Active Shape Structural Model

Dissertation

zum Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg
von

M.Sc. Stephan Al-Zubi
geboren am 3. März 1973 in Syrien

Gutachter:
Prof. Dr. Klaus-Dietz Tönnies
Prof. Dr. Siegfried Stiehl
Prof. Dr. Xiaoyi Jiang

Ort und Datum des Promotionskolloquiums:
Magdeburg, 18. November 2004

To my mother...

Acknowledgments

I would like to thank very much Fitsum Admasu and Beate Traoré for reviewing my thesis and their continuous support during my writing. I also like to thank my best friends Ahmed Ghoneim and specially Moftah Al-Zobi for their help and support.

Abstract of the Dissertation
by
Stephan Al-Zubi

Abstract

This thesis proposes a new shape model called the Active Shape Structural Model (ASSM). The ASSM combines both statistical and structural a-priori knowledge about shape variation. The statistical a-priori knowledge models co-variations between two or more parts of the shape structure (e.g. co-deformation, joint articulation). The structural a-priori knowledge specifies which structural parts can be statistically related.

The a-priori knowledge enables the ASSM to model a larger class of problems than structural or statistical models alone. Pure statistical models would have to use a complex distribution function to model shapes consisting of articulated parts like the human body. Pure structural models can decompose complex shapes into parts but cannot validate this decomposition against the allowed co-variations between those parts.

Combining both structural and statistical a-priori knowledge results in interesting properties of ASSM such as multi-resolution of part variation depending on its context, completing missing structures and resolving conflicting interpretations using the shape's largest context.

These properties of ASSM are demonstrated on two applications: Sketch recognition and ant recognition. Sketches demonstrate ASSM well because they have clearly defined structures that exhibit statistical variation for a single user, multiple users and depending on the co-variation with other parts in the sketch. The structural co-variation between multiple users was used in a new application called biometric recognition algorithm. In this case the structural relationship between drawing primitives are used as the secret information between the user and the biometric system. Experiments show that the ASSM can utilize well it's prior knowldge in recognizing, and correcting sketches as well as achieving good discrimination between users in biometric sketches. The ASSM was compared to a pure statistical represntation and shown to be capable of effeciently reprsentating valid states of training data. After demonstrating the ASSM framework within the domain of online sketches, it was next used for ant segmentation. This is because ants both have articulated parts and different structural templates are needed to represent different ant types. Experiments show that co-variation between parts can be succesfully used for both template selection and finding effeciently the articulated parts. All these applications show that utilizating prior knowledge in the form of co-variation between shapes templates can lead to a better repersntation, reconstrcution, recognition, and correction of shapes.

Kurzfassung

Diese Dissertation stellt ein neues Shape-Modell - ACTIVE SHAPE STRUCTURAL MODEL (ASSM) genannt - vor. Beide Formen des a priori Wissens über Shape-Variationen, statistisches und strukturelles, werden in ASSM vereint. Das statistische a priori Wissen bildet Co-Variationen zwischen zwei oder mehreren Teilen der Shape-Struktur (d.h. Co-Deformation, Gelenkverbindung) nach. Das strukturelle a priori Wissen benennt die strukturellen Bestandteile mit statistischem Bezug.

Das a priori Wissen befähigt ASSM zur Bearbeitung einer breiteren Problematik als lediglich struktureller oder statistischer Modelle. Rein statistische Modelle müssten eine komplexe Distributionsfunktion verwenden, um Shapes darzustellen, die aus Gelenkteilen bestehen wie der menschliche Körper. Rein strukturelle Modelle können komplexe Shapes in Teile zerlegen , nicht aber die Zerlegung gegenüber den gestatteten Co-Variationen zwischen diesen Teilen bewerten.

Die Kombination von strukturellem und statistischem a priori Wissen führt zu interessanten Eigenschaften von ASSM wie Mehrfachauflösungen von Teilvariationen in Abhängigkeit vom Kontext, der Vervollständigung fehlender Strukturen und der Analyse widersprüchlicher Interpretationen unter Verwendung des umfangreichsten Shape-Kontextes.

Diese Eigenschaften des ASSM werden anhand zweier Anwendungen demonstriert: Skizzenerkennung und Ameisenerkennung. ASSM wird durch Skizzen sehr gut dargestellt, da diese klar definierte Strukturen besitzen, die statistische Variationen für einen Einzelnutzer, mehrere Nutzer / Mehrfachnutzer und abhängig von den Co-Variationen mit anderen Teilen der Skizze aufweisen. Die strukturelle Co-Variation zwischen mehreren Nutzern fand Anwendung als biometrischer Erkennungsalgorithmus. In diesem Fall wird das strukturelle Verhältnis zwischen geometrischen Objekten als Geheiminformation zwischen dem Nutzer und dem biometrischen System verwendet. Experimente zeigen, dass ASSM sein Vorwissen sowohl zur Erkennung und Korrektur von Skizzen gut nutzbar machen kann als auch ein gutes Unterscheidungsvermögen zwischen den Nutzern in biometrischen Skizzen erreicht.

ASSM wurde mit einer rein statistischen Darstellung verglichen und erwies sich als leistungsstark bei der Präsentation gültiger Versuchswerte. Nach der Präsentation des ASSM - Systems innerhalb der Domain der Online-Skizzen, wurde es als Nächstes bei der Erkennung von Ameisen eingesetzt. Das ergibt sich, da Ameisen einerseits über Gelenke verfügen und andererseits unterschiedliche Struktur-Templates erforderlich sind, um verschiedene Kategorien von Ameisen darzustellen. Experimente zeigen, das Co-Variationen zwischen Teilen erfolgreich zur Auswahl eines Templates und dem rationellen Auffinden der Gelenkteile genutzt werden können. All diese Anwendungsmöglichkeiten zeigen, dass die Verwertung von Vorwissen in Form von Co-Variation zwischen Shape-Templates zu einer verbesserten Präsentation, Rekonstruktion, Erkennung und Korrektur von Shapes führen kann.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Computer vision is an interdisciplinary research field concerned with enabling machines to understand visual data as humans can. Although a lot of progress has been achieved in the past decades, this field has fallen short in defining a generally applicable human vision theory. However, several successful solutions for specific applications have emerged in various fields such as motion tracking, medical image processing, biometry, robot vision.

Generally the image understanding process consists of two steps as depicted in fig. 1.1: Feature extraction and analysis [36].

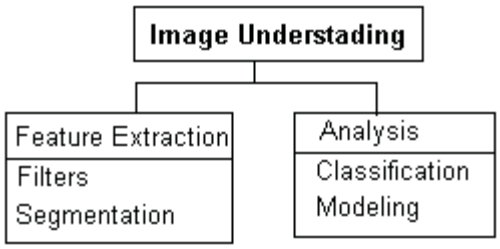


Figure 1.1: Image understanding process

In the feature extraction step, the image is subjected to some transform in order to extract useful information. This is where various filters can be applied to detect and enhance specific features such as edges, corners, and texture and to suppress noise. In addition to image filtering, segmentation methods group connected regions of the image with some local homogeneity criteria. Both filtering and segmentation enable the extraction of image features to support image understanding.

After filtering and segmenting the image, the second step of image understanding is the analysis which is finding semantic interpretations of the extracted features. At this step prior knowledge about the problem is essential. The prior knowledge is embedded in two ways:

1. A classifier is applied to extracted features to make a decision about a specific object in the image. For the classifier to work successfully, a good feature selection is a necessary condition.

2. A model instance is applied to the image. The model is a 2D or 3D representation of an image object which defines allowable constraints and variations of that object.

When applying a model to the image, a fitting process brings the model as close to the image object as possible. The advantage of models is that they can find relevant image features and exclude features that result from noise and other artifacts.

Feature extraction is computing relevant features in an image. There are important image patterns used for feature extraction:

1. Texture: a texture is a repeating pattern within an image region.
2. Shape features: the geometry of an object usually is described as a spatial distribution of landmarks and image attributes. It is the main feature used for object recognition and comparison [76].

Texture usually describes a connected region in image space with homogeneous properties. Textures alone cannot convey the information necessary to understand the image. To get the full information we must look at the complement of textures. These are the object boundaries where discontinuities occur. The topology of the boundary defines the shape of an object.

The shape of an object can be either a continuous surface in 3-D space or a continuous 2-D curve. The problem with 3-D objects is that their shapes change according to the view point when projecting them on a 2-D image plane. This general problem can be simplified by ignoring the 3rd dimension in some applications like satellite images or assembly lines [7]. In such cases the object does not change its shape much or a constant view is always assumed. In these cases the silhouettes of objects contain sufficient information for recognition. By reducing a 3-D surface to a 2-D boundary we can attain a simplification that makes modeling of these shapes less complex. This is the approach used in this thesis.

There were several solutions proposed to model shapes such as: principal warps by Bookstein [11], hierarchical cylinders by Marr [49], Fourier descriptors [79] and many other approaches which all have in common that a global measure is defined thought to be capable of representing a-priori knowledge. Localization is done by multi-resolution (i.e. frequency decomposition) such as multi-scale medial axis by Pizer [34, 41]. The main purpose of such approaches is classification (i.e. the approach is successful if sufficient information is represented for different shape classes). Feature vectors for classes may also be generated for deterministic shapes (e.g. airplanes), however, model based approaches offer more advantage because it can fit a set of plausible shapes and offer a complete characterization of the fitted shapes. Model based approaches imply classification and not the other way around. The focus of this thesis will be on the model based approach and not classification.

When looking at shapes in general we can recognize two aspects which we will call morphology and structure. Morphology is concerned with the degree of shape variations. These variations can be due to noise, object deformation or change of view point. As an example of that, consider the variation of rigid shapes like bones between different individuals. We may also have soft shapes like muscles that change in time.

The structural aspect of shapes means that a shape can be naturally divided into parts. This implies an abstraction of shape into a simplified representation, which makes comparison between shapes easy. In general shapes can be classified by structure into three categories:

1. Shapes with non-deterministic structure, for example, cancer lumps in digital mammography.
2. Shapes with a fixed structure, for example, the Corpus Callosum in the human brain which has basically the same structure across human population.
3. Shapes that have variable structure. This means shapes can vary in types of parts and the relations between them. For example, mechanical assemblies.

Generally shapes with non-deterministic structure are processed by shape descriptors which extract feature vectors measuring attributes of the whole shape like area, compactness, fractal dimension, etc. [76, 33]. These vectors are then classified to determine useful interpretations, for example, if a tumor in a mammogram is malignant or not. Also, shape models can be applied to those shapes to determine shape boundary. In such cases the model uses built-in smoothness constraints to constrain the search for the correct boundary. The smoothness assumption is used to find optimal boundaries in the presence of noise and to track moving boundaries.

Shapes with a fixed structure are usually processed by an alignment shape model (also called registration). An alignment shape model assumes that there is a model which has to be warped onto the image [33, 76]. The warping process can be done using built-in smoothness constraints or by statistical analysis of shape samples to find the main modes of shape variation. Statistical analysis of deformation is superior to smoothness constraints because it fits the shape in the direction of maximum deformation. This feature of statistical methods results in noise robustness and stability. The problem with statistical methods is that they require a large sample space to accurately model the variation depending on complexity of object shape. If no sufficient sample space is available for fitting, smoothing constraints provide the continuity conditions needed.

Shapes with variable structures [76, 33] are represented with a model which depicts structure in the form of a graph. Graph theoretic methods can be used to match and compare shapes. In some models structural constraints are represented using some form of grammar such as tree or graph grammar. The abstraction of a shape to a graph makes shape comparison and recognition easy even in cases where certain structures are added or missing.

To compare objects we need a similarity measure. The similarity measure must utilize both structural and morphological information between shapes. This is because structural measures alone may not distinguish between shapes of same structure but different morphology. The opposite is also true. Additionally, structural knowledge can be used to correct noisy or erroneous morphology and vice versa. This kind of informa-

tion correction motivates the use of a new shape model that can utilize both types of information.

When looking at the literature of shape models, we can roughly characterize them into those that represent morphology and those that represent structure. These models can also be characterized as those that apply some specific prior knowledge (model) about the shapes they are trying to find and those that do not.

Morphological shape models can roughly be subdivided by their use of prior knowledge into statistical and dynamic models. Statistical models can acquire prior knowledge using representative training samples. They determine the way shapes change from these samples and then try to fit new shape instances using that knowledge. On the other side,dynamic models try to fit shapes by a time-evolving boundary that moves based on some physical model that may not directly be related to the shapes themselves. The physical model is used in these cases for two purposes: To maintain a noise-robust smooth boundary and to represent changes in shape in a coarse-to-fine manner.

The main advantage of statistical models is that no assumption about prior knowledge is needed. All the prior knowledge can be automatically acquired by training samples.

Structural shape models can also be divided by their use of prior knowledge. The structural models that use prior knowledge are able to describe combinatorial constraints between shape parts. This means a specification of connectivity relations between part types. The part connections are represented usually by means of a shape grammar or a graph. Structural models that do not use prior knowledge of part-connectivity are usually concerned with abstracting a complex shapes into a set of simple connected parts. Geons and generalized cylinders [49] are capable of dividing a shape into simple representations of components.

This thesis tries to look at some interesting and representative shape models in each category of the mentioned shape models rather than doing a complete survey of all these models. The reason for doing that is to get a good idea of how these methods generally work and more importantly in order to understand the role prior knowledge plays in morphological and structural models. This gives rise to the idea that we can find a shape representation that can apply prior knowledge both at structural and morphological levels. The main point is that a shape model that embeds prior knowledge at both structural and morphological levels can represent shapes that models in either category cannot do. Some illustrative applications will show and explain this point.

The main questions addressed by this thesis are:

1. Can we find a shape representation capable of modeling both quantitative and qualitative features? Specifically, can we find a shape model capable of capturing both statistical deformation and structural aspects of shape?

2. What advantages does such a shape model offer that other models do not?

3. What class of applications and problems can this shape model be applied to and to what extent can it be successful?

4. What implications does shape noise have on the correction capabilities of the structural and statistical representation of shape?

This thesis will introduce a new shape model called Active Shape Structural Model (ASSM). The main goal of ASSM is to find a shape representation that adds non-statistical predetermined prior knowledge to statistical knowledge learned automatically. This addition enables ASSM to generalize the class of shape representations where only structural knowledge is predefined. The ASSM represents shapes by fusing knowledge of both structural and statistical shape features. In the ASSM we define structural features of a shape as the abstraction of the shape to a connected set of sub-shapes. The structure of a shape can vary by varying the connections between these sub-shapes or by varying the sub-shapes themselves as depicted in fig. 1.2.

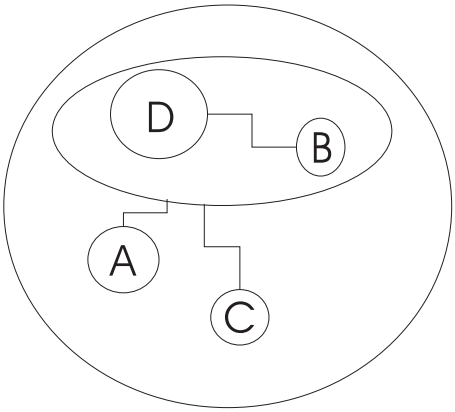


Figure 1.2: Representation of a hierarchical structure of a shape where letters represent some atomic sub-shape and circles and links represent groupings and relations between them

We define the statistical variations of a shape as the elastic variations of a fixed structured shape. Having a fixed structure makes it possible to find corresponding landmarks in a population of shape samples and therefore align them for analysis as depicted in fig. 1.3.

Figure 1.3: Elastic variation modes of a fixed structured shape obtained by principal component analysis

ASSM represents the elastic covariations between sub-shapes and how they combine together. This enables the statistical representation of a shape within its context of connected shapes. Therefore, the ASSM represents shapes in a hierarchical and multi-

resolution (of deformation) manner. ASSM applies prior knowledge about reconstructed shapes by testing allowable combinations of sub-shapes and their elastic covariations. This prior shape knowledge is applied in a bottom up fashion where contextual shape information is used to eliminate false interpretations and recognize bigger shapes.

This thesis is structured as follows: In Chapter 2 the state of the art in shape models is surveyed. Following the survey, comparisons between these models are made to demonstrate the gap that ASSM can fill in shape representation.

Chapter 3 contains the method description and how it can be adapted to various problems.

Chapter 4 of the thesis will demonstrate applications of ASSM in the domains of sketch recognition and biometrics. Sketches were chosen as applications because they are rich in structural and statistical knowledge. We can use this structural knowledge as a secret information between the user and the computer therefore defining a new authentication algorithm for biometry.

Chapter 5 will discuss the ASSM model and its properties and it will be compared to the other models surveyed in chapter 2.

The thesis concludes by a discussion of these experiments and future applications planned.

2 State of The Art

A large number of shape representations has been developed in recent years for different applications. Some of these applications are:

- Segmentation (e.g., segmentation of blood vessels).
- Registration of medical images (e.g., creating a brain atlas).
- Motion tracking (e.g., tracking lip movement).
- 3D stereo vision (e.g., finding the corresponding edges on the left and right images).
- 3D surface reconstruction and representation (e.g., reconstruction 3D objects from there 2D projections).
- Content based image retrieval (e.g., image database for retrieval of silhouettes of marine life).
- Shape similarity (e.g., using medial axes of shapes to compare their structural similarity).
- Computer graphics and shape synthesis (e.g., L-systems used to generate plant growth patterns).
- Multi-resolution representation of shape (e.g., progressive meshes that can morph between different resolutions of a polygonal mesh).

Generally shape models can be divided into four classes as depicted in fig. 2.1:

- Statistical models
- Structural models
- Dynamic models
- Hybrid models

Statistical models use statistical methods on a population of shape samples to derive variation modes of these shapes. This enables them to predict variations of new samples which gives them robustness to noise.

Structural models are more concerned with abstracting the shape into a simpler graph of structural features. This enables easy comparison of shapes.

Dynamic models are used for segmenting and tracking boundaries and have the ability to evolve in time while maintaining smoothness and continuity constraints which insure an optimal segmentation.

Finally, hybrid models combine features of two more shape classes.

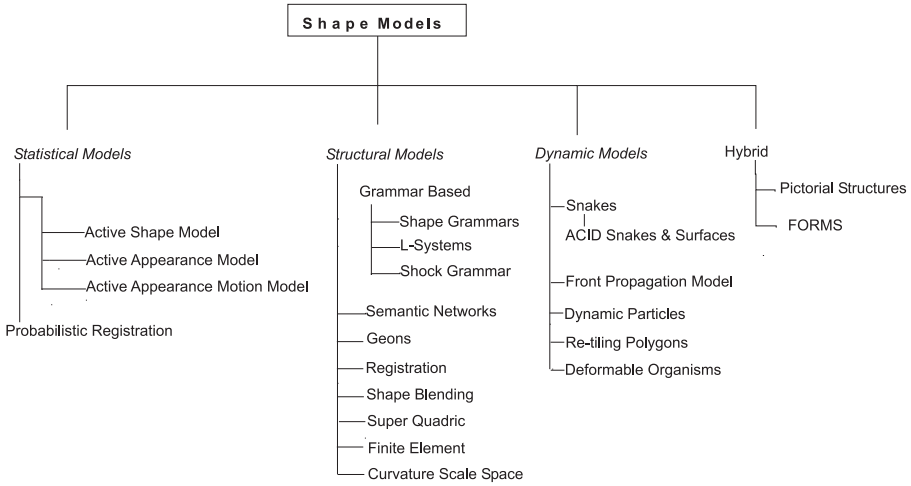


Figure 2.1: Classification of shape models

In the following sections the most important shape models of each class are summarized. These models will be compared in the next chapter. This comparison will motivate the Active Shape Structural Model (ASSM).

2.1 Statistical Models

The main property of this class of shape models is that they statistically describe variations of shape features and textures. They require a large sample space depending on the shape which may not be always available in all applications. Given that a sufficient number of samples are acquired, these samples must be aligned to a standard reference frame. This alignment eliminates all the extrinsic differences between samples leaving only the intrinsic variations. The alignment is mainly done manually by marking corresponding shape landmarks. Automatic landmarking is possible by defining a standard object centered coordinate system for each object [64, 28, 56].

The following sections describe these statistical shape models progressing from the simplest to the most complicated. Active shape Model (ASM) is concerned only with point distributions of landmarks. Active Appearance Model (AAM) adds texture features to ASM. Active Appearance Motion Model (AAMM) adds analysis in time to AAM. Probabilistic registration represents the local variation of gray values and shift vectors for registering brain volumes per each voxel.

2.1.1 Active Shape Model (ASM)

Cootes [20, 21, 25, 26] describes a statistical model which represents a fixed shape with a set of landmarks. The landmark coordinates constitute a point distribution model (PDM). An optimization is performed using gradient decent on principal component analysis of shape variations. This guides the optimization in direction of maximum change.

Applications Fitting human faces, 2D images of the knee cartilage, corpus callosum [25].

Method Given a population of m shape instances where each instance has n landmark points, we define the shape vector of a shape instance i as $\mathbf{x}_i = (x_1, y_1, x_2, y_2 \dots x_n, y_n)^T, 1 \leq i \leq m$. We align every instance $\mathbf{x}_i$ to an initial instance $\mathbf{x}_0$ that minimizes the distance $D = \|\mathbf{x}_i - \mathbf{x}_0\|$. This aligns all instances to a common reference frame eliminating translation and rotation (also scaling if we set $\|\mathbf{x}_i\| = 1$). The mean shape $\bar{\mathbf{x}}$ and covariance matrix S are computed as follows

$$\bar{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \tag{2.1}$$

$$S = \frac{1}{m-1} \sum_{i=1}^{m} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{2.2}$$

The first t eigenvectors $\phi_i, i = 1 \dots t$ and their corresponding eigenvalues $\lambda_i, i = 1 \dots t$ are computed from S. Using the matrix $\mathbf{\Phi} = [\phi_1, \phi_2 \dots \phi_t]$, we can approximate any shape instance $\mathbf{x}$ as

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{b} \tag{2.3}$$

$\mathbf{b}$ is a t-dimensional vector that represents the parameter space of the deformable model. The number of eigen values t is chosen to explain the desired percentage of shape variation. The values of $\mathbf{b}$ are constrained within a specified valid parameter space. For example, we can constrain $b_i \leq 3\sqrt{\lambda_i}, 1 \leq i \leq t$ to be the space of values which do not deviate by more than $\pm$ 3 standard deviations along each mode of variation as depicted in Fig. 2.2.

The measure of fit between the model $\mathbf{x}$ and the image $\mathbf{y}$ is the gradient image profile along a line orthogonal to the boundary passing the landmark point as depicted in Fig. 2.3.

The active shape model is a search algorithm that fits the model to a new shape instance. It is defined as follows:

1. Initialize $\mathbf{b}$ to zero.

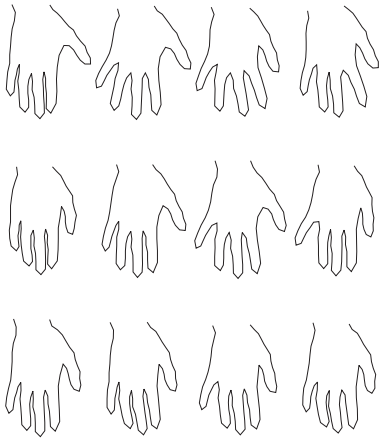


Figure 2.2: Effect of varying the first three hand shape parameters between ± 3 standard deviations (from [19])

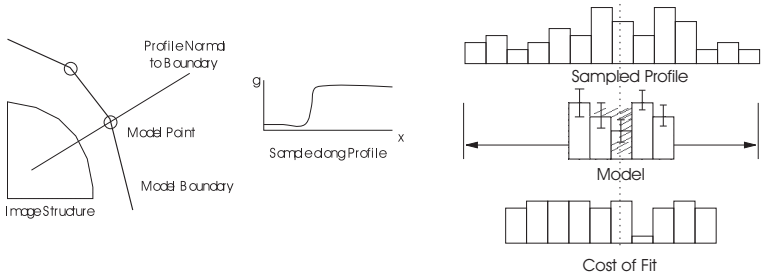


Figure 2.3: Sampled profile of gradient values on an orthogonal line to the landmark boundary (from [19])

2. Generate the model instance $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi b}$.

3. Optimize the rigid body transform parameters $\mathbf{\Theta}_{rigid} = (\Delta x, \Delta y, \theta)$ and scale parameter s for the best fit between $\mathbf{x}$ and $\mathbf{y}$.

4. Find $\mathbf{b}$ that makes $\mathbf{x}$ best fit the image $\mathbf{y}$ satisfying the constraints on $\mathbf{b}$.

5. If not converged return to step 2.

ASM can be adapted for multi-resolution search as follows: A Gaussian pyramid is constructed from the image $\mathbf{y}$. The active shape model is run from coarse to fine levels where the model intensity profile of each level is compared with the image at that resolution. Coarser levels will compare profiles which span more of the original image thus reducing the likelihood of getting stuck at local minima.

2.1.2 Active Appearance Model (AAM)

Cootes [42, 23, 30, 24, 22] extended the active shape model to include texture information of the whole image instead of just the edge profile along landmark.

Applications Segmentation of faces.

Method Initially image instances are landmarked. All instances are aligned to a reference image by ASM warping the grey level information to a standard frame. The aligned images are sampled to form a texture vector $\mathbf{g}$. This vector is normalized to zero mean and 1 standard deviation to eliminate illumination artifacts.

The texture vector $\mathbf{g}$ is modeled using principal component analysis as

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g \tag{2.4}$$

where $\mathbf{P}_g$ are the orthogonal modes of variation and $\mathbf{b}_g$ are texture parameters.

The shape parameters $\mathbf{b}_s$ from the active shape model and the texture parameters $\mathbf{b}_g$ are concatenated into one vector $\mathbf{b}$ as follows

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} \tag{2.5}$$

$\mathbf{W}_s$ is a diagonal matrix of weights. It compensates for the difference in weights between shape and texture parameters. The RMS change in $\mathbf{b}_g$ per unit change in the shape parameter $\mathbf{b}_s$ gives the estimated weight $\mathbf{W}_s$ applied to $\mathbf{b}_s$. By applying PCA on the vector $\mathbf{b}$ that has zero mean we get

$$\mathbf{b} = \mathbf{P}_c \mathbf{c} = \begin{pmatrix} \mathbf{P}_{cs} \\ \mathbf{P}_{cg} \end{pmatrix} \mathbf{c} \tag{2.6}$$

This allows us to express shape and texture parameters directly in terms of $\mathbf{c}$ as

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{W}_s^{-1} \mathbf{P}_{cs} \mathbf{c} = \bar{\mathbf{x}} + \mathbf{Q}_s \mathbf{g} \tag{2.7}$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{P}_{cg} \mathbf{c} = \bar{\mathbf{g}} + \mathbf{Q}_g \mathbf{c} \tag{2.8}$$

The measure of fit between the model and the sample image is the difference in grey value between the image $\mathbf{I}_i$ and the model $\mathbf{I}_m$.

$$\Delta = \|\mathbf{I}_i - \mathbf{I}_m\| \tag{2.9}$$

The active appearance model is a search algorithm that fits the model to the image. It is derived as follows: Given the model parameters vector $\mathbf{p}$ that includes $\mathbf{c}$, rigid body

and image intensity parameters, define the residual error vector $\mathbf{r}(\mathbf{p}) = \mathbf{g}_s - \mathbf{g}_m$ where $\mathbf{g}_s$ is the warped image under the parameter space $\mathbf{p}$. Using Taylor expansion we can write

$$\mathbf{r}(\mathbf{p} + \partial\mathbf{p}) = \mathbf{r}(\mathbf{p}) + \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\partial\mathbf{p} \tag{2.10}$$

where the ij^{th} element of the matrix $\frac{\partial \mathbf{r}}{\partial \mathbf{p}}$ is $\partial r_i / \partial p_j$.

By equating $\mathbf{r}(\mathbf{p} + \partial\mathbf{p})$ to zero we get the RMS solution

$$\partial\mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p}) \tag{2.11}$$

$$\mathbf{R} = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T \quad \frac{\partial \mathbf{r}}{\partial \mathbf{p}} \right)^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T \tag{2.12}$$

$\mathbf{R}$ is assumed to be constant and estimated by averaging its value from a training sample of several typical images. Using $\mathbf{R}$ and the residual error $\mathbf{r}$ a gradient descent algorithm is applied until Δ falls below a defined threshold.

2.1.3 Active Appearance Motion Model (AAMM)

This is an extension of AAM to which the time dimension is added. This is achieved by concatenating shape and texture vectors at specific time landmarks.

Applications Time sequences of cardiac images of the left ventricular area (2D) using MR and ultrasonic sequences (echocardiograms).

Method In [12, 53] a time sequence of the left ventricle area of the heart is normalized to 16 frames such that the end-systolic and end-diastolic frames map to the same frame number. The stack of those 2D images is considered to be a single data sample. The landmark coordinates and grey level vectors for all the time phases are concatenated to form a single vector used for active shape model.

$$\mathbf{x} = [\underbrace{x_{11}, y_{11} \ldots x_{1n}, y_{1n}}_{phase1}, \underbrace{x_{21}, y_{21} \ldots x_{2n}, y_{2n}}_{phase2} \cdots \underbrace{x_{N1}, y_{N1} \ldots x_{Nn}, y_{Nn}}_{phaseN}] \tag{2.13}$$

$$\mathbf{g} = [\underbrace{g_{11} \ldots g_{1n}}_{phase1}, \underbrace{g_{21} \ldots g_{2n}}_{phase2} \cdots \underbrace{g_{N1} \ldots g_{Nn}}_{phaseN}] \tag{2.14}$$

The Active Appearance Model is applied the usual way to $\mathbf{x}$ and $\mathbf{g}$. This results in a time-continuous segmentation for the complete cardiac cycle.

2.1.4 Probabilistic Registration

Chen [18] proposes to learn statistically the variation between different brain volumes. This variance is the a-priori knowledge used to improve future registrations. This goes beyond the smoothness constraints usually found in other registration algorithms.

Applications Elastic registration of brain volumes.

Method The author eliminates extrinsic (rotation and translation) and intrinsic (elastic) differences between a training set of volumes. He then defines two Gaussian distributions:

- Density variations: For every *voxel* in the atlas, the gray level distribution is defined for the corresponding voxel in each registered sample as shown in fig. 2.4

$$P(\Delta I \mid \mathbf{D}) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(\Delta I-\mu)^2}{2\sigma^2}} \tag{2.15}$$

  where $\Delta I = I_s - I_a$ is the voxel intensity difference between the subject and the atlas respectively. $\mathbf{D}$ is the 3-D deformation vector. μ is the mean intensity difference between the atlas and the subject at that voxel and σ^2 is the variance.

- Geometric variations: For every voxel the 3D shift vector is modelled as a multivariate Gaussian distribution as depicted in fig. 2.5

$$P(\mathbf{D}) = \frac{1}{\sqrt{(2\pi)^3 \mid \mathbf{\Phi} \mid}} e^{-\frac{(\overrightarrow{\Delta \mathbf{v}}-\overrightarrow{\omega})^T \mathbf{\Phi}^{-1} (\overrightarrow{\Delta \mathbf{v}}-\overrightarrow{\omega})}{2}} \tag{2.16}$$

  where $\overrightarrow{\Delta \mathbf{v}}$ is the 3D displacement between sample and atlas, $\overrightarrow{\omega}$ is the mean 3D displacement and $\mathbf{\Phi}$ is the 3x3 covariance matrix.

