Chapter 3

Approach: The *UpWrite*

As discussed throughout Chapter 1, a general approach is sought which should also be automatic, efficient and able to extract information in a syntactic way. In view of the approaches to natural language and stroke detection outlined in Section 1.1 and discussed in Section 2.8, the framework used is based on the *UpWrite*. Natural language and stroke detection are referred to without citation here.

3.1 The UpWrite

This general process considers data as a set of points in Euclidean space $\mathbb{R}^{\mathbf{N}}$, providing a means for amalgamating points according to structure and representing them in a generally higher dimensional Euclidean space $\mathbb{R}^{\mathbf{M}}$. It is a topological process which goes from an atomic level to a molecular one, thus grouping subsets of points into single higher level entities. The process may be repeated on these higher level entities and is hence hierarchical in nature.

For the purpose of better defining the UpWrite for practical use on region based images it is decomposed into three steps, namely

- **Localisation.** Local measurements or models of data points.
- **Amalgamation.** A method of combining these models into *chunks* based on structure.
- **Representation.** An effective way to represent each chunk as a new, possibly coloured and weighted, data point.

3.1.1 Localisation

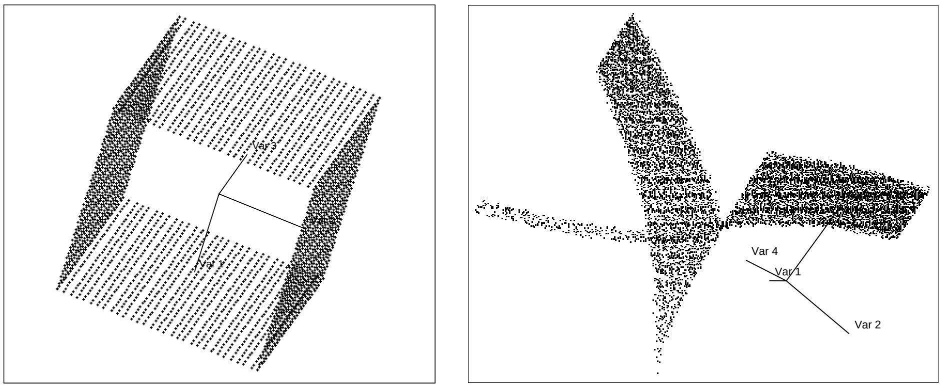
Consider the set of data points $\mathbf{P} \subseteq \mathbb{R}^{\mathbf{N}}$, for example a grey scale image containing points with values for intensity and spatial position. The localisation process aims to extract information by examining *local regions* of this Euclidean feature space. A local region may be defined as all the points within some distance with respect to a chosen or appropriate metric. A simple example is a small circular neighbourhood of defined *resolution radius* surrounding a pixel which uses the standard Euclidean metric.

The localisation process needs in some way to model or represent this region of space for further processing. Often a particular region is unsuitable due to sparseness (lack of data points) or lack of appropriate structure. These regions should be modelled (or labelled) accordingly and possibly omitted from the amalgamation process. For example, considering a binary image and the neighbourhood defined above, each local region consists of a solid region, a line segment, noise, a combination of these, or is void of data points. In this case the line segment and solid region can be appropriately labelled as a one dimensional structure embedded in a two dimensional space and a two dimensional structure, respectively.

Central geometric moments are a useful and simple tool for modelling local regions, providing a measure of density, centre of mass, and spread or elongation along various axes.

3.1.2 Amalgamation

Amalgamation of local models is ideally performed using only the structure inferred from these models. For example the points lying on the surface of a cube in $\mathbb{R}^3$ are highly structured, with local regions corresponding to an embedded two dimensional surface as depicted by Figure 3.1(a). In this case *adjacent* local neighbourhoods form similar distributions of points allowing for each face to be amalgamated into a single object. Many highly structured datasets contain lines, planes or in general arbitrary dimensional manifolds embedded in $\mathbb{R}^{\mathbf{N}}$. Figure 3.1(b) shows 2 two dimensional manifolds (or surfaces) and a single one dimensional manifold (a line)[1].



(a) open cube (b) manifolds in $\mathbb{R}^4$

FIGURE 3.1: The structure of two artificial datasets containing one and two dimensional manifolds embedded in higher dimensional spaces.[1]

[1] Figure 3.1(b) is obtained by mapping $[0,1] \times [0,1]$ and $[0,1]$ into $\mathbb{R}^4$ using mathematical functions and adding small perturbations of noise. There is actually only a single two dimensional manifold, although the functions produce a singularity forming two distinctly apparent structures

In principle, the amalgamation stage should be able to combine local neighbourhoods containing such structures without assuming too much about the dataset. It is important for an algorithm to be able to determine the confidence of a particular grouping and identify when apparent structure is the result of artifacts such as noise. One possibility is to perform analysis at multiple scales and correlate results to distinguish real structure from incidental structures resulting from artifacts. Another is to derive a variety of amalgamation procedures and either choose the *best result* or continue processing along multiple paths. Here the best result is the one which the algorithm is most confident about. It is suggested that the human brain processes visual information in a similar way, performing recognition by examining input at multiple scales and processing in multiple ways.

