Chapter 5

Segmentation of newspaper images

5.1 Context and simplifications

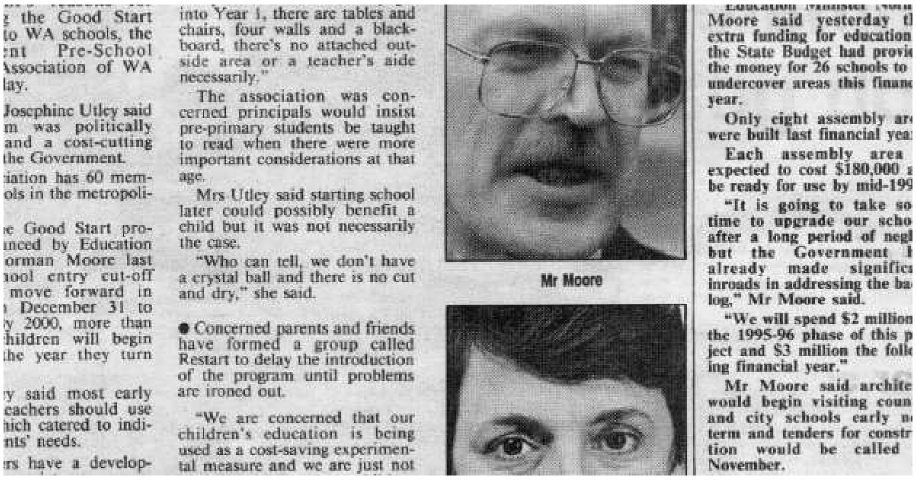
This problem deals with segmenting mixed-mode images into regions possibly corresponding to *background*, *pictures* and *text*. Mixed-mode segmentation is discussed in Section 2.5.7.

As with anomaly detection the restriction is made to grey scale images, however the complexity of the problem is increased. Images now contain multiple types of segments with no assumptions made pertaining to the number of objects in the image or the relative size of each object type. No assumption is made about the background colour, the relative contrast or the orientation of the image. Hence, white text on a black background, very dull or very bright images, and pages at an angle of 45 degrees are all within scope. Consequently invariances to intensity inversion, illumination, contrast and rotation are required. Insensitivity to noise and adaptability to image resolution are also desirable.

However, the limitation is made to images containing some amount of background and text regions. Without this restriction it is difficult to provide contrast invariance since there is no base level to use as a reference for background or text characteristics. Furthermore it is assumed that text has the greatest contrast at the appropriate resolution and the background has the least intensity variation. The final assumption is similar to assuming that text and background contain the highest and lowest frequency components respectively.

5.2 Example data and applications

Images scanned from newspapers and magazines are appropriate. The image in Figure 5.1 scanned from *The West Australian* is used throughout to illustrate different processing techniques. Notice that the image contains two different styles of font, a vertical line and the pictures contain features with a high contrast such as the frames of the glasses, the detailed eyes and the hair line. Important features not presented in this example include large headline fonts and pictures containing constant intensity regions. Appropriate images are presented along with results.



g the Good Start
to WA schools, the
ent Pre-School
Association of WA
lay.

Josephine Utley said
m was politically
and a cost-cutting
the Government.

iation has 60 mem-
ols in the metropoli-

e Good Start pro-
nced by Education
orman Moore last
hool entry cut-off
move forward in
December 31 to
y 2000, more than
hildren will begin
the year they turn

y said most early
eachers should use
hich catered to indi-
nts' needs.

rs have a develop-

into Year 1, there are tables and chairs, four walls and a black-board, there's no attached out-side area or a teacher's aide necessarily."

The association was concerned principals would insist pre-primary students be taught to read when there were more important considerations at that age.

Mrs Utley said starting school later could possibly benefit a child but it was not necessarily the case.

"Who can tell, we don't have a crystal ball and there is no cut and dry," she said.

● Concerned parents and friends have formed a group called Restart to delay the introduction of the program until problems are ironed out.

"We are concerned that our children's education is being used as a cost-saving experimental measure and we are just not

Mr Moore

Moore said yesterday t
extra funding for education
the State Budget had provi
the money for 26 schools to
undercover areas this financ
year.

Only eight assembly ar
were built last financial yea

Each assembly area
expected to cost $180,000
be ready for use by mid-199

"It is going to take so
time to upgrade our scho
after a long period of negl
but the Government
already made significa
inroads in addressing the ba
log," Mr Moore said.

"We will spend $2 million
the 1995-96 phase of this p
ject and $3 million the foll
ing financial year."

Mr Moore said archite
would begin visiting coun
and city schools early n
term and tenders for constr
tion would be called
November.

FIGURE 5.1: A newspaper image appropriate for mixed-mode segmentation.

The practical applications are discussed in Section 1.5.2 and are further apparent from Section 2.5.7. In summary, the decomposition into text and pictures provides a preprocessing tool for skew analysis, a method for determining the appropriate compression regime and, with further decomposition, input into an optical character recognition system.

5.3 Local feature extraction

For newspaper segmentation it is reasonable to choose local descriptions identifying texture-like properties at some scale. Under the framework of the UpWrite this involves representing a local neighbourhood of pixels as a point in a higher dimensional feature space. Assuming an effective local representation, large or sizeable homogeneous objects will form separable clusters in this space. Furthermore edges between regions, which are high in contextual information, will be anomalous to these clusters and hence detectable.

As outlined in Chapter 1 one of the principal objectives is to develop a set of generic methods which are reusable across a number of different applications and problem areas. The feature extraction techniques applied to anomaly detection are sufficiently general making few assumptions about image content. Consequently the same techniques are used.

The process, as described in Section 4.3, uses a block of windows to map an image in $\mathbb{R}^3$ into a series of measurement ($\mathbf{m}$) based points in $\mathbb{R}^{\mathbf{k}^2+2}$.

$$\mathbf{I}[\mathbf{i}, \mathbf{x}, \mathbf{y}] \quad \longmapsto \quad \mathbf{P}[\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{\mathbf{k}^2}, \mathbf{x}, \mathbf{y}]$$

This local feature extraction method is based on any chosen measurement which constitutes the lowest level primitive. For this restricted problem the simplest measurement adequate to provide a basic distinction between the textural regions of a newspaper is used. This is the standard deviation of the pixel intensity within each window.

$$\mathbf{m}_{window} = \sigma = \sqrt{\frac{(\sum \mathbf{i}^2)}{\mathbf{n}^2} - \left(\frac{\sum \mathbf{i}}{\mathbf{n}^2}\right)^2}$$

The typical values of these window measurements for $\mathbf{n} = 7$ are illustrated in Figure 5.2. In this example values are in the range $\mathbf{m}_{window} \in [1.61, 90.41]$ and scaled accordingly between black and white.

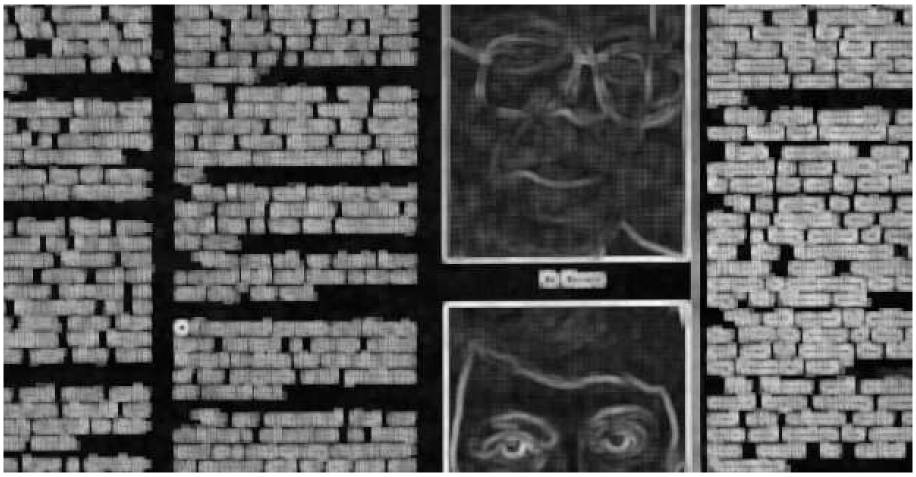


FIGURE 5.2: Value using a standard deviation window measurement. The white regions correspond to areas where the standard deviation of the intensity in a small local window is higher.

5.4 The feature space

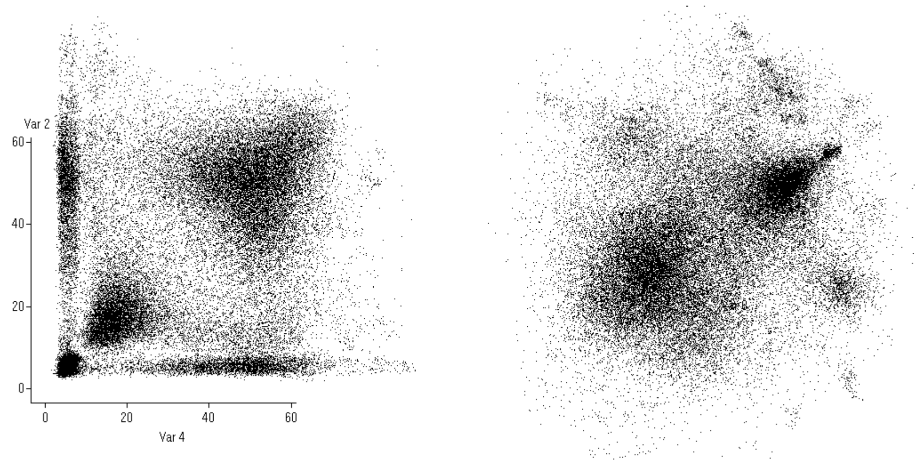
5.4.1 General characteristics

The feature space consists of clusters or at worst dense regions corresponding to segments of the image containing background, text, the left to right transition from background to text (left edge), the right edge and so on. Figure 5.3 shows example projections of the feature space using $\mathbf{k} = 3$, $\mathbf{n} = \mathbf{s} = 7$ and $\delta_x = \delta_y = 2$ for the default newspaper image[1] (Figure 5.1). To maintain consistency with Section 4.4 a projection is shown onto the plane of measurements 2 and 4 and arbitrarily, but orthogonal to the $(\mathbf{x},\mathbf{y})$ plane.

To gain an insight into the structure of the feature space regions of feature vectors have been manually coloured[2] under the first projection and re-projected onto the $(\mathbf{x},\mathbf{y})$ plane. As is intuitively expected the green points correspond to text, the red to pictures

[1]The values $\delta_x = \delta_y = 2$ are used in most illustrations to reduce the size of the dataset for real-time projection and manipulation using *xgobi* [63].

[2]The *brush* function of *xgobi* allows points to be coloured, assigned symbols or glyphs, or erased.

