6th Int. Conf. on Computer Analysis and Patterns , Prague, Czech Republic, September 6-8 1995

Spatial and Feature Space Clustering: Applications in Image Analysis

J Matas and J Kittler

University of Surrey, Guildford, Surrey GU2 5XH, United Kingdom

Abstract

We propose a novel approach to image segmentation, called *feature and spatial domain* clustering. The method is devised to group pixel data by taking into account simultaneously both their feature space similarity and spatial coherence. The FSD algorithm is practically application independent. It has been successfully tested on a wide range of image segmentation problems, including grey and colour image segmentation, edge and line detection, range data and motion segmentation. In comparison with existing segmentation approaches, the method can resolve image features even if their distributions significantly overlap in the feature space. It can distinguish between noisy regions and genuine fine texture. Moreover, if required, FSD clustering can produce partial segmentation by identifying salient regions only.

1 Introduction

Many low and intermediate level image analysis tasks in computer vision are essentialy image segmentation problems. The aim of image segmentation is to identify, in the image, regions which can be associated with perceptually meaningful scene primitives. This can mean either a complete image partitioning or the extraction of a subset of such regions selected according to some criteria. As pixels in such regions represent the same physical properties or phenomena, they are expected to exhibit a degree of similarity, i.e. a clustering tendency, in the feature space characterising the relevant property. However, a distinctive aspect of image data is its spatial ordering. Thus in the data analysis terms, image segmentation is a process of clustering *spatially indexed data*. Consequently, the grouping of pixels into clusters must take into account not only their similarity in the feature space but also the requirement of their spatial coherence.

Classically, image segmentation problems have often been viewed as purely feature space clustering problems [6]. Accordingly, the vector of image features observed at each pixel is considered as a point in the feature space. Although spatial coherence is not a constraint explicitly built in, each cluster in the feature space is expected to group pixels which on the grounds of their homogeneous properties will come from coherent regions in the image. For low dimensional feature spaces the clustering can be performed efficiently from a statistical summary of the data in terms of the histogram. By analysing the histogram one can determine the groups of pixels satisfying the homogeneity property and thus obtain the corresponding segmentation.

The global statistical analysis of the image data in the feature space has the advantage of providing a good assessment of the clustering tendency of the data. However, feature

space clustering does not automatically guarantee spatial coherence of the cluster based pixel groups. If measurements overlap in the feature space, the segmentation results obtained by this approach can be disappointing, giving a very noisy appearance. This problem has been tackled in the literature in a number of different ways. One possibility is to augment the feature vector either by measurements on the neighbouring pixels, or by pixel coordinates which will encourage spatial consistency of the segmentation result. The former approach has been particularly popular among the remote sensing community [16]. However this dramatically increases the dimensionality of the segmentation problem and the associated computational complexity. Alternatively, one may deal with the issue of spatial coherence by means of postprocessing. Morphological filtering techniques are one of the examples of methods specifically developed for this purpose. The main problem with the postprocessing refinement of the raw segmentation is that at this stage it is impossible to distinguish between noisy labelling and fine genuine image structures such as those characterised by texture.

On the other hand, region growing methods stress spatial coherence. Techniques vary, but typically, from an arbitrary starting point in the image an initially small region is grown to subsume neighbouring regions, provided the appended pixels satisfy a prespecified similarity measure. A number of termination methods have been devised to stop the region growing process [5]. By definition, the extracted regions are spatially coherent. However, the use of local statistics to compute the similarity measure often results in an excessive sensitivity to thresholds leading to oversegmentation or undersegmentation of the image data.

In this paper we argue that, in order to achieve successful segmentation (either image partitioning or selection of salient regions) one should exploit simultaneously global and local statistics that can be computed from the image, together with pixel connectivity information. The global statistics of the clustered features defines a model that can be used in local statistical testing to determine the segment identity of individual pixels. The idea of performing image segmentation by means of a parallel process of feature space analysis under spatial coherence constraints has been voiced by Ballard [1]. However, his paper only identified the issues without advocating any specific solutions to address them. We have developed an effective algorithm that achieves the objectives of simultaneous feature space-image space clustering. Moreover, we show that the proposed approach is completely general and consequently applicable to a wide spectrum of problems, ranging from grey and colour image segmentation, to edge and line extraction, range data and motion segmentation. In other words, the approach defines a unified framework for a family of image segmentation problems. It can also be used either to achieve a complete segmentation of the image or select a subset of image regions according to a prespecified criterion. Currently, region saliancy is used to define priority.

