eliminate bin.

xix) Calculate initial affine transform T_B using matches in bin B. Matches from all non-eliminated bins are added to bin B.

xx) For each bin B, identify correlation between blob interest points and those in bin B. For interest points A1 and B1 in the test set and A1' and B1' in the training set:

a. Scale test: Compute ratio of the scale at A1 to the (Euclidean) distance A1B1, and the corresponding ratio for A1' and B1'. If the ratio > 0.2, eliminate the bin.

b. Orientation test: Compute relative orientation of the points (difference between the points' orientation according to SIFT vs. the line segment between the points. If the difference of differences is > exponent of 0.2, eliminate the bin.

xxi) For all bins that voted for a particular object pose, compute Lowe's probability [see 14] p_L; if $p_L < 0.95$, eliminate the bin.

xxii) For all matches in all bins that voted for the object pose, compute pixel correlation r_P using TB. If $r_P < 0.2$, eliminate this candidate object pose.

Remaining best-ranked candidates correspond to the guesses of the system of each object blob's identity.

3. Results

Table 1 lists the twenty-four objects used in the reported experiments; each was a plastic toy measuring a few inches in each dimension. Many looked very similar, differing only in 1-2 features.

For each object, four static views were used for initial training (left side, right side, top, bottom). Figure 1 illustrates some representative instances of the objects and some of the view data.

brown frog	harley motorcycle	black car
green frog	ducati motorcycle	orange car
pale frog	orange motorcycle	black truck
brown lizard	blue motorcycle	red truck
chameleon	purple motorcycle	
dinosaur	yellow dirt bike	basketball
gecko		rugby ball
green turtle		soccer ball
orange turtle		
tortoise		
crocodile		

Table 1: The 24 objects used in the experiments.

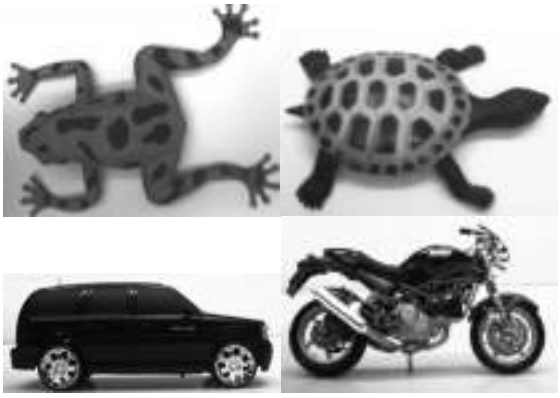

Figure 1: Representative training images (brown frog, green turtle, black truck, ducati motorcycle).

For each experiment, four of the twenty-four objects were chosen and suspended from the mobile in arbitrary positions. Figure 2 illustrates a sample set of closeup snapshots taken from sequential frames of one mobile video episode. It can be seen that the objects have changed very slightly in rotation, lighting, and occlusion.

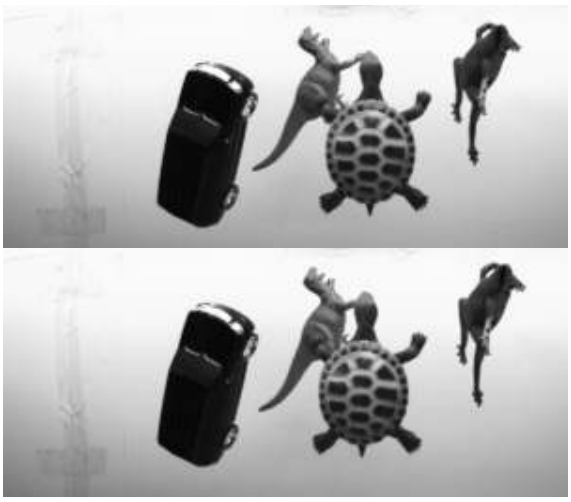

Figure 2. Sequential frames from a sample video.

From the raw inputs (Figure 2), the images undergo a series of transforms partially illustrated in Figure 3: processed outlines after dilated gradient mask (top), initial blobs after filing and smoothing (middle) and with preliminary uncorrected bounding boxes (bottom). Further details and intermediate transforms were described in the extended algorithm specification in Section 2.