Registration is defined as finding $\mathbf{D}$ for every voxel that maximizes the posterior probability $P(\mathbf{D}|\Delta I)$. Using Bayes rule we can write

$$P(\mathbf{D}|\Delta I) = \frac{P(\Delta I|\mathbf{D})P(\mathbf{D})}{P(\Delta I)} \tag{2.17}$$

Maximizing $P(\mathbf{D}|\Delta I)$ is equivalent to minimizing the Mahanalobis distance

$$\frac{(\Delta I-\mu)^2}{2\sigma^2} + \frac{(\overrightarrow{\Delta \mathbf{v}}-\overrightarrow{\omega})^T \mathbf{\Phi}^{-1} (\overrightarrow{\Delta \mathbf{v}}-\overrightarrow{\omega})}{2} \tag{2.18}$$

By taking the first derivative of eq. 2.18 with respect to 3D displacement we get the gradient vector for the gradient descent algorithm that tries to predict the direction of deformation

$$\overrightarrow{\nabla} = \frac{\Delta I-\mu}{\sigma^2} \overrightarrow{\nabla I} + \mathbf{\Phi}^{-1} (\overrightarrow{\Delta \mathbf{v}}-\overrightarrow{\omega}) \tag{2.19}$$

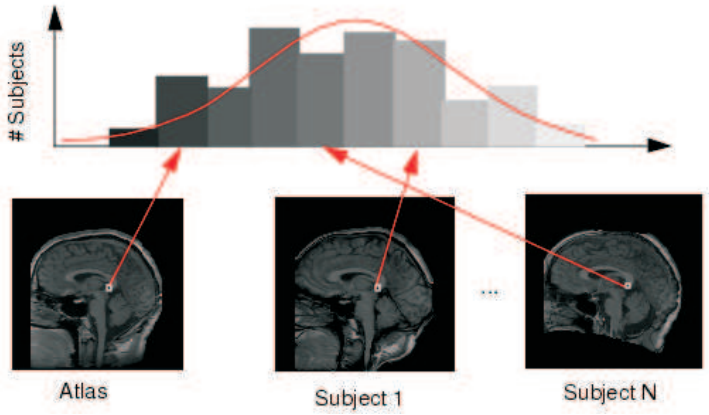


Figure 2.4: Gray level distribution per voxel (from [18])

where $\overrightarrow{\nabla I}$ is the image gradient. The use of eq. 2.19 resulted in a 34% error reduction in comparison with the method not using statistics. Smoothness constraint between one shift vector and its neighbors leads to an improved and more consistent solution. He proposes two methods: one to extend eq. 2.15 and eq. 2.16 to a multivariate Gaussian that involves blocks of $N \times M \times K$ neighbors which requires large overhead. Another approach is using a weighted sum of eq. 2.15, 2.19 over a neighborhood $\sum_{i,j,k} w_{i,j,k} \overrightarrow{\nabla I}_{i,j,k}$ in order to smooth displacements.

2.2 Structural Models

These models represent or extract structural features of shapes. They do that in the following ways:

- Abstract shapes as a graph of atomic shape substructures.
- Align a predefined model to a given shape using rigid and elastic deformations.
- Represent structure as a feature vector which can be compared to feature vectors of other shapes.
- Analyze shape structure at multiple levels of resolution.

The following sections summarize these models.

2.2.1 Grammar Models

Grammars can be used to generate and analyze shapes [36, 70, 35]. The main hypothesis is that shapes can be described as a composition of smaller shape atoms. Shape rules

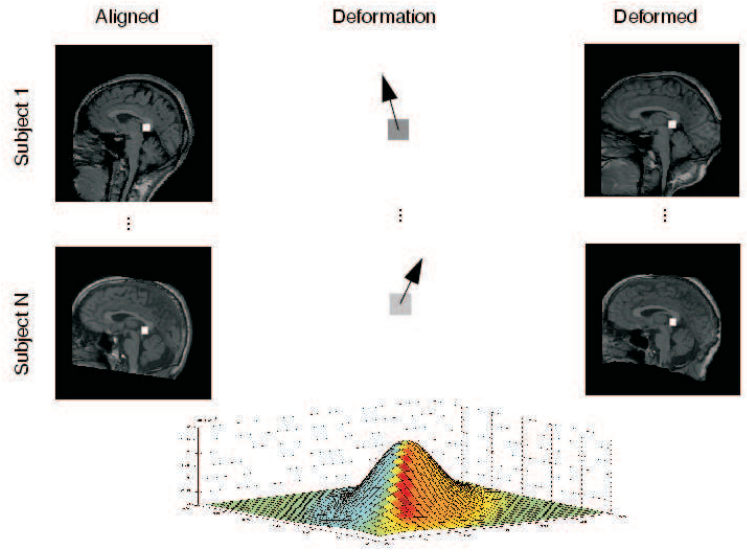


Figure 2.5: Geometric distribution per voxel (from [18])

define how the shape atoms can combine. The following sections will introduce two models that successfully employ shape grammars for analysis and synthesis.

Shape Grammars: Tree, Network

High dimensional grammars enable the recursive description of patterns that are spatially related to each other by higher order relations other than simple concatenation. It is possible to represent high dimensional grammars by a standard grammars with special nonterminal denoting positional information between parts. However, it is more compact and expressive to use high dimensional grammars.

Applications High dimensional grammars can be used to generate patterns used in the field of art and design [70]. They can also be used to describe structural relations between components such as an electronic circuit board and shapes of chromosomes [36, 35]. Another application is in assembly systems in which allowable connectivity between different part types is described by grammar [7].

In the following paragraphs, a brief description of both web and tree grammars with examples will be shown.

A web grammar is defined by the tuple $G = (N, \Sigma, P, S)$ where N is a set of nonterminal symbols, Σ is a set of terminal symbols, S is the start symbol and P is a set of productions of the form (α, β, ϕ) where α represents the sub-web to be replaced by β and ϕ is the function which specifies the embedding of β into the web occupied by α. As an example

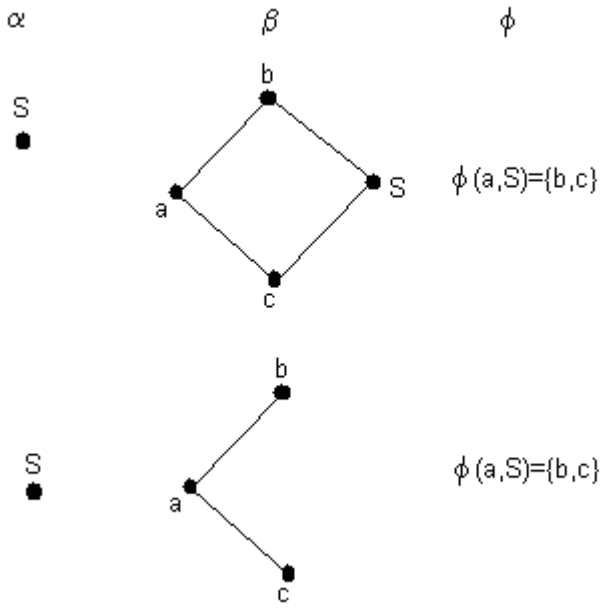


Figure 2.6: Productions used by a web grammar (from [36])

consider the web grammar defined by $N = \{S\}$,$\Sigma = \{a, b, c\}$ and P is the set of triples shown in fig. 2.6.

The embedding function ϕ states that $\alpha = S$ can be rewritten as β by connecting node a of β to the neighbors of S labeled a, b. Fig. 2.7 shows an example generated by applying this grammar.

A tree grammar is a special case of web grammars in which each production spawns a sub tree form a single non-terminal node. A typical production can be seen in fig. 2.8 where a is a terminal and $A_1, A_2, \ldots A_n$ are non-terminals.

In the following section we will discuss more specialized types of grammars used in shape representation.

Shock Grammar

Shape is described by four atomic units called shocks [67, 68]. A shock is the time evolving medial axis of a shape formed by colliding propagation fronts emitted from the shape boundary towards its center. Shape grammar describes how these shock types can combine and used to trim impossible combinations. The robust shock graph facilitates easy comparison between shapes and also reconstruction of the shape. By blurring the boundary with a smoothing kernel and observing the effect of that on shock, a multi-resolution description of shape is possible.

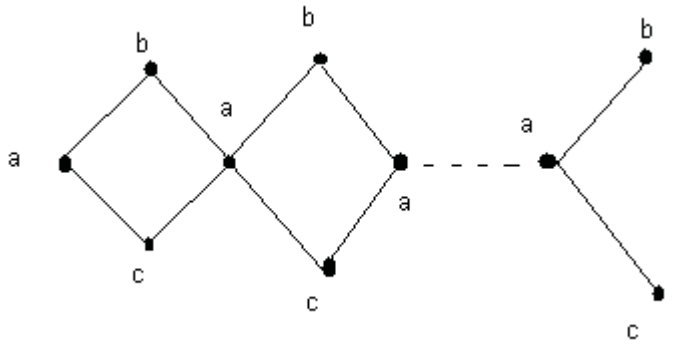


Figure 2.7: Pattern generated from a web grammar (from [36])

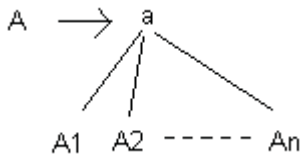


Figure 2.8: A tree grammar production(from [36])

Applications 2D images of simple geometric shapes (like a rectangle or a dumbbell), more complex shape like a hand and a comparison between different hands shapes and hand shapes occluded by objects.

Method There are 4 types of shocks defined as depicted in fig. 2.9:

- A first order shock is formed when the colliding fronts originating at the boundaries form a medial axis propagating with a finite speed in the direction where shape boundaries are widening. First order shocks correspond to protrusions where opposite boundaries form a bottle neck shape.
- A second order shock is a point where two opposite boundaries form a neck (local minimum) or where propagating fronts collide first and grow in opposite directions.
- A third order shock corresponds to parallel boundaries or where fronts collide forming the medial axis at the same instant in time.
- A forth order shock corresponds to a circle or when the fronts collide to a single point.

A shape is described as a sequence of shocks called shock groups. The shock grammar defines rules which describe the formation of consistent shock groups. The nonterminal set of the shock grammar is $N = \{S_1, S_2, S_3, S_4, S_I, E\}$ where S_1, S_2, S_3, S_4 are the shock

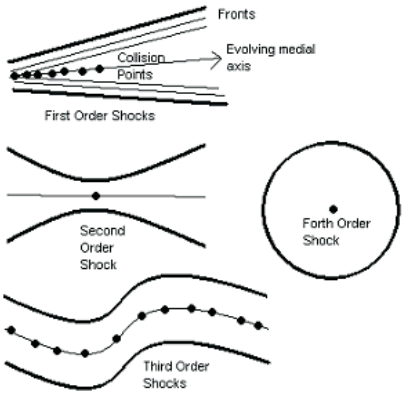


Figure 2.9: Shock types

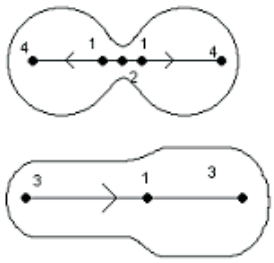


Figure 2.10: Examples of shock groups

types and S_I is the start symbol and E describes the end of a time evolving shock group. $\Sigma = \{S_T\}$ is the terminal symbol. The rules of a shock group are

$$\begin{aligned} R = \{&S_I \rightarrow S_1E,\ \ S_I \rightarrow S_2E,\ \ S_I \rightarrow S_3E,\ \ S_I \rightarrow S_4, \\ &S_1E \rightarrow S_1S_1E,\ \ S_1E \rightarrow S_1S_3E,\ \ S_1E \rightarrow S_4, \\ &S_2E \rightarrow S_2S_1E, \\ &S_3E \rightarrow S_3S_1E,\ \ S_3E \rightarrow S_3T, \\ &S_4 \rightarrow S_4S_T\} \end{aligned} \tag{2.20}$$

For example, fig. 2.10 shows two shapes which can be derived by applying the shock grammar as follows: The first shape can be derived $S_I \Rightarrow S_2E \Rightarrow S_2S_1E \Rightarrow \ldots \Rightarrow S_2[S_1 \ldots S_1]E \Rightarrow S_2[S_1 \ldots S_1]S_4 \Rightarrow S_2[S_1 \ldots S_1]S_4S_T$. The second shape is derived $S_I \Rightarrow S_3E \Rightarrow S_3S_1E \ldots S_3S_1 \ldots S_1E \Rightarrow S_3S_1 \ldots S_1S_3ES_3S_1 \ldots S_1E \Rightarrow S_3S_1 \ldots S_1S_3S_T$.

The shock grammar is used to define a procedure to prune impossible shock configurations. The procedure is as follows:

- A first order shock should be appended at the end of an existing first order branch as long as it both maintains the continuity of orientation and has a finite speed.

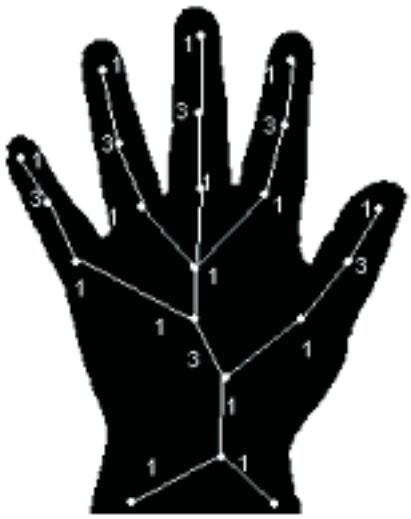


Figure 2.11: An example of a shape described by shocks (from [68])

Otherwise, a new first order shock should be initiated. If a first order shock remains isolated for a long time it should be terminated.

- A second order shock hypothesis should be discarded if it is not initial (i.e. a first order shock flows into it).
- A first order shock terminating or emanating from the middle (not the end point) of a third order shock should be deleted. A first order shock that terminates or emanates from a third order shock without maintaining continuity of orientation should be deleted.
- Two third order shocks should be grouped together if they are neighbors and their orientation is continuous. Groups of third order shocks should never intersect other groups of third or first order shocks. A third order shock that remains isolated as a single point should be interpreted as a forth order shock.
- A forth order shock that is connected to a second, third or forth order shock should be terminated.

Shapes are decomposed as shock groups to facilitate easy comparison with other shapes (by just comparing topology or shock graphs) and also shape reconstruction (by reversing the fronts from the medial axis). Fig. 2.11 depicts an example of a hand described by shock groups.

Multi-resolution (Shape Diffusion)is achieved by a curvature deformation transform (equivalent to Gaussian smoothing of the shape boundary) being applied on the shape. The significance of a shock group is proportional to its survival with increasing amounts of curvature deformation. For example a shape described by a shock group (4 - 1 - 2 - 1 - 4) can be shape diffused gradually to (4 - 1 - 3 - 1 - 4) and then with increasing

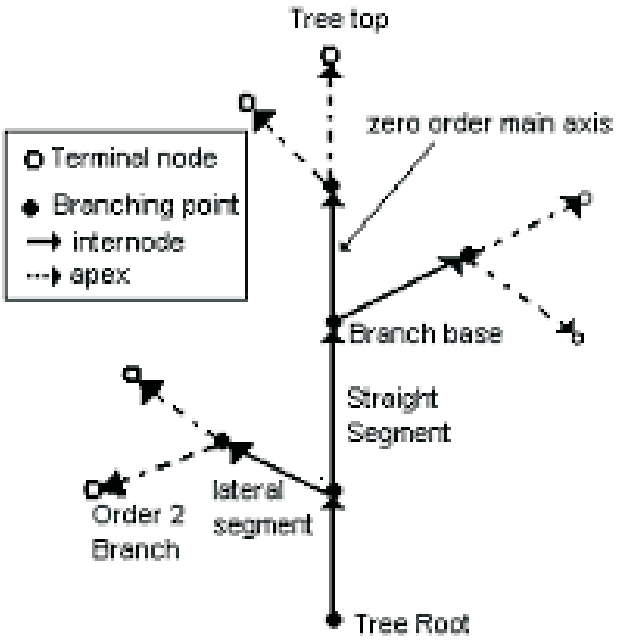


Figure 2.12: An axial tree (from [61])

diffusion to (3 - 2 - 4) which means the right forth shock is more resistant to diffusion than the left forth shock. Eventually the shape will diffuse to a single forth type shock (4).

L-Systems in Graphics

L-systems are used in computer graphics to generate branching structures in plants [61, 60]. L-Systems are a parallel context sensitive tree grammar (augmented with geometric attributes) that can simulate inheritance from ancestor branches to descendants and signals moving between neighboring branches. Discrete time evolution and probabilities of selecting multiple productions are also some modelling capabilities of L-Systems. This facilitates the description and simulation of complex plant growth patterns.

Applications Generation of plant growth patterns (e.g. sequential overlapped growth of flowering patterns).

Method An axial tree fig. 2.12 is a special type of a rooted tree where each node has at most one outgoing straight segment. All remaining edges from that node are called lateral segments. A sequence of segments is called an axis if

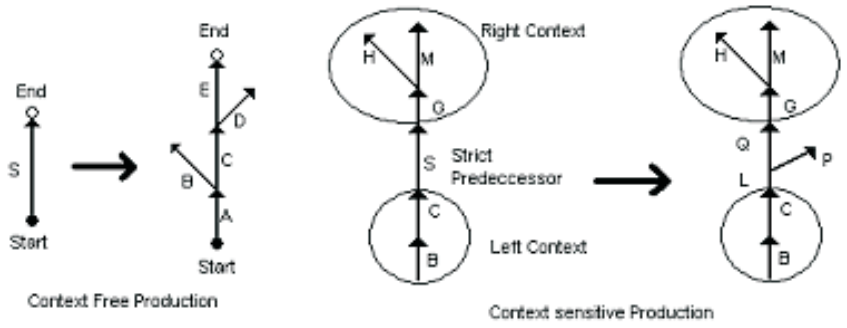


Figure 2.13: Productions (from [61])

1. The first segment is the root of the tree or a lateral segment of some node.
2. Each subsequent segment is a straight segment.
3. The last segment is not followed by a straight segment.

The axis together with its descendants forms a branch. A branch itself is an axial tree. Axes are given an order number where the axis originating at the root has order zero. An axis originating at a lateral segment of an n-order parent is of order $n + 1$.

A context free tree production as depicted in fig. 2.13 replaces a labelled edge called the predecessor with axial tree called the successor in such a way that the starting nodes of the predecessor is the same as the starting node of the successor and the same is true for the end node. A context sensitive production defines the predecessor in three parts: an axial tree l called the left context, an edge S called the strict predecessor and a right tree r called the right context.

An L-system G consists of:

1. A set of edge labels called the alphabet denoted by V.
2. An initial axial tree ω called axiom.
3. A set of tree productions P. Given G an axial tree T_2 is directly derived from a tree T_1 by simultaneously replacing each edge in T_1 by its successor defined by a production in P. L-systems are parallel rewriting rules.

A string generated by L-systems is rendered into a plant shape by using a logo like 3D turtle. The turtle has a state consisting of position, orientation and other attributes like color width etc. The L-systems are extended to generate segments with a set of attributes attached which are interpreted as commands to move the turtle and draw the object. For example: The first production in fig. 2.13 can be represented as $S \rightarrow S[-S]S[+S]S$ where $[\dots]$ enclose a nested branch and $-$ means turn left 60° and $+$ means turn right

60°. The state of the mouse is saved on a stack upon entering a branch and restored when finished with drawing that branch to its original position and orientation.

2.2.2 Alignment Models and Registration

This class of shape models consists of a single shape A that needs to be aligned to an image B. The problem is to find a transform T such that the similarity beween $T(A)$ and B is maximized. There are several types of transformations T possible:

1. Rigid body: In this case the only allowed transform is a combination of translation, rotation and scale such that the model shape is preserved.
2. Affine: In this case a transform is allowed by optimizing all parameters in affine space. This allows a greater freedom such as the model can be squashed and sheared.
3. Elastic: This includes the first transform type and additionally we allow local deformations for the model. Generally these models maintain smoothness constraints between local neighbours.

The techniques used in registration are very diverse and varied. As an example of elastic registration, consider the alignment of of the 2D figure in fig.2.14. In this case the space is deformed by B-splines. By placing an equidistant matrix of control points then displacing them around, we are able to generate a smooth deformation of the model that fits the image [29]. Generally the deformation space described with 3D objects is described as

$$T(x,y,z) = T_{global}(x,y,z) + T_{local}(x,y,z) \tag{2.21}$$

where T_{global} is the rigid body transform and T_{local} is an elastic transform described as

$$T_{local}(x,y,z) = \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(u)B_m(v)B_n(w)\phi_{i+l,j+m,k+n} \tag{2.22}$$

where Φ is a $n_x \times n_y \times n_z$ mesh of control points $\phi_{i,j,k}$ with uniform spacing δ and
$i = \lfloor \frac{x}{n_x} \rfloor - 1, j = \lfloor \frac{y}{n_y} \rfloor - 1, k = \lfloor \frac{z}{n_z} \rfloor - 1,$
$u = \frac{x}{n_x} - \lfloor \frac{x}{n_x} \rfloor, v = \frac{y}{n_y} - \lfloor \frac{y}{n_y} \rfloor, w = \frac{z}{n_z} - \lfloor \frac{z}{n_z} \rfloor$

B_l is the l^{th} basis function of a cubic spline

$$B_0(u) = (1-u^3)/6 \tag{2.23}$$
$$B_1(u) = (3u^3 - 6u^2 + 4)/6 \tag{2.24}$$
$$B_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6 \tag{2.25}$$
$$B_3(u) = u^3/6 \tag{2.26}$$

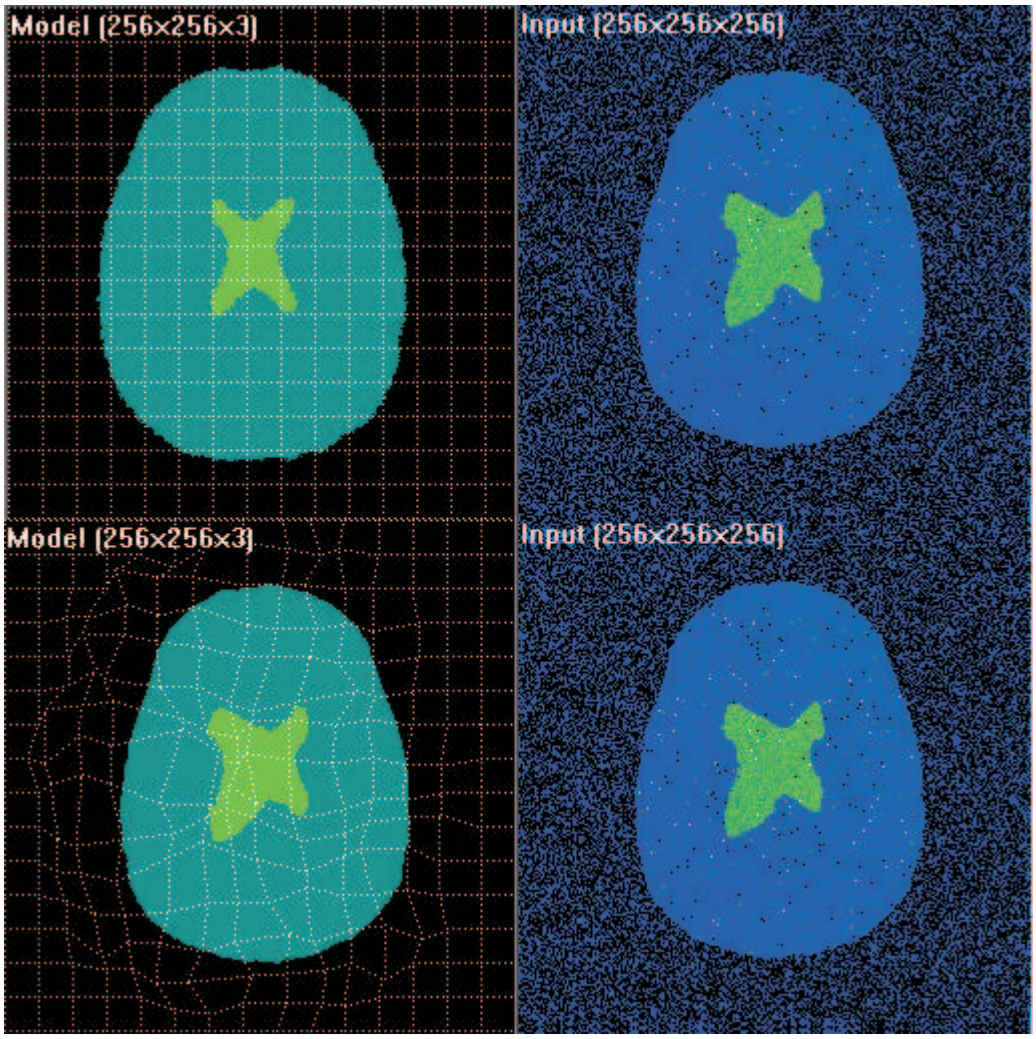


Figure 2.14: Elastic registration using B-splines showing the model before and after displacing the control points

The methods for matching similarity measures are also very diverse. A notable example is normalized mutual information [71] used to match images of different modalities. This measure is defined as

$$Y(A,B) = \frac{H(A) + H(B)}{H(A,B)} \tag{2.27}$$

where $H(A), H(B)$ are image entropies and $H(A,B)$ is the mutual entropy defined from the mutual histogram of A, B obtained from co-occurrence of intensity values of pixels at the same location.

Registration have extensively been used in medicine for example to diagnose abnormalities such as [17, 4, 5] and for matching mammography images [29] and faces [37].

2.2.3 Geons

Biederman [10] proposed a theory of Recognition by Components (RBC). The basic idea is that a structurally complex 3D object can be decomposed into a set of primitive solids called Geons. The Geon decomposition is done in two steps:

1. Object decomposition into segments or parts.
2. Geon identification for each part.

Object decomposition can be done in two ways:

1. Region based: They find image regions that correspond to object surface patches. These patches are then grouped based on Geon surface configurations.
2. Boundary based: The object surface is decomposed at points of high curvature. Each segment is fitted to the most appropriate Geon.

An example of boundary based segmentation is [78] which uses a physical model of electric charge density distribution to estimate surface curvature. This is due to the fact that convex parts have a high charge density and concave parts have a low charge density. After decomposing the object at points of high concavity, seven parametric Geon types are fitted to each segment. The most appropriate Geon type is selected for each segment providing a qualitative description and the Geon parameters provide for quantitative measurements.

Applications Range data of 3D solids.

Method The decomposition process uses a finite element model to calculate charge density distribution. The charge density is determined by the electric potential between all the points on the surface. The electric potential between a point charge q at $\mathbf{r}'$ on the surface and a reference point $\mathbf{r}$ as seen in fig. 2.15 is given by

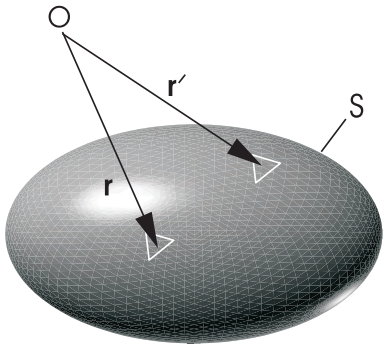


Figure 2.15: The observation point $\mathbf{r}$ and the charge source point $\mathbf{r}'$ on the surface S of an ellipsoid. O is the origin (from [78]).

$$\phi(\mathbf{r}) = \frac{q}{4\pi\epsilon_0}\frac{1}{|\mathbf{r}-\mathbf{r}'|} \tag{2.28}$$

where ϵ_0 is a constant.

The object surface is modelled as a set of finite elements of connected triangles $T_k, k = 1 \ldots N$. Each triangle is assumed to have a constant charge density $\rho_k, k = 1 \ldots N$. Because of that we may take $\mathbf{r}_i, i = 1 \ldots N$ as the observation point on each surface element. The potential at each surface element is then

$$V = \sum_{k=1}^{N} \rho_k \int_{T_k} \frac{1}{|\mathbf{r}_i - \mathbf{r}'|} dS', i = 1 \ldots N \tag{2.29}$$

V is assumed to be constant. This is because according to physics, all points on a charged conductor in equilibrium are at the same electric potential. The total charge on the surface Q is assumed to be known. We write Q as

$$Q = \sum_{k=1}^{N} \rho_k S_k \tag{2.30}$$

We obtain from eq. 2.29 and eq. 2.30 a set of linear equations with $N + 1$ unknowns, $\rho_1 \ldots \rho_N, V$. Solving for the charge densities we get a measure of curvature where convex points have a high charge density and concave points have a low charge density.

After determining charge density distribution, the decomposition algorithm iteratively splits the object at closed low charge density boundaries. The tracing of each boundary begins by selecting a starting triangle as follows:

1. Its charge density must be a local minimum.
2. The charge density must be below a certain threshold.

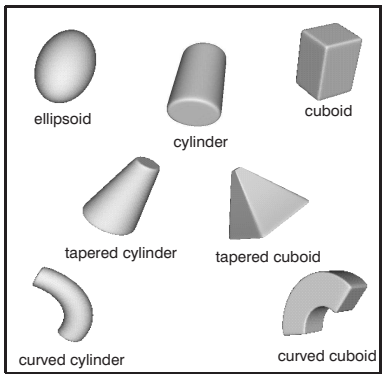


Figure 2.16: The seven parametric geons (from [78])

3. The triangle and its neighbors must not have been visited before checking that the same boundary will not be traced again.

The algorithm begins from the starting triangle by finding the neighbor with the lowest charge density and marking it so that it will not be visited again. This is repeated until we return to the starting triangle thus tracing a complete boundary. Boundary tracing is repeated until the object is fully decomposed into subparts.

After splitting the objects into subparts, the Geon identification algorithm tries to fit seven parametric shapes shown in fig. 2.16 to each part. Each Geon has a vector $\mathbf{a}_i, i = 1 \dots 7$ of model parameters that describe the size and deformation information for that Geon. The fitting function used to optimize these parameters is a weighted sum of the distance between the Geon and object surfaces as well as the difference in normals between the two surfaces.

2.2.4 Generalized Cylinders

A deformable model for reconstructing 3D surfaces from 2D projections of objects was developed by Terzopoulos et. al. [74]. This is achieved through a symmetry seeking deformable cylinder that is shaped both by the silhouette of the object and its internal symmetry forces. The user also plays a role in shaping the deformable cylinders by initializing them and interactively modifying them.

Applications 3D reconstruction of objects from a single image, 3D reconstruction of objects through stereo images, motion tracking.

Method A generalized deformable cylinder consists of a tube and a spine. The spine and the tube are geometrically represented as a mapping from object coordinates to cartesian 3D space as depicted in fig. 2.17. The spine is mapped from $s \in [0, 1]$ into $\mathbb{R}^3$: $\mathbf{v}^S(s,t) = (X(s,t), Y(s,t), Z(s,t))$. The sheet is defined by the bivariate mapping from

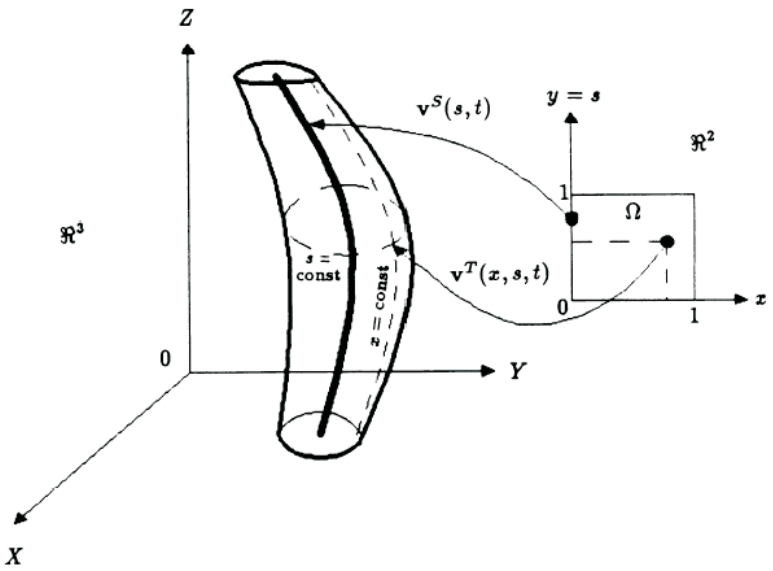


Figure 2.17: The geometric structure of a deformable cylinder (from [74])

$(x, y) \in [0, 1]^2$ into $\mathbb{R}^3$: $\mathbf{v}^T(x, y, t) = (X(x, y, t), Y(x, y, t), Z(x, y, t))$. The sheet is folded into a tube by defining two boundary conditions: $\mathbf{v}^T(0, y, t) = \mathbf{v}^T(1, y, t)$, $\frac{\partial \mathbf{v}^T}{\partial x}|_{(0,y,t)} = \frac{\partial \mathbf{v}^T}{\partial x}|_{(1,y,t)}$. The tube and the spine are coupled together by setting $y \equiv s$.