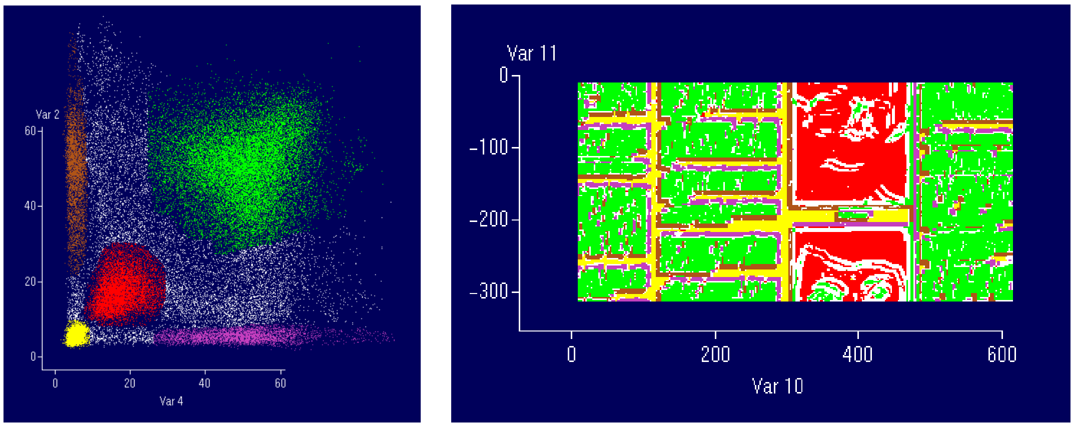


(a) (b)

FIGURE 5.3: Projection of the feature space for a single image.

and the yellow to background. The outlying clusters correspond to the transition between regions where neighbouring windows clearly occupy different image classes. For example the brown *cluster*, characterised by a large top window measurement and a small left one, is the combined result of bottom edges, right edges and other similar local features. The points between the coloured *clusters* indicate transition regions, noise and general non-conformity.

It is important to note that this is only one projection, hence the clusters do not necessarily exist in $\mathbb{R}^9$ and furthermore many transition regions with variations along hidden axes are incorrectly coloured. For example, the brown cluster contains at least two *clusters* (bottom edges and right edges). Nevertheless the structure observed from Figure 5.4 is promising.

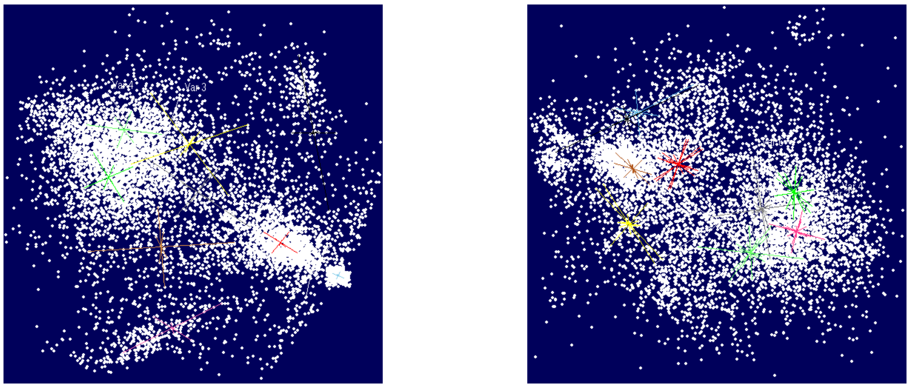


(a) (b)

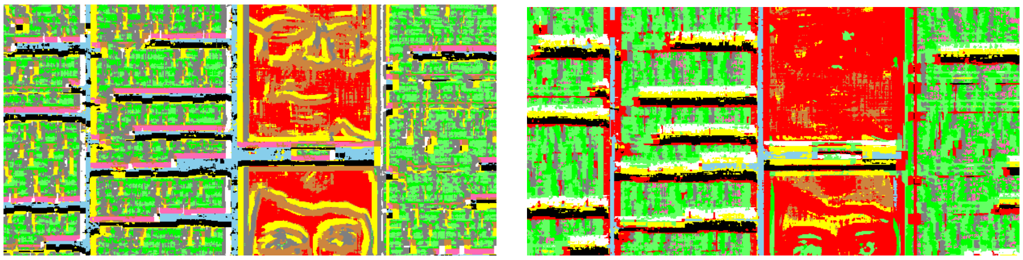
FIGURE 5.4: Projection from Figure 5.3(a) with points manually coloured and re-projected onto $(\mathbf{x},\mathbf{y})$ offsets.

5.4.2 Clustering

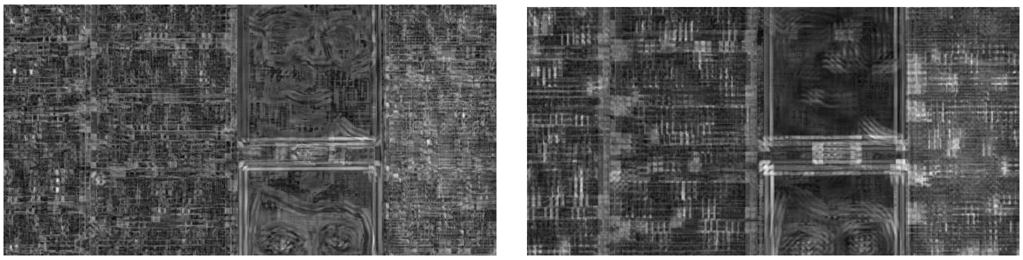
Given the success of manually labelling regions, clustering of the feature space is considered. This is performed using Gaussian mixture modelling with the EM algorithm as described and fully summarised in Section 2.1.5. Two feature spaces are considered with blocksizes $\mathbf{k}=2$ and $\mathbf{k}=3$ each for $\mathbf{n}=\mathbf{s}=7$ and $\delta_x=\delta_y=2$. In both cases 10 mixtures ($\mathbf{K}=10$) are used and Gaussian models are initialised with even weights, identity covariance matrices and means centred on randomly selected data points. Using fewer mixtures, e.g. $\mathbf{K}=5$, produces inferior results.



(a) Feature space with final mixtures represented by coloured lines.



(b) Point closest to each model.



(c) distance to this model, in the range $[0.08, 5.29]$ and $[0.56, 7.38]$ respectively.

FIGURE 5.5: Results of applying the EM algorithm on the feature space for $\mathbf{k}=2$ (left) and $\mathbf{k}=3$ (right).

Figure 5.5 shows the results of the EM algorithm after 100 iterations after which the model parameters clearly stabilise. The projection of the feature spaces includes lines corresponding to the eigenvectors of $\mathbf{C}_i$ from the mean μ_i coloured uniquely for each mixture $i \in \{1..\mathbf{K}\}$. By taking the Mahalanobis distance[3] of each data point with respect to each model, the point is assigned to the closest mixture. This is coloured accordingly in Figure 5.5(b) with the corresponding Mahalanobis distance shown in Figure 5.5(c). The brighter regions show where points fail to agree well with any of the mixtures and typically occur in transition regions for this type of feature space. Throughout the colours are manually assigned to roughly match those used previously.

The results suggest that although structure does exist it is not clear how to use this to classify local regions. It is promising that for $\mathbf{k}=3$ the picture regions are reasonably identified and not affected by the glasses or hair line. Unfortunately distinct clusters do not result from, and are not restricted to, individual region types.

5.4.3 Rule based segmentation

In view of the structure and inference made [303] from the feature space a rule based classification scheme is proposed. For simplification $\mathbf{k}=2$ is chosen with rules based on a combination of human intuition and eye balling the data using *xgobi*. The following algorithm is applied to each feature vector.

1. Set **class** $=0$ for unknown.

2. Obtain the next data point $\mathbf{p} \in \mathbf{P}$ from its symbolic representation.

3. If the magnitude of one or more window measurements is sufficiently high it possibly corresponds to an *edge* class. Since the variance of pixel intensity was originally considered, the relationship below is examined. If true, further rules determine if $\mathbf{p}$ corresponds to an edge. Each of these is considered independently and hence $\mathbf{p}$ may be assigned to more than one edge class.

$$\textit{if } (\mathbf{magnitude} > \mathbf{thres}_2) \ \textit{then} \qquad \text{where } \mathbf{magnitude} = \sqrt{\sum_{i=i}^{\mathbf{k}^2=4} \mathbf{p}_i^2}$$

   (a) If the variance of *all* top window measurements is sufficiently *low* and the variance of *any* of the bottom window measurements is sufficiently *high* set **class** to include the *top edge* class. This is achieved by the following pseudo code

$$\textit{if } (\ (\mathbf{p}_1 < \mathbf{thres}_1) \ \textit{and} \ (\mathbf{p}_2 < \mathbf{thres}_1) \ \textit{and} \ ((\mathbf{p}_3 > \mathbf{thres}_1) \textit{or} (\mathbf{p}_4 > \mathbf{thres}_1))\ )$$
$$\textit{then } \mathbf{class} = \mathbf{class} | = \text{TOP}$$

   (b) Similarly for the right edge class

$$\textit{if } (\ (\mathbf{p}_4 < \mathbf{thres}_1) \ \textit{and} \ (\mathbf{p}_2 < \mathbf{thres}_1) \ \textit{and} \ ((\mathbf{p}_3 > \mathbf{thres}_1) \textit{or} (\mathbf{p}_1 > \mathbf{thres}_1))\ )$$
$$\textit{then } \mathbf{class} = \mathbf{class} | = \text{RIGHT}$$

[3] See Section 4.5.1 or 2.1.5.

Algorithm class name	ID	Class type	Colour	Description or local characteristics
TOP	8	top edge	green	top edge of region
BOTTOM	4	bottom edge	red	bottom edge of region
LEFT	2	left edge	blue	left edge of region
RIGHT	1	right edge	yellow	right edge of region
LEFT | TOP	10	top left edge	cyan	both a top and left edge
RIGHT | TOP	9	top right edge	black	both a top and right edge
LEFT | BOTTOM	6	bottom left edge	magenta	both a bottom and left edge
RIGHT | BOTTOM	5	bottom right edge	orange	both a bottom and right edge
BACKGROUND	16	background	white	constant low variance
PICTURE	17	picture	pink	low variance, hence a picture
UNSURE	18	picture, text	light gray	a local region of mid-range variance which is possible a picture or text
TEXT	19	text	gray	high variance, hence text
default	0	-	-	this never occurs

TABLE 5.1: Colour assignments for rule based classification [303].

(c) Similarly for the bottom edge class

$$\textit{if}\ (\ (\mathbf{p}_3 < \mathbf{thres}_1)\ \textit{and}\ (\mathbf{p}_4 < \mathbf{thres}_1)\ \textit{and}\ ((\mathbf{p}_1 > \mathbf{thres}_1) or (\mathbf{p}_2 > \mathbf{thres}_1))\ )$$
$$\textit{then}\ \mathbf{class} = \mathbf{class}| = \text{BOTTOM}$$

(d) Similarly for the left edge class:

$$\textit{if}\ (\ (\mathbf{p}_1 < \mathbf{thres}_1)\ \textit{and}\ (\mathbf{p}_3 < \mathbf{thres}_1)\ \textit{and}\ ((\mathbf{p}_2 > \mathbf{thres}_1) or (\mathbf{p}_4 > \mathbf{thres}_1))\ )$$
$$\textit{then}\ \mathbf{class} = \mathbf{class}| = \text{LEFT}$$

4. *if* (**class** = 0) the point does not correspond to an edge class and requires classification as *background*, *picture* or *text*. This is based purely on the magnitudes of the variance window measurements.

