Robust Analysis of Feature Spaces: Color Image Segmentation

Dorin Comaniciu Peter Meer
Department of Electrical and Computer Engineering
Rutgers University, Piscataway, NJ 08855, USA

Keywords: *robust pattern analysis, low-level vision, content-based indexing*

Abstract

A general technique for the recovery of significant image features is presented. The technique is based on the mean shift algorithm, a simple nonparametric procedure for estimating density gradients. Drawbacks of the current methods (including robust clustering) are avoided. Feature space of any nature can be processed, and as an example, color image segmentation is discussed. The segmentation is completely autonomous, only its class is chosen by the user. Thus, the same program can produce a high quality edge image, or provide, by extracting all the significant colors, a preprocessor for content-based query systems. A 512×512 *color image is analyzed in less than 10 seconds on a standard workstation. Gray level images are handled as color images having only the lightness coordinate.*

1 Introduction

Feature space analysis is a widely used tool for solving low-level image understanding tasks. Given an image, feature vectors are extracted from local neighborhoods and mapped into the space spanned by their components. Significant features in the image then correspond to high density regions in this space. Feature space analysis is the procedure of recovering the centers of the high density regions, i.e., the representations of the significant image features. Histogram based techniques, Hough transform are examples of the approach.

When the number of distinct feature vectors is large, the size of the feature space is reduced by grouping nearby vectors into a single cell. A discretized feature space is called an accumulator. Whenever the size of the accumulator cell is not adequate for the data, serious artifacts can appear. The problem was extensively studied in the context of the Hough transform, e.g. [5]. Thus, for satisfactory results *a feature space should have continuous coordinate system.* The content of a continuous feature space can be modeled as a sample from a multivariate, multimodal probability distribution. Note that for real images the number of modes can be very large, of the order of tens.

The highest density regions correspond to clusters centered on the modes of the underlying probability distribution. Traditional clustering techniques [6], can be used for feature space analysis but they are reliable only if the number of clusters is small *and* known a priori. Estimating the number of clusters from the data is computationally expensive and not guaranteed to produce satisfactory result.

A much too often used assumption is that the individual clusters obey multivariate normal distributions, i.e., the feature space can be modeled as a mixture of Gaussians. The parameters of the mixture are then estimated by minimizing an error criterion. For example, a large class of thresholding algorithms are based on the Gaussian mixture model of the histogram, e.g. [11]. However, there is no theoretical evidence that an extracted normal cluster necessarily corresponds to a significant image feature. On the contrary, a strong artifact cluster may appear when several features are mapped into partially overlapping regions.

Nonparametric density estimation [4, Chap. 6] avoids the use of the normality assumption. The two families of methods, Parzen window, and k-nearest neighbors, both require additional input information (type of the kernel, number of neighbors). This information must be provided by the user, and for multimodal distributions it is difficult to guess the optimal setting.

Nevertheless, a reliable general technique for feature space analysis can be developed using a simple nonparametric density estimation algorithm. In this paper we propose such a technique whose robust behavior is superior to methods employing robust estimators from statistics.

2 Requirements for Robustness

Estimation of a cluster center is called in statistics the multivariate location problem. To be robust, an estimator must tolerate a percentage of outliers, i.e., data points not obeying the underlying distribution

of the cluster. Numerous robust techniques were proposed [10, Sec. 7.1], and in computer vision the most widely used is the *minimum volume ellipsoid* (MVE) estimator proposed by Rousseeuw [10, p. 258].

The MVE estimator is affine equivariant (an affine transformation of the input is passed on to the estimate) and has high breakdown point (tolerates up to half the data being outliers). The estimator finds the center of the highest density region by searching for the minimal volume ellipsoid containing at least h data points. The multivariate location estimate is the center of this ellipsoid. To avoid combinatorial explosion a probabilistic search is employed. Let the dimension of the data be p. A small number of $(p+1)$-tuple of points are randomly chosen. For each $(p+1)$-tuple the mean vector and covariance matrix are computed, defining an ellipsoid. The ellipsoid is inflated to include h points, and the one having the minimum volume provides the MVE estimate.

