

# Active Shape Models and the Shape Approximation Problem

A. Hill, T. F. Cootes and C. J. Taylor  
Department of Medical Bio-Physics  
University of Manchester  
ah@sv1.smb.man.ac.uk

## Abstract

The Active Shape Model(ASM) is an iterative algorithm for image interpretation based upon a Point Distribution Model. Each iteration of the ASM has two steps: Image data interrogation followed by shape approximation. Here we consider the shape approximation step in detail. We present a new method of shape approximation which uses *directional constraints*. We show how the error term for the shape approximation problem can be extended to cope with directional constraints and present iterative solutions to the 2D and 3D problems. We also show how the error term can be modified to allow a closed solution in the 2D case.

## Keywords

Statistical Shape Models, Flexible Templates, Active Shape Models.

## 1 Introduction

We have previously described a generic approach to interpreting both 2D and 3D images [2–5]. The essential components of the system are a compact model representing the shape of a set of variable objects - a Point Distribution Model (PDM) - and an iterative method of image search. The combination is known as an Active Shape Model (ASM). A PDM is a statistical shape model which is generated from a set of annotated examples of the objects(s) to be modelled. The ASM is an iterative procedure which locates an instance of a PDM in a given image. Each iteration of the ASM has two steps: image data interrogation followed by shape approximation. In the first step a new instance of the model in the image is proposed. In the second step the proposed shape is approximated as closely as possible whilst applying shape constraints captured by the PDM.

Here, we consider the shape approximation step of the ASM in detail. We present a new approach to the problem which uses *directional constraints* to reduce the number of iterations required for the ASM to converge and increases the accuracy of the interpretation. We show how the representation of the errors we wish to minimise when approximating a given image shape can be expressed in a more formal way than previously and can be extended to cope with the case of

directional constraints. Iterative solutions for the most general case are presented for both the 2D and 3D problems. We also show that, by modifying the error term slightly, a closed, rather than iterative, solution to the shape approximation problem can be achieved in 2D.

## 2 Background

We will briefly review how a PDM is generated and how the ASM image search procedure locates an instance of the PDM within an image. For a detailed description the reader is referred to [2–4].

### 2.1 Point Distribution Models

A PDM is generated via a principal components analysis of a training set of  $N$  object descriptions  $\{\mathbf{y}_i, (1 \leq i \leq N)\}$ . An object description,  $\mathbf{y}_i$ , is simply a labelled set of points  $\{\mathbf{y}_{i,j}, (1 \leq j \leq n)\}$  which we will call *landmarks*. The analysis involves; aligning the set of examples into a common frame of reference,  $\{\mathbf{x}_i = \text{aligned}(\mathbf{y}_i), (1 \leq i \leq N)\}$ ; calculating the mean of the aligned examples,  $\bar{\mathbf{x}}$ , and the deviation from the mean of each aligned example  $\delta\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}}$ ; calculating the eigensystem of the the co-variance matrix of the deviations,  $\mathbf{C} = (1/N) \sum_{i=1}^N \delta\mathbf{x}_i \delta\mathbf{x}_i^T$ . The  $t$  principal eigenvectors of the eigensystem are then used to generate examples of the modelled objects via the expression :

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \quad (1)$$

where  $\mathbf{b}$  is a  $t$ -element vector of shape parameters and  $\mathbf{P}$  is a  $(2n \times t)$  in 2D or  $(3n \times t)$  in 3D matrix of  $t$  eigenvectors.

By selecting  $\mathbf{b}$  from a pre-defined shape vector space, established from the set of training examples, new instances of the modelled object(s) can be generated. This enables the PDM to represent previously unseen examples and forms the basis for locating examples of the modelled object(s) in unseen images via the ASM.

### 2.2 Active Shape Models

The ASM uses an iterative algorithm for locating an instance of a PDM in a given image assuming some initial guess of the shape,  $\mathbf{b}$ , and pose,  $Q$ , of the object(s) in the image is available. At each iteration the current instance of the PDM is projected into the image via  $Q$  and the image data in the vicinity of the model instance is interrogated, usually along profiles normal to the boundary/surface - see figure 1. This leads to a new proposed set of landmarks within the image. The ASM approximates this proposed model instance as closely as possible while remaining within the shape vector space. This approximation then provides the starting point for the ASM at the next iteration.