   (a) A background class is assumed to have low measurements within all windows.

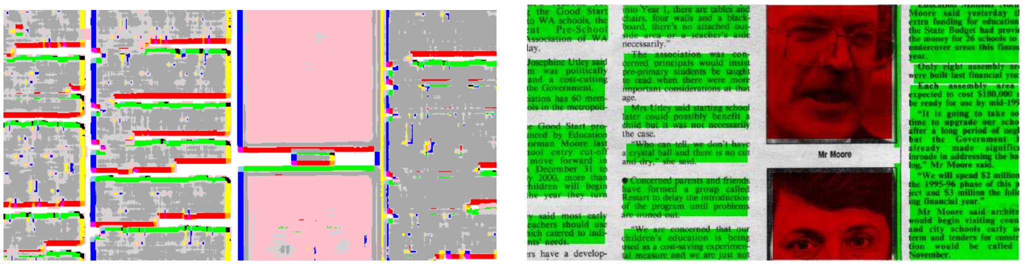
   $$\textit{if}\ (\mathbf{p}_i < \mathbf{thres}_1)\ \forall\ i \in \{1..4\}\ )\ \textit{then}\ \mathbf{class} = \text{BACKGROUND}$$

   (b) Other regions are classified based on the magnitude of the point as defined above.
   else if (**magnitude** $>$ **thres**$_3$) *then* **class** = PICTURE

   (c) *else if* (**magnitude** $>$ **thres**$_4$) *then* **class** = UNSURE

   (d) *else* **class** = PICTURE

5. Store the result of **class**, with the colour indicated in Table 5.1.



(a) rule based classification (b) corresponding result

FIGURE 5.6: Segmentation using rule based method ($\mathbf{n} = 7$).

The algorithm uses four static thresholds, set to $\mathbf{thres} = \{10, \sqrt{1000}, \sqrt{5000}, \sqrt{8000}\}$ for the illustrated results.

The performance is clearly superior to manual colouring or clustering. For a large windowsize ($\mathbf{n} = 7$) this algorithm successfully locates pictures, paragraphs of text and well defined edges. By using $\mathbf{n} = 4$ it adapts well to a finer scale, locating edges between major regions as well as many between lines of text and single well spaced words. Using similar heuristic rules [303] full region identification is achieved as shown in Figure 5.6(b).

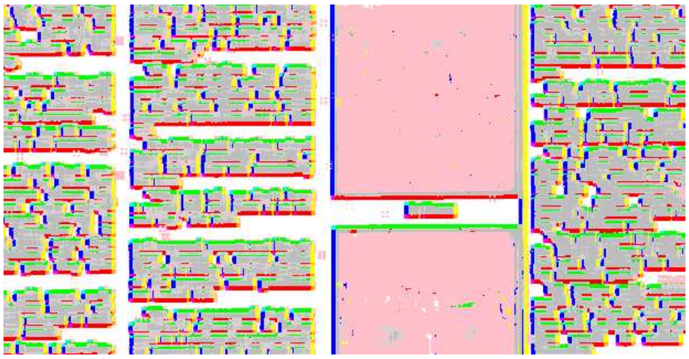


FIGURE 5.7: Rule based segmentation at a finer scale ($\mathbf{n} = 4$).

5.4.4 Feature vector parameterisation

So far, analysis of the feature space indicates structure which is extractable using specific rules for a defined simple neighbourhood complexity ($\mathbf{k} = 2$). The rule based scheme is not general. It cannot be adapted to a new form of data automatically through training, is sensitive to changes in image contrast, and does not support an arbitrary blocksize. Furthermore it lacks rotational invariance and cannot easily be adapted to arbitrary rotations. An alternative *parameterisation*[4] is proposed.

[4] Actually, the scheme proposed is an alternative representation and not strictly a parameterisation since it does not provide a one-to-one correspondence with the feature vector. However, it will be referred to as a parameterisation for convenience.

For this parameterisation it is assumed that window measurements are non-negative, restricting the feature space to the first *quadrant*. A feature vector $\mathbf{p} \in \mathbf{P}$ taken from a homogeneous section of the image $\mathbf{I}$ will have similar window measurements and therefore lie along the vector $\vec{\mathbf{1}} = [\mathbf{1}, \mathbf{1}, \ldots, \mathbf{1}]$, regardless of the dimension. The orthogonal distance $\mathbf{d}$ from $\mathbf{p}$ to $\vec{\mathbf{1}}$ is defined based on the length of the projection of $\mathbf{p}$ into $\vec{\mathbf{1}}$.

$$\mathbf{d} = ||\mathbf{p} - \mathbf{l}\,\vec{\mathbf{1}}|| \qquad \text{where } \mathbf{l} = ||\mathrm{proj}_{\vec{\mathbf{1}}}\mathbf{p}||$$

Given a small value of $\mathbf{d}$, the value $\mathbf{l}$ gives an indication of the image's class (for example *picture* or *text*). Large values of $\mathbf{d}$ indicate a less regular region, possibly relating to a boundary or *edge* class.

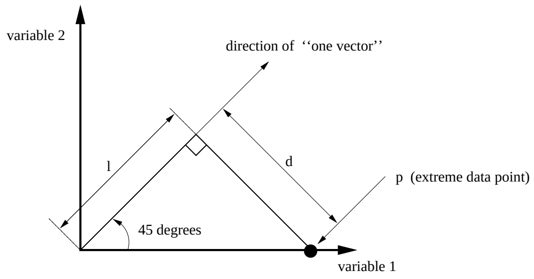


FIGURE 5.8: The relationship between $\mathbf{l}$ and $\mathbf{d}$ for an extreme point in two dimensions.

Although the measurements $\mathbf{d}$ and $\mathbf{l}$ seem to form a natural parameterisation, the value of $\mathbf{d}$ is restricted by the geometric structure of the feature space and its value is dependent on the dimension of the feature space. Figure 5.8 shows the limiting value for $\mathbf{d}$ in two dimensions ($\mathbf{N} = 2$) illustrating the enforced condition $\mathbf{d} \leq \mathbf{l}$. In general the limiting value occurs when all but one of the vector elements are 0. Assigning the remaining element to $\mathbf{N}a$ enforces the length $\mathbf{l} = a\sqrt{\mathbf{N}}$. As examples

$$\mathbf{p} = (\sqrt{2}, 0) \qquad a = \tfrac{\sqrt{2}}{2} = \tfrac{1}{\sqrt{2}} \qquad \mathbf{l} = ||(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})|| = a\sqrt{2} = 1$$
$$\mathbf{p} = (1, 0, 0, 0) \qquad a = \tfrac{1}{4} \qquad \mathbf{l} = ||(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})|| = a\sqrt{4} = \frac{1}{2}$$

Consequently, for an extreme data point, the orthogonal distance $\mathbf{d}$ is equal to

$$\sqrt{(\mathbf{N}-1)a^2 + (\mathbf{N}a - a)^2} = a\sqrt{(\mathbf{N}-1) + (\mathbf{N}-1)^2} = a\sqrt{\mathbf{N}(\mathbf{N}-1)}$$

Hence the general restriction is

$$\mathbf{d} \leq \frac{a\sqrt{\mathbf{N}(\mathbf{N}-1)}}{a\sqrt{\mathbf{N}}}\mathbf{l} \qquad i.e.\ \mathbf{d} \leq \left(\sqrt{\mathbf{k}^2 - 1}\right)\mathbf{l}$$

The above shows a dependency between the blocksize $\mathbf{k}$ and the measurements $\mathbf{d}$ and $\mathbf{l}$. This restriction causes a shearing effect in the parameter space and is unnecessarily dependent on the dimension (and hence the blocksize $\mathbf{k}$.) An alternative measurement for $\mathbf{l}$, which has no obvious dependencies *with respect to* $\mathbf{d}$, is given by the difference between the first two Minkowsky metrics (defined on page 2.1.5) for vector $\mathbf{p}$,

$$\eta = ||\mathbf{p}||_1 - ||\mathbf{p}||_2 = \sum_{i=1}^{\mathbf{k}^2} |\mathbf{p}_i| - \left(\sum_{i=1}^{\mathbf{k}^2} \mathbf{p}_i^2 \right)^{\frac{1}{2}}$$

The new parameter space is formed with the variables η and $\mathbf{d}$. Each is scaled according to $\mathbf{P}$ such that the majority (99.8%) of parameterised points lie between 0 and 100. This parameterisation is effective across numerous types of grey scale images and is not dependent on the dimension of the feature space. In summary, the process maps each feature vector as follows

$$\mathbf{P}[\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{\mathbf{k}^2}, \mathbf{x}, \mathbf{y}] \quad \longmapsto \quad \Phi[\eta, \mathbf{d}, \mathbf{x}, \mathbf{y}]$$

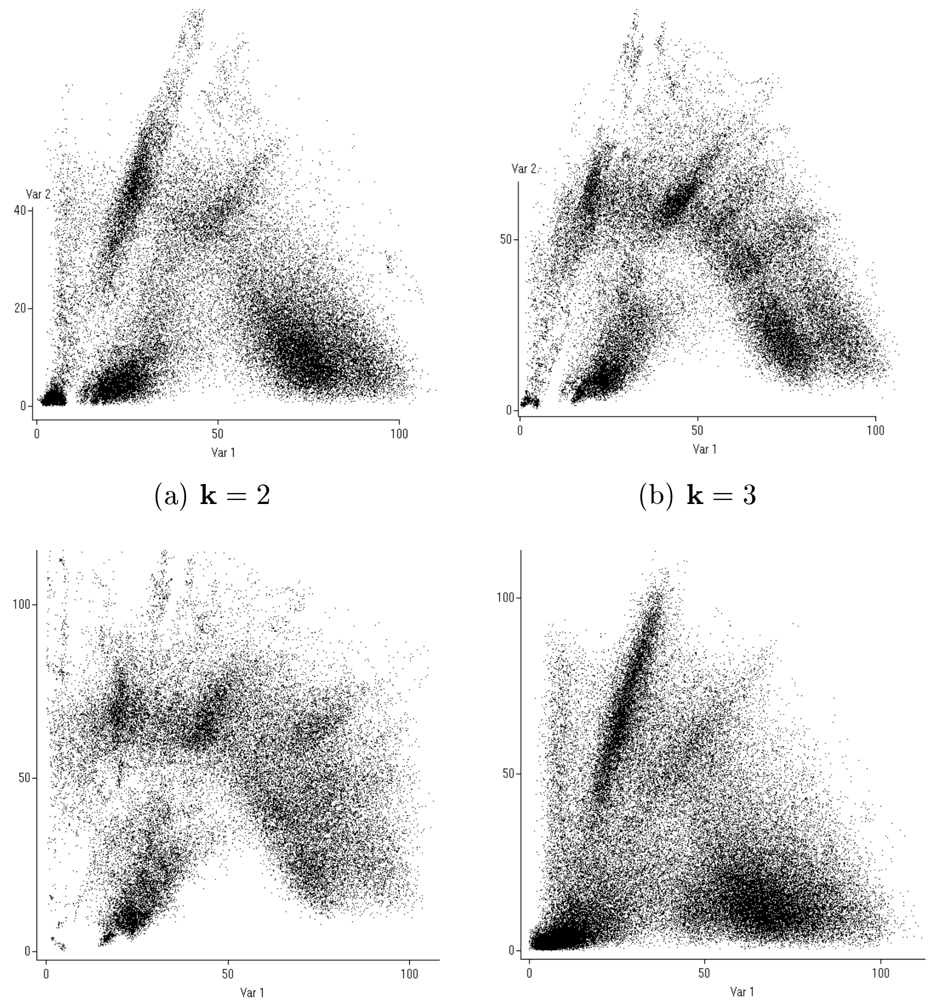
Figure 5.9 shows examples of the parameter space for $\mathbf{n} = 7$ and $\mathbf{k} \in \{2..4\}$. The clusters clearly apparent for $\mathbf{k} = 2$ and $\mathbf{k} = 3$ correspond to background (top left), pictures (bottom middle), text (right), edges (top left) and transition regions between text and edges (top). These have been verified using *xgobi*. For $\mathbf{n} = 2$ parameter values are more susceptible to noise and local variation due to the limited neighbourhood complexity of the original feature vectors. On the other hand for $\mathbf{n} = 4$ the mask is quite large (28×28 pixels) which results in a large proportion of placements between clearly defined regions.