Based on MVE, a robust clustering technique with applications in computer vision was proposed in [7]. The data is analyzed under several "resolutions" by applying the MVE estimator repeatedly with h values representing fixed percentages of the data points. The best cluster then corresponds to the h value yielding the highest density inside the minimum volume ellipsoid. The cluster is removed from the feature space, and the whole procedure is repeated till the space is not empty. The robustness of MVE should ensure that each cluster is associated with only one mode of the underlying distribution. The number of significant clusters is not needed a priori.

The robust clustering method was successfully employed for the analysis of a large variety of feature spaces, but was found to become less reliable once the number of modes exceeded ten. This is mainly due to the normality assumption embedded into the method. The ellipsoid defining a cluster can be also viewed as the high confidence region of a multivariate normal distribution. Arbitrary feature spaces are not mixtures of Gaussians and constraining the shape of the removed clusters to be elliptical can introduce serious artifacts. The effect of these artifacts propagates as more and more clusters are removed. Furthermore, the estimated covariance matrices are not reliable since are based on only $p+1$ points. Subsequent postprocessing based on all the points declared inliers cannot fully compensate for an initial error.

To be able to correctly recover a large number of significant features, the problem of feature space analysis must be solved in context. In image understanding tasks the data to be analyzed originates in the *image domain*. That is, the feature vectors satisfy additional, spatial constraints. While these constraints are indeed used in the current techniques, their role is mostly limited to compensating for feature allocation errors made during the *independent* analysis of the feature space. To be robust *the feature space analysis must fully exploit the image domain information.*

As a consequence of the increased role of image domain information the burden on the feature space analysis can be reduced. First *all* the significant features are extracted, and *only after then* are the clusters containing the instances of these features recovered. The latter procedure uses image domain information and avoids the normality assumption.

Significant features correspond to high density regions and to locate these regions a search window must be employed. The number of parameters defining the shape and size of the window should be minimal, and therefore whenever it is possible *the feature space should be isotropic.* A space is isotropic if the distance between two points is independent on the location of the point pair. The most widely used isotropic space is the Euclidean space, where a sphere, having only one parameter (its radius) can be employed as search window. The isotropy requirement determines the mapping from the image domain to the feature space. If the isotropy condition cannot be satisfied, a Mahalanobis metric should be defined from the statement of the task.

We conclude that robust feature space analysis requires a reliable procedure for the detection of high density regions. Such a procedure is presented in the next section.

3 Mean Shift Algorithm

A simple, nonparametric technique for estimation of the density gradient was proposed in 1975 by Fukunaga and Hostetler [4, p. 534]. The idea was recently generalized by Cheng [2].

Assume, for the moment, that the probability density function $p(\mathbf{x})$ of the p-dimensional feature vectors $\mathbf{x}$ is unimodal. This condition is for sake of clarity only, later will be removed. A sphere $\mathcal{S}_{\mathbf{x}}$ of radius r, centered on $\mathbf{x}$ contains the feature vectors $\mathbf{y}$ such that $\| \mathbf{y}-\mathbf{x} \| \leq r$. The expected value of the vector $\mathbf{z} = \mathbf{y}-\mathbf{x}$, given $\mathbf{x}$ and $\mathcal{S}_{\mathbf{x}}$ is

$$\begin{aligned} \mu = E[\,\mathbf{z}|\mathcal{S}_{\mathbf{x}}\,] &= \int_{\mathcal{S}_{\mathbf{x}}} (\mathbf{y}-\mathbf{x}) p(\mathbf{y}|\mathcal{S}_{\mathbf{x}}) d\mathbf{y} \qquad (1) \\ &= \int_{\mathcal{S}_{\mathbf{x}}} (\mathbf{y}-\mathbf{x}) \frac{p(\mathbf{y})}{p(\mathbf{y} \in \mathcal{S}_{\mathbf{x}})} d\mathbf{y} \end{aligned}$$

If $\mathcal{S}_{\mathbf{X}}$ is sufficiently small we can approximate

$$p(\mathbf{y} \in \mathcal{S}_{\mathbf{X}}) = p(\mathbf{x}) V_{\mathcal{S}_{\mathbf{X}}} \qquad \text{where} \quad V_{\mathcal{S}_{\mathbf{X}}} = c \cdot r^p \tag{2}$$

is the volume of the sphere. The first order approximation of $p(\mathbf{y})$ is

$$p(\mathbf{y}) = p(\mathbf{x}) + (\mathbf{y} - \mathbf{x})^T \nabla p(\mathbf{x}) \tag{3}$$