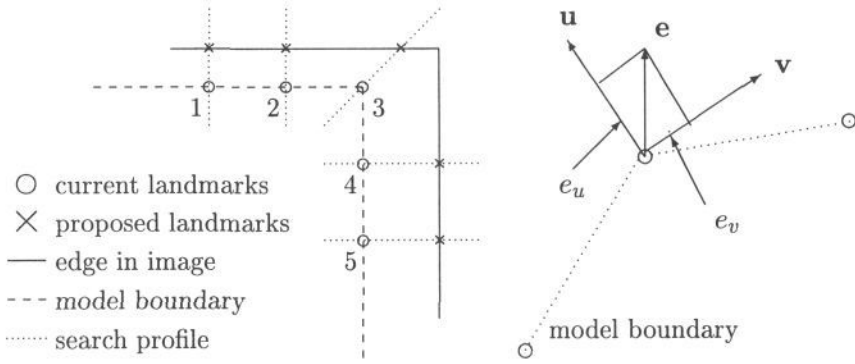


Figure 1: Searching for New Landmarks      Figure 2: Re-defining the Error Vector

### 3 The Shape Approximation Problem

The shape approximation problem we consider is simply stated : find a set of values,  $\{Q, \mathbf{b}\}$ , which best approximate a set of landmarks defined in the image frame,  $\{\mathbf{x}_i^I, (1 \leq i \leq n)\}$ . We use a least squares measure of the errors between the landmarks proposed in the image and the landmarks represented by  $\{Q, \mathbf{b}\}$  as the criterion for the best approximation. In this section we show how to generate the simplest form of the error term to be minimised. We then consider how the introduction of directional constraints re-defines the error vectors and extend the error term accordingly.

#### 3.1 The Standard Error Term

The  $i^{th}$  model point is defined by :

$$\mathbf{x}_i^M = \bar{\mathbf{x}}_i + \mathbf{P}_i \mathbf{b} \quad (2)$$

where  $\mathbf{P}_i$  is formed from the appropriate rows of  $\mathbf{P}$  in equation(1) associated with the  $i^{th}$  point. If  $\mathbf{b}$  is a  $(t \times 1)$  vector, then  $\mathbf{P}_i$  is a  $(2 \times t)$  matrix in 2D and a  $(3 \times t)$  matrix in 3D. The pose,  $Q$ , can be used to transform the image shape into the model frame or *vice versa*. We must decide which frame the errors are to be measured in. This is important because, as Horn[6] points out, if we find the pose which minimises the errors resulting from transforming the image points into the model frame,  $Q_1$  say, and the corresponding pose which minimises the errors resulting from transforming the model points into the image frame,  $Q_2$  say, then  $Q_1 \neq Q_2^{-1}$ . We wish to approximate the shape in the image frame as closely as possible since the aim of the ASM is to locate the instance of the model in the image. The error term we wish to minimise is thus :

$$E^2 = \sum_{i=1}^n \|\mathbf{e}_i^I\|^2, \quad \mathbf{e}_i^I = \mathbf{x}_i^I - Q(\mathbf{x}_i^M) \quad (3)$$

## 3.2 Directional Constraints

Consider the situation shown in figure 1. If we use a standard least-squares approach to minimise equation(3) we find that model points {1,2} are prevented from moving closer to their target points because of the increased vertical error in points {4,5}. Similarly, points {4,5} are prevented from moving closer to their target points because of the increase in the horizontal error of points {1,2}. However, points {1,2} lie on a horizontal edge and points {4,5} lie on a vertical edge. We would be willing, therefore, to allow points {1,2}/{4,5} to move freely in a horizontal/vertical direction, as long as the distance to their target boundaries were reduced. This is the so called aperture problem discussed in [1]. Below, we describe a general framework for measuring the errors along pre-defined unit vectors and weighting these modified errors in such a way that points are allowed to move more freely around the boundary/surface of the modelled object(s).

### 3.2.1 Directional Constraints in 2D

For the  $i^{th}$  point we have an error between the current model point and its corresponding target point of  $\mathbf{e}_i$  and a pre-defined unit vector,  $\mathbf{u}_i$ , normal to the boundary along which we want to measure the error i.e  $e_{u,i} = \mathbf{u}_i^T \mathbf{e}_i$  (see figure 2). To measure the error along a direction tangential to the boundary we use  $e_{v,i} = \mathbf{v}_i^T \mathbf{e}_i$  where  $\mathbf{v}_i = [-u_{i,y}, u_{i,x}]^T$ . The total weighted error term for the  $i^{th}$  point is now given by :

$$\begin{aligned} \alpha_i e_{u,i}^2 + \beta_i e_{v,i}^2 &= \alpha_i \mathbf{e}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{e}_i + \beta_i \mathbf{e}_i^T \mathbf{v}_i \mathbf{v}_i^T \mathbf{e}_i \\ &= \mathbf{e}_i^T \mathbf{W}_i \mathbf{e}_i, \text{ where } \mathbf{W}_i = \alpha_i \mathbf{u}_i \mathbf{u}_i^T + \beta_i \mathbf{v}_i \mathbf{v}_i^T \end{aligned} \quad (4)$$

By varying  $\alpha_i, \beta_i$  we can change the relative importance of the errors normal and tangential to the boundary at the  $i^{th}$  point.