Combining the parameter space of 5 different images still results in clusters, although the distinction between background and pictures is not clear. This results from pictures containing non-varying regions and backgrounds which are subtly textured. In general, it suggests the need to consider them relative to the context of each single image.

5.4.5 Summary and discussion

The feature space contains significant structure as outlined by manually colouring, clustering and applying specified rules for a restricted case. One approach is to perform supervised training by providing labelled examples of different edges, background, pictures and text, using a classifier such as a neural network. However, the selection of these classes, particularly edges, is subjective. Furthermore the resulting regions are sparse, dependent on dimension (or blocksize) and would require large amounts of appropriate training data. The proposed alternative is a parameterisation which concisely captures appropriate structure for this problem and adds desirable invariances.

The final $\eta - \mathbf{d}$ parameterisation makes use of the fact that image regions which are homogeneous with respect to the applied window measurement lie close to the one vector $\vec{\mathbf{1}}$ as outlined in Section 4.4. By adding additional assumptions the structure of the feature space is simplified. Firstly the parameterisation combines clusters associated with edges,



(a) $\mathbf{k} = 2$ (b) $\mathbf{k} = 3$

(c) $\mathbf{k} = 4$ (d) $\mathbf{k} = 2$, for 5 images

FIGURE 5.9: Feature space parameterisation using $\mathbf{n} = \mathbf{s} = 7$, $\delta_x = \delta_y = 2$ and varying blocksizes $\mathbf{k}$. The final image shows the space formed from 5 different images. Throughout all figures η are shown on the horizontal axis.

thus removing the distinction between different edges and adding a level of rotational invariance. Secondly, it scales values with respect to the current image which adds an invariance to relative contrast. However this relies on the image containing some regions of text and background as is assumed for this restricted problem. Obviously invariances to mean intensity levels and inverted images are handled by the chosen window measurement.

Overall the final parameterisation is general and permits training. It provides a tool for simplifying the feature space and forms clusters, or at worst dense regions, which correspond to meaningful local image characteristics. Hence it is used to identify objects.

5.5 Identification of text, picture, background and other regions

Five region classes are considered based on the rule based system and structures previously outlined. There are background, pictures, text, edges and other regions. The final type corresponds to non-homogeneous text, transitions or highly contrast sections of pictures, or thin, weak or double edges. Each class corresponds to a region of Φ present across different images as apparent from Figure 5.9. These are modelled using positive semi-definite quadratic forms, allowing individual points to be classified.

In general, training is performed by modelling the parameter space of a pre-classified image through taking geometric moments. However, in the case of mixed-mode images, it is desirable to model classes according to the relative structure within each image. Hence points $\mathbf{p} \in \Phi$ with a high certainty of belonging to a specific class are used to form, or refine, the class models. These models are characterised by a mean μ and covariance matrix $\mathbf{C}$ which define a positive semi-definite quadratic form

$$\mathbf{q}_{class} = (\mathbf{p} - \mu)^T \mathbf{C}^{-1} (\mathbf{p} - \mu)$$

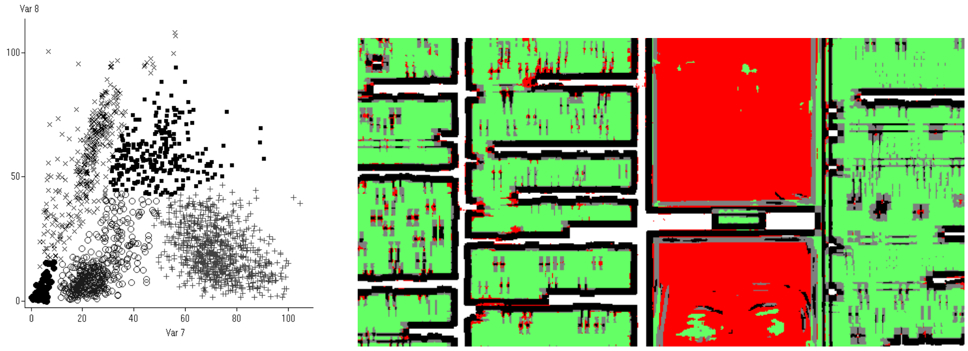
Each parameterised point $\mathbf{p} \in \Phi$ is assigned a likelihood to each possible class. This likelihood is given by $\mathbf{c}_{class} = e^{\frac{1}{2}\mathbf{q}_{class}}$. Points located at the centre of this model are assigned a likelihood of $e^0 = 1$, those at a distance of one standard deviation (with respect to the model) a value of $e^{-\frac{1}{2}\times 1} = 0.61$ and so forth. Basically this provides an initial classification Υ

$$\Phi[\eta, \mathbf{d}, \mathbf{x}, \mathbf{y}] \quad \longmapsto \quad \Upsilon[\mathbf{c}_{background}, \mathbf{c}_{picture}, \mathbf{c}_{text}, \mathbf{c}_{edge}, \mathbf{c}_{other}, \mathbf{x}, \mathbf{y}]$$

Using this, points are assigned according to the maximum likelihood to produce an *initial point based classification*. This model is equivalent to a Quadratic neural network [302] in which each neuron computes a quadratic form that is a function of its inputs. Consequently the decision boundaries are piecewise quadratic as opposed to traditional neural networks which are piecewise linear (more correctly piecewise affine). This is apparent in Figure 5.10 which shows the initial point based classification of the default newspaper image.

As stated above the models are formed specific to the structure observed in Figure 5.9. For example the *other* class uses only those points where $\mathbf{d} \geq 50$ and $\eta > 0.5\mathbf{d} + 10$ (a line with gradient 2). However it should be noted that the initial classification identifies edges where $\mathbf{d} < 50$ by using the refined class model for this particular image. Similarly the refined model adapts to the decision boundary between other classes, such as picture and background, according to the density of parameter space regions.

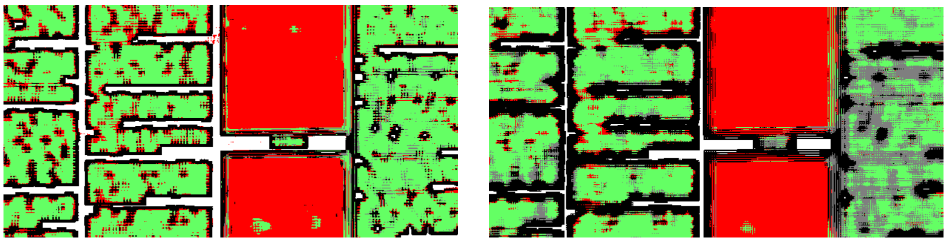
As seen in Figure 5.11 this initial classification method is robust in terms of feature extraction parameters for similar sized masks. This illustrates the effectiveness of the



(a) parameter space $(\eta, \mathbf{d})$ (b) parameter space $(\mathbf{x}, \mathbf{y})$

FIGURE 5.10: Initial classification of the parameter space using $\mathbf{k} = 2$ and $\mathbf{n} = \mathbf{s} = 7$.

parameterisation scheme even when transforming points from $\mathbb{R}^{36}$ to $\mathbb{R}^2$ (or $\mathbb{R}^{38}$ to $\mathbb{R}^4$). In this case the 36 separate 3×3 pixel window measurements are unlikely to be large and still maintain similar magnitudes, i.e. form large η and small $\mathbf{d}$ values. Consequently the model for text has a mean with a higher $\mathbf{d}$ value and hence lies close to the *other* class model increasing the chances of misclassification. The mean of the text model is $(63.6, 46.7)$ as opposed to $(67.7, 19.6)$ when $\mathbf{k} = 2$ and $\mathbf{n} = \mathbf{s} = 7$ are used (apparent from Figures 5.9(a) and (d)). This indicates the lack of points in the region which is expected to result from text. Similarly background regions are more likely to be affected by noise for large $\mathbf{k}$. Nevertheless the initial classification is acceptable.



(a) $\mathbf{k} = 3$, $\mathbf{n} = \mathbf{s} = 4$ (b) $\mathbf{k} = 6$, $\mathbf{n} = \mathbf{s} = 3$

FIGURE 5.11: Initial classification using different feature extraction parameters for a similar mask size.

5.6 Object description, representation and initial classification

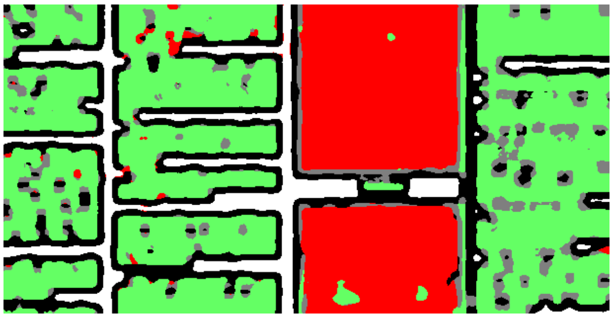
As is clear from the last section the identification of objects for this problem equates to locating clusters in the parameter space without considering spatial offset. Hence the individual objects correspond to the 5 classes which are represented by $\mu_i, \mathbf{C}_i \in \Phi$ for each class i.

Alternatively each spatially disjoint connected region of the same class may be considered an object. In this case the mean intensity, mean local standard deviation and a shape description may be appropriate to represent the object. In general there is no need for this since the problem is restricted to identifying and classifying regions and not to examining them further for higher level structure.

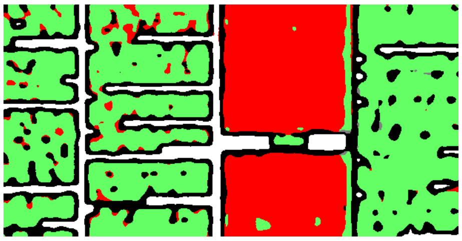
5.7 Spatial neighbourhood considerations

Although the nature of the feature extraction process naturally considers spatial neighbourhoods, the initial point based classification does not consider spatial neighbours. Consequently point based classification results suffer from small anomalous classifications within otherwise correctly classified regions such as the text (green) and edges (black) in the pictures of Figure 5.10(b).