We discuss the relationship of the proposed method to the perceptual grouping approach based on robust statistical testing. We show that the proposed method is more resilient to contamination due to clutter and consequently it can resolve image features to much enhanced resolution. The practical significance of the proposed method is that it can distinguish between noisy regions and genuine texture. This is particularly important for correctly segmenting textured motion fields (resulting from object transparency).

The paper is organised as follows. In the next section we first overview one specific feature space clustering algorithm, namely the graph theoretic (GT) clustering algorithm [9], which is a basic bulding block used in the proposed segmentation method. The novel feature and spatial domain (FSD) clustering method is then introduced. In Section 3 we illustrate the main differences between GT and FSD clustering on a set of test problems. In Section 4 we describe a number of diverse experiments in image segmentation to demonstrate the properties and versatility of the proposed procedure. Finally, conclusions are drawn in Section 5.

2 Algorithm

A large number of image segmentation problems involve low dimensional feature spaces and huge quantities of pixel data. In such situations it is most efficient to base the cluster analysis of the data on its statistical summary in the form of a histogram. Accordingly the clustering problem can be viewed as one of partitioning the feature space into regions over which the histogram is locally unimodal.

The problem of unimodal cluster separation has received significant attention in the pattern recognition community [9][7]. We adopted the Graph-theoretical clustering method of Koontz and Fukunaga. The method can be outlined as follows (for details and comparison with other approaches see [4]).

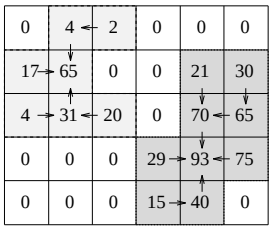

Fig. 1.: Graph-theoretical clustering in 2D space. Arrows symbolise links to bins with the largest count in local neighbourhood. Different shading highlights the two clusters of the example. Although the example used 4-neighbourhood, in this particular case (and generally with well separated clusters) the result is fairly insensitive to the size of searched neighbourhood. In this example, identical clustering is obtained with 8 and 12 neighbourhoods.

Algorithm 1: **Graph-theoretical clustering**

1. Compute feature histogram. In each bin, maintain a list of pixels voting for the bin.
2. For each bin, find the bin with maximal count in a given neighbourhood. Store a link to this bin.
3. Such links form a forest, with a root of each tree in a local maximum. The set of pixels voting for bins in a single tree form a unimodal cluster.

The graph-theoretical clustering algorithm requires two parameters: the size of a histogram bin and the size of the neighbourhood considered when searching for maximal bins in local neighbourhood. Methods for automatic selection of these parameters are given in [7].

The result of the graph-theoretical clustering for a simple example is shown in Figure 1. The arrows symbolise links to maximal bins in the local neighbourhood (in this case 4-neighbourhood). Shaded areas correspond to clusters detected by the method.

In our implementation we compute the links to maximal neighbourhood bins in a single sweep. In the same pass we insert indices of all local maxima in a list. We sort the list according to the bin count. Next, bins belonging to the cluster associated with the local maximum currently at the top of the list are obtained by traversing the tree rooted at the local maximum. Note that this implementation first outputs the cluster with a mode in the global maximum, other salient clusters follow. The algorithm therefore does not need to segment the feature space completely. The process can be terminated if eg. a cluster with interesting properties was found or a sufficiently large number of pixels have been analysed. The process can also be easily tuned to serve a top-down process by eg. considering maxima only from a selected part of the feature space, corresponding to a specific object.

Algorithm 2: **Clustering in Feature and Spatial Domains**

1. Perform graph-theoretical clustering.
2. Terminate if:
   (a) No local maximum exists.
   (b) Global maximum is bellow a threshold or the required number of regions is found.

   else select the current global maximum.
3. Traverse the tree rooted at current maximum to obtain a list of bins that belong to a unimodal cluster (identical to GT-clustering)
4. Check local (ie. connected component) consistency.
   (a) Backproject all pixels voting into bins of the unimodal cluster in the image. Perform connected componet analysis.
   (b) Sort the list of connected components by size.
   (c) Histogram computed on the largest connected component forms the initial model of the cluster distribution (in feature space).
   (d) Iterating through the sorted list, statistically test whether the current model of cluster distribution is consistent with the distribution computed on the next connected component. Votes from accepted components are removed from the global histogram.
5. Iteratively add pixels that are *both* in spatial and feature space neighbourhoods of the cluster.
6. Update the list of local maxima and go to step 2

The complexity of computing connected components is near linear [18]. The histogram comparison is carried out by a modified method described in [14] or by histogram intersection [19].