In practice, the amalgamation process should be simple and assume little about the form of the data. The location of a one dimensional manifold in $\mathbb{R}^2$ or $\mathbb{R}^3$ provides a more general method for finding lines in images than looking for connected pixels for example. If necessary, the process may require training to distinguish acceptable structure (e.g. distinguishing smooth triples from irregular ones [287], Section 2.8.3).

3.1.3 Representation for further UpWrites or classification

Under this framework, amalgamated objects are represented by a single point in a generally higher dimensional Euclidean space $\mathbb{R}^{\mathbf{M}}$ along with an optional weight and discrete label or colour. The label allows objects at any particular level of abstraction to be partially classified or distinguished. For example, the amalgamated manifolds from Figure 3.1(b) could be labelled as one and two dimensional even though the same method of representation is used. However, in this case it may be desirable to include the value as part of the feature vector and let further processing, classification or clustering automatically distinguish between one and two dimensional manifolds.

The final object representation completes the process in which a set of data points is *Up Written* to a single entity. Once all objects from the dataset have been been represented, these entities form a feature set which can undergo further higher level UpWrites through localisation and amalgamation in this new Euclidean space. Hence, the framework is hierarchical with the identification and representation of increasingly higher level (or more abstract) objects.

Ideally, a simple dataset (such as an image containing pixels) is transformed into a single data point which appropriately represents the structure of the image. Hence, in principle, i.e. given sufficiently powerful amalgamation and representation schemes, a set of images of the same class would produce a cluster in the final representation space. For example, 100 images of squares and 100 images of triangles would form two clusters allowing the classification of an unknown image according to its distance from these two clusters. The same applies to pictures of cats and dogs.

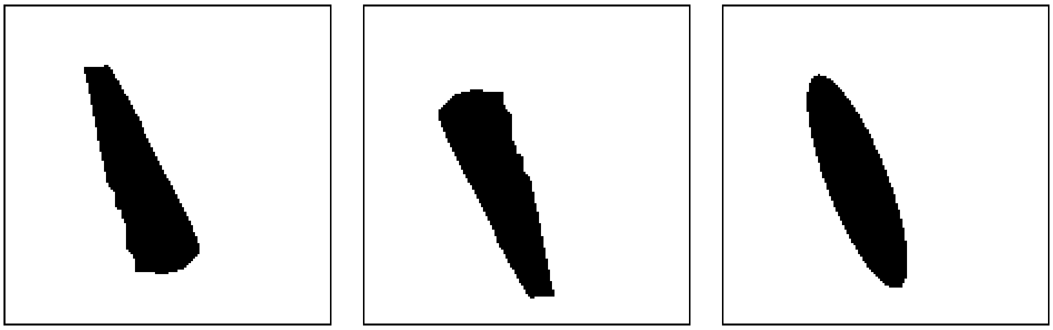
Alternatively it may be more practical to perform a defined number of UpWrites before classifying objects that are sub-parts of the entire dataset. For example, the analysis of stroke images by McLaughlin [285] [8] forms data points representing lines, ellipses or circles for classification. No attempt is made to group multiple objects within the same image into a single image representation since for the examined data and problem it is impractical. However, given the problem of recognising machine parts on a production line, a further UpWrite would be appropriate.

In general the UpWrite provides a framework for bottom-up syntactic analysis. Whether it is more appropriate to say an image is X or that the image consists of objects X, Y and Z is dependent on the particular application.

3.2 The DownWrite

The DownWrite is conceptually opposite to the UpWrite. It takes a single higher level entity and reconstructs an appropriate set of primitives. The DownWrite process is similar to stochastic rewrite grammars used for reconstructing natural language from a language model, such as that used by MegaHAL [279] (see Section 2.8.2).

Consequently, while applying the UpWrite is typically a deterministic process, the DownWrite process involves making model based choices. A choice may be based on a probability distribution, or made to produce the most simple or general set of primitives which agree with the UpWrite representation. For example, given a shape represented by geometric moments, a simple and general DownWrite produces an ellipse with the same number of data points, centroid and with the same orientation and relative spread. Figure 3.2 shows two shapes which have identical low order moments and a reconstructed image using these.



(a) plane wing (b) manipulation of (a) (c) DownWrite

FIGURE 3.2: An example of the DownWrite using up to second order geometric moments to represent the object's shape.

The DownWrite in Figure 3.2 assumes a uniform distribution and contains the *correct* number of points, centroid, orientation and ratio between major and minor axes. It does not however necessarily have the same directional standard deviations as the original. An alternative DownWrite approach, also assuming a uniform distribution, is to ensure that the reconstructed shape has the same centroid and covariance matrix (orientation and deviations) but not necessarily the same number of points. Enforcing all of these constraints to completely satisfy the model's parameters cannot be done using a uniform distribution and hence requires other assumptions. This example illustrates the difficulty of defining an appropriate DownWrite process even for simple representations.

