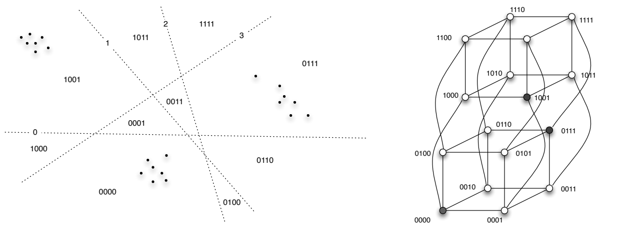
Fast Segmentation via Randomized Hashing

Camillo J. Taylor
http://www.cis.upenn.edu/~cjtaylor/

Anthony Cowley
acowley@seas.upenn.edu

GRASP Laboratory
University of Pennsylvania
Philadelphia PA, USA



(a) (b)

Figure 1: This figure depicts a simplified 2D version of the randomized hashing scheme. Figure (a) depicts a 2D feature space fractured into regions by a set of randomly chosen splitting planes. Each region is associated with a hash code indicating where it falls with respect to the splitting planes. The set of all hash codes can be associated with the vertices of a hypercube as shown in Figure (b) here the shading of the nodes indicates how many feature vectors are hashed to that code. The segmentation scheme proceeds by identifying local maxima in this hash code space.

Segmentation, the problem of breaking an image into coherent regions is, of course, a fundamental problem in Computer Vision. This paper proposes a new approach to the segmentation problem that leverages ideas developed in the Theoretical Computer Science literature to derive a new feature space based clustering algorithm that is amenable to real time implementation.

Each pixel in the image is described by a feature vector which encodes a set of properties used to describe that pixel. In all of the experiments described in this paper we employ a simple color descriptor vector but one could equally easily use more sophisticated feature vectors such as a histogram of color values or a vector of texture coefficients.

These vectors are then hashed using a set of randomly chosen splitting planes to yield binary vectors. Salient clusters in the hash space are automatically identified by considering the populations associated with various hash codes. Figure 1 shows a simplified view of this procedure in two dimensions.

The notion of randomized hashing has been employed before most notably by Indyk and Motwani in the context of Locality Sensitive Hashing [2]. These authors used a similar approach to hash a set of vectors into a set of discrete bins in order to accelerate the search for nearest neighbors. Their approach leveraged the fact that this randomized hashing procedure tends to preserves locality so points that are near to each other in the feature space are hashed to the same bin with high probability. The proposed segmentation scheme leverages the same phenomenon for a different purpose - namely to cluster the feature vectors into groups. The entire scheme is outlined below in pseudo-code.

Algorithm 1 Segmentation via Randomized Hashing

1: Hash each feature vector to an n-bit code using the n randomly chosen splitting planes
2: Maintain a count of the number of feature vectors mapped to each hash code
3: Identify local maxima in the code space - these are the cluster centers
4: Assign each feature vector to the closest local maxima
5: Run connected components on the labeled pixels to identify coherent connected components.

In order to characterize the performance of the proposed segmentation scheme, experiments were carried out using the Berkeley Segmentation Database [3] . The method was also compared directly to the Mean Shift procedure [1]. These experimental results demonstrate that the scheme can produce results comparable to other state of the art methods. Figure 2 shows some of the results obtained with our method.

Figure 2: This figure compares the output of the automated segmentation procedure to human labeled segmentations. The first row contains the input imagery, the second row contains human segmentations while the third row contains machine segmentations.

A significant advantage of the proposed segmentation scheme is that the computational effort required scales linearly in the number of pixels and the operations required are simple and regular. In order to demonstrate this a real time version of the scheme was implemented on a Macbook Pro laptop computer. This implementation was used to segment 640 by 480 video frames at a rate of 10 frames per second using a single core of an Intel Core 2 Duo processor running at 2.33 GHz. This rate includes the time taken for all phases of the algorithm, image acquisition, randomized hashing, local maxima detection and connected components processing. Since almost all of the steps in the procedure are embarrassingly parallel, the algorithm is a well suited to implementation on modern multi-core processors and GPUs and should be amenable to further acceleration.

[1] Dorin Comaniciu, Peter Meer, and Senior Member. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[2] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, New York, NY, USA, 1998. ACM. ISBN 0-89791-962-9. doi: http://doi.acm.org/10.1145/276698.276876.

[3] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *in Proc. 8th Int'l Conf. Computer Vision*, pages 416–423, 2001.