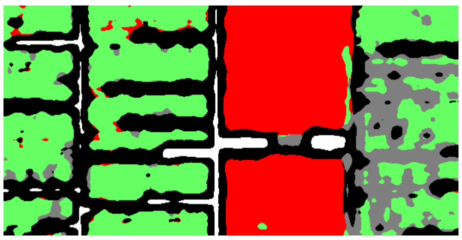
One approach is to handle this during post-processing using mathematical morphology or similar techniques. However this does not consider the likelihood of the classification or the relative position of points in the parameter space. For example, a region consisting of 10 *green* and 10 *red* points may be formed by either text or pictures. If all 10 green points lie very close to the decision boundary and the red points are a sufficient distance from the boundary, the region is probably part of a picture.



(a) $\mathbf{k} = 2$, $\mathbf{n} = \mathbf{s} = 7$



(b) $\mathbf{k} = 3$, $\mathbf{n} = \mathbf{s} = 4$



(c) $\mathbf{k} = 6$, $\mathbf{n} = \mathbf{s} = 3$

FIGURE 5.12: Final classification using a 10^{th} order neighbourhood, and a smoothing deviation of 3.

A better approach is to examine the spatial neighbourhood of Υ rather than the discrete maximum likelihoods. An n^{th} order spatial neighbourhood[5] of Υ is considered. The likelihoods $\mathbf{c}_i$ in this neighbourhood are weighted by a Gaussian smoothing function and summed to form new likelihoods for the neighbourhood. This is equivalent to numerous convolutions with a truncated or spatially contained Gaussian smoothing function over each class likelihood. The maximum of these provides the final classification.

The final results in Figure 5.12 show a significant improvement over the initial point based classification.

5.7.1 Some other results

All results presented from this point on are processed with identical parameters, as used in Figure 5.12(a). It should be noted that the images vary significantly in size and resolution and hence better results are obtainable by appropriately adjusting $\mathbf{n}$. This is not done to avoid subjective parameter choices. The size of the default image is 625×324 pixels.

(a) 592×711 pixel image (b) initial (c) final

FIGURE 5.13: Initial and final classification illustrating the effects of scale.

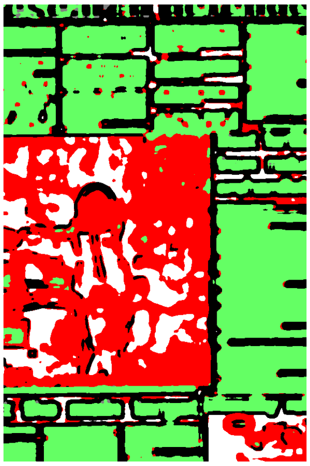
The result in Figure 5.13 *successfully* identifies meaningful image regions. The standard text and headlines are identified correctly, but the "\$600" text at the bottom is too large for the resolution parameter ($\mathbf{n}$). Regions in the picture are appropriately misclassified as background due to local characteristics.

Figure 5.14 illustrates the method's rotational invariance. It should be noted that the bias of the mask's orientation (maximised at this angle) causes slightly poorer results. If required this bias can be overcome by using circular windows, perhaps arranged at the centre and vertices of a hexagon. Edge effects are also apparent near the bottom left of the image.

Figures 5.15 and 5.16 provide examples of where the assumptions are breached and the method fails. In the first case the black section of the picture has less variation than

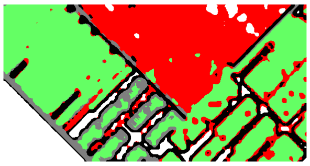
[5] The local region is considered in terms of neighbourhood order rather than the points within a defined distance. This is because at the time the code was applied to random field segmentation using ICM and hence reused. It is equivalent to a threshold of Euclidean distance as presented in Figure 2.33 on page 68.

(a) 598 × 896 pixel image



(b) final result

(c) rotated section



(d) rotated result

FIGURE 5.14: An illustration of rotational invariance.

the rippled textured background, hence causing portions of the background to be labelled as picture and visa versa. Nevertheless the result is reasonable. In the final example, the pictures have high contrast and contain text-like local characteristics. This shows the limitations of the method which fails to consider higher level structure. The position and alignment of the pictures, which cause a human to identify them as such, are not considered by the algorithm.

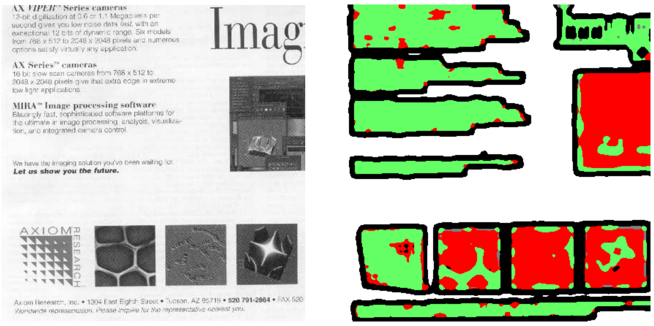
5.8 Post processing and presentation of results

Although the final local classifications are impressive it is desirable to automatically outline distinct text and picture regions from the original image. This is achieved by locating a

(a) 530 × 361 pixel image

(b) final result

FIGURE 5.15: The undesirable effects of the background assumptions.



(a) 428 × 447 pixel image (b) final result

FIGURE 5.16: An example in which the method fails.

sizeable classified region and performing a simple rectangle expanding flood fill algorithm, stopping when an insufficient proportion of unsuitably classed points is met. This *post-processing* is specific to newspaper segmentation and adds the additional assumption that regions are oriented at right angles.

5.8.1 Simple region identification algorithm

In summary the algorithm consists of two steps [303].

1. Locate all *sufficiently large rectangular picture* regions based on the final classifications. The variable **smallestpic** defines the width of the smallest acceptable picture.

2. Similarly locate sufficiently large rectangular *text* regions for unclassified regions (i.e. those which have not already been identified as pictures.) A different variable **smallesttext** is used since *sufficiently large* may vary between these classes.

The location of rectangular regions for each class is performed as follows.

- Sequentially scan through the classified data points locating square seed regions consisting entirely of the appropriate class. The seeds are sized according to **smallestpic** or **smallesttext**.

- Set the variables **top**, **bottom**, **left** and **right** to define the region.

- Repeat the following until the region no longer grows, i.e. there is no change in **top**, **bottom**, **left** or **right**.

    1. **top** = **up**()
    2. **left** = **back**()
    3. **bottom** = **down**()
    4. **right** = **forward**()

- The resultant rectangle corresponds to the final **class**. The classification of its points is replaced with a **used** state to avoid incorporation into other identified regions. A copy of the original image is also coloured to produce an output image.

- Continue to scan sequentially through the image to locate all appropriate regions.

The routine **up**() makes use of inputs **class**, **left**, **right**, **top** and the classified **image** map as follows. Routines **back**(), **down**() and **forward**() have the same structure.

- *if* **top** $= 0$ *return*. A simple boundary constraint.

- Examine the initial classifications immediately *above* the current semi-classified rectangular region. These are the values of **image** at the vertical position specified by **top** and between horizontal positions **left** and **right**, i.e.

$$\vec{\mathbf{x}} = [\mathbf{image}_{\mathbf{top}-\mathbf{1},\mathbf{left}}, \mathbf{image}_{\mathbf{top}-\mathbf{1},\mathbf{left}+\mathbf{1}}, ..., \mathbf{image}_{\mathbf{top}-\mathbf{1},\mathbf{right}}]$$

- Calculate the proportion of the elements within $\vec{\mathbf{x}}$ which are *acceptable* for the current final **class**. If the proportion is *sufficiently high* set **top** $=$ **top** $- 1$.

$$\begin{aligned} &\textit{if} \left(\sum_{i=left}^{right} (\mathrm{acceptable}(\vec{\mathbf{x}}_{\mathbf{i}}) - \textit{unacceptable}(\vec{\mathbf{x}}_{\mathbf{i}})) \right) \geq \mathbf{threshold} \\ &\textit{then } \mathbf{top} = \mathbf{top} - 1 \end{aligned}$$

  For pictures, *background*, *picture* and *other* regions are acceptable. For text, *other* and *text* regions are acceptable while *background* is unacceptable to encourage termination of final text regions.

- Return the value to **top** indicating the *new* (possibly unchanged) boundary.

This algorithm is heuristic and very simple. It makes no direct use of edges which intuitively appear to hold the majority of boundary information. The growing phase only considers the proportion of classed points in any one direction and hence has the potential to swallow small text regions within images. This makes it only applicable for images

FIGURE 5.17: Large disjoint picture and text regions.

containing regular shaped pictures. These and other points are discussed more thoroughly in [303].

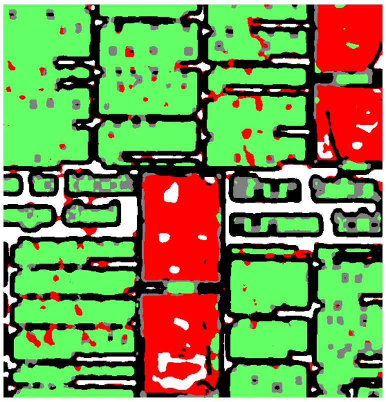
Overall it is not general or particularly intelligent, but it is effective. Figure 5.17 shows the resulting picture and text regions using this algorithm. Throughout algorithm parameters are set to **smallestpic** = 20, **smallesttext** = 7 and **threshold** = 80%.

5.8.2 Final results

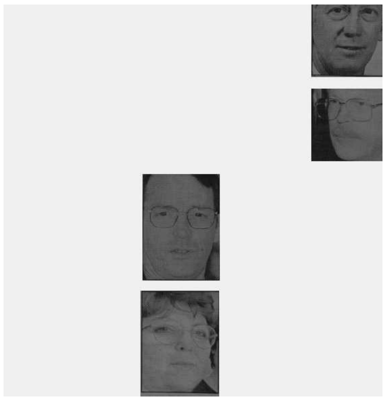
The final segmentation results are illustrated on *all* 20 samples images investigated, processed using exactly the same program with identical parameters.

The first example, taken from the same newspaper source, shows the ability to isolate pictures and the majority of text regions (Figure 5.18). The variation in image brightness and the range of text sizes are within the algorithm's capabilities.