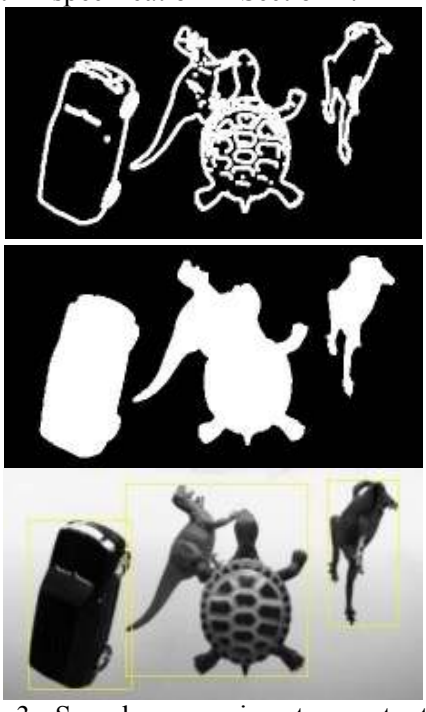

Figure 3. Sample processing stage outputs (see text)

Two natural measures arise from the experiments: the ability of the method to locate and segment objects (i.e., to correctly define bounding boxes), and the ability to identify a particular object within a bounding box. (It should be noted that the algorithm has no knowledge of the number of objects that will appear).

For 23 of the 24 objects, the method correctly identified bounding boxes in all tested instances i.e., precision = recall = 1.0 for 23 objects. One object, the soccer ball, was never correctly segmented, so for this object, precision = recall = 0.

Figure 4 shows an instance of an unsuccessful segmentation of a soccer ball.

Figure 4. Unsuccessful segmentation (see text).

Once objects were successfully segmented and bounded, the algorithm attempts to label them as one of the 24 objects on which it had been trained. In this task there are in principle no false positives, only correct identifications or false negatives. As described in Section 2, the algorithm rank-orders potential names for identified objects; we measured the percent of instances in which the algorithm identified the correct label as the top rank order (70.8%) and when the correct label was either of the top two rank-ordered identifiers (83.3%).

Figure 5 illustrates three particular sample instances of successfully segmented objects, showing correctly-segmented frames in the presence of occlusion. In the figure, the yellow bounding boxes arise from stage 1 of the algorithm (Section 2 above) whereas the blue boxes are the result of stage 2 of the algorithm.

Figure 5. Instances of correct segmentation in the presence of occlusion (see text).

Figure 6 shows an instance of the final output of a frame analysis, in which the selected label is added above each corresponding bounding box. In the example shown, the method has found three correct (black truck, green turtle, brown frog) and one incorrect (chameleon) object label. The mislabeling of the dinosaur as a chameleon is presumably due to extensive feature similarities between the two: long snout, clawed toes, tail, scales).

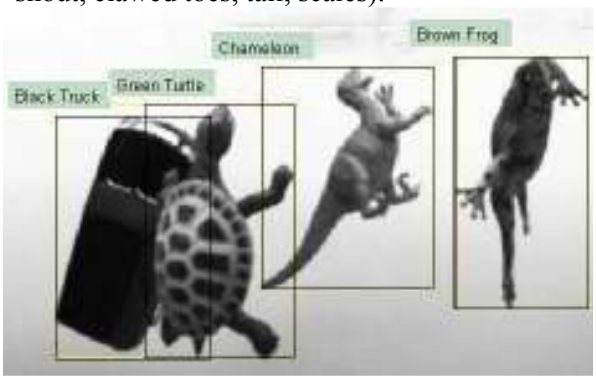

Figure 6. Final object labeling.

4. Discussion

Real visual processing typically entails objects seen from multiple viewpoints, in different lighting, changing over time, partially occluded by other objects. We have described a family of algorithms that addresses processing of objects under these relatively real-world conditions.

In the process, we introduced a novel task, in which objects suspended from a mobile revolve and rotate; the aim of the algorithm is to recognize and track the objects over time. The algorithm incorporates characteristics of a number of prior published methods, combining them in a novel fashion to enable this newly introduced task.

We showed that in this challenging task, the algorithm performs very well across a set of objects in the presence of ongoing changes in location, lighting, occlusion, size, and direction of view. Having been recognized, each object was successfully tracked through the 1-min video.

It is notable that most work in moving videos has taken the approach of using improved static object recognition in order to achieve better tracking over time e.g., [1, 2]. In such cases, images are matched against training data to update their positions in successive video frames. Feature descriptors such as SIFT [3], SURF [4], and GLOH [5] perform well for static image matching, including good tolerance for different viewpoints and lighting [6-8].