3 Test Cases

This section presents five 'test cases', ie. synthetically generated examples of simple yet common circumstances where feature space clustering alone cannot in principle result in error-free segmentation. This is due to the fact that the class distributions are chosen so that they overlap in the feature space.

For simplicity the problems are presented in the context of intensity image segmentation where dimensionality of the feature space is 1 and the data are taken from a 2D domain. The images contain rectangular patches with pixel values drawn from different Gaussian distributions. As discussed in detail bellow, the 'cases' differ in the character of the global histogram, number of distributions involved and adjacency relation of the rectangular patches. The distributions are chosen to highlight a particular limitation of the purely feature-based clustering methods like Modesp [8] or the Graph-theoretical (GT) clustering [9] [4]. In particular we shall compare the results of the proposed FSD method with GT clustering, but the conclusions drawn, in general, are expected to hold for a large class of feature space clustering algorithms.

Problem 1. Two non-adjacent regions A and B on a black background are shown in Figure 2 (a). The two overlapping clusters shown in Figure 3 (a) form a single unimodal cluster in feature space. The situation arises eg. when the clusters are normally distributed and $\mu_1 - \mu_2 \leq \sigma_1 + \sigma_2$ In this case, GT-clustering detects a single cluster (see Figure 2 (b)), regardless of the fact that each region has significantly different statictical properties. This can be shown either by a Chi-square test on the feature (ie. grey-level) distributions of the regions or simply by comparing interval estimates of the means and standard deviations of the two distributions. The distributions used in the example were $N(100, 10)$ and $N(120, 15)$. In Step 4.d, FSD clustering rejects the hypothesis that the clusters from the two regions are identical and produce the correct segments (see Figure 2 (c)). Note that in the case of the GT-clustering method, a change in the level of noise could qualitatively affect the segmentation result. The FSD method is considerably more stable. Using the spatial context, it resolves the single modal histogram into two mixture components (Figure 3 (b)) characterising the respective statistical properties of the two regions.

Problem 2. Two non-adjacent regions A and B on a black background are shown in Figure 2 (d). The regions form a histogram with two overlapping modes as shown in Figure 3 (c). The distributions used were $N(100, 10)$ and $N(140, 10)$. GT-clustering segments the image data by partitioning the histogram at the valley point between the two modes. This results in pixel misclassification (oversegmentation) illustrated in Figure 2 (e). In the case of the FSD method, in Step 4.d small regions are rejected on the grounds that their pixels are not from different distributions. In Step 5 the small regions are then incorporated into a neigbouring larger region on the grounds of spatial adjacency to deliver an error free segmentation of the image as shown in Figure 2 (f).

Problem 3. Figure 2 (g) shows two adjacent regions A and B generated according to the same distributions as in Problem 2. A third region C distributed as $N(120, 10)$, not adjacent to either A or B, overlaps in the feature space heavily both with A and B. Globally, the feature histogram is unimodal, as apparent from Figure 3 (e). As in Problem 1 the GT-clustering throws all pixel data into the same foreground cluster (see Figure 2 (h)). The proposed method resolves the data into three separate clusters

in agreement with human perception. This is depicted in Figures 2 (i) and 3 (f). The example shows how the GT result depends on non-local context.

Problem 4. Identical to Problem 2, but the two regions with the overlapping histograms are adjacent. It is interesting to observe the behaviour of the segmentation methods on the boundary shown in Figures 2 (k) and (l). In particular, the position of the boundary produced by the FSD method can be slightly biased because of the inherent ambiguity of region membership of pixels which are adjacent both spatially and in the feature space. The FSD algorithm will attempt to grow a region as far as possible. The bias will reflect the order in which regions are processed.

Problem 5. Identical to Problem 2, but a number of small regions with distribution shown in Figure 3 (g) are added to the image (see Figure 2 (m)). This simulates the effect of textured region. Here we illustrate the case when postprocessing by morphological filtering would destroy useful image content. Without the filtering, the GT-clustering method produces a noisy segmentation as shown in Figure 2 (n). In contrast, the FSD method treats differently the pixels with feature values lying in the overlapping tails of the cluster distributions and those exhibiting distinct feature space identity, regardless of the size of the region they form as can be seen in Figure 2 (o).

