Real-time Learning and Detection of 3D Texture-less Objects: A Scalable Approach

Dima Damen[1]
damen@cs.bris.ac.uk

Pished Bunnun[2]
pished.bunnun@nectec.or.th

Andrew Calway[1]
andrew@cs.bris.ac.uk

Walterio Mayol-Cuevas[1]
wmayol@cs.bris.ac.uk

[1] University of Bristol
Bristol, UK

[2] National Electronics and Computer Technology Center
Bangkok, Thailand

We present a method for the learning and detection of multiple rigid texture-less 3D objects intended to operate at frame rate speeds for video input. The method is geared for fast and scalable learning and detection by combining tractable extraction of edgelet constellations with library lookup based on rotation- and scale-invariant descriptors. Most shape-based detectors either require offline training or scale linearly as more objects are being searched for, or commonly both. To address speed and scalability in learning and testing, this paper proposes the use of predefined paths that specify the relative direction between edgelets, and importantly, make the search tractable for real-time operation. The traced edgelets are represented by a simple to compute transformation invariant descriptor, that is used as an index to similarly stored descriptors, in a way that revisits geometric hashing. The approach learns object views in real-time, and is generative - enabling more objects to be learnt without the need for re-training. During testing, a random sample of edgelet constellations is tested for the presence of known objects.

A key and distinguishing element of the method is the use of *path tracing* for both training and testing (Fig. 1). Each path defines the relative direction between the constellation's constituent edgelets. This introduction of paths is critical; as it limits the number of possible constellations and allows tractable generation of a library of descriptors. For a constellation of n edgelets, a **path** Θ is a sequence of angles $\Theta = (\theta_0, .., \theta_{n-2})$. From any starting edgelet, the base angle θ_0 specifies the direction of a tracing vector v_1, initially with unspecified length, relative to the orientation of the starting edgelet. If this tracing vector intersects with another edgelet in the edge map, then the edgelet is added to the constellation.

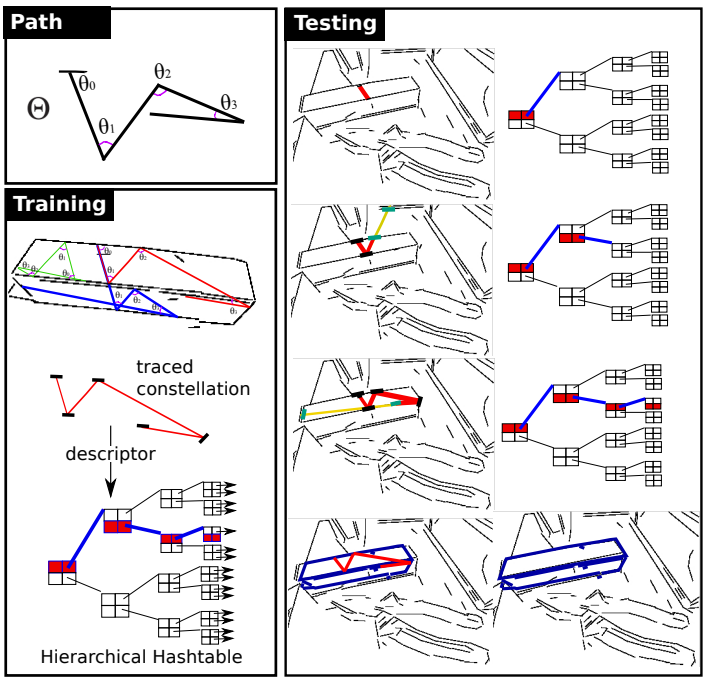

Figure 1: For a given tracing path, constellations of edgelets are traced out from training views exhaustively, and an affine-invariant descriptor for each constellation is inserted into a quantised hierarchical hash table. For a test image, constellations are traced out using the same path. Candidate detections are found by comparing the descriptor to the hash table, tested using all the training edgelets, and refined using iterative closest edgelet.

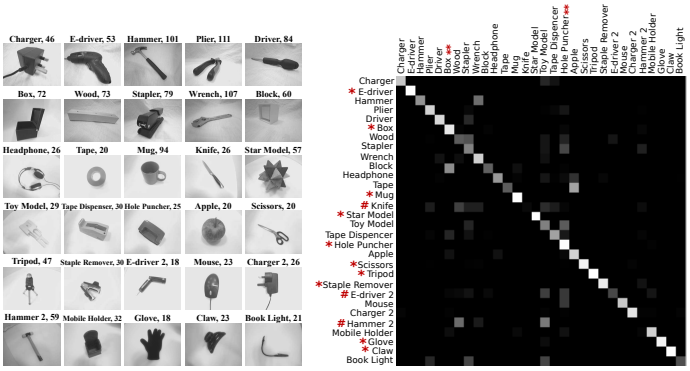

Figure 2: Thirty texture-less objects in the dataset along with the confusion matrix from 10 runs. Ten objects achieved recall > 90% (*), two of them with precision < 90% (**). Three objects are difficult to detect with recall < 30% (#).

The next tracing vector v_2 then has the direction θ_1 relative to v_1, i.e. $cos(\theta_1) = (v_1 \cdot v_2)/(|v_1||v_2|)$. This process continues until the constellation has n edgelets.

For a traced constellation, the descriptor specifies the relative orientations and distances between the consecutive edgelets in the constellation's tuple. By keeping a comprehensive library of descriptors for all constellations guided by one path Θ from all starting edgelets, it is sufficient to extract one constellation using the same path from the object in the test image to produce a candidate detection that is verified using the rest of the view edgelets. Several paths are used and a separate library is built for each chosen path. The choice of paths is discussed in the paper.

The method is tested on a dataset of 30 texture-less objects. It trades recall for speed, testing a sample of edgelet constellations in each processed frame. The method is tested at frame rates varying from 1 to 17 fps. At 7fps, recall of 50% (precision = 74%) was achieved when 30 objects were learnt (1433 views - Fig. 2). As the number of objects in the library increases from 1 to 30, the increase of detection time is dependent on the shape's ambiguity rather than the number of objects.

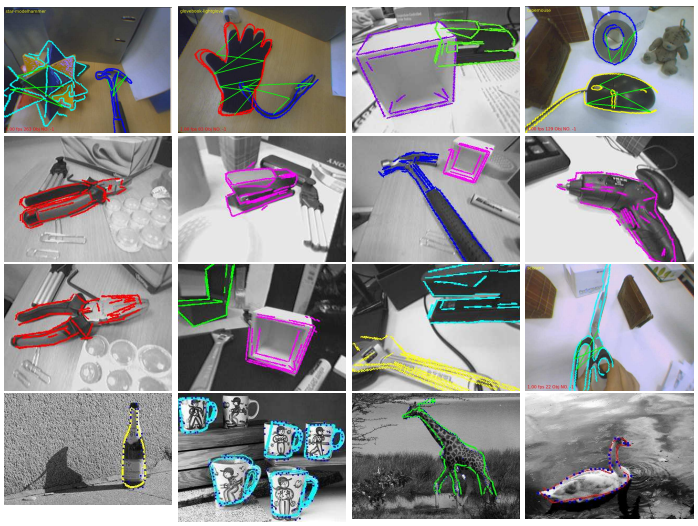

Figure 3: Sample set of results on the tools and ETHZ datasets.