where $\nabla p(\mathbf{x})$ is the gradient of the probability density function in $\mathbf{x}$. Then

$$\mu = \int_{\mathcal{S}_{\mathbf{X}}} \frac{(\mathbf{y} - \mathbf{x})(\mathbf{y} - \mathbf{x})^T}{V_{\mathcal{S}_{\mathbf{X}}}} \frac{\nabla p(\mathbf{x})}{p(\mathbf{x})} d\mathbf{y} \tag{4}$$

since the first term term vanishes. The value of the integral is [4, p. 535]

$$\mu = \frac{r^2}{p+2} \frac{\nabla p(\mathbf{x})}{p(\mathbf{x})} \tag{5}$$

or

$$E[\,\mathbf{x} \mid \mathbf{x} \in \mathcal{S}_{\mathbf{X}}\,] - \mathbf{x} = \frac{r^2}{p+2} \frac{\nabla p(\mathbf{x})}{p(\mathbf{x})} \tag{6}$$

Thus, the mean shift vector, the vector of difference between the local mean and the center of the window, is proportional to the gradient of the probability density at $\mathbf{x}$. The proportionality factor is reciprocal to $p(\mathbf{x})$. This is beneficial when the highest density region of the probability density function is sought. Such region corresponds to large $p(\mathbf{x})$ and small $\nabla p(\mathbf{x})$, i.e., to small mean shifts. On the other hand, low density regions correspond to large mean shifts (amplified also by small $p(\mathbf{x})$ values). The shifts are always in the direction of the probability density maximum, the mode. At the mode the mean shift is close to zero. This property can be exploited in a simple, adaptive steepest ascent algorithm.

Mean Shift Algorithm

1. Choose the radius r of the search window.
2. Choose the initial location of the window.
3. Compute the mean shift vector and translate the search window by that amount.
4. Repeat till convergence.

To illustrate the ability of the mean shift algorithm, 200 data points were generated from two normal distributions, both having unit variance. The first hundred points belonged to a zero-mean distribution, the second hundred to a distribution having mean 3.5. The data is shown as a histogram in Figure 1. It should be emphasized that the feature space is processed as an ordered one-dimensional sequence of points, i.e., it is continuous. The mean shift algorithm starts from the location of the mode detected by the one-dimensional MVE mode detector, i.e., the center of the shortest rectangular window containing half the data points [10, Sec. 4.2]. Since the data is bimodal with nearby modes, the mode estimator fails and returns a location in the trough. The starting point is marked by the cross at the top of Figure 1.

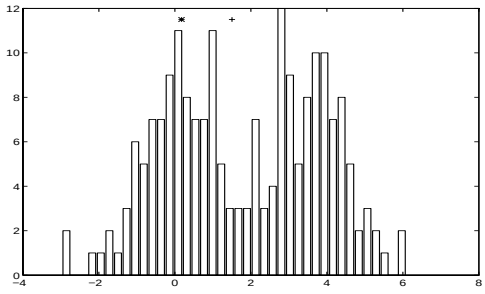


Figure 1: An example of the mean shift algorithm.

In this synthetic data example no a priori information is available about the analysis window. Its size was taken equal to that returned by the MVE estimator, 3.2828. Other, more adaptive strategies for setting the search window size can also be defined.

Table 1: *Evolution of Mean Shift Algorithm*

Initial Mode	Initial Mean	Final Mean
1.5024	1.4149	0.1741

In Table 1 the initial values and the final location, shown with a star at the top of Figure 1, are given.

The mean shift algorithm is the tool needed for feature space analysis. The unimodality condition can be relaxed by randomly choosing the initial location of the search window. The algorithm then converges to the closest high density region. The outline of a general procedure is given below.

Feature Space Analysis

1. Map the image domain into the feature space.
2. Define an adequate number of search windows at random locations in the space.
3. Find the high density region centers by applying the mean shift algorithm to each window.
4. Validate the extracted centers with image domain constraints to provide the *feature palette*.
5. Allocate, using image domain information, all the feature vectors to the feature palette.

The procedure is very general and applicable to any feature space. In the next section we describe a color image segmentation technique developed based on this outline.