The geometric structure of a cylinder undergoes dynamic deformation influenced by intrinsic (symmetry seeking) and extrinsic forces. This dynamic behavior is described by

$$\mu \frac{\partial^2 \mathbf{v}}{\partial t^2} + \gamma \frac{\partial \mathbf{v}}{\partial t} + \frac{\delta \xi(\mathbf{v})}{\delta \mathbf{v}} = \mathbf{f}(\mathbf{v}) \tag{2.31}$$

where $\mathbf{f}(\mathbf{v})$ is the net extrinsic force acting on the deformable body. μ is the mass density function of the body and γ is the viscosity function of the ambient medium. $\frac{\delta \xi(\mathbf{v})}{\delta \mathbf{v}}$ is the variational derivative of the strain energy ξ which expresses the elastic force internal to the body. The variational derivatives of a spine (ξ^S) and a tube (ξ^T) are expressed as

$$\frac{\delta \xi^S}{\delta \mathbf{v}} = \frac{\partial^2}{\partial s^2}(w_2 \frac{\partial^2 \mathbf{v}}{\partial s^2}) - \frac{\partial}{\partial s}(w_1 \frac{\partial \mathbf{v}}{\partial s}) \tag{2.32}$$

$$\begin{aligned} \frac{\delta \xi^T}{\delta \mathbf{v}} = & \frac{\partial^2}{\partial x^2}(w_{20} \frac{\partial^2 \mathbf{v}}{\partial x^2}) + 2\frac{\partial^2}{\partial x \partial y}(w_{11} \frac{\partial^2 \mathbf{v}}{\partial x \partial y}) \\ & + \frac{\partial^2}{\partial y^2}(w_{02} \frac{\partial^2 \mathbf{v}}{\partial y^2}) - \frac{\partial}{\partial x}(w_{10} \frac{\partial \mathbf{v}}{\partial x}) - \frac{\partial}{\partial y}(w_{01} \frac{\partial \mathbf{v}}{\partial y}) \end{aligned} \tag{2.33}$$

where the weights $w_1, w_2, w_{01}, w_{10}, w_{20}, w_{02}, w_{11}$ are functions of material coordinates and time that control tension and rigidity.

The extrinsic forces consist of the symmetry seeking forces, user interaction forces and image forces. The symmetry seeking forces maintain the spine at the center of the tube. To define these forces we first define the tube's centroid as

$$\overline{\mathbf{v}}^T(s) = \frac{1}{l}\int_0^l \mathbf{v}^T|\frac{\partial \mathbf{v}^T}{\partial x}|dx \tag{2.34}$$

$$l = \int_0^1 |\frac{\partial \mathbf{v}^T}{\partial x}|dx \tag{2.35}$$

We then define the tube's radial vector with respect to the spine as $\mathbf{r}(x,s) = \mathbf{v}^T(x,s) - \mathbf{v}^S(s)$ and the unit radial vector $\hat{\mathbf{r}} = \mathbf{r}/|\mathbf{r}|$ and the mean radius as

$$\bar{r}(s) = \frac{1}{l}\int_0^1 |r||\frac{\partial \mathbf{v}^T}{\partial x}|dx \tag{2.36}$$

The spine is forced to be in an axial position inside the tube the tube by introducing the following two forces on the spine and the tube respectively

$$f_a^S(s,t) = a(\overline{\mathbf{v}}^T - \mathbf{v}^S) \tag{2.37}$$

$$f_a^T(x,s,t) = -(a/l)(\overline{\mathbf{v}}^T - \mathbf{v}^S) \tag{2.38}$$

where $a(s)$ controls the strength of the force.

To make the tube seek radial symmetry around the spine we define the following force

$$f_b^T(x,s,t) = b(\overline{r} - |\mathbf{r}|)\overline{\mathbf{r}} \tag{2.39}$$

where $b(s)$ controls the strength of the force.

Finally an expansion/contraction force is introduced around the tube

$$f_c^T(x,s,t) = c\hat{\mathbf{r}} \tag{2.40}$$

where c(s) controls the strength of the force and if $c < 0$ then the cylinder deflates and it inflates when $c > 0$.

The total force for the spine and tube thus becomes

$$\mu\frac{\partial^2 \mathbf{v}^S}{\partial t^2} + \gamma\frac{\partial \mathbf{v}^S}{\partial t} + \frac{\delta \xi^S}{\delta \mathbf{v}^S} = \frac{\delta P^S}{\delta \mathbf{v}^S} + f_a^S \tag{2.41}$$

$$\mu\frac{\partial^2 \mathbf{v}^T}{\partial t^2} + \gamma\frac{\partial \mathbf{v}^T}{\partial t} + \frac{\delta \xi^T}{\delta \mathbf{v}^T} = \frac{\delta P^T}{\delta \mathbf{v}^T} + f_a^T + f_b^T + f_c^T \tag{2.42}$$

where $\frac{\delta P^S}{\delta \mathbf{v}^S}, \frac{\delta P^T}{\delta \mathbf{v}^T}$ are the variational derivatives of extrinsic image forces acting on the bodies.

The extrinsic image force of the tube $\frac{\delta P^T}{\delta \mathbf{v}^T}$ is computed from the potential function P^T. This potential attracts the tube to single object silhouettes . It defined as

$$P^T(\mathbf{v}^T) = \beta|\nabla(G_\sigma * I(\Pi[\mathbf{v}^T]))| \tag{2.43}$$

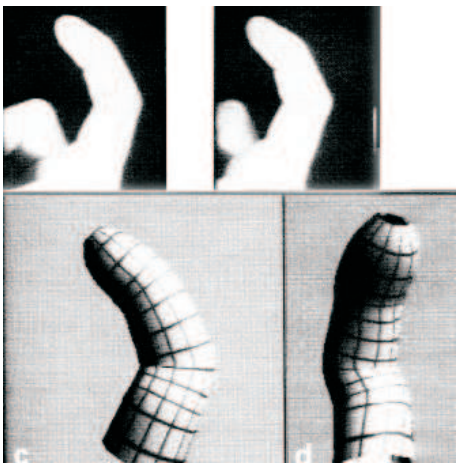


Figure 2.18: 3D reconstruction of a finger from a stereo pair (from [74])

$\Pi[\mathbf{v}^T]$ expresses the projection of the deformable tube $\mathbf{v}^T(x,s)$ to the image plane. $G_\sigma * I$ is the convolution of the image with a Gaussian smoothing filter with σ and ∇ is the gradient operator. $\beta(x,s)$ is the weighting function which is nonzero for occluding boundaries of the tube

$$\beta(x,s) = \begin{cases} 1, & \text{if } |\mathbf{i}.\mathbf{n}| < \tau \\ 0, & \text{otherwise} \end{cases} \tag{2.44}$$

where $\mathbf{n}$ is the unit normal over the surface of the tube and $\mathbf{i}$ is the unit vector from the imaging focal point to any point on the tube.

The image potential defined in eq. 2.43 is used to reconstruct the deformable tube from a single image. This can be generalized to multiple stereo images by adding a new potential for each image and just summing them. Fig. 2.18 depicts a finger reconstructed from a stereo pair. This idea can be carried out further to track moving objects in an image sequence. This is achieved by initializing the deformable cylinders from the first image. Subsequent images exert time continues image forces that drive the cylinder to new deformations.

A disadvantage of deformable generalized cylinders presented here is that they require initial spines to be set by the user.

2.2.5 Shape Blending

DeCarlo et al. [27] proposed to model a shape as set of primitive surface patches smoothly interpolated with one another. Holes in the shape are defined as a topological operation. A top down fitting process begins with an ellipsoid wrapped around the object and end up with a graph describing the main constituent blended components of the shape.

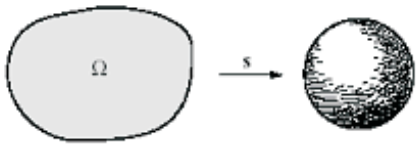


Figure 2.19: Shape map (from [27])

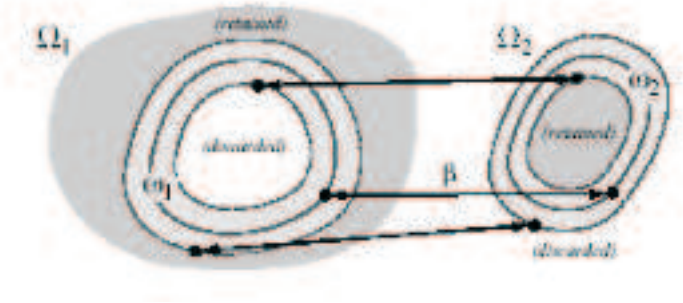


Figure 2.20: Glued domains defined by the equivalence $\sim$ (from [27])

Applications Range data consisting of primitive solids like a box attached to a cylinder, Simple solids with holes and complex solids like a mannequin.

Method Shape is defined as a function $s : \Omega \rightarrow \mathbb{R}^3$ where Ω is the parametric domain of the shape mapped by s to a 3D point in space as depicted in fig. 2.19.

Blending two shapes $s_1 : \Omega_1 \rightarrow \mathbb{R}^3, s_2 : \Omega_2 \rightarrow \mathbb{R}^3$ involves two steps:

- Specifying the retained subsets of each shape's domain and glueing overlapping domains together.
- Defining the interpolation function between the two shape surfaces on the glued domains.

To glue two domains we define closed curves $\kappa_1 \subset \Omega_1, \kappa_2 \subset \Omega_2$ as depicted in 2.20. Overlapping neighbors between Ω_1, Ω_2 are defined around κ_1, κ_2: $\omega_1 \subset \Omega_1, \omega_2 \subset \Omega_2$. An equivalence relation maps between ω_1, ω_2: $\mathbf{u}_1\ \mathbf{u}_2 \Leftrightarrow \beta(\mathbf{u}_1) = \mathbf{u}_2, \mathbf{u}_1 \in \omega_1\, and\, \mathbf{u}_2 \in \omega_2$. The domain of the blended shape is $\Omega^* = (\Omega_1 \cup \Omega_2)/\sim$.

A surface-blending function $\alpha : \omega_1 \rightarrow [0, 1]$ performs the smooth join between the two original shapes as depicted in fig. 2.21. We define the blended shape as follows

$$s(u) = \begin{cases} s_1(u), u \in \Omega_1 - \omega_1 \\ s_2(u), u \in \Omega_2 - \omega_2 \\ s_1(u)\alpha(u) + s_2(u)(1 - \alpha(u)), u \in \omega_1 \end{cases} \tag{2.45}$$

The algorithm begins by fitting an ellipsoid to the object. A blending region is created along the surface boundary separating outward and inward boundary forces as depicted

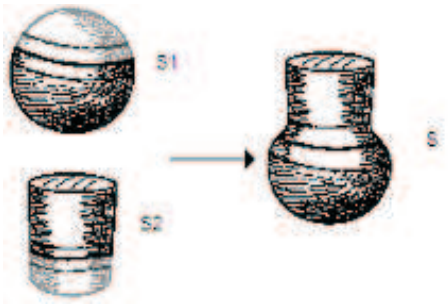


Figure 2.21: Geometric blending of two shapes (from [27])

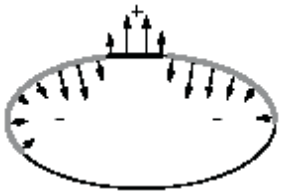


Figure 2.22: A blending surface created along the boundary between outward and inward forces (from [27])

in fig.2.22. This results in protrusions being separated as blended shape components. Whenever the surface self-intersects, a hole is created. The fitting process results in creating a graph describing the main shape components as depicted in fig.2.23

2.2.6 Super Quadric Model

Terzopoulos and Metaxas in [73, 52] propose a mixed model to represent a single object with no parts, which consists of:

1. A global superquadric which facilitates recognition and comparison between shapes.
2. A local deformation model in the form of a displacement field that fits the finer details of the object.

The fitting process is modeled as a dynamic system.

Applications Range data from simple objects like an egg, a mug to more complex shapes like a doll.

Method Fig. 2.24 depicts the super quadric model which is a closed surface with parametric coordinates $\mathbf{u} = (u, v)$ defined on a domain Ω. The positions of points on the model are given as a time varying function of u

$$\mathbf{x}(\mathbf{u}, t) = (x_1(\mathbf{u}, t), x_2(\mathbf{u}, t), x_3(\mathbf{u}, t))^T \tag{2.46}$$

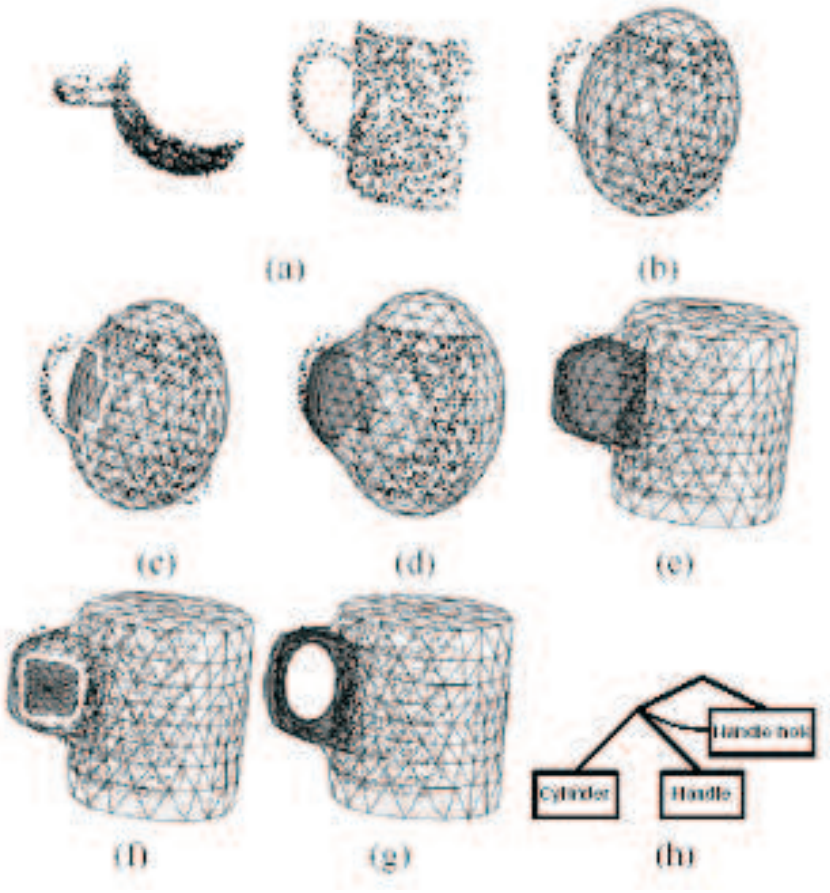


Figure 2.23: Fitting of a cup and the resulting shape component graph (form [27])

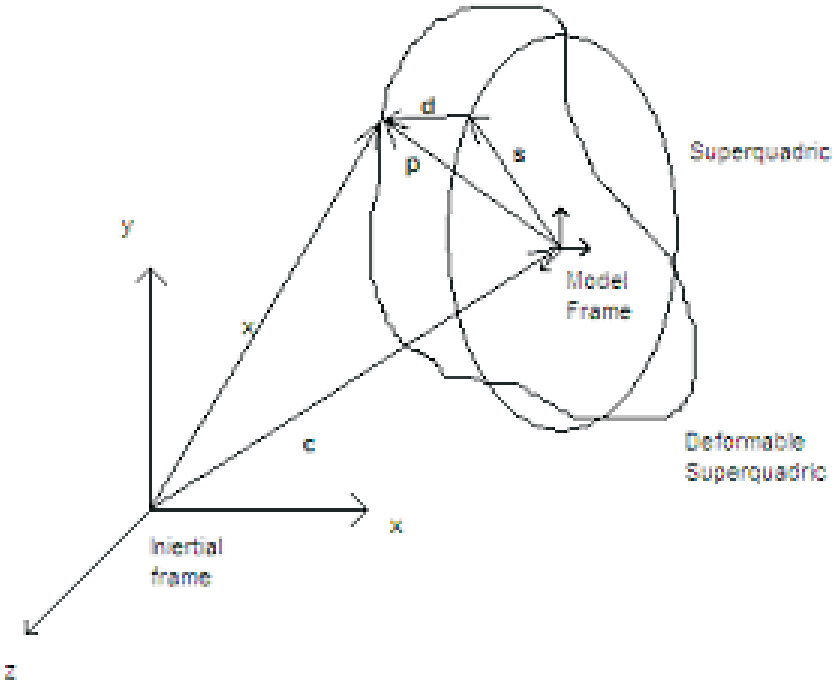


Figure 2.24: Geometry of a deformable super quadric (from [73])

The transform between the model frame of reference and world coordinates is expressed using the translation vector $\mathbf{c}(t)$ and the rotation matrix $\mathbf{R}(t)$

$$\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p} \tag{2.47}$$

The position $\mathbf{p}$ is expressed as the sum of the super-quadric shape $\mathbf{s}(\mathbf{u}, t)$ and a displacement vector $\mathbf{d}(\mathbf{u}, t)$

$$\mathbf{p} = \mathbf{s} + \mathbf{d} \tag{2.48}$$

The super-quadric ellipsoid $\mathbf{s}$ is defined as

$$\mathbf{s} = a \begin{pmatrix} a_1 C_u^{\tau_1} C_v^{\tau_2} \\ a_2 C_u^{\tau_1} S_v^{\tau_2} \\ a_3 S_u^{\tau_1} \end{pmatrix}, -\pi/2 \leq u \leq \pi/2, -\pi \leq v \leq \pi \tag{2.49}$$

$$S_w^{\tau} = sign(sin(w))|sin(w)|^{\tau} \tag{2.50}$$

$$C_w^{\tau} = sign(cos(w))|cos(w)|^{\tau} \tag{2.51}$$

The parameter space of $\mathbf{s}$ is defined as

$$\mathbf{q}_s = (a, a_1, a_2, a_3, \tau_1, \tau_2) \tag{2.52}$$

The displacement field $\mathbf{d}$ is expressed as a sum of basis functions $\mathbf{b}_i(\mathbf{u})$

$$\mathbf{d} = \sum_i \mathbf{b}_i q_i \tag{2.53}$$

The vector $\mathbf{q}_d(t) = (\ldots, q_i(t), \ldots)^T$ is the vector of generalized coordinates that depend on time only. The basis functions $\mathbf{b}_i$ are collected as a diagonal matrix $\mathbf{S}$ so we can rewrite $\mathbf{d}$ as

$$\mathbf{d} = \mathbf{S}\mathbf{q}_d \tag{2.54}$$

The parameter space is defined by the vector $\mathbf{q} = (\mathbf{c}^T, \theta^T, \mathbf{q}_s^T, \mathbf{q}_d^T)^T$ where θ are rotational parameters. By assuming no mass to the dynamic system we can write the dynamic equation of the model as

$$\mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}_q \tag{2.55}$$

where $\mathbf{C}, \mathbf{K}$ are damping and stiffness matrices and $\mathbf{f}_q$ are the image forces. This equation can be expressed as a time step sequence as follows

$$\mathbf{q}^{(t+\Delta t)} = \mathbf{q}^{(t)} + \Delta t \mathbf{C}^{-1}(f_q^{(t)} - \mathbf{K}\mathbf{q}^{(t)}) \tag{2.56}$$

The displacement field $\mathbf{d}$ is kept smooth by maintaining the constraint

$$\varepsilon(\mathbf{d}) = \int w_1(u)((\frac{\partial \mathbf{d}}{\partial u})^2 + (\frac{\partial \mathbf{d}}{\partial v})^2) + w_0(u)\mathbf{d}^2 du \tag{2.57}$$

The image force is described as a sum of two terms:

1. A short term force resulting from gradient information after convolving the image with a Gaussian filter $P(x, y) = \|\nabla(G_\sigma * I)\|$.
2. A long range force that determines the distance between the image surface and the nearest point on the deformable boundary $f(\mathbf{u}_\gamma) = \beta\|\mathbf{r} - \mathbf{x}(\mathbf{u}_\gamma)\|$ where $\mathbf{r}$ is a data point and $\mathbf{x}(\mathbf{u}_\gamma)$ is the nearest point on the model surface.

The importance of this model is its multi-resolution property which enables it to capture global shape features important for image description and comparison. At the same time, it has the ability of reconstructing the shape using the local model.

2.2.7 Finite Element Model

Sclaroff [66, 57, 65] describes a deformable finite element model which is fitted dynamically to a single object with no specified structure. Modal analysis finds the coordinates

of the fitted model in the eigen space of free vibration modes. This representation of the fitted model enables easily the recognition of objects with great accuracy (e.g. facial recognition).

Applications Fitting and recognition of human faces. Segmenting and fitting of human figures.

Method The fitting model (usually an ellipsoid) is modelled as a connected set of (n) nodes. The energy of the whole shape is described as

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R} \tag{2.58}$$

where

- $\mathbf{U}$ is a $3n \times 1$ vector of $(\Delta x, \Delta y, \Delta z)$ displacements of the n nodal points from their initial positions.
- $\mathbf{M}, \mathbf{C}, \mathbf{K}$ are $3n \times 3n$ matrices describing mass, damping and material stiffness of the whole system.
- $\mathbf{R}$ is the $3n \times 1$ vector describing the x,y and z components of the forces acting on the nodes.

The system eventually comes to a state of rest when it satisfies the equilibrium equation

$$\mathbf{K}\mathbf{U} = \mathbf{R} \tag{2.59}$$

The purpose is to find an orthogonal transformation matrix that diagonalizes eq. 2.58 which decouples the degrees of freedom, finds a closed form solution, and reduces computation. The transformation matrix $\mathbf{\Phi}$ is applied as follows

$$\mathbf{U} = \mathbf{\Phi}\tilde{\mathbf{U}} \tag{2.60}$$

Substituting eq. 2.60 into eq. 2.58 yields

$$\begin{aligned}
\tilde{\mathbf{M}}\ddot{\tilde{\mathbf{U}}} + \tilde{\mathbf{C}}\dot{\tilde{\mathbf{U}}} + \tilde{\mathbf{K}}\tilde{\mathbf{U}} &= \tilde{\mathbf{R}} && (2.61)\\
\tilde{\mathbf{M}} &= \mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi} && (2.62)\\
\tilde{\mathbf{C}} &= \mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi} && (2.63)\\
\tilde{\mathbf{K}} &= \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi} && (2.64)\\
\tilde{\mathbf{R}} &= \mathbf{\Phi}^T\mathbf{R} && (2.65)
\end{aligned}$$

The transform $\mathbf{\Phi}$ is obtained by solving the eigen value problem

$$\mathbf{K}\phi_i = \omega_i^2 \mathbf{M}\phi_i \tag{2.66}$$

Solving eq. 2.66 yields $3n$ eigen solutions such that all eigen vectors ϕ_i are orthonormalized

$$\mathbf{\Phi} = [\phi_1, \phi_2 \dots \phi_{3n}] \tag{2.67}$$

$$\Omega^2 = \begin{pmatrix} \omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \ddots & \\ & & & \omega_{3n}^2 \end{pmatrix} \tag{2.68}$$

$$\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} = \Omega^2 = \tilde{\mathbf{K}} \tag{2.69}$$

$$\mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi} = \mathbf{I} = \tilde{\mathbf{M}} \tag{2.70}$$

The damping matrix $\mathbf{C}$ is made diagonalizable by restricting it to the form

$$\mathbf{C} = a_0 \mathbf{M} + a_1 \mathbf{K} \tag{2.71}$$

Using the transform Φ, eq. 2.58 is reduced to $3n$ individual equations

$$\ddot{\tilde{u}}_i(t) + (a_0 + a_1\omega_i^2)\dot{\tilde{u}}_i(t) + \omega_i^2 \tilde{u}_i(t) = \phi_i^T \mathbf{R}(t), i = 1, \dots, 3n \tag{2.72}$$

or

$$\ddot{\tilde{\mathbf{U}}} + (a_0 \mathbf{I} + a_1 \Omega^2)\dot{\tilde{\mathbf{U}}} + \Omega^2 \tilde{\mathbf{U}} = \mathbf{\Phi}^T \mathbf{R}(t) \tag{2.73}$$

The modal coordinate system defines the free vibration modes such that the first 6 modal coordinates correspond to translational and rotational rigid body transforms. Higher vibration modes convey information about the object's shape.

The force $\mathbf{R}$ is defined as a set of virtual springs between the model node (x_k, y_k, z_k) and the corresponding point at the object's surface (x_k^w, y_k^w, z_k^w) . The magnitude of the forces is defined as

$$(\gamma_{3k}, \gamma_{3k+1}, \gamma_{3k+2})^T = (x_k^w, y_k^w, z_k^w)^T - (x_k, y_k, z_k)^T, 1 \leq k \leq n \tag{2.74}$$

The fitting problem is finding the displacement that satisfies the equilibrium condition in eq. 2.59. This is simply the solution of $\mathbf{U} = \mathbf{K}^{-1}\mathbf{R}$ that requires an iterative method because $\mathbf{K}$ has a high dimension. A direct non-iterative closed form solution exists by converting the equilibrium into modal coordinates

$$\tilde{\mathbf{U}} = \tilde{\mathbf{K}}^{-1}\tilde{\mathbf{R}} = (\mathbf{\Omega}^2)^{-1}\tilde{\mathbf{R}} \tag{2.75}$$

To establish the correspondence between the model and the object, an ellipsoidal coordinate system is defined on the object whose origin is the center of mass and whose axes correspond to the axes of inertia of the object. The springs are then attached to the model from this object centered coordinate system.

The main significance of modal analysis is the ability to recognize and compare objects. This is because high frequency modal vectors are discarded and only a finite low order frequency eigen vectors are used which characterize the main features of the shape. This multi-resolution property enables the comparison of two objects simply by taking the dot product of their modal coordinates under the same model. The closer the dot product is to 1, the more similar the objects are. $\mathbf{\Omega}$ is pre-computed once for the model which enables efficient fitting on many objects. The similarity equation is

$$\varepsilon = \frac{\tilde{\mathbf{U}}_1 \cdot \tilde{\mathbf{U}}_2}{\|\tilde{\mathbf{U}}_1\|\|\tilde{\mathbf{U}}_2\|} \tag{2.76}$$

2.2.8 Curvature Scale Space

The main idea is to use Gaussian multi-resolution in shape analysis. This is done by smoothing the boundary of a 2D shape by a Gaussian kernel. A curvature scale space graph (CSS) is formed from zero crossings of the boundary curve. The longer the zero crossing survives increasing smoothing the more important it is as a shape feature. This enables the analysis of shape with high robustness against boundary noise.

Applications Detection of corners of planar curves [62], detection of malignant melanomata by measuring border irregularity [46] and efficient indexing of marine life silhouettes for content based image retrieval [54].

Method As depicted in fig. 2.25, a curvature scale space graph is constructed from a planar curve defined by a parametric curve $(x(t), y(t))$ where t is normalized to be in some fixed interval such as $[0, 1]$. The planar curve is subjected to a Gaussian smoothing kernel of width σ. A curvature scale space graph consists of u as the horizontal axis and σ as the vertical axis. A point plotted at some (u, σ) represents the zero crossing of the curvature of the smoothed boundary. The segments between those zero crossings represent concave (negative curvature) and convex (positive curvature) parts. If we sort the local maxima in the CSS graph from the highest down, we end up with main zero crossings of the shape that represent its most prominent features. The smallest local maxima represent insignificant or noisy features. This interesting property gives CSS

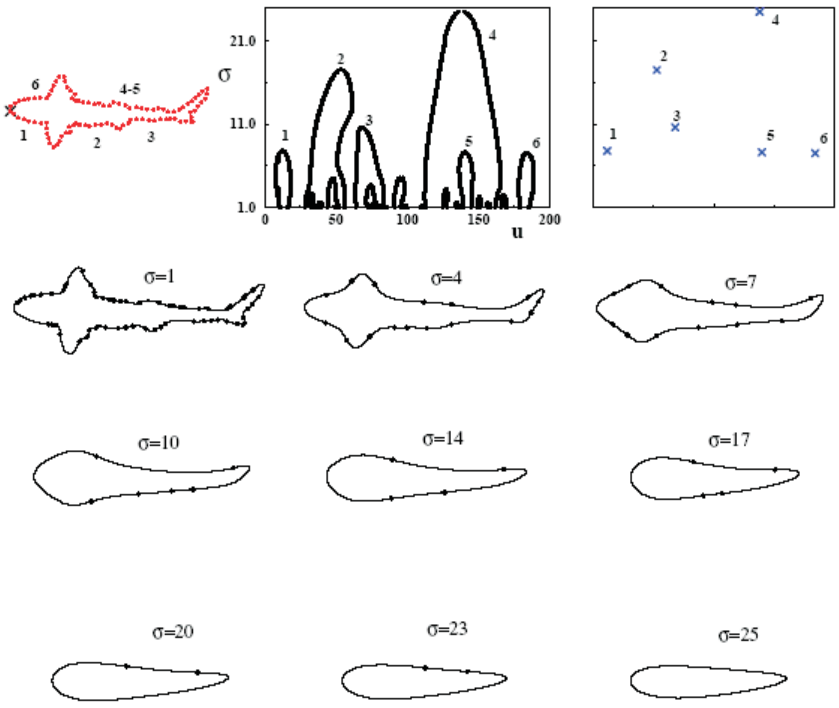


Figure 2.25: Curvature scale space graph and curve evolution (from [54])

robustness to noise and ability to measure important features of the shape and compare shapes.

2.3 Dynamic Models

After the selective survey of structural models, now we begin describing some representative dynamic models.

2.3.1 Snakes

The model introduced by Kass et. al. [40] defined a dynamic or active contour trying to locally minimize its energy. The snake-energy is defined as the integral sum of three weighted forces: the snakes internal force, the image forces, and external forces. The internal forces represent built-in smoothness constraints that try to minimize both stretching and bending of the active contour. The image forces are the image features that attract the snake towards edges. The external forces represent high level user interaction which pulls the snake away from a local minimum.

Figure 2.26: Tracking lips using snakes (from [40])

Applications Segmentation of medical images, 3D Stereo correspondence analysis and motion analysis (e.g. mouth movement tracking as depicted in fig. 2.26).

Method Given a parametric representation of a snake as $\mathbf{v}(s) = (x(s), y(s))$, the snake equation can be written as

$$E^*_{snake} = \int_0^1 E_{snake}(\mathbf{v}(s))ds = \int_0^1 E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s))ds \quad (2.77)$$

where $E_{int}, E_{image}, E_{con}$ are the internal, image, and external energies, respectively.

The internal energy of the snake is written as

$$E_{int} = (\alpha(s)|\mathbf{v}_s(s)|^2 + \beta(s)|\mathbf{v}_{ss}(s)|^2)/2 \quad (2.78)$$

where $\alpha(s)$ and $\beta(s)$ are relative weights that control stretching and bending forces, respectively.

The image force is defined as

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term} \quad (2.79)$$

where $w_{line}, w_{edge}, w_{term}$ are weights and $E_{line}, E_{edge}, E_{term}$ are energies calculated from intensity, gradient, and corner image feature, respectively.

User interaction is introduced by defining springs and volcanos. A spring is a force between the snake point $\mathbf{x}_1$ and an external point $\mathbf{x}_2$ described as $-k_1(\mathbf{x}_1 - \mathbf{x}_2)^2$. A volcano is a repulsion force described as k_2/r^2 where r is the distance between an external point and the snake. User interaction can help the snake correct its solution. This is because snakes are local optimizers that require a good initialization.

The curve moves using discrete time step Euler equations.

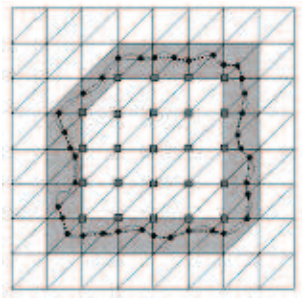


Figure 2.27: Simplex (ACID) grid with snake nodes sampled from the intersection (from [51])

2.3.2 ACID Snakes and Surfaces

McInnerney and Terzopoulos [51] suggest the use of a particle system snake expanding by an inflation force. Lagrangian motion equations model the dynamic behavior. The image space is divided into triangles or simplexes (ACID grid) as depicted in fig. 2.27. The nodes of the snake are re-parameterized after M time steps by a new set of nodes computed by intersecting the evolving snake with the edges of the ACID grid. They generalize this approach to 3D in [50]. The advantages of such an approach are:

1. It simulates the topological adaptivity of propagating fronts.
2. It enables user interaction and smoothness constraints of traditional snakes that front propagation does not provide.
3. It enables the merging and splitting of different snakes and solves problems like self intersection using the ACID grid that maintains one intersection point per edge.

Applications Complex topologies considered for segmentation are 2D and 3D images of blood vessels, corpus callosum, cross sections of vertebra and MR brain images.