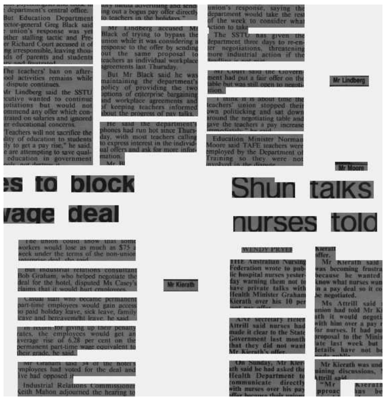
(a) 559 × 578 pixel image



(b) final point classification



(c) picture regions



(d) text regions

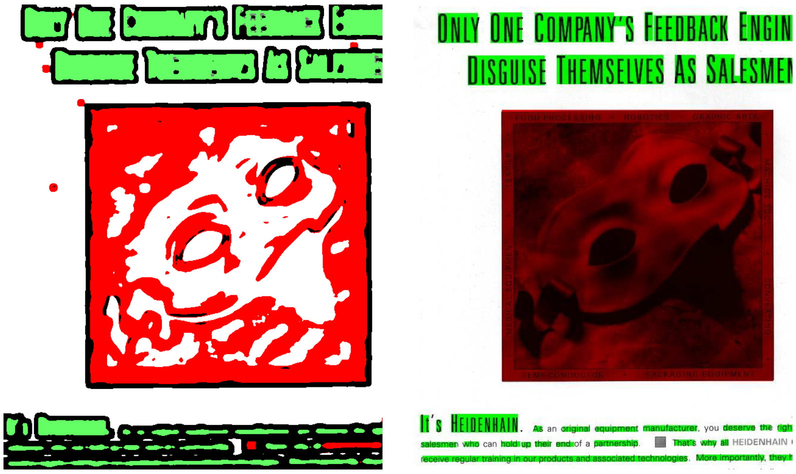
FIGURE 5.18: A full example which demonstrates invariance to image brightness.

Figure 5.19, from a computer magazine, provides the only example where post-processing completely fails. Although the final classification is impressive, clearly outlining all appropriate regions, the irregular shaped picture causes undesirable expansion across regions

FIGURE 5.19: A 657 × 618 pixel image not suited to post-processing.

which are mostly background. Hence the small isolated text regions are swallowed by the picture which is not determined by the strong edge regions.

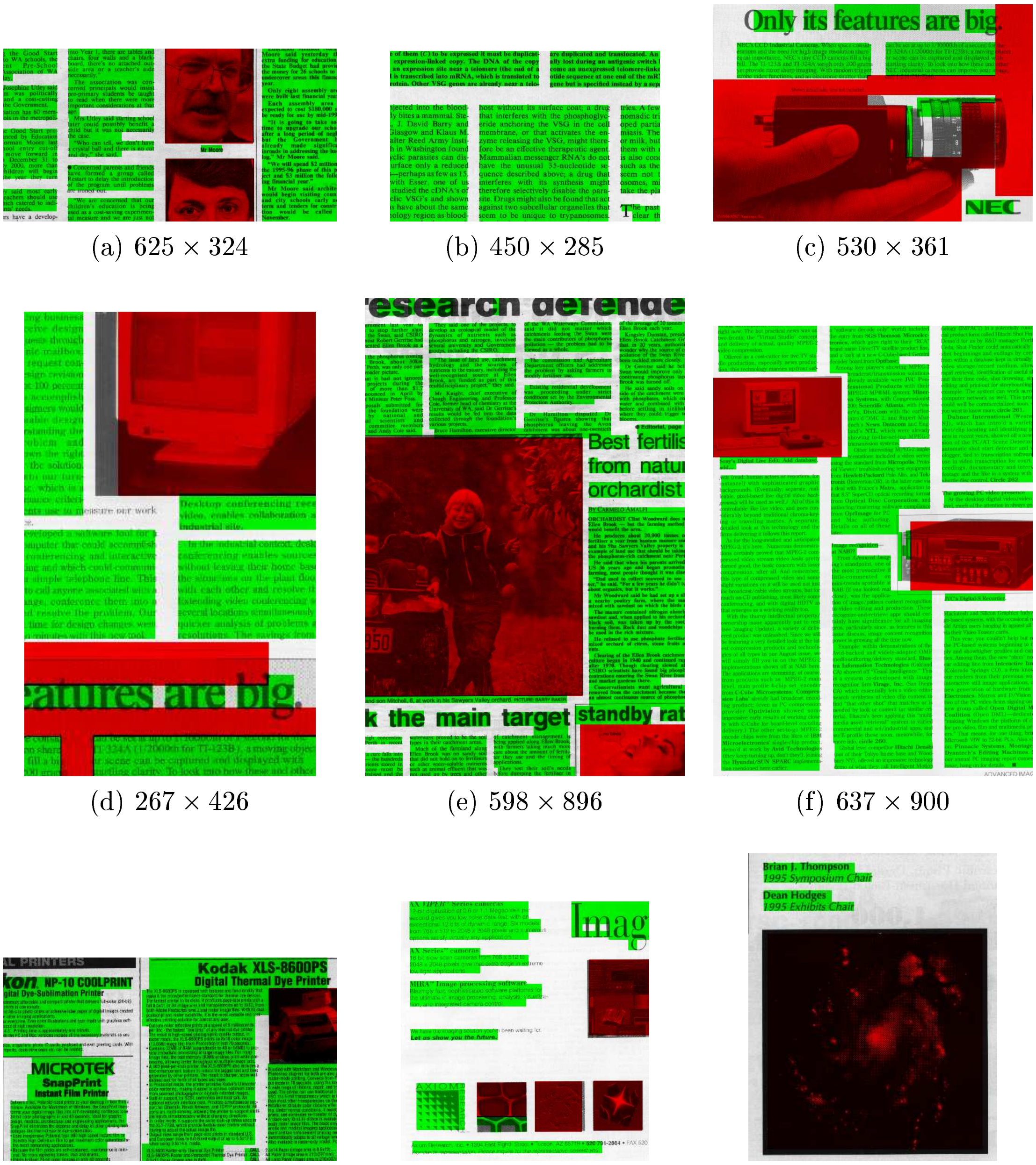
In contrast, the result in Figure 5.20 benefits from the post-processing assumption of large regular rectangular pictures. The regions *incorrectly* classified as background are caused by constant and slowly varying picture regions. Despite this, the picture is identified correctly after post-processing.



(a) 635 × 775 pixel image (b) result

FIGURE 5.20: Segmentation of an image with very smooth picture regions.

The remaining 17 results appear in Figure 5.21, with selected pictures not shown previously reproduced in Figure 5.22 to further outline local characteristics. Overall the results are particularly good over a range of different scales (as indicated) and sources of printed text. The images in (f), (i), (l) and (p) were originally scanned, or chosen, for their pictures which contain fine text-like characteristics. These results are particularly promising. Figures 5.21(b), (j), (n) and (q) demonstrate the ability to locate text regions with the absence of pictures. The opposite is not within the scope since it breaches the original assumptions.

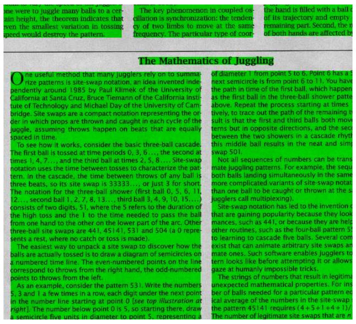


(a) 625 × 324 (b) 450 × 285 (c) 530 × 361

(d) 267 × 426 (e) 598 × 896 (f) 637 × 900

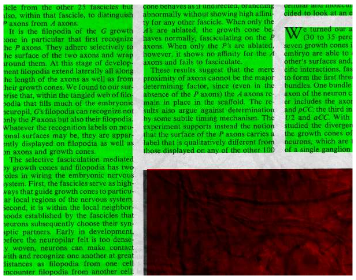
(g) 503 × 307 (h) 428 × 447 (i) 325 × 403

FIGURE 5.21: Final segmentation results. [part 1]

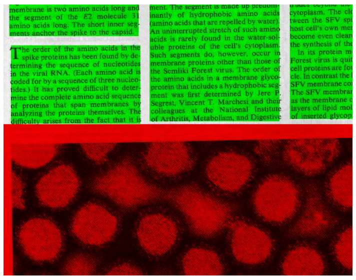


(j) 632 × 571

(k) 592 × 711

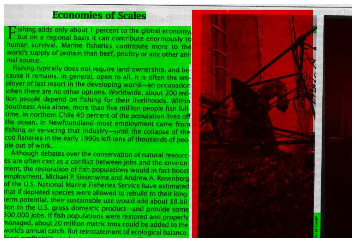


(l) 603 × 468



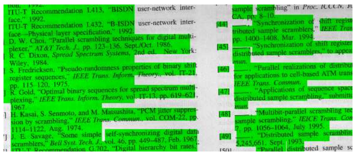
(m) 607 × 471

(n) 553 × 421



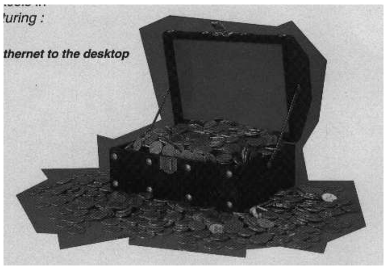
(o) 639 × 428

(p) 969 × 450



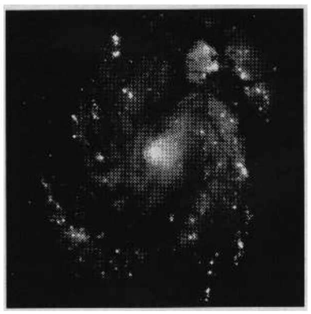
(q) 578 × 243

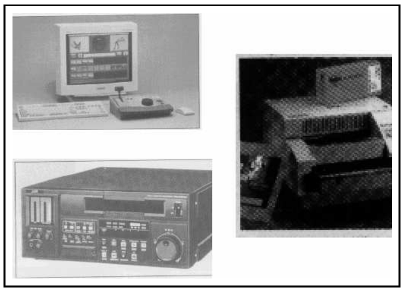
FIGURE 5.21: Final segmentation results. [part 2]



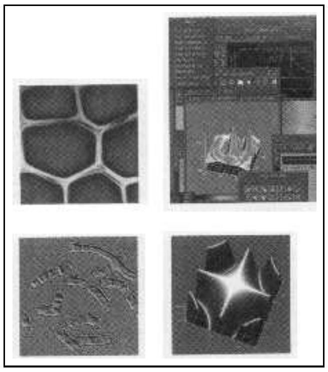
from Figure 5.19(b)

5.20(b)

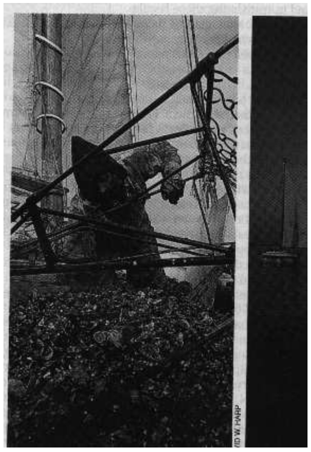

5.21(i)

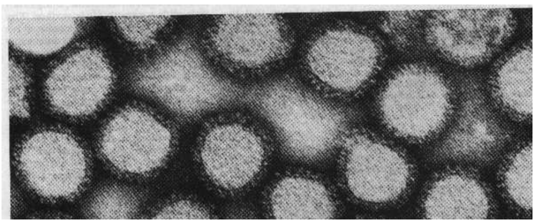

5.21(f) and (g)


5.21(d)


5.21(h)


5.21(p)


5.21(o)


5.21(m)

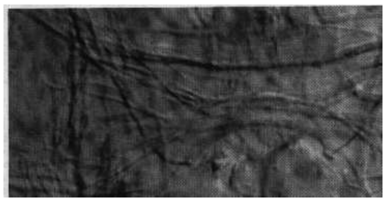

5.21(l)

FIGURE 5.22: Original grey scale regions classified as picture.