4 Color Image Segmentation

Image segmentation, partioning the image into homogeneous regions, is a challenging task. The richness of visual information makes bottom-up, solely image driven approaches always prone to errors. To be reliable, the current systems must be large and incorporate numerous ad-hoc procedures, e.g. [1]. The paradigms of gray level image segmentation (pixel-based, area-based, edge-based) are also used for color images. In addition, the physics-based methods take into account information about the image formation processes as well. See, for example, the reviews [8, 12]. The proposed segmentation technique does not consider the physical processes, it uses only the given image, i.e., a set of RGB vectors. Nevertheless, can be easily extended to incorporate supplementary information about the input. As homogeneity criterion color similarity is used.

Since perfect segmentation cannot be achieved without a top-down, knowledge driven component, a bottom-up segmentation technique should

- only provide the input into the next stage where the task is accomplished using a priori knowledge about its goal; and
- eliminate, as much as possible, the dependence on user set parameter values.

Segmentation resolution is the most general parameter characterizing a segmentation technique. While this parameter has a continuous scale, three important classes can be distinguished.

Undersegmentation corresponds to the lowest resolution. Homogeneity is defined with a large tolerance margin and only the most significant colors are retained for the feature palette. The region boundaries in a correctly undersegmented image are the dominant edges in the image.

Oversegmentation corresponds to intermediate resolution. The feature palette is rich enough that the image is broken into many small regions from which any sought information can be assembled under knowledge control. Oversegmentation is the recommended class when the goal of the task is object recognition.

Quantization corresponds to the highest resolution. The feature palette contains all the important colors in the image. This segmentation class became important with the spread of image databases, e.g., [3, 9]. The full palette, possibly together with the underlying spatial structure, is essential for content-based queries.

The proposed color segmentation technique operates in any of the these three classes. The user only chooses the desired class, the specific operating conditions are derived automatically by the program.

Images are usually stored and displayed in the RGB space. However, to ensure the isotropy of the feature space, a uniform color space with the perceived color differences measured by Euclidean distances should be used. We have chosen the $L^*u^*v^*$ space [13, Sec. 3.3.9], whose coordinates are related to the RGB values by nonlinear transformations. The daylight standard D_{65} was used as reference illuminant. The chromatic information is carried by u^* and v^*, while the lightness coordinate L^* can be regarded as the relative brightness. Psychophysical experiments show that $L^*u^*v^*$ space may not be perfectly isotropic [13, p. 311], however, it was found satisfactory for image understanding applications. The image capture/display operations also introduce deviations which are most often neglected.

The steps of color image segmentation are presented below. The acronyms **ID** and **FS** stand for image domain and feature space respectively. All feature space computations are performed in the $L^*u^*v^*$ space.

1. [**FS**] *Definition of the segmentation parameters.* The user only indicates the desired class of segmentation. The class definition is translated into three parameters

   - the radius of the search window, r;
   - the smallest number of elements required for a significant color, N_{min};
   - the smallest number of contiguous pixels required for a significant image region, N_{con}.

The size of the search window determines the resolution of the segmentation, smaller values corresponding to higher resolutions. The subjective (perceptual) definition of a homogeneous region seems to depend on the "visual activity" in the image. Within the same segmentation class an image containing large homogeneous regions should be analyzed at higher resolution than an image with many textured areas. The simplest measure of the "visual activity" can be derived from the global covariance matrix. The square root of its trace, σ, is related to the power of the signal (image). The radius r is taken proportional to σ. The rules defining the three segmentation class parameters are given in Table 2. These rules were used in the segmentation of a large variety images, ranging from simple blood cells to complex indoor and outdoor scenes.

When the goal of the task is well defined and/or all the images are of the same type, the parameters can be fine tuned.

Table 2: *Segmentation Class Parameters*

Segmentation Class	Parameter		
	r	N_{min}	N_{con}
Undersegmentation	0.4σ	400	10
Oversegmentation	0.3σ	100	10
Quantization	0.2σ	50	0

2. [**ID+FS**] *Definition of the search window.*
The initial location of the search window in the feature space is randomly chosen. To ensure that the search starts close to a high density region several location candidates are examined. The random sampling is performed in the image domain and a few, $M = 25$, pixels are chosen. For each pixel, the mean of its 3×3 neighborhood is computed and mapped into the feature space. If the neighborhood belongs to a larger homogeneous region, with high probability the location of the search window will be as wanted. To further increase this probability, the window containing the highest density of feature vectors is selected from the M candidates.