Method The snakes nodes are indexed as $\{\mathbf{x}_i(t) = (x_i(t), y_i(t)), i = 1 \dots N-1\}$ that form a closed boundary. The equation of motion is described by the balance of internal and external forces

$$\gamma_i \frac{d\mathbf{x}_i}{dt} + a\alpha_i + b\beta_i = \rho_i + \mathbf{f}_i \tag{2.80}$$

where γ_i is the internal damping coefficient and α_i maintains even distances between nodes and β_i maintains resistance to bending deformations

$$\alpha_i(t) = 2\mathbf{x}_i(t) - \mathbf{x}_{i-1}(t) - \mathbf{x}_{i+1}(t) \tag{2.81}$$

$$\beta_i(t) = 2\alpha_i(t) - \alpha_{i-1}(t) - \alpha_{i+1}(t) \tag{2.82}$$

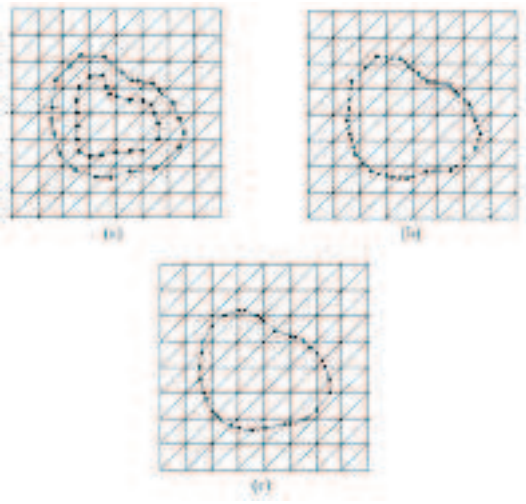


Figure 2.28: T-Snake re-parametrization (a) T-Snakes expands darning deformation step (b) new nodes are sampled by intersection with ACID grid (c) new T-snake formed (from [51])

The inflation force pushes the boundary outwards until it hits the edges defined by the direction normal to the node $\mathbf{n}_i(t)$ and F that maintains the direction within the object region

$$\rho_i(t) = qF(I(\mathbf{x}_i(t)))\mathbf{n}_i(t) \tag{2.83}$$

$$F(I(x,y)) = \begin{cases} +1, & I(x,y) \geq T \\ -1, & otherwise \end{cases} \tag{2.84}$$

The external force $\mathbf{f}_i$ is maintained by a Gaussian smoothed gradient. The discrete time step equation becomes

$$\mathbf{x}_i^{(t+\Delta t)} = \mathbf{x}^{(t)} - \frac{\Delta t}{\gamma}(a\alpha_i^{(t)} + b\beta_i^{(t)} - \rho_i^{(t)} - \mathbf{f}_i^{(t)}) \tag{2.85}$$

The algorithm alternates between calculating deformations using eq. 2.85 within M-Time steps then it re-parameterize the T-Snake nodes using the ACID grid as shown in fig. 2.28 such that the element cannot be moved to make a grid vertex outside the T-snake when it was inside. By maintaining a list of inside vertices, we keep track of the interior region of the snake. Self intersections and merging T-snakes are automatically handled in the ACID grid by maintaining a single intersection point when the T-snake intersects itself at the edge several times or by deleting the intersection points crossing the edge that come from colliding T-snakes thus merging them. The re-sampling of particles by automatically using the grid is an improvement over the dynamic particles approach.

2.3.3 Front Propagation Model

Malladi et al. [48] propose to define a closed curve that expands as a moving front where each point on the surface has a velocity proportional to curvature which gets stopped at boundaries.

Applications Complex topologies considered for segmentation are 2D images of blood vessels and other organs like cross sections of liver.

Method Let $\gamma(0)$ be a closed initial curve in 2D. Let $\gamma(t)$ be the family of curves generated by moving $\gamma(0)$ along its normal vector field with speed $F(\kappa)$ as a given scalar function of curvature κ. Let $\mathbf{x}(s,t)$ be the position vector which parameterizes $\gamma(t)$ by $0 \leq s \leq S$.

A level set $\Psi(\mathbf{x},t) = d$ assigns each point $\mathbf{x}$ in space a real value d which represents the distance to the surface or curve $\gamma(t)$. A positive d means a point outside the curve and a negative d means a point inside the curve.

$$\gamma(t) = \mathbf{x}|\Psi(\mathbf{x},t) = 0 \tag{2.86}$$

Define $\mathbf{x}(t)$ as a point on the front $\gamma(t)$ and $|\mathbf{x}_t| = F(\mathbf{x}(t))$ and the vector $\mathbf{x}_t$ normal to the front at $\mathbf{x}(t)$, then

$$\Psi(\mathbf{x}(t),t) = 0 \tag{2.87}$$

By differentiating 2.87 with respect to t and applying the chain rule we get

$$\Psi_t + \sum_{i=1}^{N} \frac{\partial \Psi}{\partial x_i}\frac{dx_i}{dt} = 0 \tag{2.88}$$

where x_i is the ith component of $\mathbf{x}$, hence

$$\Psi_t + F|\nabla\Psi| = 0 \tag{2.89}$$

The discrete finite difference form of eq. 2.90 can be evolved in time using a uniform grid of spacing h where at every location i,j

$$\frac{\Psi_{i,j}^{n+1} - \Psi_{i,j}^{n}}{\Delta t} + F(\nabla_{i,j}\Psi_{i,j}^{n}) = 0 \tag{2.90}$$

where $\nabla_{i,j}$ is the finite difference approximation of Ψ and $F = F_0 + F(\kappa)$ where F_0 is a constant. The direction of propagation is given by $\mathbf{n} = \nabla\Psi$. The curvature is given by

$$\kappa = -\frac{\Psi_{xx}\Psi_y^2 + 2\Psi_x\Psi_y\Psi_{xy} + \Psi_{yy}\Psi_x^2}{(\Psi_x^2 + \Psi_y^2)^{3/2}} \tag{2.91}$$

To stop the propagating front at edge boundaries one approach considered is to multiply F by κ_I which depends on the image gradient convolved by a Gaussian kernel

$$F_I(x,y) = \kappa_I(x,y)F \tag{2.92}$$
$$\kappa_I(x,y) = e^{-|\nabla G_\sigma * I(x,y)|} \tag{2.93}$$

2.3.4 Dynamic Particles

Szeliski et al. [72] defines a dynamic system of evolving oriented particles expanding into the object surface. New particles are added dynamically to the set. Potential internal forces are defined to maintain an even and smooth distribution of particles on the surface. A triangulation algorithm then links the particles to form polygon mesh of object surface.

Applications Complex topologies considered for segmentation are 3D images vertebra and other solid objects like a mug and toroidal structures.

Method An oriented particle system defines for each particle a state $(\mathbf{p}_i, \mathbf{R}_i)$ where $\mathbf{p}_i$ is the position and $\mathbf{R}_i$ is the 3×3 rotation matrix that defines the orientation of the particles coordinate frame and the third column defines the normal $\mathbf{n}_i$. Given two particles i, j, we define $\mathbf{r}_{i,j} = \mathbf{p}_i - \mathbf{p}_j$. Potentials were defined to maintain the geometric smoothness between particles and they are:

1. Long range attraction forces and short range repulsion forces

$$\phi_{i,j}(\mathbf{r}_{i,j}) = A\|\mathbf{r}_{i,j}\|^{-n} - B\|\mathbf{r}_{i,j}\|^{-m} \tag{2.94}$$

2. Co-planarity potential

$$\phi_P(\mathbf{n}_i, \mathbf{r}_{i,j}) = (\mathbf{n}_i \cdot \mathbf{r}_{i,j})^2\psi(\|\mathbf{r}_{i,j}\|) \tag{2.95}$$

   where ψ is a monotone decreasing function.

3. co-normality potential

$$\phi_N(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{i,j}) = \|\mathbf{n}_i - \mathbf{n}_j\|\psi(\|\mathbf{r}_{i,j}\|) \tag{2.96}$$

4. co-circularity potential

$$\phi_C(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{i,j}) = ((\mathbf{n}_i + \mathbf{n}_j) \cdot \mathbf{r}_{i,j})^2 \psi(\|\mathbf{r}_{i,j}\|) \tag{2.97}$$

The overall interaction between two particles is the weighted sum of all the forces

$$E_{i,j} = \alpha_{i,j}\phi_{i,j}(\mathbf{r}_{i,j}) + \alpha_P\phi_P(\mathbf{n}_i, \mathbf{r}_{i,j}) + \alpha_N\phi_N(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{i,j}) + \alpha_C\phi_C(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{i,j}) \tag{2.98}$$

The total internal energy is computed by summing over all the inter-particle energies. Euler's method is used to model the dynamic behavior of particles. As particles expand and reach surface boundaries new particles are inserted using two rules:

1. If two neighboring particles have a sufficient distance between them say $d_{min} \leq d \leq d_{max}$ and the candidate particle midway between them is at least further by $0.5d_{min}$ from any other particle.
2. If the number of immediate neighbors of a particle falls with the range $n_{min} \leq n_N \leq n_{max}$ and the angle between two successive neighbors is within a suitable range $\theta_{min} \leq \Delta\theta \leq \theta_{max}$ projected into the particles local x, y -plane, then particles are added to fill the gap.

2.3.5 Re-tiling Polygons

Turk [75] defines a method for computer graphics to reconstruct polygonal meshes of the same shape at different resolution levels. The method uses a sampling method similar to dynamic particles with distances weighted by local curvature so that the areas of higher curvature get more vertices. The number of vertices determines the resolution level. A local tessellation algorithm constructs a triangular mesh from these vertices for rendering.

Applications Re-tiling complex 3D objects like molecules, vases.

Method The surface to be re-sampled is randomly sprayed initially with a user specified number of vertices. A relaxation algorithm is applied to repel each point from its neighboring points. The basic operation is to project all neighbors of a point to a plane tangent to that point. The repelling force of each neighbor is calculated and the point is moved in the direction of total force. The radius of repulsion is adjusted such that they are reduced at surfaces with high curvatures. This has the effect of concentrating vertices at surface points with high curvature thus preserving the geometry of the samples shape.

Shape is sampled at different resolutions according to the following procedure as depicted in fig. 2.29:

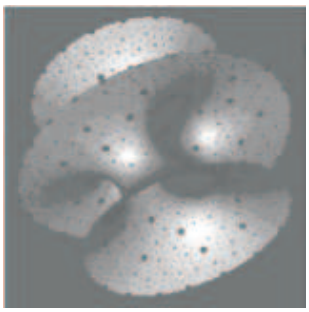


Figure 2.29: Shape sampled at 3 resolutions where level 1-vertices are the larger points and so on (from [75]).

1. The user initially places n_1 vertices and runs the relaxation algorithm.
2. A new set of n_2 vertices are sprayed on the surface and the relaxation algorithm is run again but fixing the positions of the n_1 vertices of the last level.
3. The process is repeated again fixing the location of the vertices of all the previous levels.

A tessellation is defined which uses the original polygonal surface to construct a triangular mesh that preserves the topology of the shape.

2.3.6 Deformable Organisms

A deformable organism [38] is structured as a muscle-actuated body whose behavior is controlled by a brain that is capable of making both reactive and deliberate decisions based on sensory data. This cognitive ability is able to evaluate the relative importance of image features at each segmentation stage to escape false interpretations that other methods tend to latch onto.

Applications Segmentation of corpus callosum in MR images of the brain.

Method The deformable organism is a layered architecture which consists of:

- Geometric representation
- Motor system
- Perception system
- Behavioral/Cognitive system.

The geometric representation of a deformable organism specifies its shape and morphology. For example, a tapered shape consisting of one medial axis and a border silhouette.

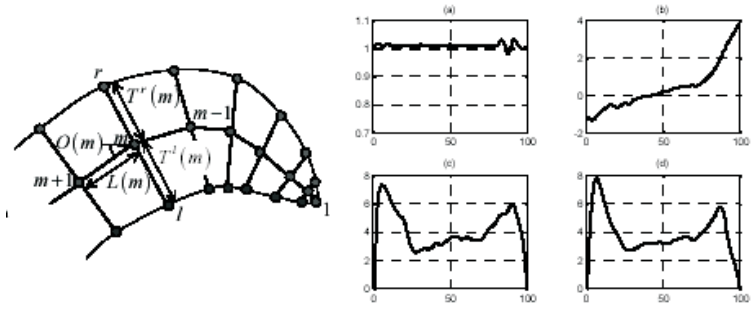


Figure 2.30: Geometric structure of a deformable organism and its length, orientation, left and right thickness profiles (from [38])

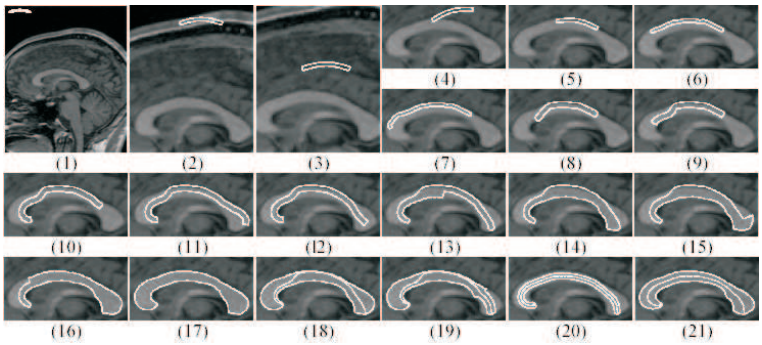


Figure 2.31: Progression of the deformable organism to segment the CC (from [38])

Such a shape can be specified by many profiles such as length, orientation, right thickness, left thickness and orientation as depicted in fig. 2.30.

The motor system is a set of parameterized procedures that implement complex transforms using deformation actuators. Deformations range from bulging one part, smoothing the boundary, elongation, turning the medial axis etc. These deformations can be described as

$$p_d = \overline{p_d} + \sum_l \sum_s [M_{dls} w_{dls} + \sum_t \alpha_{dslst} k_{dlst}] \tag{2.99}$$

where p is the shape profile, d is the transform type, $\overline{p}$ is the average shape profile, k is an operator profile, l, s are location and scale of the deformation, t is the operator type (e.g. Gaussian, triangular), α is operator amplitude, M are the variation modes for a specific d, l, s and w contains the variation mode weights.

The perception system consists of image sensors placed at medial or boundary points that can measure anything from intensity, gradient, Hough transform etc.

The behavior/cognitive system is a plan to carry out active search for image features by collecting information from sensory data and triggering deformation controllers to bring the organism closer to its target segmentation. An example of this behavior is depicted in fig. 2.31.

2.4 Hybrid Models

In the last three sections we discussed the main ways shapes have been modeled. Statistical approaches predict changes in shape based on training data. Structural approaches divide shapes into smaller components thus enabling comparison and simplification of representation. Dynamic models can track and segment complex shapes. In certain problem domains the requirements of a shape model may not be enough to be fulfilled by either one of these models alone. In such a case, a new representation that combines one or more of the three representations is needed. Two models will be reviewed in this section: Pictorial Structures [31] and FORMS [80]. They have elements of both statistical and structural models. But as we shall see Pictorial Structures model deformation between parts a fixed structure. FORMS models viariable deformable parts but no co-deformation between them is described. This will gives rise to the ASSM modelwhich can do both.

2.4.1 Pictorial Structures

The basic idea is to combine statistical and structural models by defining the shape as a tree of structural components and modeling the relation between each edge connected pair statistically. The maximum likelihood algorithm computes the optimal parameters for the model. After building the model, a MAP distribution is used for matching. By sampling from the MAP distribution the model implements a global search strategy. It utilizes the relationships between parts to find more plausible and accurate matches.

Applications Finding facial landmarks and pose estimation of articulated human body [31]. See figures 2.32 & 2.33.

Method The model defines a graph $G = (V, E)$, where the vertices $V = \{v_1, \ldots, v_n\}$ correspond to shape parts and every edge $(v_i, v_j) \in E$ indicates a statistical dependency between parts v_i, v_j. Specifically, an instance of a shape is given by $L = \{l_1, \ldots, l_n\}$ where l_i is a random variable which indicates the location of part v_i in the image I. The prior distribution over the object configurations $p(L|\theta)$ is a Markov Random Field with the structure specified by the graph G. Using Bayes rule, the prior distribution of an object configuration given an observed image I is

$$p(L|I, \theta) \propto p(I|L, \theta)p(L|\theta) \tag{2.100}$$

where $p(I|L, \theta)$ is the probability of observing an image I given the object configuration L. θ are the model parameters. Assuming the parts don't overlap and that each shape part has appearance parameters $u = \{u_i|v_i \in V\}$, we can define $p(I|L)$ as the product of individual likelihoods

Figure 2.32: Detection of landmarks in human faces (from [31])

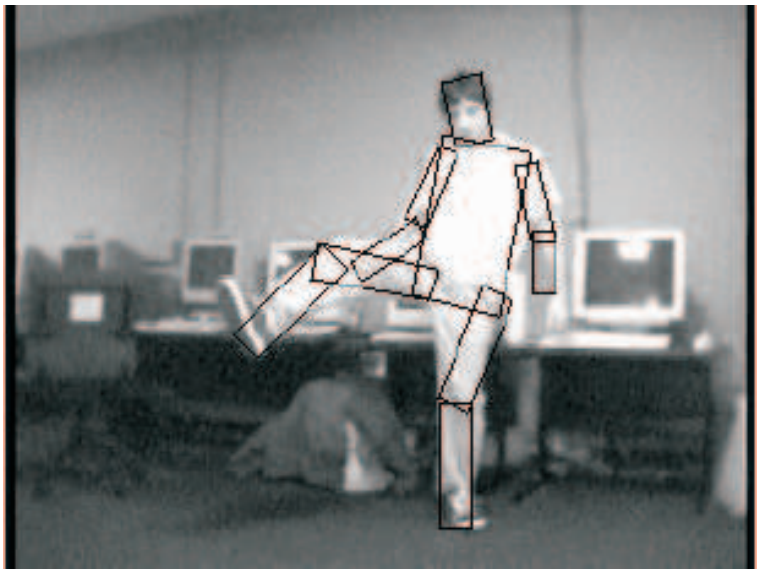


Figure 2.33: Estimating pose for a moving human body (from [31])

$$p(I|L,u) \propto \prod_{i=1}^{n} p(I|l_i,u_i) \tag{2.101}$$

Since the dependencies between parts form a tree the prior joint distribution between parts $p(L)$ is expressed as

$$p(L) = \frac{\prod_{(v_i,v_j)\in E} p(l_i,l_j)}{\prod_{v_i\in V} p(l_i)^{deg(v_i)-1}} \tag{2.102}$$

where $deg(v_i)$ is the degree of vertex v_i in the adjacency graph. The prior distribution between two parts is defined as a normal distribution as follows

$$p(l_i,l_j|c_{ij}) \propto N(T_{ij}(l_i) - T_{ji}(l_j), 0, \Sigma_{ij}) \tag{2.103}$$

where T_{ij}, T_{ji} and $\Sigma_{i,j}$ are the connection parameters encoded by c_{ij}. The functions T_{ij} and T_{ji} together represent the relative locations between parts v_i and v_j. Σ_{ij} measures the spring stiffness connecting v_i and v_j.

After specifying the model, we must find the optimal model parameters θ. To do that a set of training images $I^1,\ldots,I^m$ and corresponding object configurations $L^1,\ldots,L^m$ are required. The appearance parameters u are learned using

$$u_i^* = arg\,max_{u_i} \prod_{k=1}^{m} p(I^k|l_i^k,u_i) \tag{2.104}$$

The connection parameters c_{ij} are learned using an ML estimate

$$c_{ij}^* = arg\,max_{c_{ij}} \prod_{k=1}^{m} p(l_i^k,l_j^k|c_{ij}) \tag{2.105}$$

The last step is to find an optimal connection tree from the completely connected graph between nodes. This is done by finding the minimum spanning tree where the weight assigned to every edge is $-log\,q(v_i,v_j)$. $q(v_i,v_j)$ represents the quality of connection between every node pair. It is defined as

$$q(v_i,v_j) = \prod_{k=1}^{m} p(l_i^k,l_j^k|c_{ij}^*) \tag{2.106}$$

After adequately estimating model parameters, we can use the model for matching. Matching is done by both finding the MAP estimate of the object location given the observed image and also by sampling object configurations from the posterior distribution. The MAP estimate is the object configuration with the highest probability given as

$$L^* = arg\,max_L p(L|I,\theta) = arg\,max_L \left(\prod_{i=1}^{n} p(I|l_i,u_i) \prod_{(v_i,v_j)\in E} p(l_i,l_j,c_{ij}) \right) \qquad (2.107)$$

By sampling the MAP estimate, we implement a global search algorithm that is not sensitive to initialization and can find object configurations wherever they are in the image.

This model has been used in two applications with increasing complexity: Facial landmark detection and pose estimation of human body movement.

In the case of facial landmarks, the location of each landmark is described only by an (x, y) position. The appearance parameters around each landmark are a vector of Gaussian derivatives of different orders, scales and orientations. The model was tested on both 5 and 9 landmarks.

The second application is pose estimation of the human body. Binary images were formed by subtracting images containing the background alone from images containing a moving person. The problem is to estimate the best pose of the articulated body given these binary images. The model is divided into articulated rectangular parts. Each rectangle is parameterized by the vector (x, y, s, θ) where (x, y) represent the location of the centroid, s is the foreshortening scale, and θ is the orientation. The geometry between parts is captured by two parameters: θ_{ij} is the relative orientation and the location of the joint connecting them is (x_{ij}, y_{ij}). The appearance parameters measure how much each part covers the foreground pixels. Experiments have shown that the mutual information derived from the structural model is a strong factor in finding the pose robustly. Problems still occur for estimating the pose of occluding parts.

2.4.2 FORMS

FORMS [80] divides shape silhouettes into parts from the medial axis transform. Each part is represented as either a deformable worm or circle. The model stores two data structures to represent the training data set: Connectivity graphs representing connected components and a table of parts addressed by the deformation parameters of each part. When matching a new shape, the model finds the best fitting shape in its database based on both the similarity of deformation parameters and the nearest matching connectivity graph. To find similar graphs, a set of graph operators is used to make the graphs corresponding to each other.

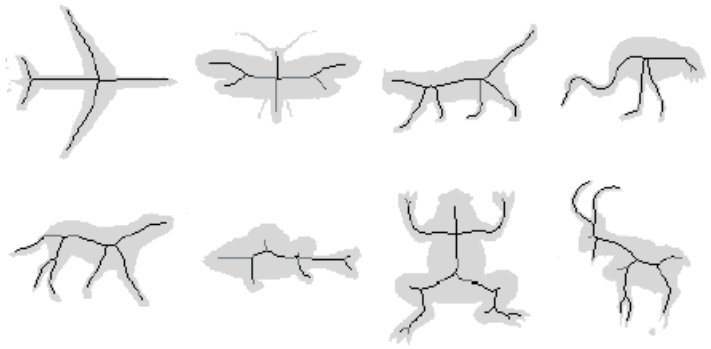


Figure 2.34: Some skeletons (from [80])

Applications Learning and matching of a database of 35 shape silhouettes [80]. Some samples are shown in fig. 2.34.

Method FORMS begins training its database in three steps: First it finds the medial axis transform for each silhouette. Then it divides the shape into parts at points where the medial axis branches. After that it builds a graph of the shape model and calculates for each divided part the deformation parameters.

FORMS defines two deformable shape primitives for each part it divides as shown in fig. 2.35: The worm (that represents elongated parts) and the circle (that represents joints and short ends).

The worm shape consists of both a single axis and the ribs. The axis is uniformly sampled by n points along its path. The coordinates of the sampled points are stored as a vector $\overrightarrow{X} = (x_1, y_1, \ldots, x_n, y_n)$.

The ribs are represented as the distance to the boundary perpendicular to the axis. The two sides are symmetric so the deformations are represented as a single vector $\overrightarrow{R} = (r_1, r_2, \ldots, r_n)$

The second shape primitive is the circle. It is represented by dividing its circumference into equal angular intervals and measuring the length of the ray from the center to the boundary forming a vector $\overrightarrow{C} = (d_1, d_2, \ldots, d_m)$.

The deformation modes of each shape part are characterized by either finite element methods in cases where no sufficient training data exist or principal component analysis otherwise.

When confronted with a new shape, FORMS splits it at junction points where the medial axis branches as shown in fig. 2.36. It builds a skeleton graph of the model and calculates the deformation parameters of each part in the skeleton. It decides if each part is a circle

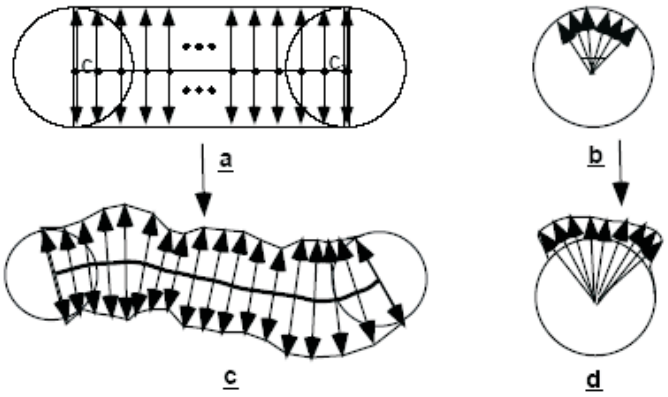


Figure 2.35: Deformable primitives: The worm and the circle (from [80])

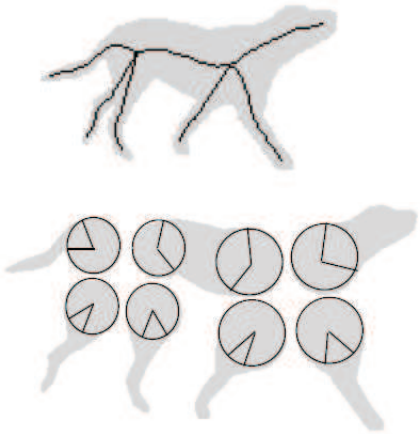


Figure 2.36: Segmentation of a dog (from [80])

or a worm based on a shortness measure. The information collected from all the shape are stored in two databases:

1. A database of skeleton graphs where each model may have several skeletons due to change of pose or viewpoint.
2. A content addressable memory called the butcher's shop as shown in fig. 2.37. Each cell in this data structure is addressed by both the deformation parameters and a label indicating what part of an object it belongs to (e.g head of a dog).

After storing all model representations in the database, we have to use this information to match new shapes. The similarity measure between a model M and a shape instance D is described by first defining a mapping function Φ from the parts of M to the base parts of D. The similarity of a single part m of M and a part d of D is defined by

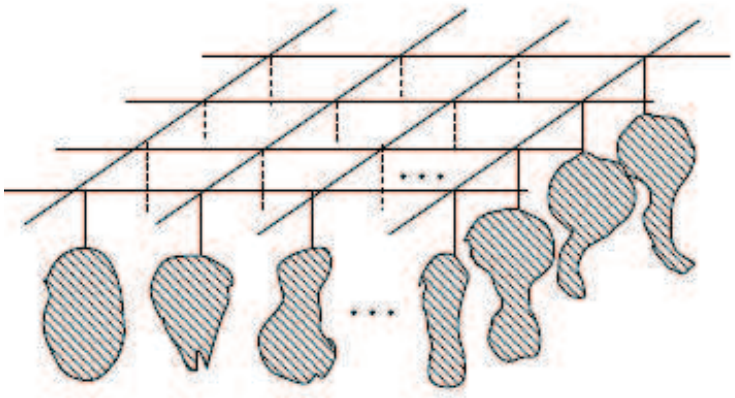


Figure 2.37: The butcher shop (from [80])

$$P_{match}[m,d] = \frac{1}{Z} exp^{-\sum_{i=0}^{k} \frac{(\alpha_i - \beta_i)}{2\sigma_i^2}} \tag{2.108}$$

where $\alpha_i, \beta_i, i = 1 \ldots n$ are the deformation parameters of m, d respectively and σ_i^2 are the variances of the deformation parameters of m.

The similarity between the model M and the shape D under the match Φ is defined by the probability

$$P_{\Phi}[M,D] = \prod_{\Phi(m_i)\neq\phi} P_{match}[m, \Phi(m)] \prod_{\Phi(m)=\phi} P_{missing}[m] \prod_{\Phi^{-1}(d)=\phi} P_{extra}[d] \tag{2.109}$$

where $P_{missing}, P_{extra}$ are the penalties of extra parts in M and D respectively. They are defined as follows

$$P_{missing}[m] = \frac{1}{Z_1} exp^{-\lambda_1 \frac{A(m)}{\frac{1}{n}\sum_{i=1}^{n} A(m_i)}} \tag{2.110}$$

$$P_{extra}[d] = \frac{1}{Z_2} exp^{-\lambda_2 \frac{A(d)}{\frac{1}{n}\sum_{i=1}^{n} A(d_i)}} \tag{2.111}$$

where λ_1, λ_2 are scaling constants and $A(m), A(d)$ are the relative importance of the part in model M and data D, respectively.

In addition to the similarity measure defined above, FORMS defines four graph operators to trim the model graph and bring it closer to the data graph. The operators are :

Cutting model branches off, merging two connected junctions, concatenating branches together, and shifting a branch to another.

The experiments show good matching results even in cases of missing parts and occlusion.

2.5 Comparison Between Shape Models

In the previous sections a detailed survey of shape models was conducted. The section in hand contains a summary of these models, advantages, limitations and application domains of each model class. From this we draw conclusions about the new shape model.

Table 2.1 shows the comparison between the different shape models. This table leads to the following observations:

Statistical models require a suitable sample space of similar shapes. The sample space must be sufficiently large to train the model on all the variations in shape. Once the training is done, fitting the model to new instances is a very stable and noise-robust process. All statistical methods require a preprocessing step of landmarking all samples. This can be time-consuming, specially if landmarking is carried out manually. In terms of application, statistical models are suitable for deformable shapes which have a fixed structure and many instances. This is why they are applied in medicine specially for segmenting specific organs such as corpus callosum in the brain and bone segmentation. In cases were motion is needed to be tracked, the object of interest must exhibit periodic motion such as the heart. Statistical models generally have a compact representation of an object instance making them suitable for data compression in cases where a database of the same object exists.

Structural shape models can be divided into constrained and unconstrained models. Constrained models are those that use structural constraints such as grammars to describe allowable shape combinations. This prior knowledge can be used to generate plausible shape structures. In addition to that, the prior knowledge can be used to correct erroneous results found by segmentation modules (for example, see [7]). The unconstrained structural models have no prior knowledge about how shapes combine but they are capable of abstracting complex structures into either a small set of connected shape primitives or a representation by a small parameter space. Structural models capable of multi-resolution shape representation can separate global from local shape features. Shape abstraction and multi-resolution give structural models the ability to compare and search different shapes. This is the application domain where structural models are mainly used. Registration methods were included into structural models because they describe structural information of a shape even for a fixed template.

Dynamic models define boundaries evolving in time attracted by both internal and external forces. Internal forces define the smoothness constraints that regularize the boundary and help the model to avoid image noise. External forces can be either image features or user interaction forces that modify the model to the desired fit. Although snakes use local

optimization which makes them sensitive to initialization, dynamic models have evolved to be more global and less initialization sensitive. This feature enables these models to find complex shape boundaries. Because of the dynamic nature of those models, they are suited for motion tracking applications. Another important application domain is the segmentation of static images when the boundary of the object contains relevant information. These models do not describe structural information about shape nor explain deformations of these objects. Because of this they cannot be used to compare shapes or compactly represent them.

From the previous paragraph, we can now formulate the following hypothesis: A more general shape model should represent the prior shape knowledge in both how shapes statistically deform and the structural relations between them. Deformation is no longer confined to a single shape but to co-deformation between several sub-shapes. This combination enables noise robustness from statistical models and validation of structure from structural models. There can be many other implications for such a shape model. One implication is the ability to verify a shape based on its context of neighboring shapes. This means that the shape's neighborhood can provide information on what the structure type is and how it deforms. Another implication is that co-deformation enables a multi-resolution representation of shape and specialization of a shape as it becomes part of a bigger shape hierarchy.

The application domains suited for such a shape model are multi-part objects consisting of reusable deformable components. This can be found in mechanical assembly systems, hand drawn sketches and for content based image retrieval for a database of connected objects such as insects or bones. In such applications multiple samples must be found.

The next chapter will describe how such a hybrid model will be realized.

Table 2.1: Comparison between shape models

	Requires a structurally invariant shape	Statistical analysis of shape	Structural description of shape	Includes prior shape knowledge	Representation by a small feature space	Multi-resolution shape representation	Model evolution by kinetic equations	Initialization sensitive	Robustness to noise	User intercation	Applications*
Active Shape Model	√	√		√	√				√		S
Active Appearance Model	√	√		√	√				√		S
Active Appearance Motion Model	√	√		√	√				√		SM
Probabilistic Registration	√	√		√					√		S
Shape Grammars: Network, Tree			√	√							G
Shock Grammar			√	√					√		SC
L-Systems			√	√							G
Semantic Networks			√	√		√					C
Registration	√			√					√		S
Geons			√		√						RC
Generalized Cylinders			√		√					√	RC
Shape Blending			√		√						C
Super Quadrics			√		√	√	√				RC
Finite Element Model			√		√		√				C
Curvature Scale Space			√		√	√			√		CD
Snakes							√	√	√	√	SM
T-Snakes/Surfaces							√		√		S
Front Propagation							√		√		S
Dynamic Particles							√		√		RS
Re-tiling Polygons						√	√		√		RS
Deformable Organisms	√	√	√	√	√	√	√		√		S
Pictorial Structures	√	√	√	√	√				√		S
FORMS		√	√	√	√		√	√	√		CD

* S=Segmentation, M=Motion tracking, C=Shape comparison, G=Shape generation, D=Content based image retrieval, R=3D Surface Reconstruction

3 Active Shape Structural Model

After the survey in the previous chapter and a comparison between shape models, this chapter will describe in detail the framework for Active Shape Structural Model (ASSM).