4 Experiments

All the experiments were carried out using the same program where only the voting procedure (approximately five to ten lines of code) was application specific. The section of the code for the FSD clustering computation which is about three orders of magnitude larger remained unchanged.

Segmentation of a gray level image. A very noisy synthetic image of Figure 4 (a) referred to as "stamp" which has been used as a benchmark in edge detection studies [13], has been used to demonstrate the properties of the FDS segmentation scheme. For comparison with the segmentation approach based on edge detection the image was first filtered with an optimal filter [13], [12] used in the Petrou-Kittler edge detector. The result obtained with the edge detector is shown in Figure 4 (e). The filtered image Figure 4 (b) is the input to the FSD and GT clustering algorithms which produce outputs shown in Figures 4 (c) and (d) respectively. Note the noisy appearance of the GT result. On the other hand, the FSD method's biased treatment of larger regions explains why the outer circular ring is about one pixel wider than its true size. From the FSD segmentation output the image edges can be obtained by extracting the region boundaries as shown in Figure 4 (f). When compared to the edge detector output we note the relative quality of the FSD performance. The edge map in Figure 4 (f) is less noisy and the boundaries are closed which may help the subsequent analysis. The edge detector output produces less noisy edge chains along strait lines. However, considering that in contrast to the edge detection method the FSD approach has no inbuilt knowledge of shape, the clustering scheme works remarkably well. In the histogram terms, the segmentation result of the two methods can be seen in Figure 4 (h) for the GT algorithm and in Figure 4 (i) for the FSD algorithm, with Figure 4 (g) showing the image histogram.

Range Data The approach was tested on depth images from the NRCC database [17]. The NRCC database contains dense image data with depth available at every

pixel location. A sample image is shown in Figure 5 (a). We estimated the depth image derivatives $\frac{dz}{dx}$ and $\frac{dz}{dy}$ using the Sobel operator. Each pair of derivatives defines a normal which is superimposed on the depth image. Note that points of constant derivative values belong to a set of pixels lying on parallel planes. The clustering was performed in the $\frac{dz}{dx}$, $\frac{dz}{dy}$ space. The resulting segmentation for the FSD and GT clustering methods is shown in Figures 5 (b) and (c) respectively. At some pixel location the depth measurements are not available. At such points estimates of the derivatives are randomly spread over the feature space and consequently the pixels are not assigned to any cluster (see the white points in Figures 5 (b) and (c).

Optic Flow We tested the suitability of the technique for motion segmentation on a pair of images 6 (a) taken from the 'Lab' sequence of images analysed in the thesis of Bober [2], page 53. The image contains a stationary background and two moving objects undergoing distinct translatory motion. The optic flow field in Figure 6 (b) was computed using a robust Hough Transform technique described in [3], with the following parameters: block size 16x16, final reslolution 0.01 pixel. The sequence was preprocessed by Gaussian filtering, sigma 3 pixels. The results of the FSD and GT clustering methods are shown in Figures 6 (c) and (d) respectively. It can be seen that both moving objects are segmented out. The differences are insignificant as the clusters corresponding to the two motions are not overlapping. In both cases the segmentation is noisy because the regions contain flow vectors which are distribution outliers. Note that these errors are the artifact of the optic flow computation rather than produced by the clustering procedures. It should also be pointed out that for nontranslatory motion, a more general motion model would have to be assumed to achieve feature space and spatial domain coherence to allow satisfactory segmentation.

Line Detection by Hough Transform In order to demonstrate a completely different application of the FSD clustering method we consider the problem of detecting straight lines in the image shown in Figure 7 (a). Edge position and direction is mapped into the ρ - θ line parameter space. The connected component analysis (Step 3.a) in this case effectively compares the properties of the individual line segments associated with the infinite straight line defined by the mode of each cluster. Note that at this stage the lines that are grouped together in a single unimodal cluster will be resolved into a set of contributing lines with distinct parameters. In this sense the proposed approach does not have the tendency to merge several lines of slightly different parameters into a long ghost line with incorrect 'averaged' parameters. The FSD clustering output is shown in Figure 7 (c) for comparison with the robust Hough transform output [15] in Figure 7 (b). There are no significant differences in performance. The slight oversegmentation produced by the Hough transform method could be reduced by threshold modification. Also note that the FSD method outputs edge chains, rather than line models. Different clusters of edge pixels are denoted by different greylevels. The main aim here is to demonstrate the versaltility of the FSD method.