In general, while amalgamating primitives the UpWrite process loses information, so the DownWrite cannot be expected to form perfect image reconstruction. This is analogous to a human examining a shelf containing a few dozen books and then being asked to describe or draw the shelf. It is most likely that the person has stored enough information to draw a shelf with approximately the same number of books and include details such as book height, thickness and perhaps colour. Although the reconstructed image generally differs from the original scene, it does provide an insight into the information captured by the UpWrite process[2]. This is of fundamental importance.

It should further be noted that the ability of the DownWrite to reconstruct an acceptable image does not necessarily imply an unsuitable UpWrite or final representation. This inability may be attributed to failure in the DownWrite process or the inability to define an appropriate DownWrite.

In summary, the DownWrite process uses the UpWrite representation of an object to reconstruct an instance of that object. The process provides a powerful insight into the information captured by the UpWrite, thus providing a higher confidence in an algorithm's performance than is obtainable by experimental classification results. In general, the object resulting from the DownWrite is not unique.

3.3 Discussion

3.3.1 Representation problems

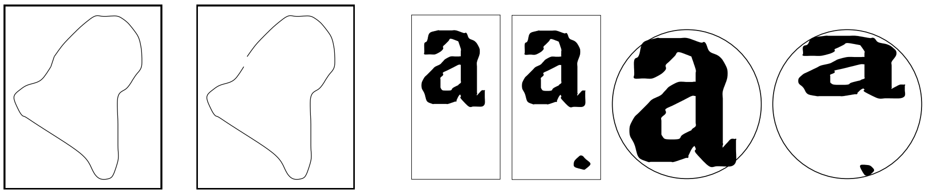
Under the framework of the UpWrite, the data must first be represented as an appropriate set of points in $\mathbb{R}^N$. As discussed in Section 1.1, decomposing images into regions is a two dimensional problem as opposed to natural language analysis and stroke detection which are one dimensional. This makes the initial representation more difficult.

For stroke detection it is sensible and convenient to represent the image as a set of points in $\mathbb{R}^2$. However for grey scale it may be more convenient to extract low level local features for initial representation rather than using the set $\mathbf{I}[\mathbf{i}, \mathbf{x}, \mathbf{y}] \subset \mathbb{R}^3$. Alternatively, this can be considered as the first UpWrite, transforming a local set of points from $\mathbf{P}$ into a higher level representation using a trivial predefined amalgamation scheme.

Regardless, a robust representation is required at each hierarchical level to ensure similar objects are not misclassified or deemed significantly different due to noise. Two example datasets are shown in Figure 3.3. In (a) the number of data points provides a better feature or representation than the length of the external perimeter. The second

[2] See also Section 1.3.

example illustrates the sensitivity of Zernike moments to the fitting of a unit disc, causing similar looking objects to have vastly different representations. In this case geometric moments are more robust with respect to noise given an already identified object.



(a) similar shapes (b) similar characters fitted with a unit disc

FIGURE 3.3: Data for which robust and non-robust representations are available.

The appropriateness of the representation governs the success of classification. Given that data is transformed from one Euclidean space to another, it is desirable for small perturbations in the data at one level to correspond to smooth transitions in the representation or feature space at the next.

3.4 Summary of approach

The framework of the UpWrite is more of a philosophy than an approach. However it does agree with both the desired properties and the objectives outlined in Section 1.1.3 and 1.4 respectively. In principle it is generic, automated, hierarchical and syntactic and potentially computationally viable. Furthermore it provides a means for classification, recognition and image reconstruction for algorithm insight.

Considering this methodology, the problem of decomposing images into sub-images is tackled by addressing a number of simplified or restricted problems, namely

- the detection and classification of anomalies in homogeneous grey scale images.
- the segmentation of mixed-mode grey scale images into predetermined regions.
- the segmentation of coloured real-world scenes into homogeneous regions.

To illustrate the common framework each restricted problem is presented in a chapter using similar section headings. This adds to the effect of drawing a parallel between different areas of work. To make the headings more interesting and specifically informative, variations of the following heading names are used.

1. **Context and simplifications:** Restrictions and general assumptions.
2. **Example data:** Example data; its purpose and applications.
3. **Local feature extraction:** Low level representation.

4. **The feature space:** Examination of structure, measurements and parameterisation of features, but not amalgamation.

5. **Object identification:** Amalgamation of features into objects.

6. **Object representation:** Object descriptions and robust representation.

7. **Higher level processing:** Additional processing, further UpWrites and post-processing.

8. **Classification:** Final object classification and results.

9. **DownWrite:** Image reconstruction as an insight into processing and appropriate representation.

10. **Comparisons and conclusions:** Examination of appropriate standard techniques, summary of results and general conclusions.