The next section will provide a brief informal introduction to ASSM. Section 2 will explain the mathematical and algorithmic formulation.

3.1 Brief Introduction

The Active Shape Structural Model (ASSM) is used to recognize, segment and reconstruct shapes [2]. It models shapes as a direct acyclic graph (DAG) of inter- and intra-connected deformable geometric structures. The deformation of these structures is determined by principal component analysis. The deformable structures used for fitting can be either atomic shapes or relations. Atomic shapes are deformable shapes of a fixed structure. Relations can be parts of bigger relations and can overlap as shown in fig. 3.1.

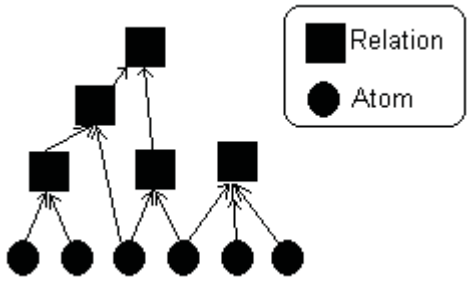


Figure 3.1: Shape representation as a direct acyclic graph of atoms and relations

The ASSM consists of a training module and a recognition module.

In the training module, the structures are specified and their statistical deformations are computed from several sample shapes. Before the sample shapes undergo statistical analysis, they must be landmarked and then aligned with each other to eliminate extrinsic differences. After that, principal component analysis of the aligned samples determines the deformation modes of the structural unit. After all shape structures have been specified and analyzed, the training module builds a shape table.

In the recognition module, the image is searched using the shape table in a bottom up fashion as depicted in fig. 3.2. The iterative search begins with some initial shape that has been fitted to the image. This fitted structure spawns new structural candidates in the image using the shape table. These candidates are fitted to the image and the best ones are selected. The search is iterated on the newly formed set of structures until the image is covered. In addition to spawning and fitting structural candidates, existing structures are grouped to bigger structures.

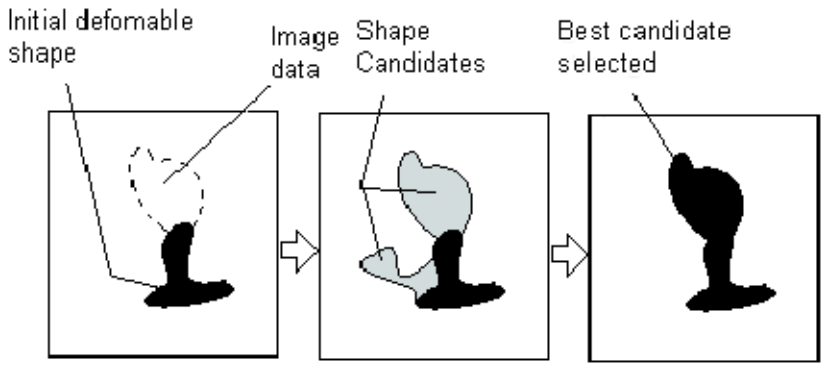


Figure 3.2: Expanding an existing shape by new structural elements then selecting the best candidate.

The relations specify both how shape atoms can be grouped together and their co-deformation. The statistical deformation of a shape atom or a relation becomes more restricted when it becomes part of a bigger relation. This is because it co-varies its deformation with other shapes within that relation. This is the multi-resolution property of the ASSM.

As an example of this property, imagine a deformable model of the hand depicted in fig. 3.3. Assume an atomic shape model of a finger that can vary in length between all possible finger shapes from the thumb to the index finger. As all five fingers are fitted to the hand image, the deformable fingers have a mutual covariance which restricts their deformation such that the thumb cannot be longer than the index. That means every finger deformable model has become more specific in its deformation by being part of a relation. Relations not only specify co-deformation within its parts but also structural constraints between the parts. The structural constraints specify relative orientation, relative scale and connectivity of these parts.

Another property of ASSM is prior knowledge. This means that prior structural and statistical knowledge is used to construct a deformable model based on the features found in image data. This prior knowledge gives ASSM capabilities to:

- Find missing structures when no sufficient evidence exists in the image.
- Find multiple interpretations for the image. After that, eliminate false ones based on image context.

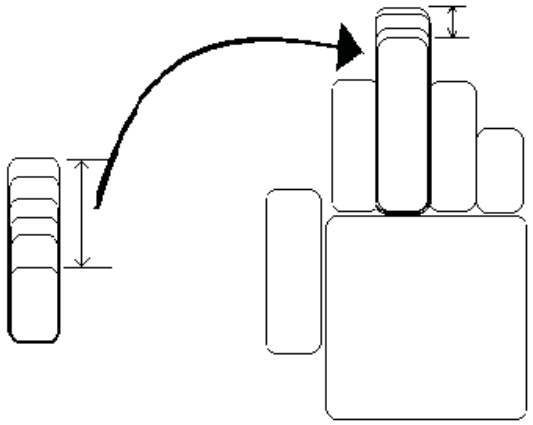


Figure 3.3: The deformable model of a finger is more constrained when it becomes part of a hand relation

To find missing structures, a partially constructed deformable model can be used to predict possible structural candidates. The deformations of these structural candidates can be computed using a regression technique which computes the deformation of the candidate as a function of the partially fitted model. The deformed candidate is accepted when a sufficient fit is achieved by its context neighbors.

An image can be interpreted by many fitting deformable shapes. If interpretations conflict with each other, then we can use relations for conflict resolution. This is done by eliminating shape atoms and their dependent relations by a quality measure. This quality measure favors larger better fitted relations over smaller badly fitted ones, therefore, a shape is not only eliminated based on its fitness to data but also with respect to other related shapes.

To sum up, ASSM is a shape model which binds deformable shapes together to build bigger shape structures. The connectivity between these structures and their co-deformations are statistically analysed. Recognition of images is conducted in a bottom up fashion where shape context eliminates false interpretations and finds missing structures. The next section will define the algorithms needed to implement ASSM.

3.2 Method

The previous section stated that the ASSM consists of a training module and a recognition module. The training module provides prior knowledge to the ASSM. The recognition module uses the prior knowledge of the ASSM to recognize and reconstruct structures from images. The next two sections will discuss these modules in detail.

3.2.1 The Training Module

A shape table of deformable templates is constructed from shape samples by the following steps:

- Complex shapes have to be subdivided into meaningful structural units. The basic types of shape units and the relations between them have to be specified.
- Shapes are landmarked either automatically or manually.
- Shapes are sampled as vectors in geometric space.
- Sampled vectors are aligned for statistical analysis.
- Principal component analysis is applied on the aligned samples and relations.
- Shape regression parameters are computed for relations.

Now each step will be discussed in detail.

First, the user must specify the structures that have to be represented with ASSM. This means the user determines the atomic deformable shape types to be represented. There are criteria to decide when to separate variations into different atoms:

1. This division usually corresponds to some functional part in the application domain for example a machine part in an assembly. In general an atomic shape type must be semantically relevant to the application domain and must have similar comparable sample space. The variation within an atom is crisp and simple enough to be represented by a simple Gaussian distribution.
2. There is a combinatorial relation between the atom and other shapes. This means the atom is reused with different shape types.
3. There is a sufficient number of training samples for this atom otherwise it should be merged with a similar shape type.
4. The atoms do not overlap too much between each other in their distributions which leads to misinterpretation.

After this, the user must specify which deformable atomic types have a relationship between them and how these relationships are modeled. A hierarchy of higher order relations is then specified in a bottom up fashion.

As an example consider a structure like the human skeleton shown in fig 3.4. The most natural structural division is to consider every bone as an atomic deformable shape. The most natural way to define a relationship is to consider any connected set of bones as a relation. The connectivity relations model the range of rotations between bones and the conformation and scale of these shapes. For instance, the fibula, tibia and femur is one relation. A global relation containing the whole skeleton can describe the conformations between all the bones.

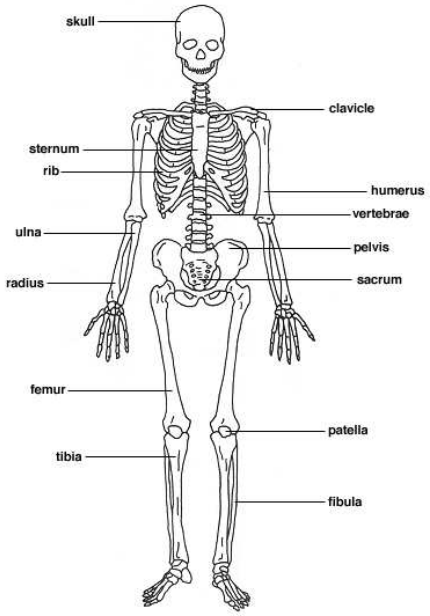


Figure 3.4: Human skeleton, each bone represents a deformable shape.

After specifying the structure, representative shape samples for every structural component is collected. Every shape sample has to be landmarked. Landmarks represent corresponding features between shapes that make them comparable to each other. Fig. 3.5 depicts a manual landmarking of fish silhouettes at corresponding features like the mouth, tail and fins. Landmarking can be done either automatically or with user interaction. When user interaction is used then the user can specify certain key points and intermediate land mark points can be automatically generated between them as depicted in fig. 3.6.

After landmarking we have to represent the geometric features of a shape into a vector. There are several methods to represent object geometry:

1. By taking the coordinates of the landmarks directly. In this case all samples of the shape have to be aligned to a common reference frame to eliminate extrinsic differences due to translation, rotation and scale. After alignment, only intrinsic differences due to shape variation remain.

2. By taking object centered coordinates. This is done by dividing the object into triangles and taking the edge lengths of these triangles. This has the advantage of translation and rotation invariant geometric representation as depicted in fig. 3.7. In fig. 3.8 we see how the variation of some substructures of the mesh can be modeled independently by varying the lengths of edges. The mesh representation is also useful in cases when initially insufficient data samples exist. In this case we can model the mesh variation as spring mass system using finite element methods. Vibration modes with low modality correspond to low frequency shape variation.

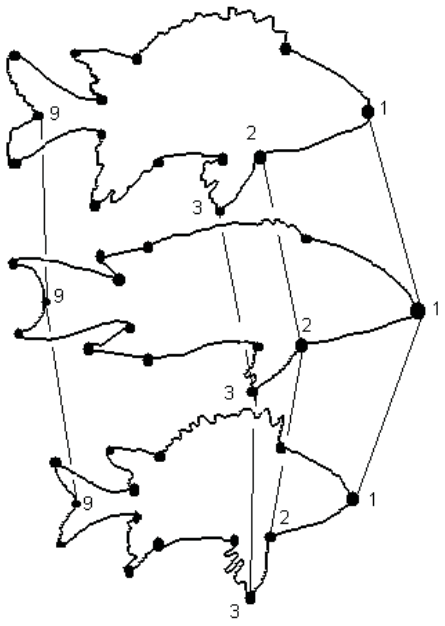


Figure 3.5: Manually landmarking silhouettes of 3 fishes such that corresponding landmarks represent the same shape feature

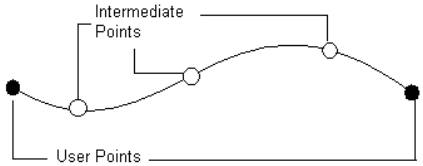


Figure 3.6: Intermediate points placed at equal distancesbetween user specified landmarks

Later, if sufficient samples are provided, the system can switch to statistical variation.

3. Automatic landmarking of a single elongated shape usually involves finding a medial axis of the shape and spawning ribs from the central axis as depicted in fig. 3.9. This is a generalized cylinder representation of the shape. In such cases we need to represent the length of ribs, orientation and length between successive medial segments as depicted in fig. 3.10. A special case is when the shape reduces to a single line then only the coordinates of the medial axis have to be represented. An example of this is a stroke made by a digital pen.

After landmarking the shapes and selecting a good appropriate representation, the shapes must be aligned together. The alignment process is not necessary in the case of geometric representations like the triangular mesh but it is necessary in cases were coordinates are represented directly. In the following discussion we assume a shape is

Figure 3.7: Fish divided into a triangular mesh

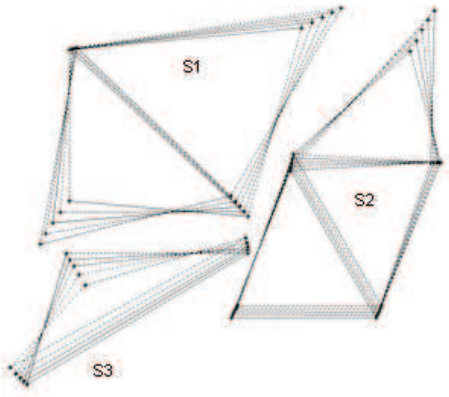


Figure 3.8: Polygon mesh representation of shape where variations in edge lengths are modelled statistically

represented as a set of n (x, y) coordinates directly. Specifically, we assume that a sample vector of landmarks $\mathbf{x}_i, 1 \leq i \leq p$ is represented as $(x_{i,1}, x_{i,2} \ldots x_{i,n}, y_{i,1}, y_{i,2} \ldots y_{i,n})^T$.

A population of samples $S = \{\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_p\}$ is iteratively aligned to an average shape $\overline{\mathbf{x}}$ by finding the transform parameters θ_i that minimizes the average Euclidian distance between the corresponding n points of $\mathbf{x}_i$ and $\overline{\mathbf{x}}$. $\overline{\mathbf{x}}$ is initialized as $\mathbf{x}_1$ and recalculated after every realignment of S. The transformation parameters θ are translation and optionally rotation, scale or all three. The alignment process is described in alg. 1:

Figure 3.9: Fish subdivided into a generalized cylinder

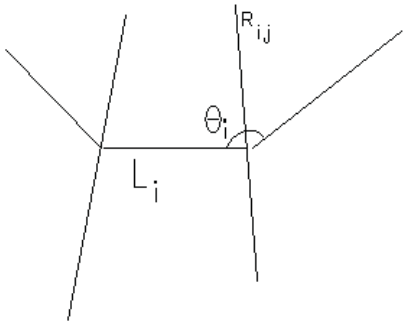


Figure 3.10: Parameters of a generalized cylinder

Data : a population of samples $\{\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_p\}$
Result : mean shape $\overline{\mathbf{x}}$, all the samples aligned
$\overline{\mathbf{x}} \leftarrow \mathbf{x}_1$;
repeat
  for *i←1* **to** p **do**
    Find rigid body transform parameters θ_i which minimize $\|T(\mathbf{x}_i, \theta_i) - \overline{\mathbf{x}}\|$; (where T is the transform function)
    $\mathbf{x}_i \leftarrow T(\mathbf{x}_i, \theta_i)$;
    $\overline{\mathbf{x}} \leftarrow \sum_{i=1}^{p} \frac{\mathbf{x}_i}{p}$;
  end
until $\overline{\mathbf{x}}$ *converges*;

Algorithm 1: Alignment of training sets

Alg. 1 always converges to the optimal solution no matter what the shapes are. This is because the optimal rotation and scale parameters are represented as an over-determined linear system where least square solution is minimized using pseudo inverse. Translation is easy to find as the difference of shape centroids. Fig. 3.11 shows an example of aligning 10 strokes with each other. After aligning all samples of S, we apply principal component analysis to yield a matrix of t principal components $\mathbf{\Phi} = [\phi_1, \phi_2 \dots \phi_t]$. The shape parameters are described by a vector $\mathbf{b}$ such that $\mathbf{x} = \overline{\mathbf{x}} + \mathbf{\Phi}\mathbf{b}$.

As an example, fig. 3.12 shows the first three variation modes of a complex two-stroke signature analyzed from 20 samples taken from the same user. Each stroke is automatically landmarked by setting the first and last points as main landmarks and placing n intermediate points between them. Even with 20 samples we see a clear pattern of variation. For instance, the first variation mode depicts the position variation of the first stroke with respect to the second. The second depicts the double loop variation of the second stroke and so on.

Relations between structural components can model many mutual attributes. These attributes are:

Figure 3.11: Alignment of 10 spring samples

Figure 3.12: First three variation modes of a signature

- Co-deformation happens if the deformation of one part is correlated with the deformation of another part. For example, if we model the variations of skeletons of different vertebrate animals, the shape of corresponding bones is highly correlated with other bone types such that the identification of one bone as belonging to a certain species will determine the deformation of the remaining corresponding bones.

- Connectivity tells which parts connect and at how and what are the connection points. This is specially appropriate for objects connecting at joints. An example of this are bones.

- Relative scale is the ratio of sizes between different parts. For example children have skulls that are relatively larger than the rest of the body in comparison with adults.

- Relative orientation: If parts are connected at joints then the relative orientation reflects the degree of freedom of these parts. As an example, the degree of freedom of an elbow joint is more constrained than the range of angles the shoulder can make.

Given those types of relations we can represent relations at many levels of complexity. The simplest representation of a relation between objects is obtained by landmarking and then aligning group of objects by their coordinates using algorithm 1. The sample vector for the object group $\{\mathbf{x}_1, \ldots \mathbf{x}_n\}$ is formed by simply concatenating their coordinates into

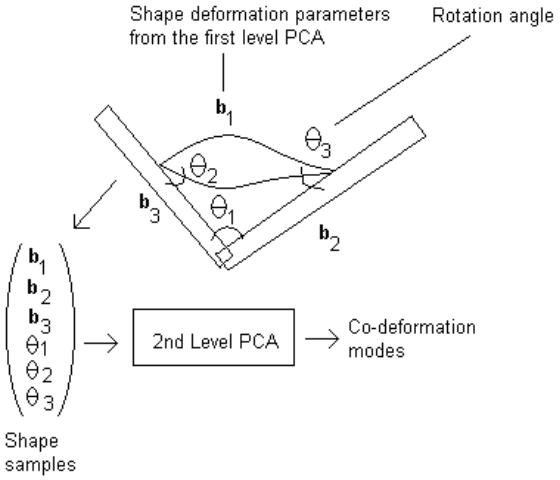


Figure 3.13: 2-Level PCA applied on three shapes

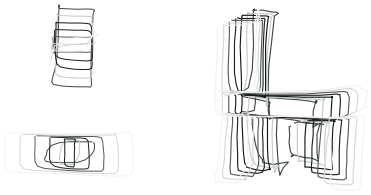


Figure 3.14: left: Variation modes of a rectangle, **right:** a chair consisting of five rectangles that have more constrained co-deformations with each other

one column vector $\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}$. This representation is suitable for cases where there is little rotational variation between the objects of one group.

If we want to model more complex orientational relations like joints between n shape atoms then we can no longer use a linear model to represent rotation . In this case we can define a 2-Level PCA on objects to decouple deformation and connectivity attributes and model each linearly as depicted in fig. 3.13 . This is done in three steps: First, deformation parameters of each shape atom is extracted $\mathbf{b}_i = \mathbf{\Phi}^T(\mathbf{x}_i - \overline{\mathbf{x}}), i = 1 \ldots n$.

The next step is to specify the binary features between each pair of shape atoms i, j, specifically, the relative scale $s_{i,j}$ (ratio of sizes) and orientation $\mathbf{\Theta}_{i,j}$ between atoms i, j. $\mathbf{\Theta}_{i,j} = (\theta_{i,j}, \Delta x_{i,j}, \Delta y_{i,j})$ specifies the diplacement $\Delta x_{i,j}, \Delta y_{i,j}$ of the joint between them and the rotation angles $\theta_{i,j}$ around this joint.

After that we collect all shape parameters and binary features into a sample vector $\mathbf{r} = (\mathbf{b}_1 \dots \mathbf{b}_n, \mathbf{\Theta}_{1,2}, \mathbf{\Theta}_{1,3} \dots \mathbf{\Theta}_{i,j} \dots \mathbf{\Theta}_{n-1,n}, s_{1,2}, s_{1,3} \dots s_{i,j} \dots s_{n-1,n})$.

Given p samples for a relation, we can convert all these samples to the vector representation as described in the previous paragraph: $\{\mathbf{r}_1 \dots \mathbf{r}_p\}$. After that, we form a data matrix R from these samples and apply again principal component analysis. In this case, the variables used for principal component analysis represent different units of magnitude. For example, they represent rotation angle in radians coupled with x, y device coordinates of a landmark. In such a case, all variables have to be normalized to form a correlation matrix. The normal form of a variable x with mean μ and standard deviation σ is $\frac{x-\mu}{\sigma}$. To normalize eigen coordinates we have to simply divide each eigen variable x_i by the square root of its corresponding eigenvalue λ_i : $\frac{x_i}{\sqrt{\lambda_i}}$.

The variation modes extracted from the second level PCA will find deformation and connectivity correlations between the objects. This is specially suitable in cases where elastic shapes interact with each other. As an example, deformation of arm muscles (contraction) is correlated with the elbow joint as shown in fig. 3.13.

After applying the principal component analysis on structural hierarchies, we observe that the variation of an object becomes more specific as it becomes a part of a relation. Fig. 3.14 shows how the variation of a rectangle is more constrained when it is part of a shape group. The significance of this is that a sub-shape changes its variation modes according to its context. This is the shape multi-resolution property of ASSM. Specifically, the variation of a deformable shape is more constrained when it is a part of bigger structural groups. This implies that as more structural parts of the shape are recognized, the more they are able to give information about each other. This feature will be explored next.

Relations can be used to predict new shapes when only some are given using a regression technique. This speeds up searching for relations and also completes missing structures in the image. Principal component regression (PCR) uses the Eigen space or the shape parameter space $\mathbf{b}$ as regression and observation variables. Shape coordinates $\mathbf{x}$ are not directly used because they have a high linear correlation. The necessary condition for regression is that the regressor objects can successfully predict the variation of observation objects. If there is no linear correlation between them then linear regression does not make sense.

Given a relation $R = \{a_1, a_2 \dots a_n\}$ between n atoms of which $A \subset R$ are regression objects and $B \subset R, A \cap B = \phi$ are observation objects and given a population of p samples, we compute a regression matrix $\mathbf{B}$ as follows:

- We align the p samples and compute the latent vectors and roots of the regressors and similarly the latent vectors and roots of the observation objects.
- For every sample $\mathbf{x}_i, 1 \leq i \leq p$, we compute the shape coordinates of regressors $\mathbf{b}_{i,A} = \mathbf{\Phi}^T(\mathbf{x}_{i,A} - \overline{\mathbf{x}}_A)$ and observation objects $\mathbf{b}_{i,B} = \mathbf{\Phi}^T(\mathbf{x}_{i,B} - \overline{\mathbf{x}}_B)$.

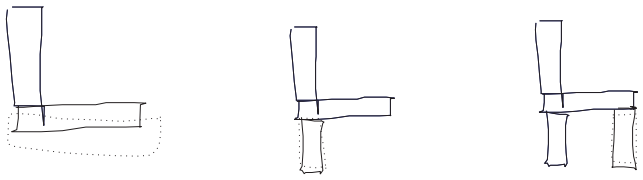


Figure 3.15: A chair modeled as a relation between rectangular single-stroke objects. PCR constructs the expected shape given 1, 2 and 3 regressor objects from left to right respectively. As we can see regression improves its fit to the original data the more regression objects are used.

- We form a regression matrix from shape parameters $\mathbf{R} = [\mathbf{b}_{1,A}^T, \ldots, \mathbf{b}_{p,A}^T]^T$ and an observation matrix of shape parameters $\mathbf{S} = [\mathbf{b}_{1,B}^T, \ldots, \mathbf{b}_{q,B}^T]^T$. Then we compute the regression matrix $\mathbf{B} = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\mathbf{S}$. Let $\theta_{A/B} = (\overline{\mathbf{x}}, \mathbf{\Phi}_x, \mathbf{\Phi}_y, \lambda_x, \lambda_y, \mathbf{B})$ be the regression parameters of A to B.
- For a relation R which consists of n objects or relations $\{r_1, \ldots, r_n\}$ we compute all the regression parameters $\theta_{A/B}, \forall A \subset R, A \neq \phi, B = R - A$.

  Fig. 3.15 shows how PCR is used to predict parts of a chair. We see the match between actual and predicted shapes increase with the number of shapes that are used for regression.

The recognition module builds a shape table that specifies all deformations and regressions of each structural element and the specific relations between these structural elements. Table 3.1 shows the data structure of the shape table. This table represents both shape atoms and relations. The first column represents an identification number for a structural element. The second and third columns list the sub and super structures to which this structural element belongs. The fourth column represents the variation modes of this shape structure. The fifth column represents regression matrices of this shape where the subscripts indicate what sub-structures map to which super-structures. The last column represents the parameters used to weight the relative importance of a given structure and also various thresholds and parameters to place on the structural element.

This shape table is used as prior knowledge to search, reconstruct and group new shapes. This is the task of the recognition module that will be presented next.

3.2.2 The Recognition Module

After constructing the shape table, we can use it to recognize and reconstruct new shapes. This is depicted in algorithm 2. The shape interpretation consists of the following steps: Finding some initial atomic shapes. Then, relations are recognized in the image by using the structural prior knowledge. Relations are also used to generate new structures. The image is then searched for evidence which supports the generated structures. When

Table 3.1: Data structure of a shape table

Structure No.	Sub-structure numbers	Super-structure numbers	Deformation parameters	Regression parameters	Weight
1	{} atomic shape	$\{3, 4\}$	$\{\overline{\mathbf{x}}_1, \mathbf{\Phi}_1, \lambda_1\}$	none	$\mathbf{w}_1$
2	{} atomic shape	$\{3\}$	$\{\overline{\mathbf{x}}_2, \mathbf{\Phi}_2, \lambda_2\}$	none	$\mathbf{w}_2$
3	$\{1, 2\}$	$\{4\}$	$\{\overline{\mathbf{x}}_3, \mathbf{\Phi}_3, \lambda_3\}$	$\{B_{1\rightarrow 2}, B_{2\rightarrow 1}\}$	$\mathbf{w}_3$
4	$\{1, 3\}$	$\{\}$	$\{\overline{\mathbf{x}}_3, \mathbf{\Phi}_3, \lambda_3\}$	$\{B_{1\rightarrow 3}, B_{3\rightarrow 1}\}$	$\mathbf{w}_4$
⋮	⋮	⋮	⋮	⋮	⋮

there is sufficient data to support the relation then it gets accepted. Finally, conflicting interpretations between candidate atoms are resolved using the atom's largest context principle. This means that candidate shapes that belong to bigger relations are favored to single atoms or atoms that belong to smaller relations. Once a candidate atom is selected for removal, all the relations it belongs to are removed. After this brief overview, the details of each step will be presented next.

Figure 3.16 shows a demonstration of the recognition algorithm which will be used to explain the algorithm as we go along each step.

Initialization: The first step of recognition is to find one or more atomic shapes candidates on the image to initialize the recognition process. This is achieved by user interaction or by placing one or more atomic shapes on the image at a position near their intended location. These initial shapes then deform into their correct positions on the image using their features.

Another way is to replace user interaction by using another algorithm to find some atomic shapes such as throwing random templates on the image and finding the largest shape from the best fitting template[8].

The third way is to look for prominent sub-shapes by throwing randomly an atomic shape template on the image. The templates are thrown in different positions, orientations and scales onto the image. These shape atoms are allowed to deform to search for matching image features. Those deformed templates that have a good fit to the image are taken as initial candidates to the image. It is not necessary initially that all the candidates are correct matches because subsequent steps will eliminate false candidates based on the matching algorithm.

Data : Shape Table (ST), Input Image (I)
Result : Recognized shape Instances (RS)
Find some initial atomic shape candidates C_0 and fit them to I;
$RS \leftarrow C_0$;
repeat
  Find all Relations C_1 that can be formed from grouping candidates in RS ;
  for $\forall r \in C_1$ **do**
    Fit r to I;
    if r *fits* **then**
      Add r to RS;
    end
  end
  Generate all potential shapes C_2 from RS using regression for relations in ST;
  for $\forall r \in C_2$ **do**
    Fit r to I;
    if r *fits* **then**
      Add r to RS;
    end
  end
until *no more shapes are added to* RS;
Find all atoms in RS that have a conflict and put it in C_3;
while C_3 *has conflicting atoms* **do**
  Find $a \in C_3$ with minimum cost of removal;
  Delete a from C_3;
end

Algorithm 2: Recognition algorithm

In the case of fig. 3.16 the triangle atom was initialized on the image.

Finding subsequent shapes: The next step is to utilize relations between shapes to find new candidates and eliminate the unlikely ones. This is done by finding sub-shape candidates that can be grouped with each other or by generating new shape candidates and trying to fit them to the image. The sub-shapes are grouped together by finding atomic shape candidates that are both spatially near each other and have the shape types that constitute a candidate relation. A candidate relation is formed when the constituent sub-shapes pass the deformation test. This means if the deformation of the relation is within a specified threshold, then the relation is accepted as a candidate relation. The candidate relation can then be fitted to image using the more specific deformation modes. The deformation modes of a relation contain more information about how individual atoms are related to each other and therefore provides better prior knowledge for fitting better to the image data.

The other way relations can be used is to generate and validate the existence of new shapes and even complete shapes that are missing in the image. This is done by taking an existing candidate subset of shapes and using the relation's regression matrix to generate a new shape candidate. The newly generated shape candidate utilizes the deformation and position information of its neighbors to construct a good initial guess of its deformation and location on the image. After that the generated shape can be locally fitted to the image. There are three possible outcomes of fitting the generated shape:

- The generated shape fits the image. In this case the generated shape is taken as a candidate to be used for further processing.
- The generated shape does not fit the image well but there is strong evidence from the generating relation that it must exist. In this case we accept the generated shape as a candidate for further processing. Strong evidence means that there is high confidence in the relation and that its existing parts fit well with the image. In this case we must set the criteria for accepting a relation where some atoms fit partially. We have to specify for each relation the minimum number of atoms for it to be accepted as a candidate.
- The generated shape and the neighborhood that generated it do not fit well the image. In this case we drop the generated shape from further processing.

In fig. 3.16, 3 potential candidates are shown. The bottom triangle is then merged with the two leg rectangles that matches a relation in the shape table.

Candidate selection: After looping the algorithm and iterating the generation and merging steps for relations, a large set of candidate sets and relations are generated. At this point a decision has to be made which candidates to keep and which to discard. This

is done by determining conflicting shape atoms candidates. A conflict occurs between two shape atoms a_i and a_j if they try to segment the same image region. In other words, if they represent two interpretations for the same part of the image. Let us define the context of a shape atom a as the set of all the super relations containing this atom such that no relation is contained within another relation $context(a) = \{r_1, r_2 \dots r_k\}$. The next step is to compute the cost of removing an atomic object a and all the relations of which it is part of. The cost function must take into account two factors: the size of the relation and the fitting cost.

The size factor means that bigger relations that contain more objects are favored to smaller relations. This is because bigger relations receive more support from their atoms. One possible way to define size is to assign weights to every atom type $w(a)$ that characterizes its importance with respect to other atomic types. The size of the relation r is simply the sum of the weights of its atoms $size(r) = \sum_{a_i \in r} w(a_i)$.

The second factor favors relations that fit better to the image I. We can describe the total cost for removing an atomic object a as

$$cost(a) = \sum_{r \in context(a)} size(r)/dissimilarity(r, I) \tag{3.1}$$

The conflict resolution algorithm is an iterative algorithm which consists of two steps: determining all conflicting atoms then removing the atom with the minimum cost and all its dependent relations. The algorithm terminates when all conflicts are resolved.

Every atom or relation generated has to be fitted to the image. The fitting function uses a similarity measure which is a weighted sum of two factors. The first factor is the image force that pulls the deformable template to the correct image feature. The image features used are application dependent. They can be edge,corner or texture features. The template can utilize [8] a number of these features at different positions.

The other factor is a measure of deformation. If too much deformation is needed to bring the shape to fit the image, then probably it is a false fit.

We define the similarity function as the weighted sum of these two factors:

$$dissimilarity(\mathbf{x}, \overline{\mathbf{x}}, \mathbf{\Phi}, \lambda, I) = f_{deformation}(\mathbf{x}, \overline{\mathbf{x}}, \mathbf{\Phi}, \lambda) + \alpha \cdot f_{image}(x, I) \tag{3.2}$$

$$f_{deformation}(\mathbf{x}, \overline{\mathbf{x}}, \mathbf{\Phi}, \lambda) = \sqrt{\sum_{i=1}^{t} \frac{b_i^2}{\lambda_i}} \tag{3.3}$$

where $(b_1, b_2 \dots b_t) = \mathbf{b} = \mathbf{\Phi}^T(\mathbf{x}_{aligned} - \overline{\mathbf{x}})$ and $\mathbf{x}_{aligned}$ is rigid body alignment of $\mathbf{x}$ to the mean shape $\overline{\mathbf{x}}$. The *dissimilarity* function used to fit both relations and atoms to the image. If the final value of the fitting process is more than a threshold τ, then

the atom or relation is accepted as a candidate otherwise it is rejected. The fitting process is a minimization process on the shape parameter space $\mathbf{b}$ and the rigid body transform parameters $\mathbf{\Theta} = (\theta, s, \Delta x, \Delta y)$ where θ are the rotation parameters, s is the scale, $(\Delta x, \Delta y)$ are the translation parameters.