Segmentation of Colour Images The RGB values of the image in Figure 8 (a) were projected into the chromatic plane to achieve illumination invariant segmentation results [11]. The underlying assumptions for successful segmentation are diffused reflection, and the ability to treat specularities and interreflections as noise as discussed in [10]. Here the FSD scheme performs significantly better (see Figure 8 (c)) than GT clustering

(Figure 8 (b)), specially in the dark and/or specular regions where clusters corresponding to different reflectances overlap. They are disambiguated by the FSD procedure by considering the spatial context.

5 Conclusions

We have proposed a novel approach to image segmentation, called *feature and spatial domain* clustering. The method is devised to group pixel data by taking into account simultaneously both their feature space similarity and spatial coherence. The FSD algorithm is practically application independent. It has been successfully tested on a wide range of image segmentation problems, including grey and colour image segmentation, edge and line detection, range data and motion segmentation. In comparison with existing segmentation approaches, the method can resolve image features even if their distributions significantly overlap in the feature space. It can distinguish between noisy regions and genuine fine texture. Moreover, if required, FSD clustering can produce partial segmentation by identifying salient regions only.

Acknowledgements

The work was carried under ESPRIT project 7108 *Vision as process*. The experiments on range data were carried out with help from Andrew Stoddart and Adrian Hilton. We thank Miroslaw Bober for providing the input data for the optical flow field segmentation and to Maria Petrou for the stamp image.

References

1. D.H. Ballard. Parameter nets. *Artificial Intelligence*, pages 235–267, 1984.
2. M. Bober. *General Motion Estimation and Segmentation form Image Sequences*. PhD thesis, VSSP Group, University of Surrey, United Kingdom, 1995.
3. M. Bober and J. Kittler. Robust motion analysis. In *CVPR94*, pages 947–952, 1994.
4. K. Fukunaga. *Introduction to Statististical Pattern Recognition*. Academic Press, 1990.
5. R.M. Haralick and Shapiro. *Computer and Robot Vision - Volume II*. Addison-Wesley, 1993.
6. Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
7. A. Khotanzad and A. Bouarfa. Image segmantation by a parallel , non-parametric histogram based clustering algorithm. *Pattern Recognition*, pages 961–973, 1990.
8. J. Kittler. A locally sensitive method for cluster analysis. *Pattern Recognition*, 8:23–33, 1976.
9. W.L.G. Koontz, P. M. Narendra, and K. Fukunaga. A graph-theoretic approach to nonparametric clustring. *IEEE Trans. Computers*, C-25:936–944, 1976.
10. J. Matas, R. Marik, and J. Kittler. Illumination invariant colour recognition. In E. Hancock, editor, *British Machine Vision Conference*. BMVA Press, 1994.
11. J. Matas, R. Marik, and J. Kittler. On representation and matching of multi-coloured objects. In *Fifth International Conference on Computer Vision (Cambridge, MA, June 20–23, 1995)*, pages 726–732, Washington, DC., 1995. Computer Society Press.
12. M. Petrou. The differentiating filter approach to edge detection. In *Advances in Electronics and Electron Physics*, volume 88, pages 297–345. Academic Press, 1994.

13. M. Petrou and J. Kittler. Optimal edge detectors for ramp edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, May 1991.
14. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
15. J. Princen. Hough transform methods for curve detection and parameter estimation. Technical report, University of Surrey, 90.
16. John A. Richards. *Remote Sensing Digital Image Analysis*. Springer, 1993.
17. M. Rioux and L. Cournmoyer. *The NRCC Three-dimensional Image Data Files*. National Research Council Canada, Ottawa, Ontario, Canada, 1988.
18. R. Sedgewick. *Algorithms*. Addison-Wesley, 2nd edition, 1988.
19. M.J. Swain. *Color Indexing*. PhD thesis, University of Rochester, 1990.