3. [**FS**] *Mean shift algorithm.*
To locate the closest mode the mean shift algorithm is applied to the selected search window. Convergence is declared when the magnitude of the shift becomes less than 0.1.

4. [**ID+FS**] *Removal of the detected feature.*
The pixels yielding feature vectors inside the search window at its final location are discarded from both domains. Additionally, their 8-connected neighbors in the image domain are also removed *independent* of the feature vector value. These neighbors can have "strange" colors due to the image formation process and their removal cleans the background of the feature space. Since all pixels are reallocated in Step 7, possible errors will be corrected.

5. [**ID+FS**] *Iterations.*
Repeat Steps 2 to 4, till the number of feature vectors in the selected search window no longer exceeds N_{min}.

6. [**ID**] *Determining the initial feature palette.*
In the feature space a significant color must be based on minimum N_{min} vectors. Similarly, to declare a color significant in the image domain more than N_{min} pixels of that color should belong to a connected component. From the extracted colors only those are retained for the initial feature palette which yield at least one connected component in the image of size larger than N_{min}. The neighbors removed at Step 4. are also considered when defining the connected components Note that the threshold is not N_{con} which is used only at the postprocessing stage.

7. [**ID+FS**] *Determining the final feature palette.*
The initial feature palette provides the colors allowed when segmenting the image. If the palette is not rich enough the segmentation resolution was not chosen correctly and should be increased to the next class. All the pixel are reallocated based on this palette. First, the pixels yielding feature vectors inside the search windows at their final location are considered. These pixels are allocated to the color of the window center without taking into account image domain information. The windows are then inflated to double volume (their radius is multiplied with $\sqrt[3]{2}$). The newly incorporated pixels are retained only if they have at least one neighbor which was already allocated to that color. The mean of the feature vectors mapped into the same color is the value retained for the final palette. At the end of the allocation procedure a small number of pixels can remain unclassified. These pixels are allocated to the closest color in the final feature palette.

8. [**ID+FS**] *Postprocessing.*
This step depends on the goal of the task. The simplest procedure is the removal from the image of all small connected components of size less than N_{con}. These pixels are allocated to the majority color in their 3×3 neighborhood, or in the case of a tie to the closest color in the feature space.

In Figure 2 the *house* image containing 9603 different colors is shown. The segmentation results for the three classes and the region boundaries are given in Figure 5a–f. Note that undersegmentation yields a good edge map, while in the quantization class the original image is closely reproduced with only 37 colors. A second example using the oversegmentation class is shown in Figure 3. Note the details on the fuselage.

5 Discussion

The simplicity of the basic computational module, the mean shift algorithm, enables the feature space analysis to be accomplished very fast. From a 512×512 pixels image a palette of 10–20 features can be extracted in less than 10 seconds on a Ultra SPARC 1 workstation. To achieve such a speed the implementation was optimized and whenever possible, the feature space (containing fewer distinct elements than the image domain) was used for array scanning; lookup tables were employed instead of fre-

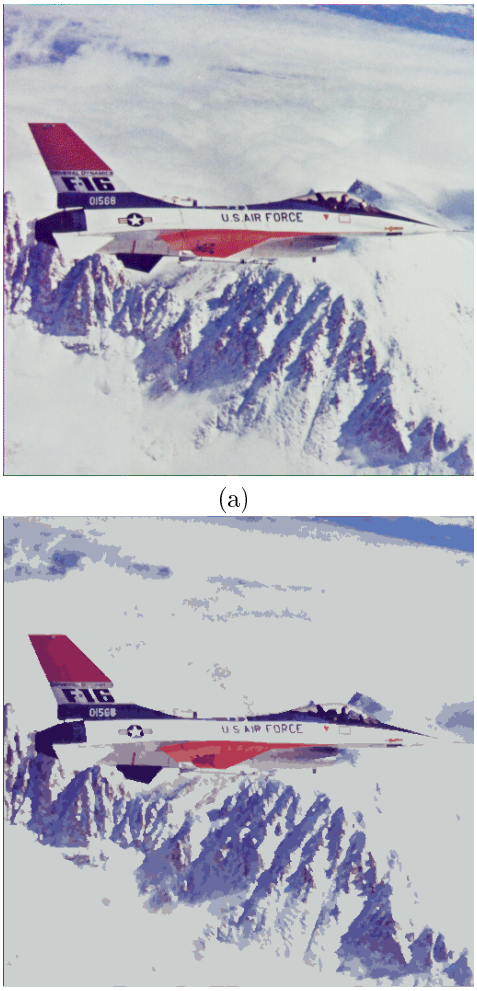
Figure 2: The *house* image, 255 × 192 pixels, 9603 colors.

quently repeated computations; direct addressing instead of nested pointers; fixed point arithmetic instead of floating point calculations; partial computation of the Euclidean distances, etc.