In fig. 3.16 we see that the circle primitive which is in conflict with the arch relation is removed. The same is true for the other circle primitive. Finally after removing false candidates we end up with the two relations shown in the figure.

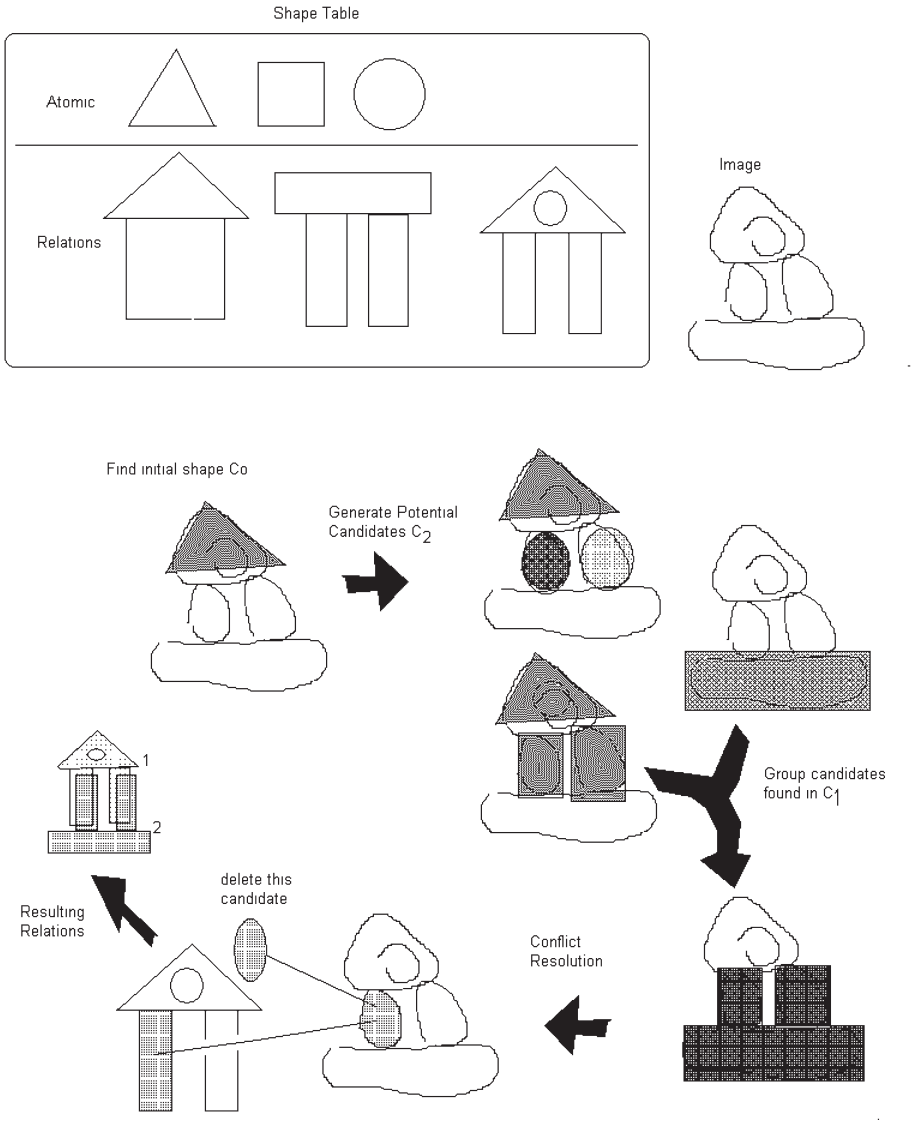


Figure 3.16: The recognition algorithm

4 Applications of ASSM

In the previous chapter a framework for ASSM was defined. This chapter demonstrates this framework on specific applications. The domain of application considered here is hand drawn sketches using digital ink and recognition of ants.

This chapter will is structured as follows: first, the reasons for choosing sketches will be explained. Subsequently, literature pertaining to sketches will be briefly surveyed. The ASSM framework will be adapted specifically to sketches. After this two applications of sketches are considered: recognition of sketches of mechanical systems and using sketches for security and authentication systems. After this, a comparison between ASSM and ASM in using sketches is made. This will demonstrate the representational abilities of ASSM. The next application demonstrates the use of ASSM framework in a more realistic application, namely, the recognition of ant images for biological classification. This is because different ant types have different structural representations. In addition to that an ant's body has a well segmented articulated structure.

The main reason why these applications were chosen is to demonstrate the hypothesis of how the prior knowledge of ASSM of both structure and morphology can help us achieve solutions to problems that the models surveyed cannot do easily. Sketches are an example where both types of knowledge are needed. Using a statistical model alone such as ASM will not represent variation and covariation between structural parts of a sketch as a single distribution.

The use of structural models alone may not be able to capture the morphology with sufficient precision as will be demonstrated in biometric sketches. This precision determines the quality of authentication.

The reconstructive abilities of ASSM both at structure and morphology enables it to find missing or bad parts of a sketch that neither a pure statistical or structural model can do. A statistical model would have to perceive missing parts as a part of its distribution rather than recognizing it as a discrete part. A structural model will not be able to reconstruct the morphology of the bad or missing part based on its surrounding shapes.

In the case of ants, a model with prior knowledge of both structure and morphology is needed. The precise knowledge of both different shape templates and how they are aligned with respect to each other is crucial to segmenting and recognizing an ant. If either component is missing as would be the case for the models surveyed, then this will not be be done easily.

Figure 4.1: A child's drawing representing a simple sketch

After explaining the reasons why these applications are chosen, we we need next to define what is a sketch. A sketch is a set of structurally variable and statistically correlated drawing primitives of different complexity.

The structural variability comes from the tendency of humans to represent objects of the real world with simple drawings. The shapes drawn usually have a simple enough representation that falls within template types. As an example consider fig. 4.1. In this case we see that the head and the eyes are represented as a circle and the upper body has a characteristic shape which is a square. Also the shape of feet is characteristic as hooks turning to the right. In general we observe that semantically similar objects in the real world have similar shape representations in the 2D sketch space.

The statistical correlation means that drawings do not consist of just simple shape templates but also that these templates are strongly related to each other. For example, we observe in fig. 4.1 the spatial positioning of different body parts and the relative scale of these parts to each other. We also see a representation of touching hands which is another relation between the persons represented here.

The shape primitives vary in complexity in fig. 4.1. This can be seen from very simple shapes like the arms to more complicated ones like the two flowers.

Hand drawn sketches were chosen to demonstrate the ASSM because they have the following properties:

1. Sketches are more suitable for shape oriented models as opposed to feature oriented models such as for cursive hand writing recognition.

2. Training and testing data are easy to generate and no special pre or post processing steps are required.

3. If we impose the constraint that no structure is smaller than a stroke, we can easily separate shape sub-structures from each other.

4. Strokes are suitable for statistical analysis because they vary shape in relationship to each other or when drawn by the same user or different users.

5. Sketches can contain missing structures or incomplete information because users tend to use them as an informal way to communicate ideas before committing to them. This means that a sketch model must be able to infer using its prior knowledge what the user means and complete the missing information.

Sketches are gaining increasing importance with the shift to pen based interface as palm and tablet computers are proliferating. Currently sketching systems are employed in the field of design such as: design of user interfaces [47], recognizing mechanical designs [6] and content based image retrieval [77]. Many sketching systems restrict sketch recognition to simple shape primitives like squares, circles, polygons or specific shapes [6, 32]. ASSM describes sketches statistically allowing complex and uniform shape description.

The application of ASSM on sketches will be demonstrated in three ways:

- Qualitatively: As a recognition algorithm for mechanical systems. This is where the ability of ASSM to construct and characterize complex relations is illustrated. All the sketches are generated by a single user.
- Quantitatively: As a biometric recognition system. This is where the structural component of sketches is applied to authenticate different users.
- Comparatively: The active shape model and ASSM will be compared. This is done by embedding all structural and relational variations in a single normal distribution. This distribution is demonstrated on an example that shows the inadequacy and complexity of this representation.

The next section demonstrates the qualitative aspect of sketches.

4.1 Sketch Recognition

In this section the adaptation of ASSM framework on sketches will be explained followed by experimental results and a discussion [3].

Sketches are represented as a sequence of strokes. A stroke is created from the moment the user puts the pen down on the drawing surface until the pen is lifted. Given this representation we make an important simplifying assumption for ASSM: No atomic shape is smaller than one stroke. This simplification avoids connectivity problems when the user connects one stroke with the next one. For example, fig. 4.1 shows two persons with the arms drawn as separate strokes and they are connected with the upper body in the third person.

The reason this simplification is made is to focus on the structuring capabilities of ASSM rather than shape morphology. Another reason is that if the user has the tendency to frequently connect two shapes adjacent in time, they can be modeled as a separate shape template.

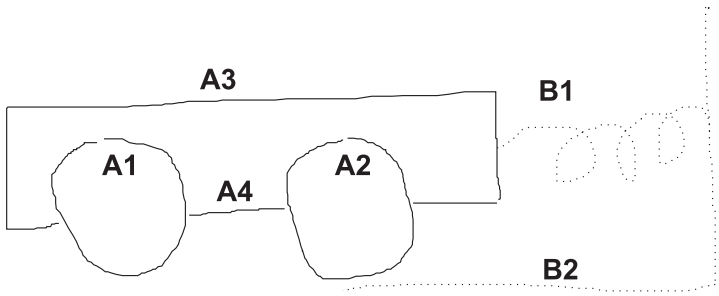


Figure 4.2: Levels of a sketch: (1) Strokes A1 A4, B1, B2 (2) Atoms: Cart, spring (3) Relations: Correlation between the length of the spring and the distance between the cart and the wall.

The sketch is represented at three levels: Stroke, object and relation as depicted on fig. 4.2. The stroke is the most basic unit of shape. An ordered list of strokes representing a single object is a shape atom. The reason for assuming stroke is that users tend to draw the same multi-stroke object in the same order. Groups of atoms, that are statistically correlated together, are combined by relations. A relation may also include other relations. The components of a relation are not drawn in any predefined order.

After defining the structure of a sketch, the training and recognition modules have to be adapted to these structures. The training module is described next.

To collect samples for the training module, we must define how to capture and landmark strokes. A stroke is captured using a digital pen. The pen's device sampled coordinates are stored as a sequence of triplets $\mathbf{s} = ((x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_p, y_p, t_p))$ where $(x_i, y_i), i = 1 \ldots p$ are the device coordinates of a stroke-point and $t_i, i = 1 \ldots p$ is the time in milliseconds from a specified time point (e.g. start of the program).

After storing the stroke's points, we use a B-spline function to interpolate between them with the time t as the parametric variable: $\mathbf{p}(t) = (x(t), y(t)), t_1 \leq t \leq t_p$. Time is used as the interpolating variable because it samples more of the curve at points of high curvature and high detail.

Using time for interpolation enables a simple solution for automatically landmarking a stroke. We use the stroke's first and last points as main landmark points and we generate a fixed number of intermediate points. Experiments have shown this to be adequate even for very complicated strokes consisting of many corners and curves. The errors resulting from displacements of corners are reduced because many landmarks concentrate around them as depicted in fig. 4.3. This is the case also when the same stroke is drawn with different speeds, scales and orientations. To generate a sample vector of n points $\mathbf{x} = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)^T$ from the stroke $\mathbf{p}$ we set $(x_i, y_i) = \mathbf{p}(t_1 + \frac{(t_p - t_1)}{n-1}(i-1))$, i=1...n.

When an atom or relation sample consists of multiple stroke $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_q$, we can create a sample vector by concatenating the corresponding landmarked vectors of these strokes

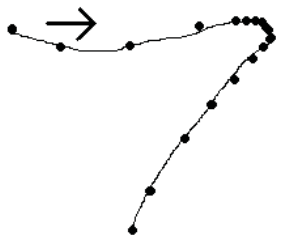


Figure 4.3: Distribution of landmarks around corners in a stroke.

$(\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_q^T)^T$. As a amall note we have to mention that this stroke ordering is done only at training but is not required for recognition.

A more complicated representation of the relation is obtained by specifying joints where parts can rotate. This is modeled using the two level PCA representation described in chapter 3. In this case the second level represents the angle alone when rigid motion is assumed or else the normalized angle and the normalized deformation parameters when nonrigid rotation is assumed. For this application, the one-level representation is assumed.

Training samples for atoms and relations are collected for a given user. Algorithm 1 can be then used to align training samples and the PCA is applied to find variation modes.

After finding the variation modes for every atom and relation, we have to compute the regression parameters for each relation. In this case if a relation R consists of atoms $R = \{a_1, a_2, \ldots, a_n\}$, we compute regression parameters for every proper subset of atoms $X \subset R, X \neq \phi$ to the rest of the relation $Y = R - X$.

When all regression parameters are computed, all the information to build the shape table in the training phase is complete.

Data : $\mathbf{\Phi}, \bar{\mathbf{x}}$, data $\mathbf{x}$
Result : Elastic fitting $\mathbf{x}_{elastic}$, rigid fitting $\mathbf{x}_{rigid}$
$\mathbf{x_0} \leftarrow \bar{\mathbf{x}}$;
repeat
  find a rigid body transform T that minimizes $\|\mathbf{T}(\mathbf{x}) - \mathbf{x_0}\|$;
  $\mathbf{x_1} \leftarrow \mathbf{T}(\mathbf{x})$;
  $\mathbf{b} = \mathbf{\Phi}^T(\mathbf{x_1} - \bar{\mathbf{x}})$;
  $\mathbf{x_2} \leftarrow \mathbf{x_0}$;
  $\mathbf{x_0} \leftarrow \bar{\mathbf{x}} + \mathbf{\Phi b}$;
until $\|\mathbf{x_2} - \mathbf{x_0}\| < \varepsilon$; ;
$\mathbf{x}_{elastic} \leftarrow \mathbf{x_0}, \mathbf{x}_{rigid} \leftarrow \mathbf{x_1}$;

Algorithm 3: Deformable shape alignment algorithm

After constructing the shape table, we must use it to recognize new sketch instances. Recognition requires the definition of an appropriate similarity function. The similarity function is defined between a shape template consisting of n strokes described by the parameters: $(\overline{\mathbf{x}}, \mathbf{\Phi}, \lambda)$ and a sequence of n data strokes in the sketch. The n data strokes are first interpolated by a B-spline for each stroke and then landmarked the same way used in the training module. When landmarking the sketch stroke sequence $\mathbf{x}$, we must use the same number of landmarks as the corresponding strokes in the shape template. This facilitates comparison between the two shapes. After sampling $\mathbf{x}$, an elastic alignment algorithm described in algorithm 3 is used. This algorithm tries to minimize the distance between the corresponding landmarks of the shape template and $\mathbf{x}$. The result of the algorithm is to compute a rigid body transform of $\mathbf{x}$ fitted to a shape template deformed from $\overline{\mathbf{x}}$. After fitting, the similarity function (actually dissimilarity as defined here) is defined as the weighted sum of both the deformation parameters and the maximum distance between corresponding landmark co-ordinates:

$$dissimilarity(\mathbf{x}, \bar{\mathbf{x}}, \mathbf{\Phi}, \lambda) = deformation(\mathbf{x}, \bar{\mathbf{x}}, \mathbf{\Phi}, \lambda) + \alpha \cdot distance(\mathbf{x}, \bar{\mathbf{x}}, \mathbf{\Phi}, \lambda), \quad (4.1)$$

$$deformation = \sqrt{\sum_{i=1}^{t} (\frac{b_i}{\lambda_i})^2} \;\; where \;\; \mathbf{b} = \mathbf{\Phi}^t(\mathbf{x}_{elastic} - \bar{\mathbf{x}}) = (b_1, b_2, ..., b_t),$$

$$distance = max_{i=1}^{p} \|u_i - v_i\| \;\; where \;\; \mathbf{x}_{elastic} = (u_1, ..., u_p), \mathbf{x}_{rigid} = \;\; (v_1, ..., v_p)$$

The specification of both the fitting and *dissimilarity* enables candidate relations and atoms to be accepted or rejected based on some threshold value τ. This threshold can be set to accept a high number of candidates because the conflict resolution step of the ASSM algorithm will be set to reject most false candidates that do not receive sufficient support from their neighbors.

4.1.1 Experimental Results

The goal of the experiments is to demonstrate the abilities of ASSM on sketches of complex mechanical systems. They demonstrate the ASSM model because objects correspond to machine parts and relations represent scale and connectivity constraints. The experiments will demonstrate:

1. fitting of atoms and finding relations.
2. resolving conflicting interpretations using context information from structure.
3. finding missing or incomplete structural elements using the context information.

Fig. 4.4 shows objects used in constructing the sketches. Binary and higher order relations analyze spatial and scale covariance between machine parts as seen in Fig. 4.5.

The ASSM model was trained with $10 - 30$ samples per object or relation drawn by a single person. 10 samples are sufficient to learn the variation modes if the user selects

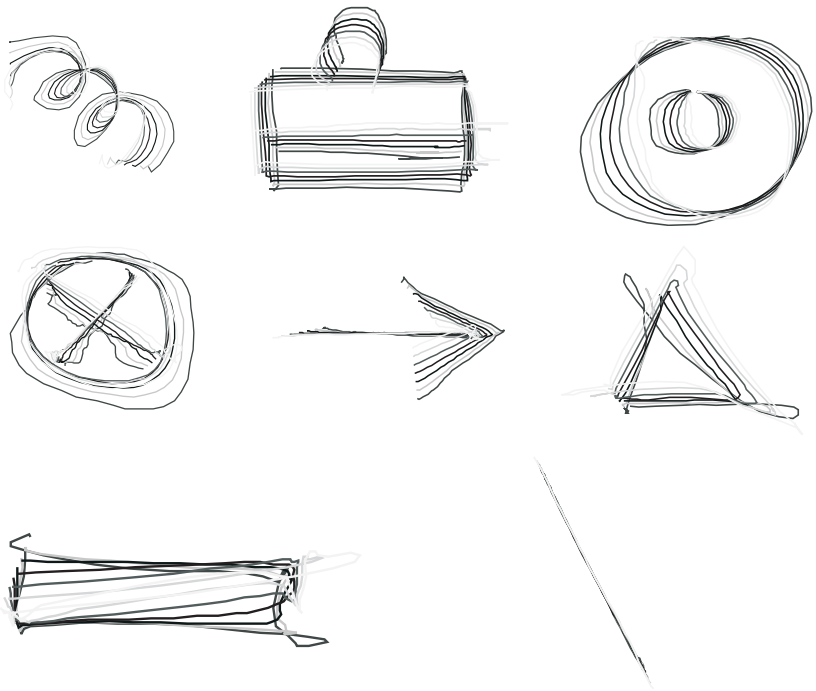


Figure 4.4: Deformable Objects: Spring, Weight, Wheel, Joint, Force, Pivot, Bar and Rope.

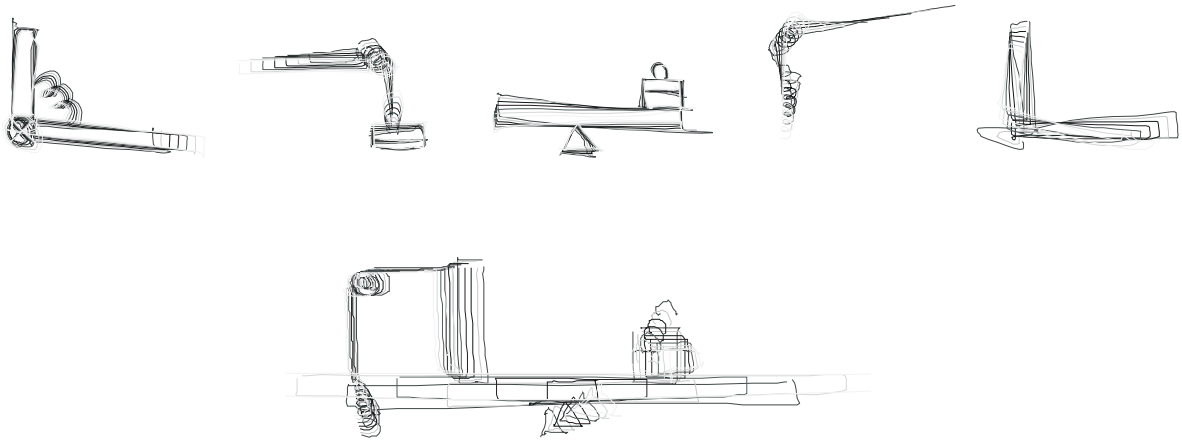


Figure 4.5: Relations: Arm, crane, lever, corner, pulley. The last relation ShockAbsorber consists of a pulley, lever and corner.

a representative training set that captures all the variability. In order to guarantee a smooth probability distribution and a smooth mean 30 samples were used. This number is also a reasonable effort for the user. The number of principal components ranged between 3 for simple shapes up to 12 for the most complex shape. The number of principal components was set to explain 95% of the variation in samples. This number was chosen because it captures the variation accurately but still retaining a low number of principal components. If 98% was chosen then the number of principal components will double and most of the higher order components do not represent significant variation but are noise artifacts. On the other hand, choosing 90% sometimes excludes some components that are relevant to the variation. Fig. 4.6 shows some training data used for machine parts. The samples' variations must reflect some functional aspect for each machine part. For example, the size and orientation of the bar object can vary. The distance between the weight and the pulley can vary in the crane relation and the position of the pivot with respect to the bar can vary in the lever.

Fig. 4.7 shows the results for interpreting a sketch. The left image shows the individual objects with the best fitting shape overlaid and the latent coordinates of each object. The right image shows relations binding these objects. The shock absorber is the largest relation binding three smaller relations.

The algorithm is well conditioned because the *dissimilarity* threshold could be set high without compromising the result. Specifically, the total variation can be set to unto 15 standard deviations from the mean and about half the length of the major shape axis without affecting the result too much. This is because most conflicting interpretations were eliminated using the shape's largest context as depicted in Fig. 4.8. In cases where no relations exist to correct false interpretations, the algorithm will have no solution but to accept the false candidates. This also applies to cases when two fitting interpretations have nearly equal costs and the false one is selected. Experiments on 5 sketch variations similar to fig. 4.7 with a total of 17 relations and 90 atoms showed that using context knowledge was able to eliminate 94% of false candidates.

Fig. 4.9 shows how shape regression can predict plausible candidates from their context for some of the relations shown in fig. 4.7. In all these cases, the relation is found by generating a shape candidate from its found set and then matching this candidate against the data. When a sufficient match exists the next candidate is generated and tested until the whole relation is found. Experiments showed that when up to 3 strokes are deleted from the original sketch of a shock absorber, the ASSM was able to find this relation and reconstruct the missing strokes. This was tested for 35 sketches of shock absorbers and only in two cases the ASSM was unable to recognize the relation. For smaller relations like the arm, the ASSM was able to find and reconstruct the relation when only one stroke is missing. This was done for a sample space of 45 arms and 4 were not recognized. Similar results hold for the other relations. This demonstrates that ASSM can generate candidate relations when sufficient support exists from the data. The decision of how many strokes must be in the sketch before a reconstruction is carried out rests with the user. If he sets too few support strokes , the algorithm

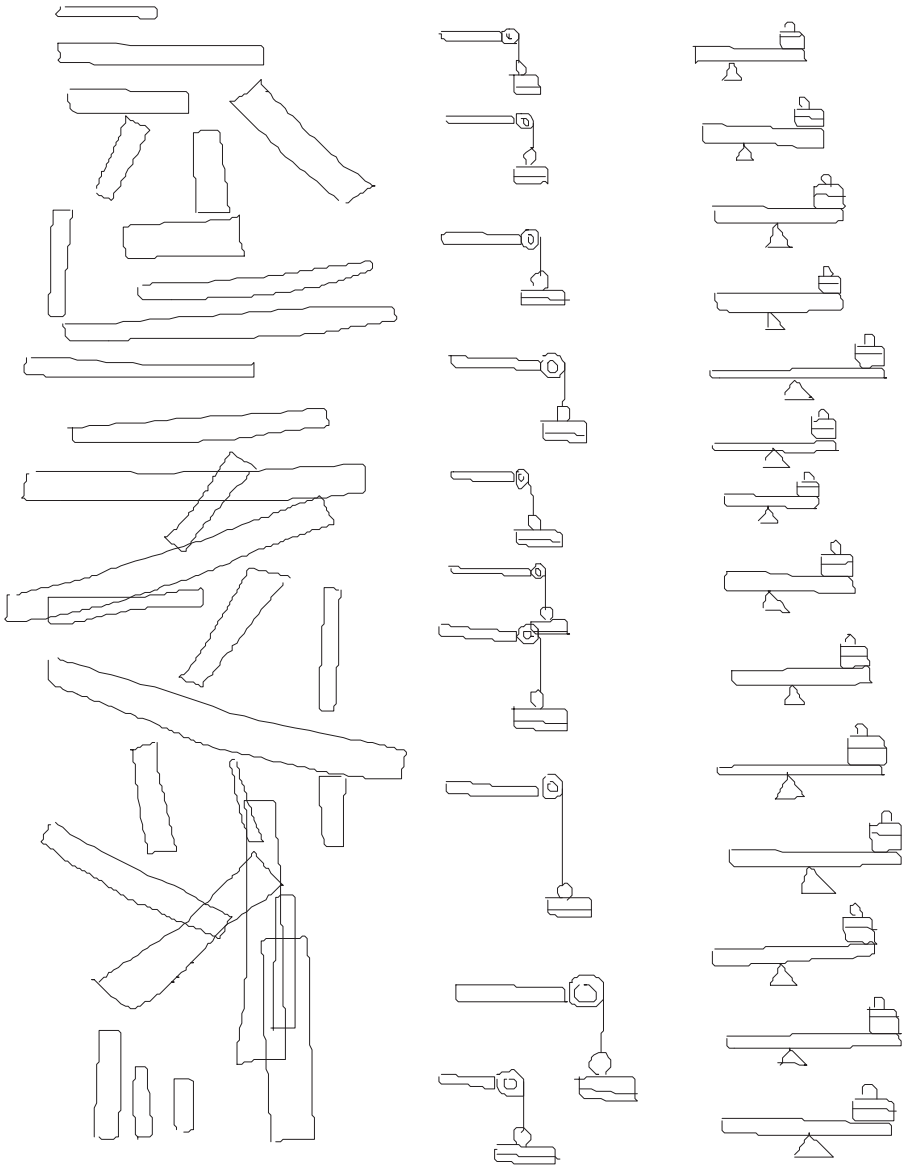


Figure 4.6: Training sample for bar, crane and pulley

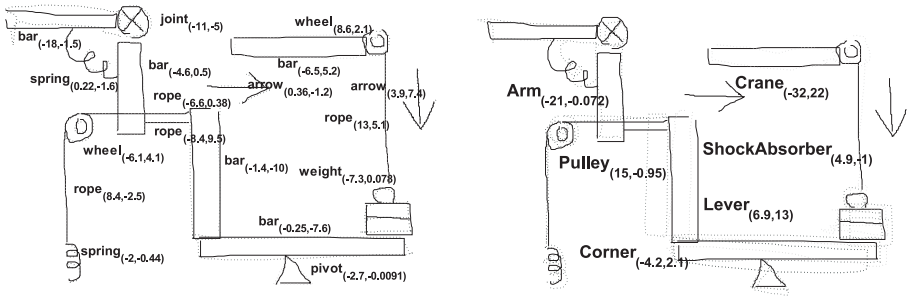


Figure 4.7: Example sketch with the overlaid fitted model (dotted lines). Left: Objects, Right: Relations. Each object or relation is characterized by its shape parameters where the first two are shown.

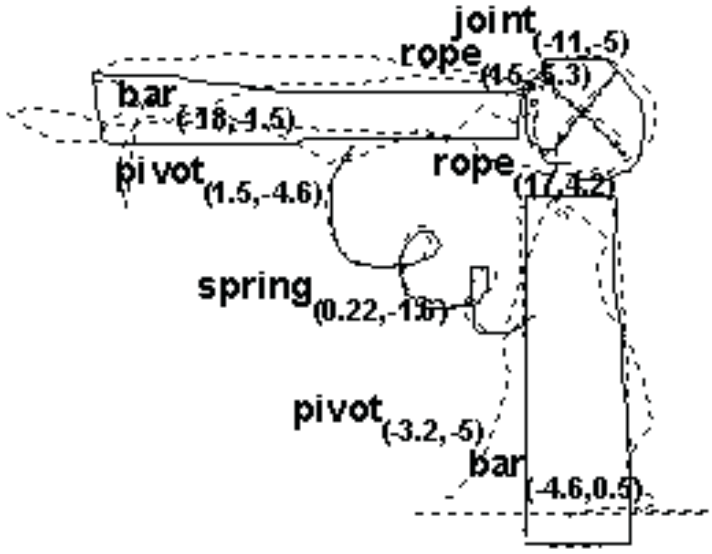


Figure 4.8: Conflicting interpretations (dotted lines) like the pivot and rope objects above are resolved using the fact that the bars and the joint are part of an arm relation which represents a larger shape context with higher confidence.

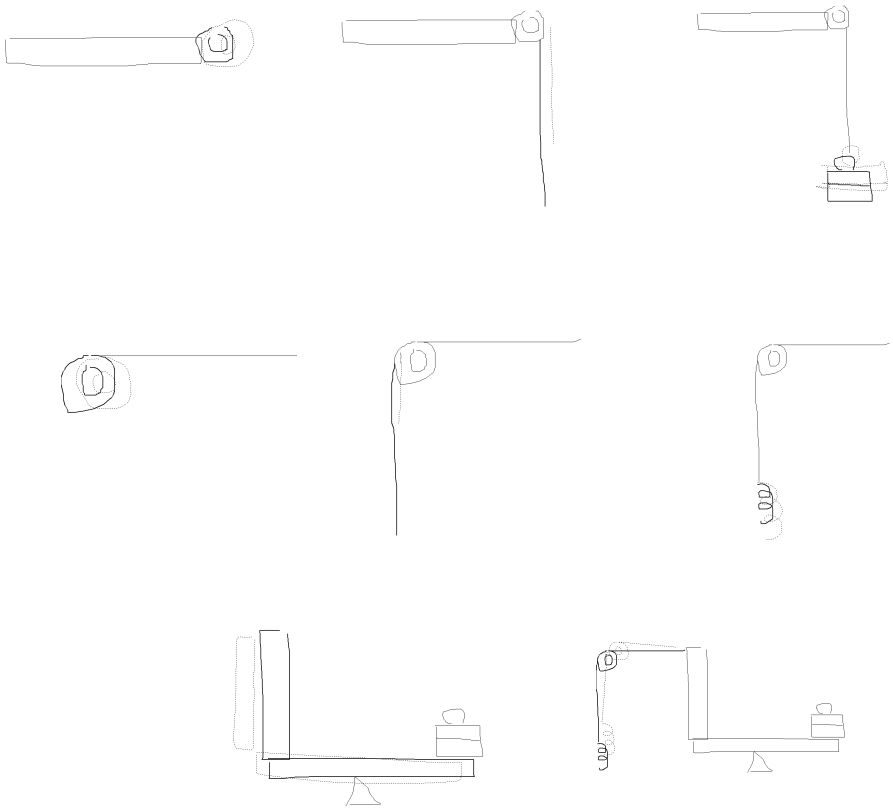


Figure 4.9: Reconstructing shapes from their context by regression. Each row shows the step by step generation and matching of relations. Each step shows the generated shape candidates (dotted curves) which is matched with the best stroke (thick curve). The remainder are the regression strokes. The last row shows how the shock absorber is generated and matches with its three subrelations

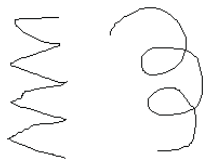


Figure 4.10: Two representations for a spring object drawn by two different users.

will generate many false candidates that fit badly to the data. For these experiments 3 missing strokes for the bigger relations of more than 10 strokes and 1 or 2 for smaller relations was adequate.

In this implementation, training took most of the processing time ,e.g., about 10 minuets on a pentium III processor with 128MB memory. Recognizing new sketches takes only a few seconds.

These experiments demonstrate the use of ASSM to characterize shape variability due to both statistical variations made by a single user and more importantly structural variations due to functionalities between machine parts drawn in no particular order.

In the next section, the extension to variations between multiple users will be studied.