This article was processed using the LaTeX macro package with LLNCS style

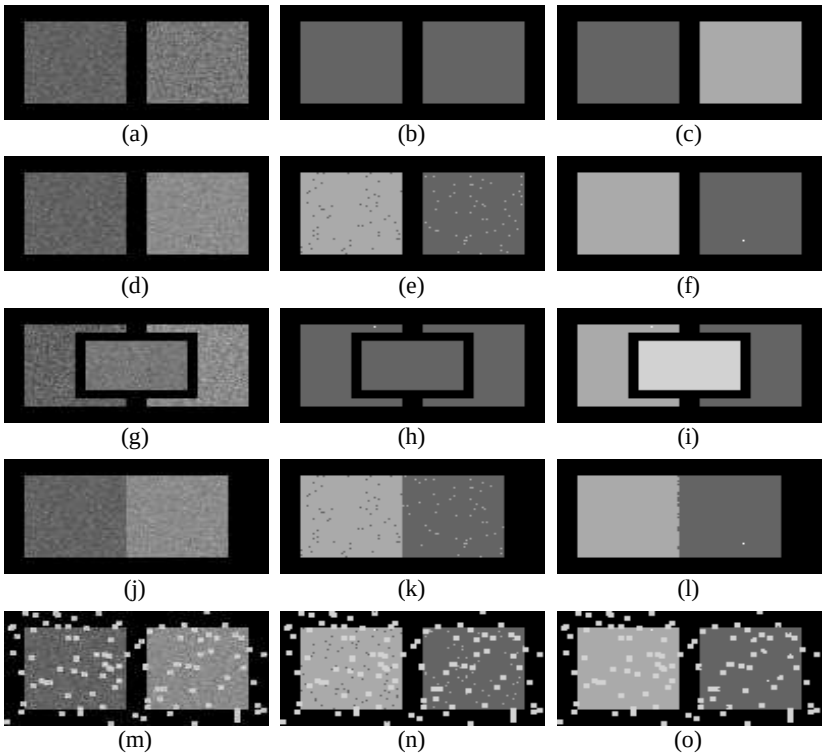

Fig. 2.: Test case images.

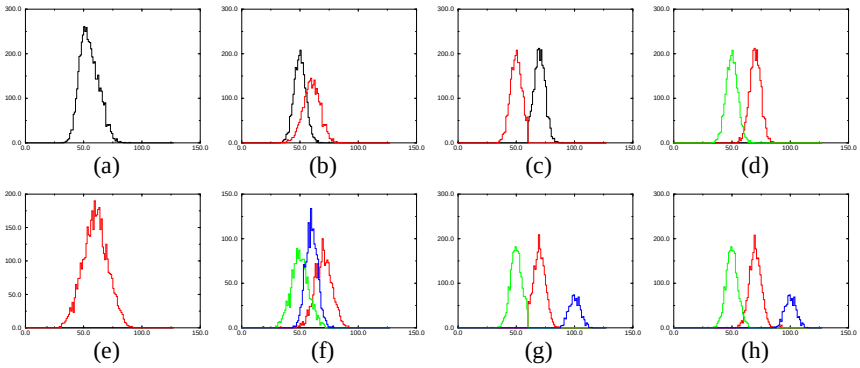

Fig. 3.: Test case image histograms.

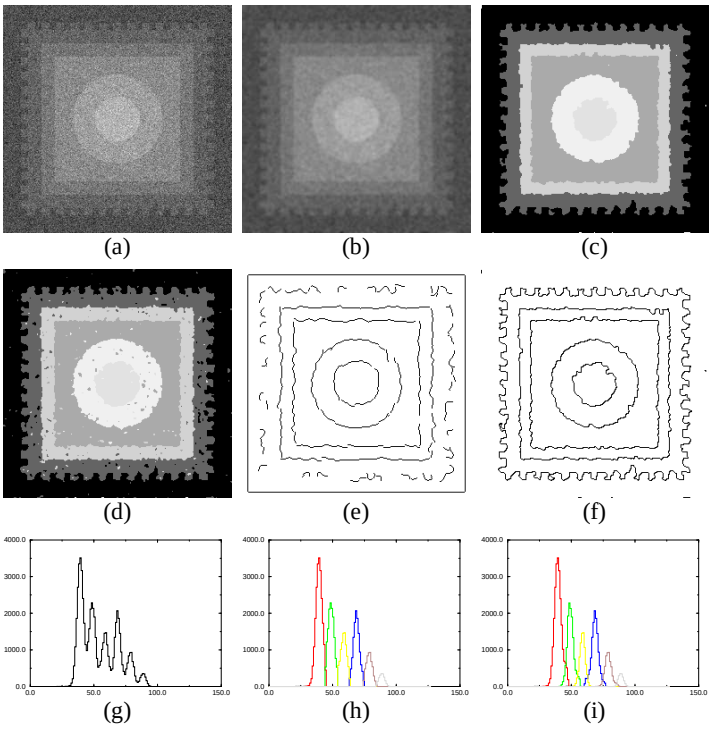

Fig. 4.: Stamp (from [12]).