The analysis of the feature space is completely autonomous, due to the extensive use of image domain information. All the examples in this paper, and dozens more not shown here, were processed using the parameter values given in Table 2. Recently Zhu and Yuille [14] described a segmentation technique incorporating complex global optimization methods (snakes, minimum description length) with sensitive parameters and thresholds. To segment a color image over a hundred iterations were needed. When the images used in [14] were processed with the technique described in this paper, the same quality results were obtained unsupervised and in less than a second. Figure 4 shows one of the results, to be compared with Figure 14h in [14]. The new technique can be used *unmodified* for segmenting gray level images, which are handled as color images with only the L^* coordinates. In Figure 6 an example is shown.

The result of segmentation can be further refined by local processing in the image domain. For example, robust analysis of the pixels in a large connected component yields the inlier/outlier dichotomy which then can be used to recover discarded fine details.

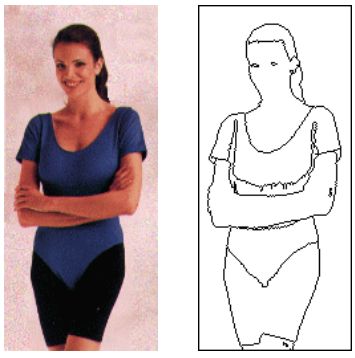
In conclusion, we have presented a general technique for feature space analysis with applications in many low-level vision tasks like thresholding, edge detection, segmentation. The nature of the feature space is not restricted, currently we are working on applying the technique to range image segmentation, Hough transform and optical flow decomposition.



(a)

(b)

Figure 3: Color image segmentation example. (a) Original image, 512 × 512 pixels, 77041 colors. (b) Oversegmentation: 21/21 colors.



(a) (b)

Figure 4: Performance comparison. (a) Original image, 116 × 261 pixels, 200 colors. (b) Undersegmentation: 5/4 colors. Region boundaries.