4.2 Biometric Sketch Recognition

Statistical methods are becoming more important in all biological fields of study. Biometry deals with the application of mathematical techniques to the quantitative study of varying characteristics of organisms, populations and species [39, 69]. Examples of using biometry are finger print, retinal and face recognition. Biometry can be used in hand drawn data because they are rich in quantitative features. This raises the potential to extend the use of sketches in biometry and use ASSM to find the relevant features. In the previous section the adaptation of ASSM on sketches has been demonstrated on drawings of mechanical systems. All the training samples were drawn from a single user. In the case of multiple users, there are many aspects to consider when examining inter-user variability. One aspect is the characteristic ways users draw the same object as can be seen in fig. 4.8. This is the dynamic aspect of user variability. A more important difference pertaining to ASSM is about structural inter-user variation. This variation has two forms:

1. Users tend to use different drawing primitives to represent the same object as shown in fig. 4.10. In this case different shape templates are used to represent the same semantic object.

2. If the drawing primitives are the same between two users, they must differ in the relationship between these structures (inter-class variation).

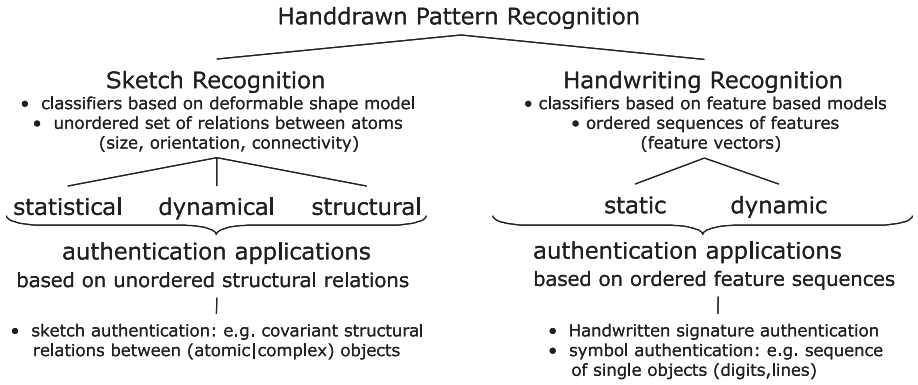


Figure 4.11: Classification of biometric sketch authentication applications

These types of variations between users enable the use of this information for user identification. This is the idea behind the application being proposed here [39, 1, 16]. The structural variation enables to apply a biometric algorithm on sketches to authenticate users. As depicted in fig. 4.11, the structural component of a sketch (containing rich information in how the shapes relate to each other) is what differentiates sketches from handwritten signatures and symbols (simple fixed drawing) [45]. To understand how this application relates to biometry, we must define what a biometric authentication system is.

The following text is quoted from [15]: "A biometric authentication system can be considered as a part of an IT infrastructure where a person is subjected to a general authentication process for receiving e.g. access rights to IT system resources, activity regulations and information non-repudiation within electronic business processes, or the permission to pass a gate or to enter a place or room. The *general authentication process* can be divided into the five subsequent phases: *enrollment*, *(biometric) authentication*, *authorization*, *access control*, and *derollment and authorization withdrawal*. During the phase of *enrollment* appropriate biometric raw data of a person is captured, the biometric signature (template) for the biometric authentication is computed, and the relevant biometric and personal data is stored in a biometric database [14]. A person's authenticity is checked by an identification (1:c) or verification (1:1) comparison of the actually computed biometric signature with the biometric signature class in the phase of *biometric authentication* potentially being combined with authentication methods based on a person's knowledge, possessions, location, and time. Implicit and explicit authorizations are given to the person in the *authorization* phase with respect to strong and weak authorizations. In the *access control* phase the access to e.g. IT system resources or activity control within electronic business processes is granted by an *access management system*. In the phase of *derollment and authorization withdrawal* a person is derolled and the person's access rights are removed".

Sketches were chosen for the biometric authentication system because they are a very simple and intuitive way to represent secret information. They are easy to remember

and draw. The structural information of sketches will be used for authentication the following way:

1. All users will be trained to draw a predefined set of shape primitives as shown in fig. 4.14.

2. They will be asked to construct a simple sketch consisting of a given number of these primitives. The way the user decides to connect, scale and orient these primitives will be the secret structural information he conveys to the authentication system.

3. The user is asked to enroll the sketch he drew several times. When he is authenticated, he has to draw the same sketch.

The enrollment process collects sketch samples from the user. These samples are aligned using alg. 1 and then PCA is applied. For each user $i = 1 \dots p$, we construct a biometric signature. This signature consists of all the parameters resulting from principal component analysis $(\overline{\mathbf{x}}_i, \mathbf{\Phi}_i, \lambda_i)$ and a threshold value τ_i for the dissimilarity measure in eq. 4.1 which minimizes the overlap between users. The output of the enrollment process is a biometric signature table $T = \{(\overline{\mathbf{x}}_i, \mathbf{\Phi}_i, \lambda_i, \tau_i) : i = 1 \dots p\}$. A problem that can occur when enrolling a new user is that his signature might be closer to one or more existing user signatures. In such a case one of two solutions can be applied:

1. Clustering/classifying without accepting a decrease of the authentication system recognition performance. Once the user i is enrolled to the already $(i-1)$ enrolled users, his biometric signature $(\overline{\mathbf{x}}_i, \mathbf{\Phi}_i, \lambda_i)$ is compared with all enrollment samples of the previous $(i-1)$ users. If the mean dissimilarity is less than three standard deviations from another users samples, user n has to re-enroll with a new sketch (pattern).

2. Clustering/classifying with accepting a decrease of the authentication system recognition performance. If the user needs to be enrolled with a fixed set of samples and the dissimilarity is less then three standard deviations, a higher false match rate can be used to enroll the new user by adjusting τ_i. To maintain the algorithms performance an additional sketch can be enrolled for user i to increase his discrimination distance from other users.

The authentication process consists of validating a user i's drawing $\mathbf{x}$ with his signature $(\overline{\mathbf{x}}_i, \mathbf{\Phi}_i, \lambda_i, \tau_i)$. This is done by fitting $\mathbf{x}$ to the shape template using alg. 3 and then calculating the dissimilarity measure d using eq. 4.1. If $d < \tau_i$ then the user is authenticated otherwise he is rejected.

In the following section both dynamic and structural aspects will be evaluated with a number of experiments.

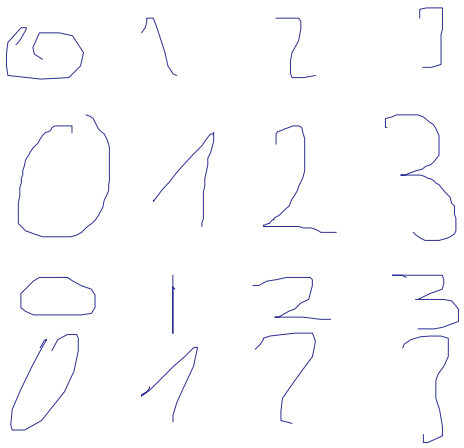


Figure 4.12: Pin samples taken from four different users.

4.2.1 Evaluation and Tests of the Biometric Sketch Recognition Algorithm

The biometric signatures are used to characterize the input of users in two ways:

1. Statistically (quantitative features): If a population of users is asked to draw exactly the same shape, the set of biometric signatures can be used to some extent for identification of users based on the characteristic way they draw these shapes. By increasing the complexity of the shape, the identification performance increases.
2. Structurally (qualitative features): A sketch additionally contains connectivity, scale and orientation relations between shapes. These relationships are represented in the biometric templates of single users and substantially improve discrimination performance in comparison to statistical features only.

Three types of experiments were done to examine these two claims:

1. Handwritten PIN number tests: For testing the statistical claim.
2. Sketch tests: For testing the structural claim.
3. Imposter tests: Test to what extent an intruder with no, partial or full knowledge about user sketches can be falsely authenticated.

Handwritten 4 digit PIN numbers tests: A population of 10 users was asked to draw 30 times the PIN number (0123). Some samples are shown in fig. 4.12. Each test used 20 randomly selected samples for training and the remaining 10 for testing. Each test was cross validated 10 times and the average error rate was computed. Each stroke was sampled by 32 points. The number of principle components was set to represent (explain) 98% of the samples and ranged between 11 to 15 principal components.

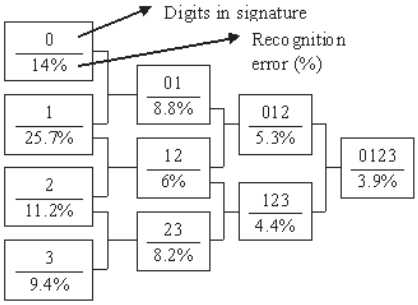


Figure 4.13: Recognition error rates decrease as more digits are combined

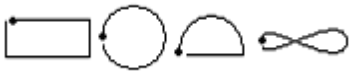


Figure 4.14: Shape types used to construct sketches: bar, wheel, base, and knot

Figure 4.13 depicts how the recognition error rate drops from worst case 25.7% for digit 1 to 3.9% for the complete PIN. The conclusion is that the error rate of a combined structure is less than the error rates of its substructures.

Sketch tests: Four basic shape types were given to 10 users as shown in fig. 4.14. Each user was given four tasks of increasing complexity to complete in his way as shown in table 4.1. Figure 4.15 shows some mean sketches drawn.

Each stroke was sampled by 16 points. For every sketch, the number of principal components was set to explain 95% of the samples. The number of principal components ranges between 10 for task 1 and 15 for task 4. The experiments were conducted on 10 users. Each user sketched each task 30 times. For every user task, 20 randomly selected samples were used for training and the remaining 10 were used for testing. The tests were cross validated 10 times and averaged.

As depicted in table 4.1, the average recognition error decreases as the complexity of the structures increases. Task 4 consisting of 11 objects had 0% error.

Imposter tests: These tests verify how often a correct user is falsely rejected for authentication and an imposter is falsely accepted. Three kinds of tests were considered:

1. The imposters have full knowledge of the sketch and trying to copy it.
2. They have partial knowledge of the sketch structure.
3. They have no knowledge of the sketch structure at all.

The full knowledge test was conducted with two imposters who tried to copy 20 times task 4 of user 8 (see fig. 4.15). The results were compared with 10 user samples and

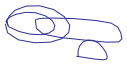
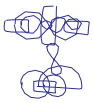
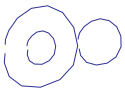
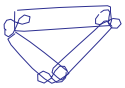
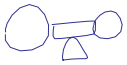
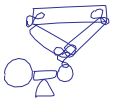
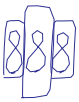
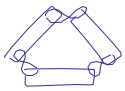
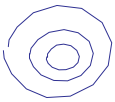
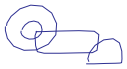
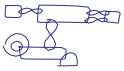
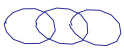
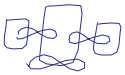
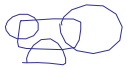
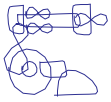
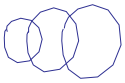
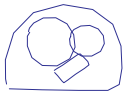
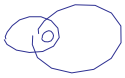
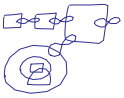
	task 1	task 2	task 3	task 4
user 1				
user 2				
user 3				
user 4				
user 5				
user 6				
user 7				
user 8				
user 9				
user 10				

Figure 4.15: Mean sketches drawn by some users

Table 4.1: Sketching tasks given to users and their recognition errors

task	description	objects	error %
1	Draw three connected wheels of different sizes	3	1.3%
2	Draw 3 connected bars one bar is bigger than the others Connect the bars to 3 knots	6	0.9%
3	Draw 2 connected wheels one wheel is bigger than the other Connect the wheels to a small bar Connect bar to a big base	4	0.7%
4	Draw Task 2 and task 3 connect them with a knot	11	0.0%

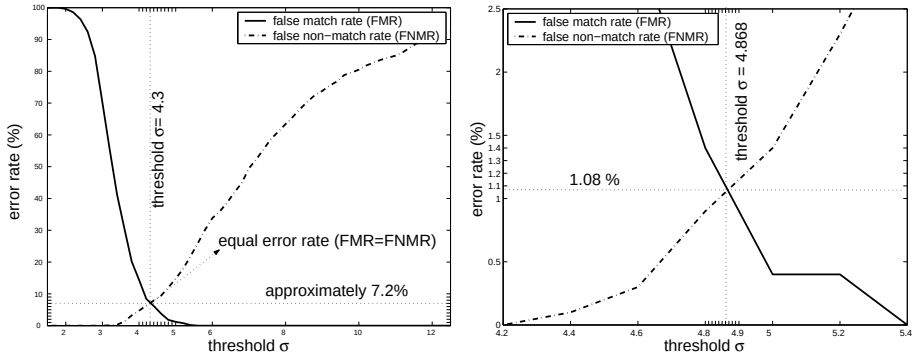


Figure 4.16: Imposter tests **left:** direct copying (task 4) **right:** last knot unknown (task 4)

cross validated 50 times. Figure 4.16 left shows the false acceptance and rejection rate graph that resulted by adjusting the threshold on the dissimilarity measure in eq. 1. As we see the point of equal error rate is about 7.2% which is due to the statistical properties which differentiate the user from imposters. For the partial knowledge test two imposters where given all the knowledge about task 4 of user 8 except the position of the last knot which has to be guessed. 20 samples were drawn and the results are depicted in figure 4.16 right. The point of equal error decreases to about 1%. Further tests with even less knowledge showed no error for this small sample set which to some extent validates the assumption that structural knowledge is unlikely to be duplicated by an imposter when he has no knowledge about it.

The previous results show that structural semantics can be used with some accuracy within an authentication system. In cases where the users draw similar structures, dynamic features can be used for discrimination. The intra-user variation of structure over time will be tested in future work. In this case the question posed will be how well do users remember the pattern they decided to enroll with after a week, month or a year. And more importantly if they do remember, how much will the patterns they

draw change over time. Another factor that has to be considered is testing on a larger set of users say 100, 1000 and more. The limited test set of 10 users presented here only shows the potential of such a system but not a full evaluation of its performance.

In the following section, ASSM and a pure statistical model will be compared in terms of shape representation.

4.3 ASM versus ASSM

This section demonstrates what happens if we do not use structural knowledge as part of ASSM and instead use only an active shape model as the shape representation. In this case all probability distributions of each structural variant in the shape have to be merged into a single probability distribution. This is because all types of shape variation have to be modeled statistically. There are three types of variation captured by ASSM: Variation of shape class, deformation modes within one shape class and covariations between shape classes. The active shape model can only model deformations of a single shape class. When including the other two variation types in ASM, a complex non-normal probability distribution is formed. If this distribution is approximated using a single principal component analysis, we end up with a model that generates and fits invalid intermediate states. Otherwise, a complex nonlinear distribution is needed to specifically capture valid states such as using a Gaussian mixture or hierarchical point distribution models [13]. The non-linear distribution however requires a large number of samples to separate valid from invalid states. The alternative is to allow too many invalid or to exclude many valid states. This problem will be illustrated on sketches similar to the ones presented in the last two sections.

Fig. 4.17 shows an example sketch consisting of 5 structural components. Sketch A is the basis and sketches B and C are structural variants of A different by two structural classes. Sketches D and E have the same structural classes as A but different covariation of parts. In this case it is simply the positions of these parts relative to each other.

Fig. 4.18 shows a scatter plot of 30 samples on the first two eigen coordinates normalized by their standard deviations. The figure shows an even distribution within two standard deviations from the mean. Subsequently we can use this distribution to generate random instances within 3 standard deviations from the mean, we get the result shown in fig. 4.19. All instances are valid variations of the same shape class.

To demonstrate what happens when varying the class type, we use instances of sketch A,B and C as training samples to principal component analysis. Fig. 4.20 shows the scatter plot of the first two normalized eigen coordinates of these samples. In this case, the distribution is clearly non-linear with three distinct clusters. Any state generated between those clusters is an improbable state as seen in fig. 4.21. Similar results are shown for positional variation by taking random instances of sketch A,D and E as shown in fig. 4.22 and fig. 4.23. Finally, fig. 4.24 and fig. 4.25 show the cases when both types

of variations are combined. The scatter plots reveal several potential problems to learn the nonlinear distribution:

1. If a cluster contains too few points, it is difficult to determine if they represent a cluster or outliers.
2. The complexity of the distribution increases with more added cases. This makes learning the nonlinear distribution increasingly difficult because it would require more training samples to crisply separate different cases.

In contrast to the non-linear distribution approach, the structural model of ASSM provides the prior knowledge that enables us to separate the non-linear distribution into smaller linear distributions. Each linear distribution requires a small set of representative training samples. The structural model specifies linearly correlated shape groups eliminating the need to learn the non-linear model.

4.4 Recognition of Ant Images

In the previous sections ASSM was described and applied to sketches. In this section, the ASSM framework will be demonstrated on other application domains. The application presented here is the recognition and classification of ants in ant image databases [8, 9]. The ant body can be subdivided naturally into parts and there is no need to create artificial structures. Another interesting thing about ants is that they can be classified by morphology into family trees. This means that some ant types are more similar to other ants the closer they are in the family tree. This feature corresponds to the multi-resolution representation of ASSM. The ant body is an articulated structure which consists of several clear segments. The shapes and number of these segments changes between species. This implies that the ant variations cannot be described by a single structural prototype but with several variants where some components are reused between these prototypes. An example is depicted fig. 4.27 where the first two ant classes share the same head prototype. Because of these two properties, we can apply an ASSM frame work that starts with a multi-shape generic ant. As the recognition process proceeds these structures begin to specialize to more specific shape types as determined by the ant type. Using the structural and statistical constraints between shapes, the best fitting shape is found and used to segment and classify the ant.

In the implementation shown here, ant images are used which are taken form a standardized side view position. This view captures best the ant shape because no body part occludes another. The ants are separated from the background image using a color feature classifier. The silhouette of the ant body is then used in subsequent analysis.

The shape representation used here is similar to deformable organisms [38] as depicted in fig. 4.26. At the base level there is the actual image data. Various filters are then applied at the next level to extract the ant silhouette from it's background and then smooth it with a Gaussian filter. The shape templates in the next level are modeled as

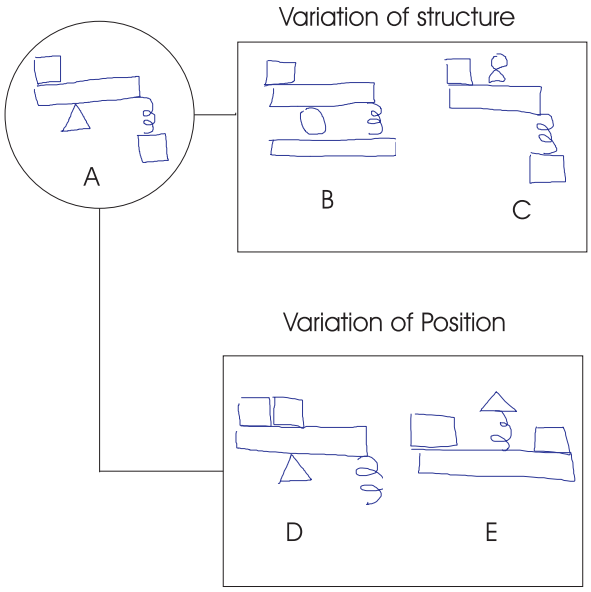


Figure 4.17: A group of sketches depicting both structural and positional variability. (A) is the Basis sketch and (B,C) are structural variants with 3 common parts. (D,E) are variants of (A) which have the same structural parts but at different positions.

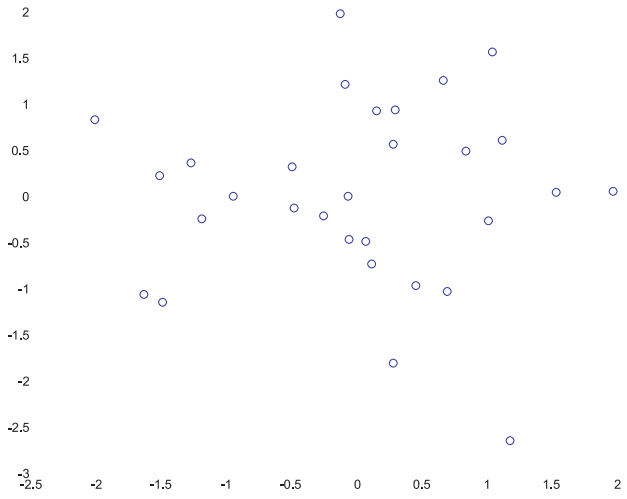


Figure 4.18: Scatter plot of the first two normalized eigen coordinates of Case A in fig 4.17 for 30 samples.

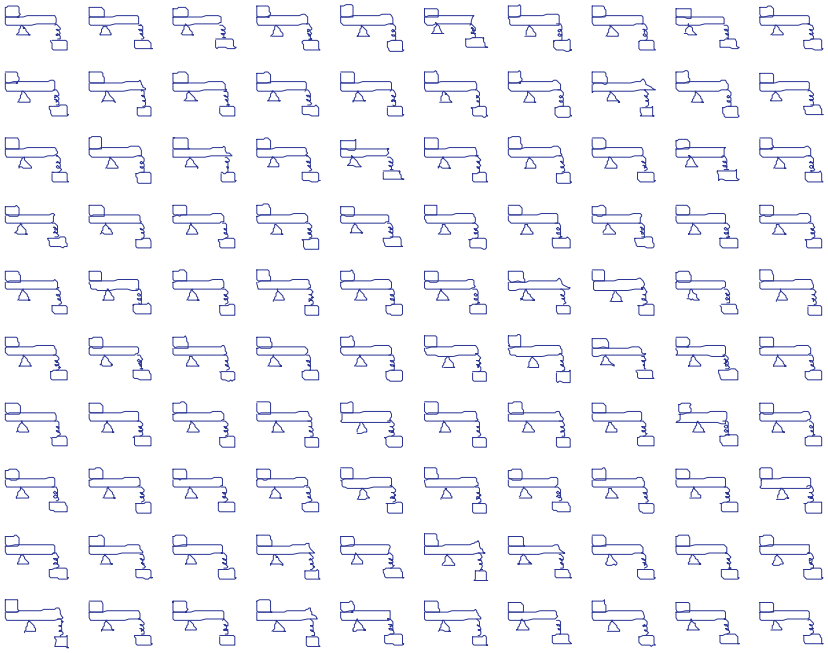


Figure 4.19: Some generated samples of Case A in fig 4.17. All samples show valid states within three standard deviation for each eigen coordinate.

Figure 4.20: Scatter plot of the first two normalized eigen coordinates learned from structural combinations (A,B,C) in fig 4.17. The plot clearly shows a non-normal distribution with three distinct clusters. The empty space between the blobs are improbable states.

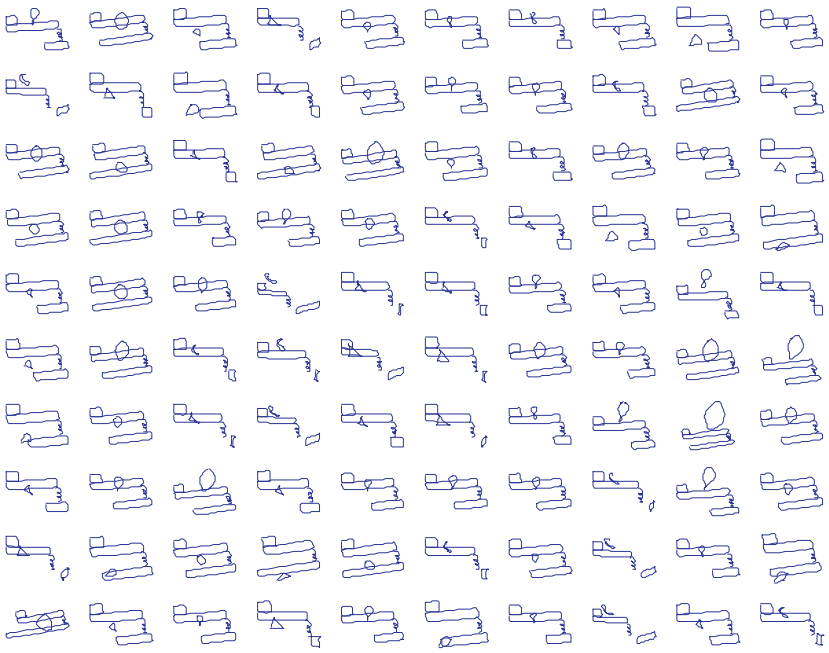


Figure 4.21: Generated samples from the distribution which is learned from structural combinations (A,B,C) shown in fig 4.17. Many intermediate invalid states appear.

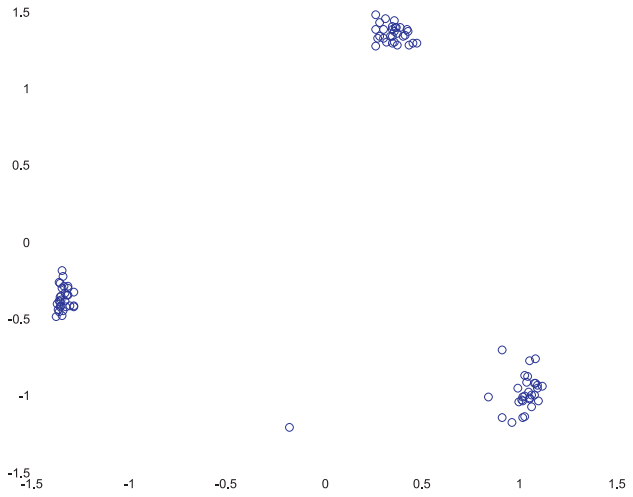


Figure 4.22: Scatter plot of the first two normalized eigen coordinates learned from positional combinations (A,D,E) in fig 4.17. The plot clearly shows a non-normal distribution with three distinct blobs. The empty space between the blobs are improbable states.

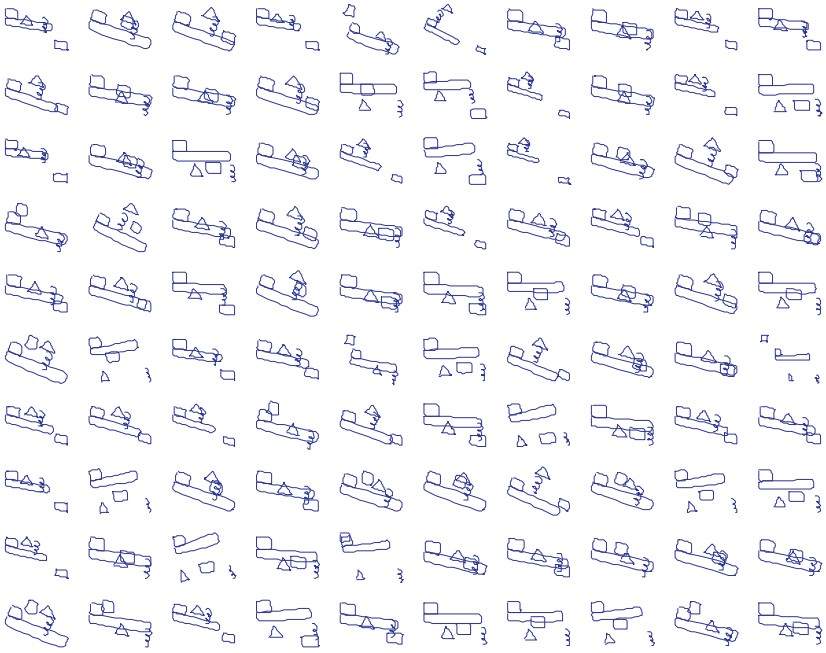


Figure 4.23: Generated samples from the distribution which is learned from positional combinations (A,D,E) shown in fig 4.17. Many intermediate invalid states appear.

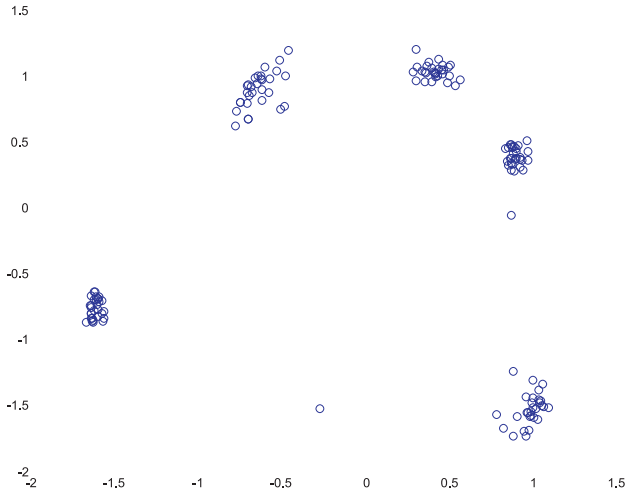


Figure 4.24: Scatter plot of the first two normalized eigen coordinates learned from combinations (A,B,C,D,E) in fig 4.17. The plot clearly shows a non-normal distribution with 5 distinct blobs. The empty space between the blobs are improbable states.

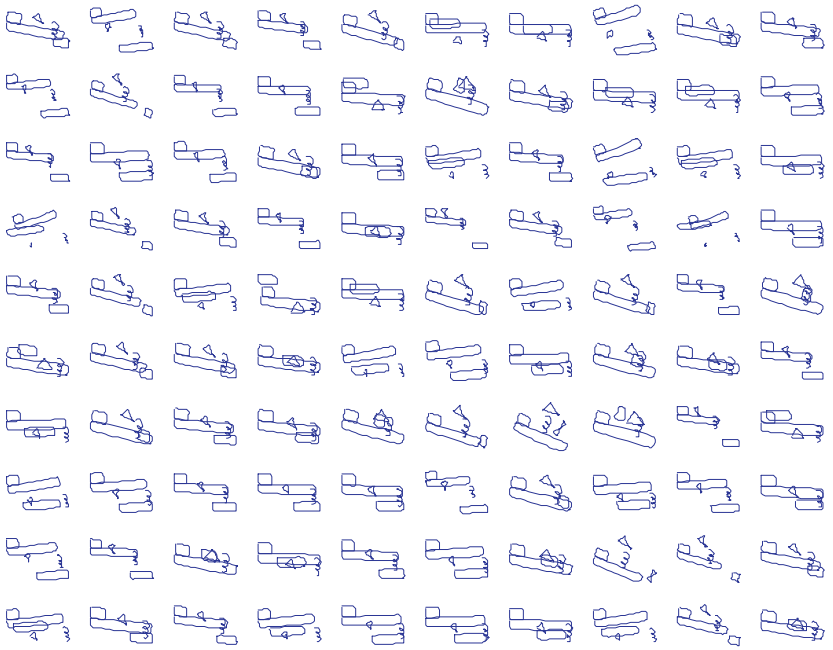


Figure 4.25: Generated samples from the distribution which is learned from combinations (A,B,C,D,E) shown in fig 4.17. Many intermediate invalid states appear.

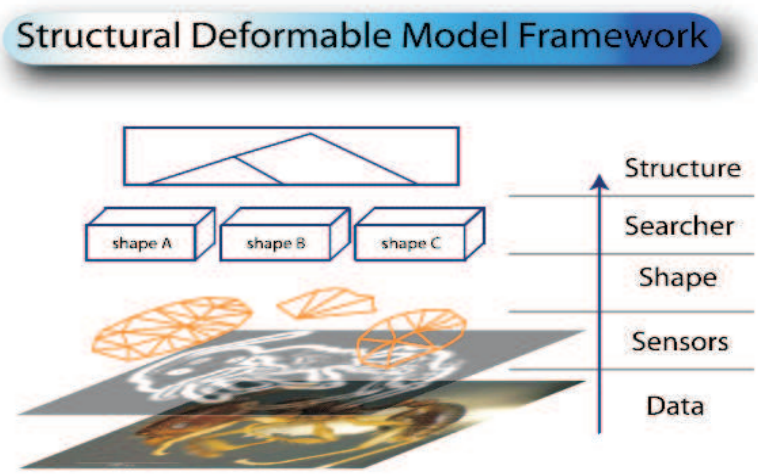


Figure 4.26: Model for recognizing images of ants (from [8])

spring mass systems moving using Euler equations in discrete time steps. Each node is assigned a mass which is affected by two kinds of forces: Sensory image forces such as edges and color information and internal spring forces. Each spring is given a rest length l_0 and a stiffness constant k. When the spring is displaced from the rest length by u, that spring will exert an opposing force $-ku$. The stiffness constant k is defined as the inverse of the variance $\frac{1}{\sigma^2}$ of previously fitted training samples. The template is designed with many internal nodes and springs that maintain the stability of its form. The total force affecting any given node is

$$\mathbf{f} = \sum_{i=1}^{n} -k_i \mathbf{u}_i + \alpha \sum_{j} \mathbf{f}_{image_j} \tag{4.2}$$

where α is a weighting constant.