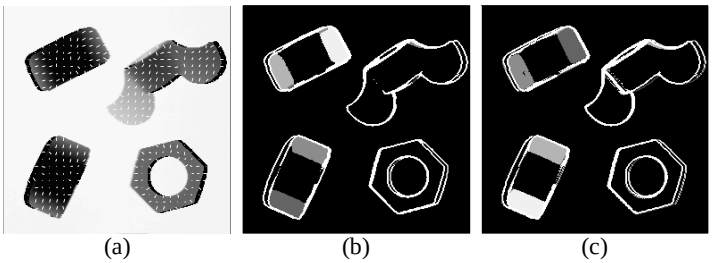

Fig. 5.: Bolts. (a) Original image. (b) FSD clustering. (c) GT clustering.

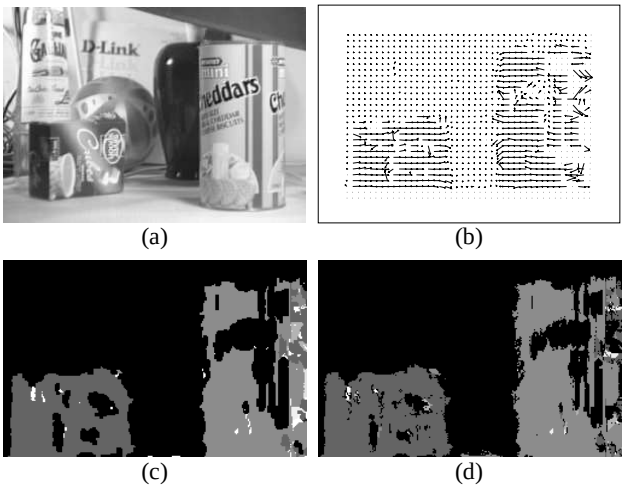

Fig. 6.: Lab sequence.

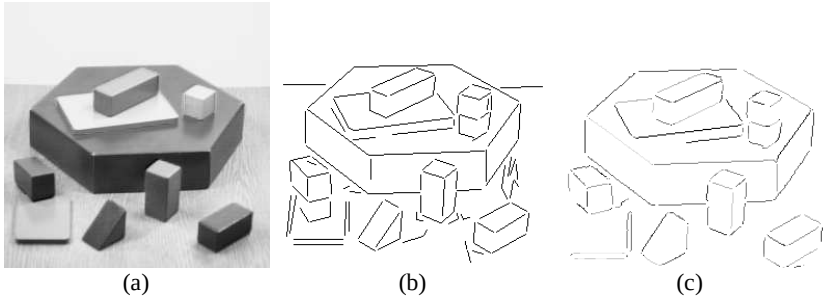

Fig. 7.: A polyhedral scene. (a) Original image. (b) Lines detected by Hough Transform (c) Results of clustering

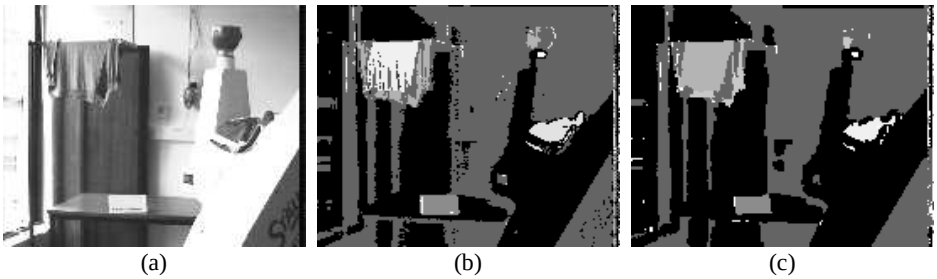

Fig. 8.: Colour. (a) Original image. (b) GT clustering. (c) FSD clustering.