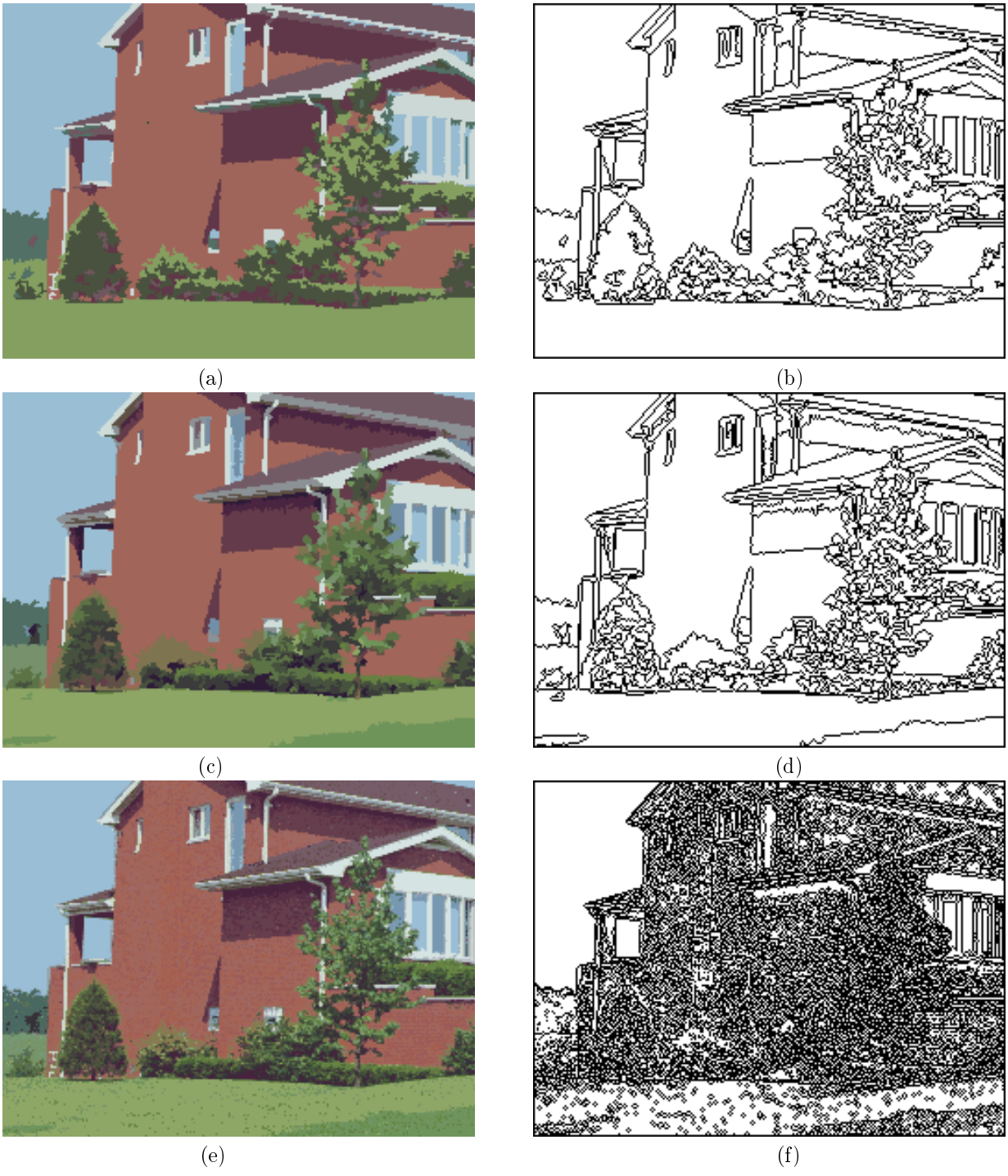


Figure 5: The three segmentation classes for the *house* image. The right column shows the region boundaries. (a)(b) Undersegmentation. Number of colors extracted initially and in the feature palette: 8/8. (c)(d) Oversegmentation: 24/19 colors. (e)(f) Quantization: 49/37 colors.

Acknowledgement

The research was supported by the National Science Foundation under the grant IRI-9530546.

References

[1] J.R. Beveridge, J. Griffith, R.R. Kohler, A.R. Hanson, E.M. Riseman, "Segmenting images using localized histograms and region merging", *Int'l. J. of Comp. Vis.*, vol. 2, 311–347, 1989.

[2] Y. Cheng, "Mean shift, mode seeking, and clustering", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, 790–799, 1995.

[3] M. Flickner et al., "Query by image and video content: The QBIC system", *Computer*, vol. 28, no. 9, 23–32, 1995.

[4] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Second Ed., Boston: Academic Press, 1990.

[5] J. Illingworth, J. Kittler, "A survey of the Hough transform", *Comp. Vis., Graph. and Imag. Proc.*, vol. 44, 87–116, 1988.

[6] A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data*, Englewood Cliff, NJ: Prentice Hall, 1988.

[7] J.-M. Jolion, P. Meer, S. Bataouche, "Robust clustering with applications in computer vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, 791–802, 1991.

[8] Q.T. Luong, "Color in computer vision", In *Handbook of Pattern Recognition and Computer Vision*, C.H. Chen, L.F. Pau, and P.S P. Wang (Eds.), Singapore: World Scientific, 311–368, 1993.

[9] A. Pentland, R.W. Picard, S. Sclaroff, "Photobook: Content-based manipulation of image databases", *Int'l. J. of Comp. Vis.* vol. 18, 233–254, 1996.

[10] P.J. Rousseeuw, A.M. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.

[11] P.K. Sahoo, S. Soltani, A.K.C. Wong, "A survey of thresholding techniques", *Comp. Vis., Graph. and Imag. Proc.*, vol. 41, 233–260, 1988.

[12] W. Skarbek, A. Koschan, *Colour Image Segmentation – A Survey*, Technical Report 94-32, Technical University Berlin, October 1994.

[13] G. Wyszecki, W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, Second Ed. New York: Wiley, 1982.

[14] S.C. Zhu, A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 18, 884–900, 1996.

(a) (b) (c)

Figure 6: Gray level image segmentation example. (a) Original image, 256×256 pixels. (b) Undersegmentation: 5 gray levels. (c) Region boundaries.