The next level in fig. 4.26 is the search algorithm. The search algorithm is done both globally and locally. Local search is used when each template is allowed to evolve in time to seek a steady state solution guided by local deformation and image forces. To overcome local minima solutions, a global search algorithm throws hundreds of such templates onto the image with different scales, orientations and positions as depicted in fig. 4.29. Each template is allowed to deform to reach a local solution. The template selected as the optimal solution is the one which fits best the image. The fitting function is measured as the weighted summation of sensor and deformation energies. In this implementation, the search algorithm begins by throwing thousands of instances of the head template on the image and selecting the best fitting template. This is because the head is the biggest component which is common to all the ants.

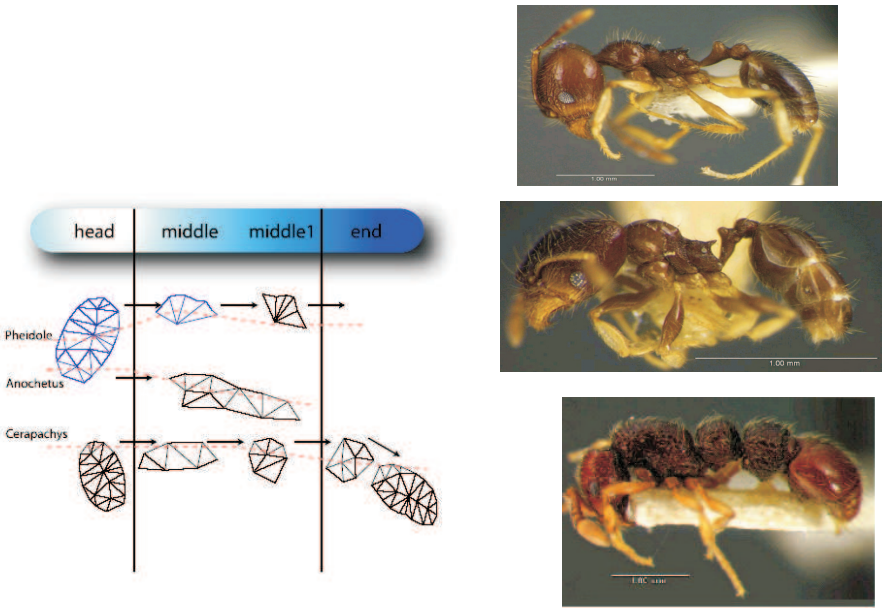


Figure 4.27: Three species of ants with different structural components. The first two types share the same head template (from [8])

The next level in fig. 4.26 is the use of structural relationships between those templates. This is achieved by a probability distribution function that defines the relative orientations, scales and positions between parts. Fig. 4.28 shows the distribution map of the thorax area relative to the head.

This means that every part is labeled by once the head atom is found, the searcher begins by throwing templates on the image according to the probability distribution specified by the head location. An example is depicted in fig. 4.28. The position distribution enables the searcher to throw random templates not evenly sampled all over the image but sampled from the probability distribution of the existing found shapes. This is depicted in fig: 4.30. In this case we see that a global uniform search in the left image found the best match at the wrong location but using the fitted head on the right the correct fit was found.

The structural variants were tested on three ant types as shown in fig. 4.27: Pheidole, anochetus and cerapachys. The first two ants share the same head and the third is the most complex consisting of five segments.

Experiments show the classification ability using these templates to recognize some ant samples. Fig. 4.31 shows generated candidates and best fitting candidate for all three template classes as applied to some samples. The probability of correct classification for the sample on the first row (Pheidole fervens) is 81% pheidole, anochetus 77%, cerapachys 55%. For the second sample pheidole subermata it is pheidole 82%, anochetus

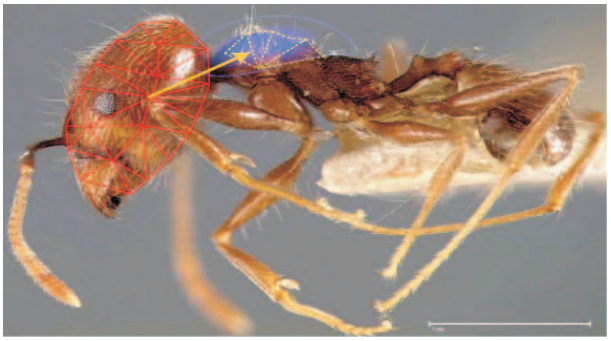


Figure 4.28: The covariance between the head and middle part enables the creation of spatial probability distribution depicted as a fuzzy cloud over the middle that enables the allocation of the chest area (from [8]).

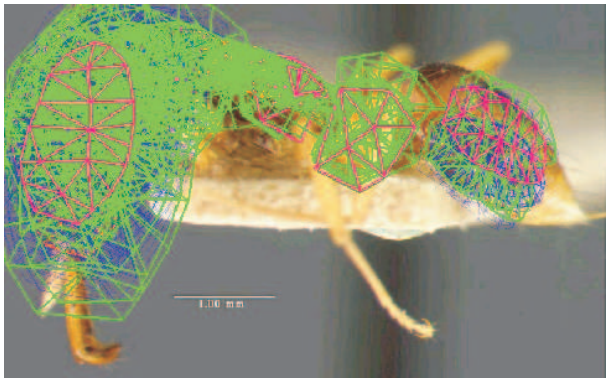


Figure 4.29: Thousands of templates are thrown on the image by applying the structural constraints between parts until the best candidate wins (from [8]).

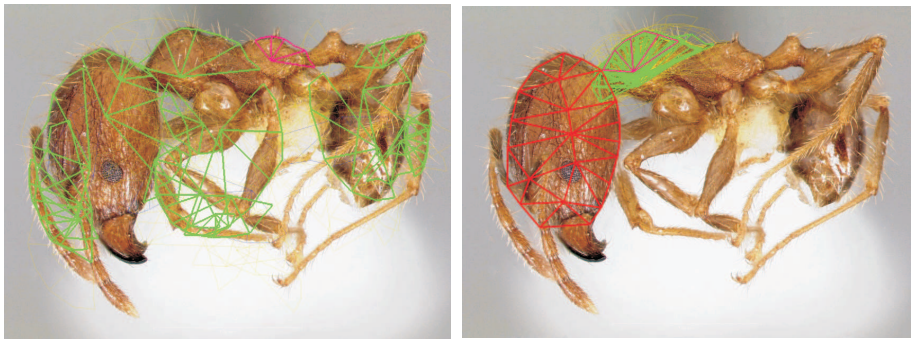


Figure 4.30: The global search for the thorax as guided without structural knowledge (left) and using the head matched (right) (from [8]).

76%, cerapachys 62%. The third sample anochetus cato is classified with probabilities pheidole 74%, anochetus 81%, cerapachys 65%.

Tests were also conducted on 75 samples of class pheidole to see if they are classified correctly. 12 samples were misclassified giving a rate of 84%.

These preliminary results show how the structural-statistical framework of ASSM can be used with some success for image classification based on the shape model coupled with sensory information and reusable structural components.

In the next chapter, the ASSM model will be compared to all the other shape models surveyed in chapter 2.

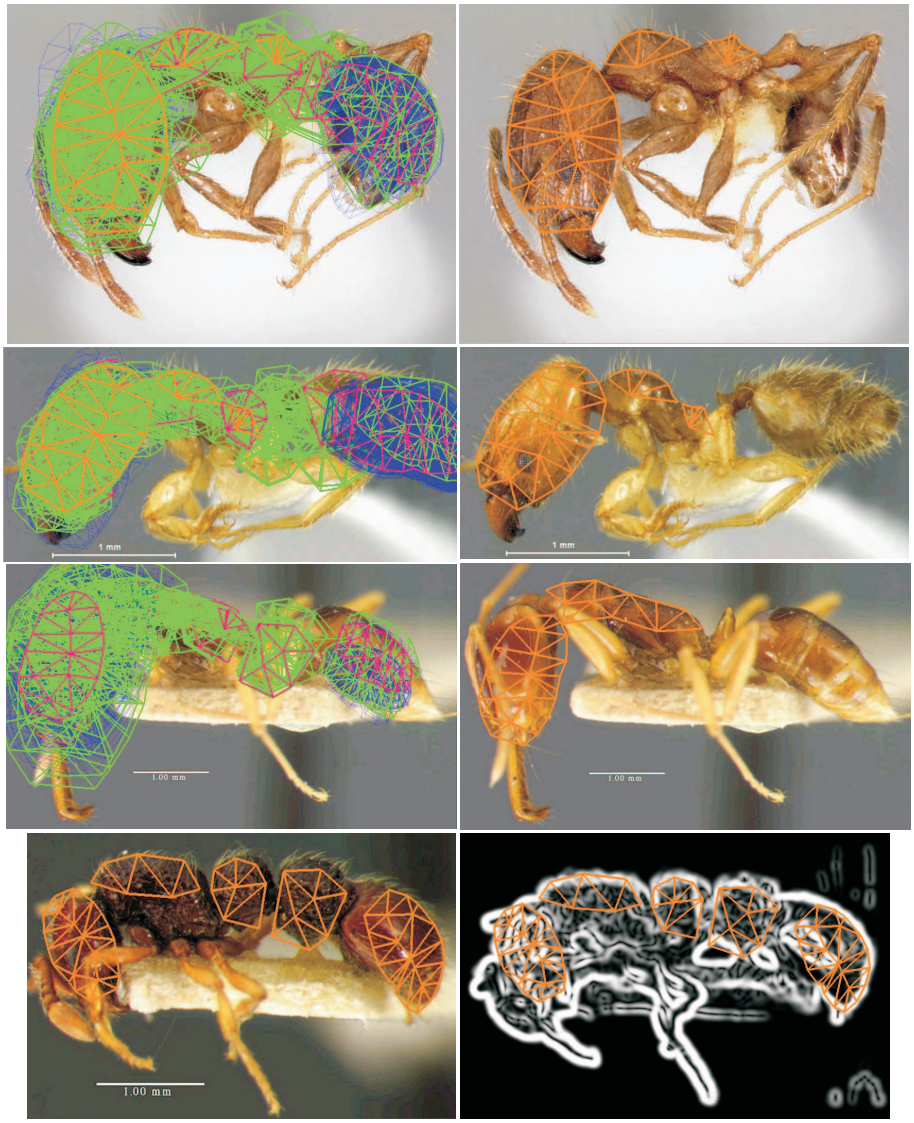


Figure 4.31: Recognition and classification results for some ant samples showing generated candidates and best fitting result. From top to bottom: Pheidole fervens, pheidole subermata, anochetus cato, Cerapachys vitiensis (from [8]).

5 Discussion

After having explained the framework for ASSM and showing some of its applications in the previous chapters, it is time now to evaluate the ASSM model with respect to the other shape models presented in Chapter 2.

Table 2.1 shows a comparison between various shape models. The following is a discussion of ASSM features with respect to some properties listed in this table.

Statistical analysis of shape: Statistical models are representation models for a single shape class. If a complex multi-part shape is modeled using a single statistical distribution, then either one obtains a simple flat distribution which allows many invalid variations of objects or one has to find a distribution which is very complicated to reflect all the valid states. The complexity of this distribution to be learned requires more training samples to reflect the true distribution; otherwise, the variations learned would not cover all the valid states. The ASSM solves this problem by dividing complex shape to a group of smaller linear distributions. This enables both complexity reduction of the distribution and accurate modeling of valid states.

Structural description of shape: Structural deformable models allowing variation modes such as geons and generalized cylinders define generic shape templates that fit a large range of shapes. These templates allow a lot of variation and can fit any configuration of shapes. This may lead to ambiguity because there can be several solutions to the fitting process. In addition to that, these templates cannot specially classify the actual structure type they segment. ASSM defines a set of more problem-specific atomic shapes. The variations of these atoms are more crisp and can therefore fit better the image. The stored shape graphs enable ASSM to find missing and extraneous structures.

Representation by a small feature space: ASSM is able to abstract any complex scene into a graph of few relations and attach a small feature vector at each relation. This is sufficient to compress the shapes scene and later reconstruct it with this representation. It also enables efficient indexing and comparison between scenes making it suitable for image databases.

Multi-resolution shape representation: Multi-resolution is a very important feature for some shape models. It represents shapes from coarse to fine levels of detail. This

representation enables the separation of macro from micro features of shape. Macro features abstract shapes to a simple, noise robust form. This facilitates easy comparison between shapes tolerant to various noise artifacts. Generally, shape models that apply multi-resolution can be divided into two categories: Those that apply a smoothing operation and those that use a hierarchical description of shapes. Examples of the first category can be seen using a curvature scale space for example [55] for matching silhouettes of fish databases. Gaussian smoothing is not the only operation that can be used. For example, Latecki [44, 43] uses discrete curve evolution which is basically removing line segments one at a time based on a cost function.

The other type of multi-resolution uses hierarchical description. This means that we divide the shape into two or more different models which when used together can reconstruct the original shape. An example of that is superquadrics by Terzopoulos [73] which uses a global superelastic to describe the global boundary of the shape then a local displacement field adjusts the boundary to represent micro features. Another example is the object shape by Pizer [58]. In this case the shape is divided into a network of medial primitives called figures. The shape is described in four levels: the object as a whole, each figure, inter-figural relations and boundary displacement for each figure.

Both multi-resolution representations of shape use no problem specific prior knowledge. This means that different levels of detail may ignore important information about the shape (e.g. smoothing may delete an important feature of a boundary as noise such as figures in a human body silhouette). Also, certain irrelevant features may be retained at low levels of detail (e.g. a shape occluded by another).

In both these cases ASSM can solve this problem by coupling predefined structure with morphology. In this case, the level of detail is represented by the degree of variation which is large for a structural element and then becomes more constrained as more structures come into play with it. This means that fine details will not be ignored as there is sufficient evidence that supports it and also irrelevant features are ignored because there is not enough context information to support it. In short multi-scale fitting in ASSM is semantically dependent on the specific shape and can be refined with more fitted shapes.

Initialization sensitivity : If a good fit is found to some initial shape, the relations can generate and correct the remaining parts. The statistical knowledge can validate the initial fitting. Finding these initial structural elements is crucial for matching.

Robustness to noise: ASSM has the ability to infer missing structures and correct existing ones based on shape context. In addition to that, ASSM has the usual robustness to noise derived from statistical models. In the case of shock grammars, the correction rules to shock graphs are restricted to the connectivity of few shape classes. This trims errors in shock graphs. Similarly ASSM can trim a wider range of erroneous shape types and find a suitable interpretation based on contextual relations.

Applications: ASSM can be used for segmentation, shape comparison, shape generation, content based image retrieval and reconstruction and compression of shape scenes.

After a general comparison of ASSM with respect to the other model features, it is necessary to compare it to the other more similar hybrid models. Pictorial structures [31] statistically model the relations between shape parts. The shape structure however remains invariable. The relations in between are constrained to be binary between parts and the shape graph is constrained to be a tree. This means there is no provision for shape multi-resolution. ASSM can both vary shapes structurally and allow a multi-resolution hierarchy of relations between them. The pictorial structures model only rigid body relations between structures like translation rotation and scale. It did not address how deformation is handled between those objects. This is because it used simple rigid shape templates like the rectangle.

FORMS [80] is the opposite to pictorial structures. It does allow structural variation of parts using skeletal graphs. The statistical model is used only to represent the variation of single parts and not the relation between them. This means FORMS matches only based on connectivity graphs and degree of fit of each part individually. In addition to that FORMS can describe only two basic deformable shape templates. This makes the shape distribution more flat and allows for ambiguity in matching. Another inherent problem with FORMS is that it relies on the medial axis transform to device the object to parts. The medial axis is inherently unstable and may easily generate extra structures because of noise. The division to parts by medial axis also may not be semantically meaningful to some applications. For example, the tail of a fish may be modeled better as a single deformable shape rather than two independent templates. The ASSM framework provides more problem-specific shape templates that rely on domain knowledge rather than the medial axis.

In addition to pictorial structures and FORMS, there are other shape models that represent soft articulated shapes such as [59] which were not listed in chapter 2. These models work by fitting deformable objects to a fixed structure. The fitting is done on each shape individually without using any explicit co-deformation information or higher order relations between more than two objects. ASSM is able to represent these kinds of relations explicitly which can guarantee a better more robust fit.

6 Conclusion and Future work

This thesis has presented a shape model framework that does not view single shapes as deformable entities but as a collective of interdependent shapes. The ASSM framework can model a broader range of problem-specific shape templates and allows structural variation of shape. It models both deformation and co-deformation of shapes and allows multi-resolution relations.

The main drawbacks of ASSM are:

- The need for a sufficient number of representative training samples. This may not always be available.
- The structural model have to be fully specified beforehand by the user and cannot be learned from training samples.

Future work will concentrate on finding solutions for those two problems. The first problem can be addressed by extending ASSM to use a dynamic model when insufficient samples exist and then gradually as more samples are collected, the model converges to statistical aspects. This is depicted in fig. 6.1.

The second problem is can ASSM learn from training samples the structural relationships. This can be done in several steps: Learn the atomic shapes and then check if these shape have connections to each other, then learn the statistics of these relations. This can be solved by finding a uniform representation for morphology and structure and not separated as they are now in ASSM. Some preliminary tests were conducted using both clustering and inductive learning systems. Clustering was used to group atoms into classes. The main problem was that atoms with similar morphology but different semantics were merged and also the same atom but with a large change in morphology

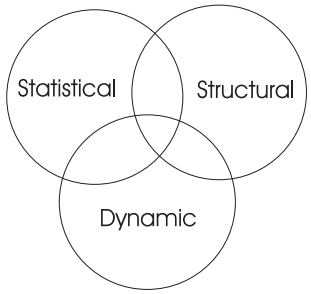


Figure 6.1: The possible overlaps of shape models

were separated into serial classes. To learn complex relations between these atoms, an inductive logic system was tried. This revealed that even in simple cases where two atoms are structurally related, the limitations were obvious. This is because in these cases predicates that describe morphology substitute exact statistical deformable models resulting in a large false acceptance rate. For more complex relations consisting of three or more atoms, the results look too bad to be useful. The conclusion is that learning a complex structure automatically is a difficult problem which is open to further research.

Another aspect of the ASSM framework is to apply it to the domain of image databases along the lines outlined in the ant database application. In the future texture aspects have to be integrated into ASSM to make it a full model that uses more image features to find a good fit. In this case ASSM will begin as a dynamic structural model and gradually shift to a statistical dynamic model as more representative samples are available. As an example [63] uses local partial AAM models linked via global constraints to avoid the under training problem. The templates considered are triplets of vertebra that overlap each other. Global constraints make local AAM templates fit with each other.

Bibliography

[1] S. Al-Zubi, A. Broemme, and K. Toennies, "Using an active shape structural model for biometric sketch recognition," in *Proceedings of DAGM*, Magdeburg, Germany, September 2003, pp. 187–195.

[2] S. Al-Zubi and K. Toennies, "Extending active shape models to incorporate a-priori knowledge about structural variability," in *Proceedings Pattern Recognition, 24th DAGM Symposium*, vol. 2449. Zurich, Switzerland: LNCS, September 2002, pp. 338–344.

[3] ——, "Generalizing the active shape model by integrating structural knowledge to recognize hand drawn sketches," in *CAIP*, ser. Lecture Notes in Computer Science, vol. 2756. Springer, 2003, pp. 320–328.

[4] S. Al-Zubi, K. Toennies, N. Bodammer, and H. Hinrichs, "Fusing markov random fields with anatomical knowledge and shape based analysis to segment multiple sclerosis white matter lesions in magnetic resonance images of the brain," in *Proceedings of SPIE (Medical Imaging)*, vol. 4684, San Diego, February 2002, pp. 206–215.

[5] S. Al-Zubi, K. D. Toennies, N. Bodammer, and H. Hinrichs, "Fusing markov random fields with anatomical knowledge and shape based analysis to segment multiple sclerosis white matter lesions in magnetic resonance images of the brain," in *Bildverarbeitung fr die Medizin*, Leipzig, March 2002, pp. 185–188.

[6] C. Alvarado and R. Davis, "Resolving ambiguities to create a natural computer–based sketching environment," *Int. Joint Conference on Artificial Intelligence*, pp. 1365–1371, 2001.

[7] C. Bauckhage, F. Kummert, and G. Sagerer, "A structural framework for assembly modeling and recognition," in *Proc. 10th International Conference on Computer Analysis of Images and Patterns (CAIP'03)*, ser. Lecture Notes in Computer Science 2756. Groningen, The Netherlands: Springer-Verlag, 2003, pp. 49–56.

[8] S. Bergner, "Structural eformable models for robust object recognition," Diploma thesis, Magdeburg University, December 2003.

[9] S. Bergner, S. Al-Zubi, and K. Toennies, "Deformable structural models," in *IEEE International Confernce on Image processing*, 2004.

[10] I. Biederman, "Human image understanding: Recent research and a theory," *Computer Vision, Graphics, and Image Processing*, vol. 32, pp. 29–73, 1985.

[11] F. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. on Pattern Anaylsis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, June 1989.

[12] H. Bosch, S. Mitchell, B. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. Reiber, "Appearance-motion models for endocardial contour detection in time sequences of echocardiograms," *SPIE proceedings in Medical Imaging*, 2001.

[13] C. Bregler and S. Omohundro, "Surface learning with applications to lipreading," *Advances in neural information processing systems*, vol. 6, 1994.

[14] A. Broemme, *A Discussion on Privacy Needs and (Mis)Use of Biometric IT-Systems*. Bratislava, Slovakia: IFIP WG 9.6/11.7 SCITS-II, 2001.

[15] ——, *A Classification of Biometric Signatures*. Baltimore, USA: IEEE Int. Conf. on Multimedia & Expo (ICME), 2003.

[16] A. Broemme and S. Al-Zubi, "Multifactor biometric sketch authentication," in *Proceedings of the BIOSIG*, A. Broemme and C. Busch, Eds., Darmstadt, Germany, 2003 2003, pp. 81–90.

[17] M. Chen, T. Kanade, D. Pomerleau, and H. Rowley, "Anomaly detection through registration," *Pattern Recognition*, vol. 32, pp. 113–128, 1999.

[18] M. Chen, T. Kanade, D. Pomerleau, and J. Schneider, "Probabilistic registration of 3-d medical images," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-99-16, July 1999.

[19] T. Cootes, "Statistical models of appearance for computer vision," Imaging Science and Biomedical Engineering, University of Manchester, Manchester M13 9PT, U.K, Tech. Rep., March 2004.

[20] T. Cootes, D. Cooper, C. Tylor, and J. Graham, "A trainable method of parametric shape description," in *2nd British Machine Vision Conference*, P. Moforth, Ed. Springer - Verlag, September 1991, pp. 54–61.

[21] ——, "A trainable method of parametric shape description," *Image and Vision Computing*, vol. 10, no. 5, pp. 289–294, June 1992.

[22] T. Cootes, G. Edwards, and C. Taylor, "A comparative evaluation of active appearance model algorithms," in *9th British Vision Conference*, M. N. P. Lewis, Ed., vol. 2. BMVA press, Sept. 1988, pp. 680–689.

[23] ——, "Active appearance models," in *5th European Conference on Computer Vision*, H.Burkhardt and B. Neumann, Eds., vol. 2. Springer, 1998, pp. 484–498.

[24] ——, "A comparative evaluation of active appearance model algorithms," *9th British Machine Vison Conference*, vol. 2, pp. 680–689, September 1998.

[25] T. Cootes and C. Taylor, "Active shape models," in *3rd British Machine Vision Conference*, D. Hogg and R. Boyle, Eds. Springer-Verlag, September 1992, pp. 266–275.

[26] ——, "Data driven refinement of active shape model search," *In British Machine Vision Conference*, 1996.

[27] D. DeCarlo and D. Metaxas, "Shape evolution with structural and topological changes using blending," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1186–1205, November 1998.

[28] M. Dickens, H. Sari-Sarraf, and S. Gleason, "A streamlined volumetric landmark placement method for building three-dimensional active shape models," in *Proceedings of SPIE on Medical Imaging*, K. M. H. Milan Sonka, Ed., vol. 4322, 2001, pp. 269–280.

[29] D.Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawks, "Nonrigid registration using free-form deformations: Application to breast mr images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 712–721, August 1999.

[30] G. Edwards, C. Taylor, and T. Cootes, "Interpreting face images using active appearance models," in *3rd International Conference on Automatic Face and Gesture Recognition*, Japan, 1998, p. 300305.

[31] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," in *IJCV*, 2003.

[32] M. Fonseca and J. Jorge, *Using Fuzzy Logic to Recognize Geometric Shapes Interactively.* IEEE International Conference Fuzzy Systems (FUZZIEEE), 2000.

[33] D. Forsyth and J. Ponce, *Computer Vision A Modern Approach*, P. Lindner, Ed. Printice Hall, Upper Saddle River, New Jersey 07458: Alan Apt, 2003.

[34] D. Fritsch, S. Pizer, B. Morse, D. Eberly, and A. Liu, "The multiscale medial axis and its applications in image registration," *Pattern Recognition Letters*, vol. 15, no. 5, pp. 445–452, May 1994.

[35] K. Fu, *Syntactic Pattern Recognition and Applications*, ser. advences in computing science and technology, R. T. Yeb, Ed. Prntice-Hall, 1982.

[36] R. Gonzalez and R. Woods, *Digital Image Processing*, 2nd ed. Addison-Wesley Publishing Company, 1992.

[37] P. Hallinan, G. Gordon, A. Yuille, P. Gibin, and D. Mumford, *Two- and Three-Dimensional Patterns of the Face.* Ak Peters, Ltd., 1999.

[38] G. Hamarneh, T. McInerney, and D. Terzopoulos, "Deformable organisms for automatic medical image analysis," in *Proc. Third International Conference on Medical Image Computing and Computer Assisted Interventions (MICCAI'01)*, Utrecht, The Netherlands, October 2001, pp. 66–75.

[39] P. Jolicoeur, *Introduction to Biometry.* Kluwer Academic/Plenum, May 1999.

[40] M. Kass, A. Witken, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Computer Vision*, pp. 321–331, 1987.

[41] R. Katz and S. Pizer, "Untangling the blum medial axis transform," *International Journal of Computer Vision - Special UNC-MIDAG issue*, vol. 55, no. 2, pp. 139–153, November–December 2003.

[42] A. Lanitis, C. Taylor, and T. Cootes, "Automatic tracking, coding and reconstruction of human faces using flexible appearance models," *IEEE Electronic Letters*, vol. 30, p. 1578 1579, 1994.

[43] L. Latecki and R. Lakaemper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 10, pp. 1185–1190, October 2000.

[44] L. Latecki and R. Lakper, "Discrete approach to curve evolution," in *Proc. of 20. DAGM-Symposium Mustererkennung (Pattern Recognition).* Stuttgart, Germany: Springer-Verlag, September 1998, pp. 85–92.

[45] F. Leclerc and R. Plamondon, *Automatic Signature Verification: The State of the Art 1989–1993.* Int. Journal of Pattern Recog. and Artificial Intelligence, 1994.

[46] T. Lee and M. Atkins, "A new approach to measure border irregularity for melanocytic lesions," in *Proceedings of SPIE in Medical Imaging*, K. M. Hanson, Ed., vol. 3979, 2000, pp. 668–675.

[47] J. Lin, M. Newman, and J. Hong, *DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design.* CHI: Human Factors in Comp. Systems, 2000.

[48] R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, 1995.

[49] D. Marr, *Vision: A Computational Investigation into the Humn Representation and Processing of Visual Information*, J. Wilson and P. Monsour, Eds. New York: W. H. Freeman and Company, 1982.

[50] T. McInerney and D. Terzopoulos, "Topology adaptive deformable surfaces for medical image volume segmentation," *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 840–850, 1999.

[51] ——, "T-snakes: Topology adaptive snakes," *Medical Image Analysis*, vol. 4, pp. 73–91, 2000.

[52] D. Metaxas and D. Terzopoulos, "Dynamic deformation of solid primitives with constraints," *Proceedings of ACM SIGGRAPH*, pp. 309–312, July 1992.

[53] S. Mitchell, B. Lelieveldt, R. van der Geest, H. Bosch, J. Reiber, and M. Sonka, "Time continues segmentation of cardiac mr images sequences using active appearance motion models," *SPIE proceedings in medical imaging*, 2001.

[54] F. Mokhtarian, S. Abbasi, and J. Kittler, "Robust and efficient shape indexing through curvature scale space," in *British Machine Vision Conference*, 1996.

[55] ——, "Robust and efficient shape indexing through curvature scale space," in *Proceedings of British Machine Vision Conference*, Edinburgh, UK, 1996, pp. 53–62.

[56] H. Overhoff, A. Mastmeyer, and J. Ehrhardt, "Automatic landmark identification in 3-d image volumes by topgraphy conserving appoximation of contour dada," in *Proceedings of SPIE on Medical Imaging*, vol. 3661, 1999.

[57] A. Pentland and S. Sclaroff, "Closed-form solutions for physically based shape modeling," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 13, no. 7, pp. 715–729, July 1991.

[58] S. Pizer, D. Fritsch, P. Yushkevich, V. Jhonson, and E. Chaney, "Segmentation, registration, and measurement of shape variation via image object shape," *IEEE Trans. on Medical Imaging*, vol. 18, no. 10, pp. 851–865, October 1999.

[59] R. Plaenkers and P. Fua, "Articulated soft objects for multi-view shape and motion capture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, 2003.

[60] P. Prusinkiewicz, J. Hanan, and R. Mech, "An l-system-based plant modeling language," in *AGTIVE*, 1999, pp. 395–410.

[61] P. Prusinkiewicz, A. Lindenmayer, and J. Hanan, "Developmental models of herbaceous plants for computer imagery purposes," *Computer Graphics (SIGGRAPH 88 Conference Proceedings)*, vol. 222, no. 4, pp. 141–150, 1988.

[62] A. Rattangsi and R. Chin, "Scale-based detection of corners of planar curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 4, pp. 430–449, April 1992.

[63] M. Roberts, T. Cootes, and J. Adams, "Linking sequences of active appearance sub-models via constraints: an application in automated vertebral morphometry," in *Proc.BMVC*, vol. 1, 2003, pp. 349–358.

[64] T. Sabisch, A. Ferguson, and H. Bolouri, "Automatic landmark extraction using self-organising maps," *Proceedings of Medical Image Understanding and Analysis*, pp. 157–160, July 1997.

[65] S. Sclaroff and A. Pentland, "Search by shape examples: Modeling nonrigid deformation," in *Proc. 28th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA., October 1994.

[66] ——, "Modal matching for correspondence and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 17, no. 6, pp. 545–561, 1995.

[67] K. Siddiqi, B., and B. Kimia, "Parts of visual form: Computational aspects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 239–251, 1995.

[68] K. Siddiqi and B. Kimia, "Toward a shock grammar for recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 1996.

[69] R. Sokal and F. Rohlf, *Biometry: The Principles and Practice of Statistics in Biological Research*. W.H. Freeman and Company, September 1994.

[70] G. Stiny and J. Gips, "Shape grammars and the generative specification of painting and sculpture," *IFIP Congress*, pp. 125–135, August 1971.

[71] C. Studholme, D. Hawkes, and D. Hill, "A normalized entropy measure for multimodality image alignemet," in *SPIE Conference on Image Processing*, vol. 3338, 1998, pp. 132–143.

[72] R. Szeliski, D. Tonnesen, and D. Terzopolulos, "Modelling surfaces of arbitrary topology with dynamic particles," in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'93)*. IEEE Computer Society Press, 1993, pp. 82–87.

[73] D. Terzopoulos and D. Metaxas, "Dynamic 3d models with local and global deformations: Deformable superquadrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 703–714, July 1991.

[74] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on deformable models: Recovering 3d shape and nonrigid motion," *Artificial Intelligence*, vol. 36, no. 1, pp. 91–123, 1988.

[75] G. Turk, "Re-tiling polygonal surfaces," *ACM SIGRAPH*, vol. 16, no. 2, pp. 55–64, 1992.

[76] S. Ullman, *High-level Vision*, 2nd ed. Cambridge, Massachusetts: The MIT Press, 1997.

[77] R. Veltcamp and M. Tanase, *Content–Based Image retrieval Systems: A Survey.* Tech. Rep. UU-CS-2000-34. Dep. of Computing Science, Utrecht Univ., 2000.

[78] K. Wu and M. Levine, "Segmenting 3d objects into geons," in *ICIAP*, 1995, pp. 321–334.

[79] D. Zhang and G. Lu, "Generic fourier descriptors for shape-based image retrieval," in *Proc. of IEEE International Conference on Multimedia and Expo (ICME2002)*, Lausanne, Switzerland, August 26–29 2002.

[80] S. Zhu and A. Yuille, "Forms: A flexible object recognition and modeling system," *Int. J. Comp. Vision*, vol. 20, no. 3, pp. 187–212, 1996.