5.8.3 Timing considerations

Since no real-time implementation is suggested the computational time required is not considered critical. The non-optimised code operating on the default image requires about 10 megabytes of memory and the following computation time on an R10000.

- 0.66 seconds to calculate window measurement required for $\mathbf{P}$. This is optimised as outlined in Section 4.3.2.
- 1.07 seconds to calculate η and $\mathbf{d}$.
- 0.15 seconds to scale η and $\mathbf{d}$ and form Φ.
- 0.17 seconds to identify regions of Φ for modelling.
- 0.50 seconds to refine class models.
- 6.4 seconds to calculate likelihoods for all points to each class, hence forming Υ. This is highly unoptimised using an arbitrary dimensional Gaussian list and symbolic image representations to form multiple images depending on the number of Gaussians in the list.
- An unrequired time to calculate the maximum likelihood and distances for the presentation of initial point based classification results. Currently this recalculates all likelihoods, but uses different code which only takes 5.5 seconds.
- 13.9 seconds for final neighbourhood smoothing based classification. Again this is highly unoptimised.
- 0.3 seconds for the simple flood fill algorithm.

In summary, the process takes about 23 seconds without optimisations or parallel considerations. Most operations can be performed in parallel.

5.9 The DownWrite

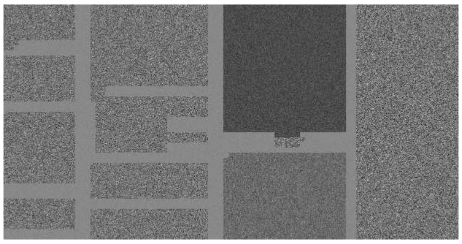
The idea behind the DownWrite, as previously outlined and seen,[6] is to construct an instance of an object to investigate the information captured by a model and provide an insight into its limitations. In this case it is difficult to recreate image sections since the *parameterisation* of feature vectors loses significant information. It is suggested that a multi-scale UpWrite into the parameter space may be required to allow for a sophisticated DownWrite. Alternatively, a DownWrite could be based on specific modelling of individual regions corresponding to sizeable disjoint pictures, text, edges and background.

However, for newspaper segmentation it is unclear why a *high quality* reconstruction is useful. Instead a simple reconstruction is considered by modelling each major disjoint

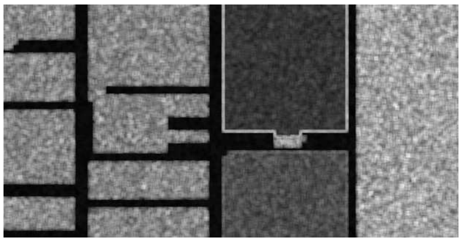
[6] Refer to Sections 3.2 and 4.9 and references within.

region by its mean μ and standard deviation σ. This is illustrated in Figure 5.23 using 5 major regions (2 text and 2 pictures) from the result in Figure 5.21(a).

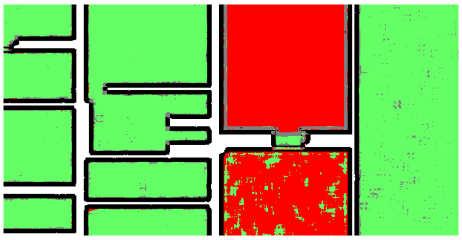
Each disjoint 7×7 window of the final classification is examined to determine the most dominant class. The model (μ, σ) for this class is used to reconstruct pixels in that window. This is repeated across the image as seen in Figure 5.23. While the image itself does not resemble the original or a newspaper for that matter, an insight into the $\eta - \mathbf{d}$ parameterisation and the overall algorithm is obtained by processing this image.



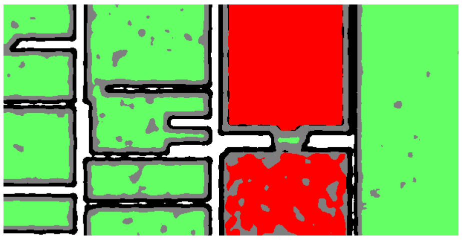
(a) image reconstruction



(b) window measurement



(c) initial classification



(d) final classification

FIGURE 5.23: The classification of a reconstructed image provides some insight into the algorithm.

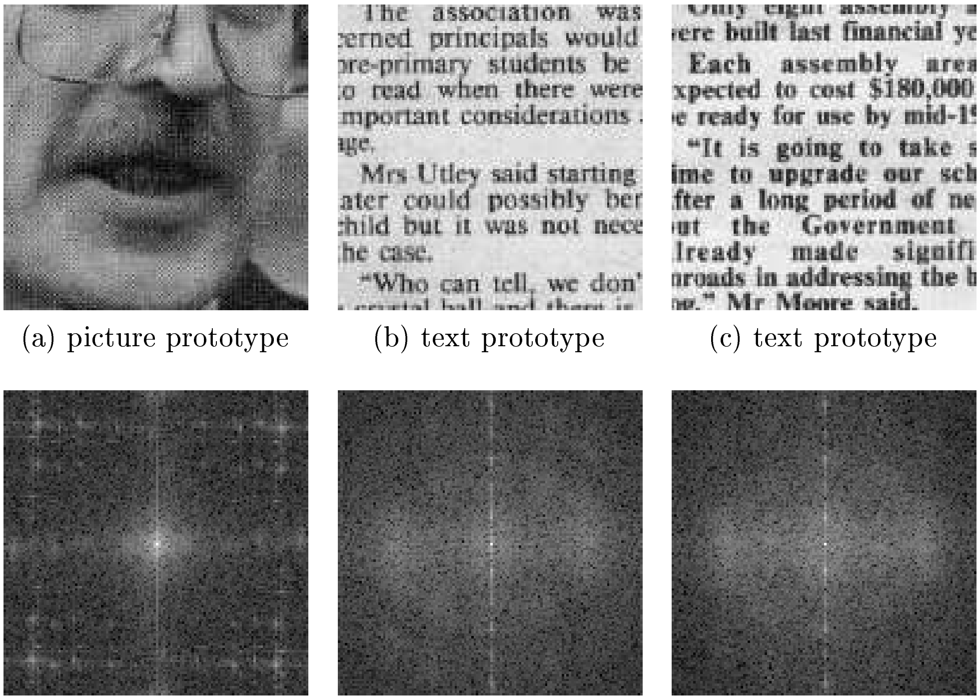
5.10 A comparison and conclusions

5.10.1 Optimal Gabor filter segmentation

Segmentation of newspaper into pictures and text using an optimally designed Gabor filter is now presented. The algorithm proposed by Dunn [134] as outlined at the end of Section 2.5.5 (page 87) is used. This requires two prototype textures and an appropriately chosen set of spatial frequencies, orientations and bandwidths.

Figure 5.24 shows proposed 128×128 pixel picture and text prototypes along with their frequency spectra. The algorithm is applied to samples (a) and (b), and (a) and (c) with orientation spacing of $15°$ covering the full $360°$ of rotation. Each orientation is examined at a total of 7 logarithmically spaced frequencies up to the limiting Nyquist sampling frequency and 35 linearly spaced bandwidths[7] σ to fully cover a range of local

[7]Values chosen limit the spatial spread of the filter to gain spatial precision. Using very small frequency bandwidths may result in better filters, but result in non-precise smooth regions which are undesirable for



(a) picture prototype (b) text prototype (c) text prototype

frequency spectra of the above; $(1 + \log(FFT))$ scaled.

FIGURE 5.24: Texture prototypes considered for designing the optimal Gabor filter.

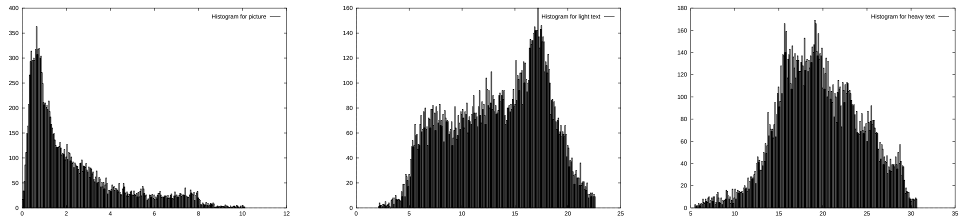
precisions.

The corresponding $\frac{360}{15} \times 7 \times 35 = 5880$ Gabor filters are used to determine the optimal filter. In each case a model (μ,σ) is formed from the magnitude of all filter outputs across each sample image. Although seemingly computationally expensive this only requires the formulation of a translated Gaussian complex image, 3 complex image multiplications, 3 inverse FFTs and 3 magnitude and statistic calculations which take a total of about 0.3 seconds on an R10000 processor using custom C libraries. Hence the modelling requires less than 40 minutes of processing time. This can obviously be reduced by intelligent filter design, although is not considered since the aim is to investigate *the optimal* Gabor filter.

Using samples (a) and (b) several different filters obtained a high ranking, each with a high frequency and low frequency bandwidth. These are at orientations of 60°, 90° and 120° and produce similar expected classification errors. However for samples (a) and (c) only the 90° filters achieved such a high ranking. Furthermore the error reported is less than a quarter of the magnitude. Using this orientation at a frequency of $\frac{3\pi}{16}$ and $\sigma = \frac{3\pi}{64}$ produces the magnitude of Gabor filters illustrated by the histograms in Figure 5.25. According to the corresponding model a threshold of 7.88 yields the optimal solution for distinguishing between samples (a) and (c). Further analysis shows that the region of the histogram between 5 and 10 in Figure 5.25(b) is a side effect of the white space between paragraphs from Figure 5.24(b). Hence, this filter which is optimal for sample (c) is also

this problem.

applicable for sample (b) and therefore is used for segmentation.