In the present paper we go the other way, using the fact that the same objects are moving incrementally over time and thus exhibit predictable constraints over moment-to-moment changes. Unlike previous research [e.g., the above plus refs 9-12], our algorithm is based not on tracking robust feature descriptors over frames but rather using consecutive time constraints to successively improve the performance of the object recognition method over elapsed time. Since consecutive frame images are similar we can track individual objects over movement. In this way we greatly increase the probability of obtaining a sequence of frames that are very similar to one or more of the few static labeled training images. We thus obtain extensive additional data with no additional training or labeling costs.

Further investigation is ongoing; it is hoped that the introduction of the new methods will spur further studies into the relationship between tracking and recognition, and the new mobile task (which we intend to make freely available online) is proffered as a potentially useful (and easily augmentable) dataset for further study of these issues.

Acknowledgments: The authors would like to thank Bennet Vance for fruitful discussion and advice, as well as his help in constructing the object recognition algorithm. A special thanks also to Javier Ruiz-del-Solar and Patricio Loncomilla for their help in understanding their L&R algorithm. Correspondence should be sent to Nimit Dhulekar.

References

[1] Ta D.N., Chen W.C., Gelfand N., and Pulli K.: SURFTrac: Efficient Tracking and Continuous Object Recognition Using Local Feature Descriptors. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2009), 2937-2944.

[2] Foresti G.L.: Object Recognition and Tracking for Remote Video Surveillance. IEEE Transactions on Circuits and Systems for Video Technology, 9 (7), (1999).

[3] Lowe D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), (2004), 91–110

[4] Bay H., Andreas E., Tuytelaars T., and Van Gool L.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding, 110(3), (2008), 346-359.

[5] Mikolajczyk K. and Schmid C.: Performance Evaluation of Local Descriptors. IEEE Transactions on Pattern Analysis & Machine Intelligence, 27(10), (2005), 1615–1630,

[6] Sivic J. and Zisserman A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision, (2003), 1470–1477.

[7] Grauman K. and Darrell T.: The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. Proceedings of IEEE International Conference on Computer Vision, 2, (2005), 1458–1465.

[8] Nister D. and Stewenius H.: Scalable Recognition with a Vocabulary Tree. In CVPR '06: Proceedings of the 2006 IEEE

Computer Society Conference on Computer Vision and Pattern Recognition, (2006), 2161-2168.
[9] Skrypnyk I. and Lowe D. G.: Scene Modelling, Recognition and Tracking with Invariant Image Features. In ISMAR '04: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR), (2004), 110-119.
[10] Liu C., Yuen J., Torralba A., Sivic J., and Freeman W.T.: SIFT Flow: Dense Correspondence Across Different Scenes. In ECCV '08: Proceedings of the 10th European Conference on Computer Vision, (2008), 28-42.
[11] Chekhlov D., Pupilli M.L., Mayol-Cuevas W.W., and Calway A.D.: Realtime and Robust Monocular SLAM Using Predictive Multi-resolution Descriptors. In 2nd International Symposium on Visual Computing, (2006), 276-285.
[12] Wagner D., Reitmayr G., Mulloni A., Drummond T., and Schmalstieg D.: Pose Tracking from Natural Features on Mobile Phones. In ISMAR '08: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, (2008), 125-134.
[13] Harris C. and Stephens M.: A Combined Corner and Edge Detector. In Proceedings of the Fourth Alvey Vision Conference, (1988), 147–151.
[14] Lowe D.: Local Feature View Clustering for 3D Object Recognition. In 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01), 1, (2001), 682-688.
[15] Loncomilla P. and Ruiz-del-Solar J.: Gaze Direction Determination of Opponents and Teammates in Robot Soccer. Lecture Notes in Computer Science, 4020, (2006), 230–242.
[16] Loncomilla P. and Ruiz-del-Solar J.: A Fast Probabilistic Model for Hypothesis Rejection in SIFT Based Object Recognition. Lecture Notes in Computer Science, 4225, (2006), 696–705.
[17] Loncomilla P. & Ruiz-del-Solar J.: Robust Object Recognition Using Wide Baseline Matching for RoboCup Applications. Lecture Notes in Computer Science, 5001, (2008), 441–448.
[18] J. Ruiz-del-Solar and P. Loncomilla, Robot Head Pose Detection and Gaze Direction Determination Using Local Invariant Features. Advanced Robotics, 23(3), (2008), 305-328.
[19] Mikolajczyk K., Tuytelaars T., Schmid C., Zisserman A., Matas J., Schaffalitzky F., Kadir T., and Van Gool L.: A Comparison of Affine Region Detectors. International Journal of Computer Vision 65(1), (2005), 43–72.
[20] Granger R.: Engines of the Brain: The Computational Instruction Set of Human Cognition. In AI Magazine, 27, (2006), 15-32.