(a) (b) (c)

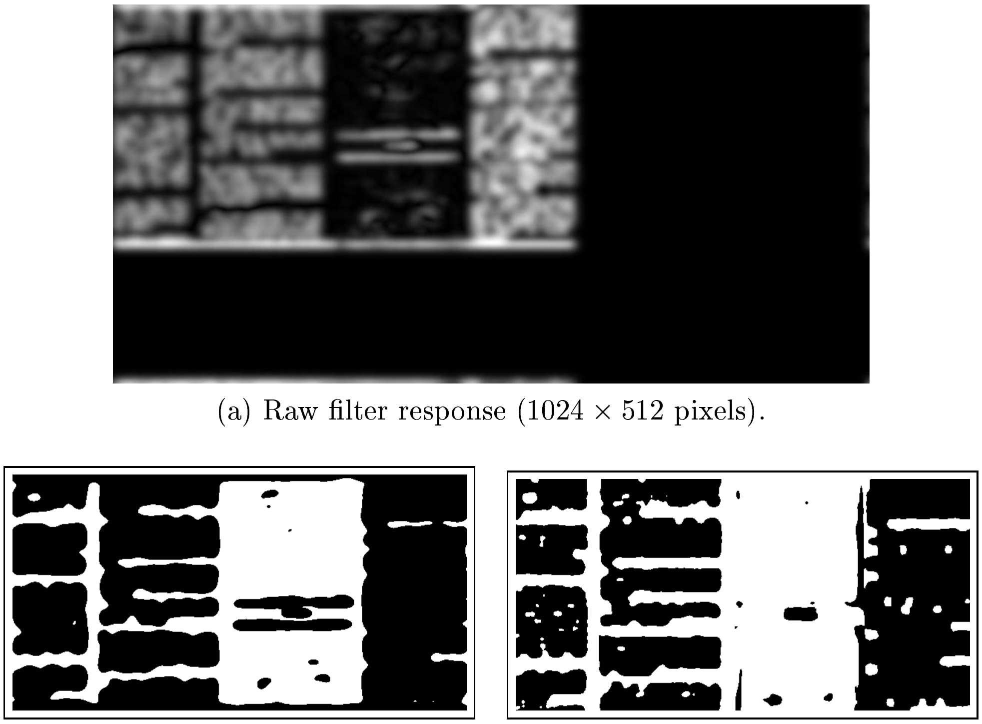
FIGURE 5.25: Histograms of the magnitude of the filter response for the optimal Gabor filter for the prototype in Figure 5.24. The filter is oriented at 90° with a frequency of $\frac{3\pi}{16}$ and frequency bandwidth $\sigma = \frac{3\pi}{64}$.

Figure 5.26(a) illustrates the raw filter response for this optimal filter, which when thresholded with the design threshold of 7.88 forms the image regions corresponding to text and picture in Figure 5.26(b). To compare it to methods presented in this chapter the models for *background*, *edge* and *other* are omitted with the final classification based on the neighbourhoods of likelihoods according to the remaining *text* and *picture* models. This is shown in Figure 5.26(c). Neither result is clearly superior, with the Gabor approach suffering from edge effects, horizontal lines within the image and tending to smooth the boundaries of text regions.

To compare the two schemes more completely the model obtained, or *trained*, on the default image is applied to several other newspaper images. The results, as seen in Figure 5.27, are somewhat subjective. The first example produces excellent results in both cases, although the Gabor filter produces some undesirable edge effects and joins small paragraphs near the top of the image, and the proposed method results in numerous small holes within paragraphs. The identification of the text "k the main target" and "standby rat" (top left image in Figure 5.27) using the proposed method is worth noting. Similar results are obtained from the second image. The proposed method identifies the brand name on the monitor as text as well as the *words* "atures are big.", but contains numerous holes within paragraphs. The final two examples containing more uniform dense text are generally handled better with the proposed method.

The results from Figure 5.27 indicate the effectiveness of both methods. Given an intelligent design and comparable implementations the two require very similar computation time[8]. However the optimal Gabor method is inferior in terms of rotational invariance since it is based on a prototype texture with a defined orientation and possibly well defined texture orientation. This is seen in Figure 5.28 where the same model is applied to

[8] Given the optimal filter parameters, the result in Figure 5.26(a) took 10.68 seconds, $2 \times 4.2 = 8.4$ of which was to calculate an optimised FFT and inverse FFT. The unoptimised proposed method takes less than 15 seconds (the final classification is reduced from 13.9 (Section 5.8.3) to 5.2 seconds when just two models are considered). Given the lack of optimisations in the proposed method, both are considered computationally equivalent.



(a) Raw filter response (1024×512 pixels).

(b) optimal threshold

(c) window based method

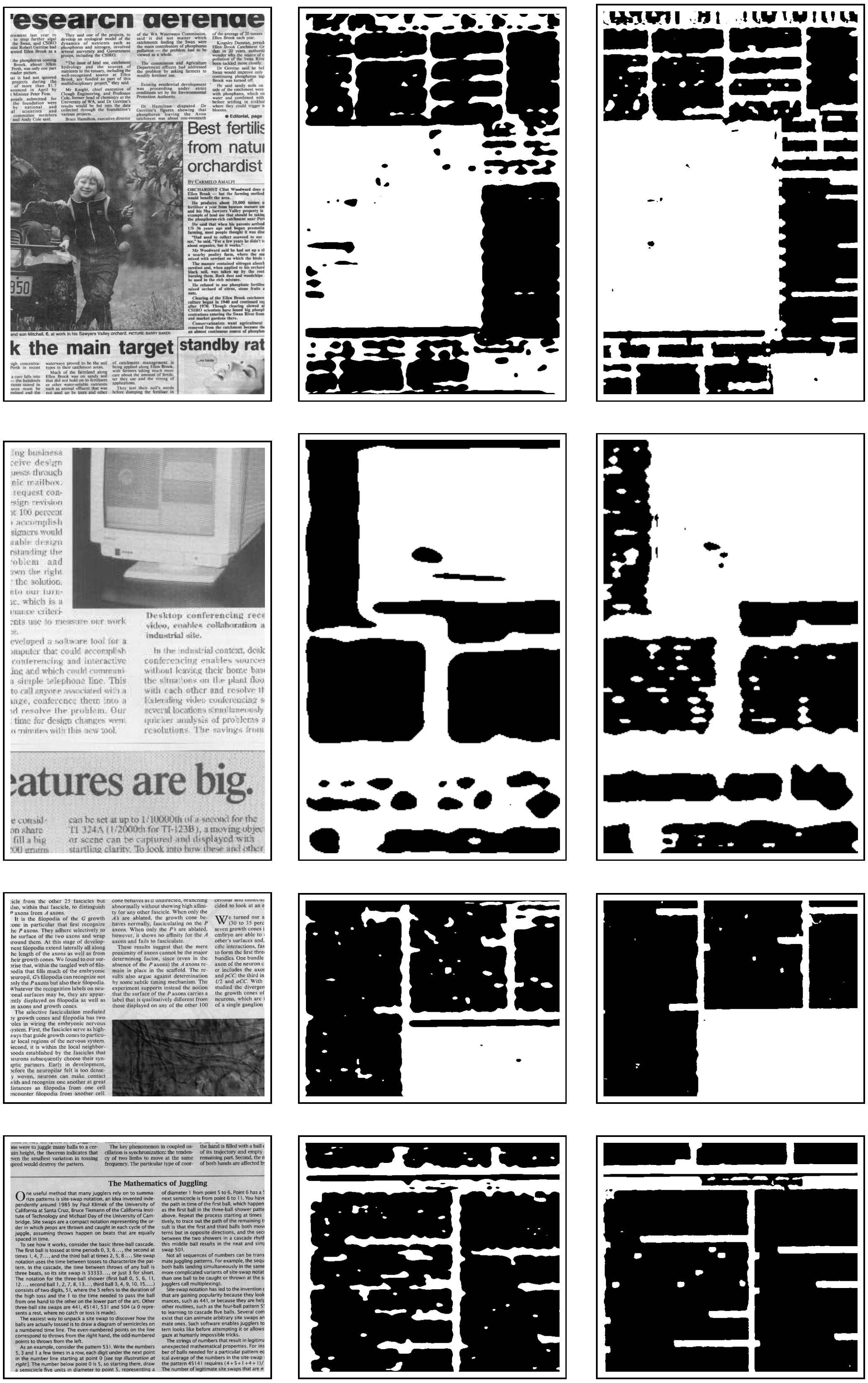
FIGURE 5.26: A comparison between the optimal Gabor filter and a simplification of the method presented in this chapter.

a rotated image.

5.10.2 Summary and additional comments

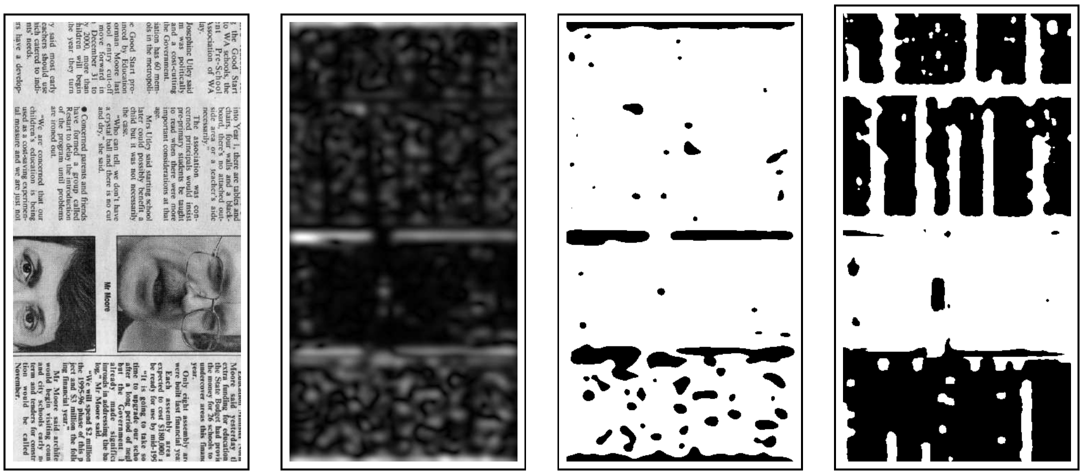
The restricted problem investigated in this chapter uses the same generic feature extraction framework successful for isolating and representing anomalies. The feature space uses the standard deviations of pixel intensity as a window measurement and contains significant structure on scanned newspaper and magazine images. A new tool is proposed which successfully captures this structure by considering a measure of local abnormality and the geometric structure of the feature space. The final two values $(\eta, \mathbf{d})$ provide a feature vector representation Φ which is robust with respect to dimension or equivalently to the blocksize $\mathbf{k}$ which governs the neighbourhood complexity captured by feature vectors. The space Φ has desirable invariances to image brightness, contrast, translation and rotation.

The segmentation of newspaper into *picture*, *text*, *edge* and *background* classes is performed by modelling clusters in appropriate regions of Φ and refining initial classifications based on these models by considering spatially close vectors $\mathbf{p} \in \Phi$. The results over 20 legitimate images are illustrated after a simple heuristic post-processing algorithm which paints picture regions red and text green. Overall these are particularly good considering that the images are from different sources and contain varying features. A simple reconstruction suggests that these segmentation techniques are applicable to other images with



(a) original image (b) optimal threshold (c) window based method

FIGURE 5.27: Results using the same models applied in Figure 5.26.



(a) (b) (c) (d)

FIGURE 5.28: The sensitivity of the optimal Gabor filter to rotation. The optimal Gabor filter is shown in (b), its threshold in (c) and the window based method in (d).

similarly well defined structures.

Two class segmentation using an optimal Gabor filter produces comparable results with respect to computational time and performance on *correctly* oriented images. Even with the proposed method restricted to two class models (*text* and *picture*) it is still arguably superior. To obtain a general Gabor based segmentation possessing desirable features such as rotational invariance, a set of well defined Gabor filters is required along with a more general classification scheme. This would be orders of magnitude more computationally expensive.

In conclusion the method is simple, effective, easily adaptable to changes in image resolution and possesses numerous invariances.