### 3.2.2 Directional Constraints in 3D

Here, we consider measuring the error along both a unit normal to the surface,  $\mathbf{u}_i$ , and in the plane tangential to the surface. The error measured along the unit normal is  $e_{u,i} = \mathbf{u}_i^T \mathbf{e}_i$ . The component of the error vector in the tangential plane is given by :

$$\mathbf{e}_{v,i} = \mathbf{e}_i - (\mathbf{u}_i^T \mathbf{e}_i) \mathbf{u}_i = \mathbf{V}_i \mathbf{e}_i \text{ where } \mathbf{V}_i = (\mathbf{I} - \mathbf{u}_i \mathbf{u}_i^T) \quad (5)$$

The total weighted error term for the  $i^{th}$  point is given by :

$$\begin{aligned} \alpha_i e_{u,i}^2 + \beta_i \mathbf{e}_{v,i}^T \mathbf{e}_{v,i} &= \alpha_i \mathbf{e}_i^T \mathbf{u}_i \mathbf{u}_i^T \mathbf{e}_i + \beta_i \mathbf{e}_i^T \mathbf{V}_i^T \mathbf{V}_i \mathbf{e}_i \\ &= \mathbf{e}_i^T \mathbf{W}_i \mathbf{e}_i, \text{ where } \mathbf{W}_i = \alpha_i \mathbf{u}_i \mathbf{u}_i^T + \beta_i (\mathbf{I} - \mathbf{u}_i \mathbf{u}_i^T)^2 \end{aligned} \quad (6)$$

## 3.3 Generalised Error Term

In the previous section we showed that directional constraints result in a symmetric weight matrix associated with each error vector. In both the 2D and 3D directional

weighting schemes the weight matrices are derived from a set of unit normals. These normals are defined by the current instance of the model which is itself defined by the shape vector,  $\mathbf{b}$ . We will use  $\{\mathbf{W}_i^M, (1 \leq i \leq n)\}$  to denote the weight matrices defined in the model frame. In order to measure the errors in the image frame the unit normals must be rotated into the image frame. It can be shown that, if  $\mathbf{R}$  is the rotation matrix associated with the pose  $Q$ , then the weight matrices defined in the image frame are given by  $\{\mathbf{W}_i^I = \mathbf{R}\mathbf{W}_i^M\mathbf{R}^T, (1 \leq i \leq n)\}$ .

The error term we wish to minimise in the general case is now given by :

$$E^2 = \sum_{i=1}^n (\mathbf{x}_i^I - Q(\mathbf{x}_i^M))^T \mathbf{W}_i^I (\mathbf{x}_i^I - Q(\mathbf{x}_i^M)) \quad (7)$$

where  $\mathbf{x}_i^M$  is defined by equation(2) and  $Q(\mathbf{x}) = s\mathbf{R}\mathbf{x} + \mathbf{t}$  where  $s$  is the scale factor,  $\mathbf{R}$  is the rotation matrix and  $\mathbf{t}$  is the translation.

## 4 A Two Stage Iterative Solution

We can minimise equation(7) using a two stage iterative scheme in which we first assume we have a known approximation to  $\mathbf{b}$  and solve for  $Q$ ; then, given an approximation for  $Q$ , solve for  $\mathbf{b}$  and so on. This is the method used in the shape approximation step of the ASM as described previously[3, 4]. The difference here is that we are going to solve for  $\{Q, \mathbf{b}\}$  with respect to equation(7), thus guaranteeing the minimisation of the errors in the image frame for a combined set of  $\{Q, \mathbf{b}\}$  values in the general case of full  $(2 \times 2)$  or  $(3 \times 3)$  weight matrices.

### 4.1 Solving for $Q$ given $\mathbf{b}$ - the Alignment Problem

Here, we must solve :

$$\text{Min } E^2 = \sum_{i=1}^n (\mathbf{x}_i^I - \{s\mathbf{R}\mathbf{x}_i^M + \mathbf{t}\})^T \mathbf{W}_i^I (\mathbf{x}_i^I - \{s\mathbf{R}\mathbf{x}_i^M + \mathbf{t}\}) \quad (8)$$

where  $\mathbf{x}_i^M$  is defined by equation(2). We consider the 2D and 3D cases separately.

#### 4.1.1 Alignment in 2D

In 2D we use  $s\mathbf{R} = \begin{bmatrix} c & -d \\ d & c \end{bmatrix}$  where  $c = s \cos(\theta)$ ,  $d = s \sin(\theta)$  and  $\theta$  is the angle of rotation. In order to minimise equation(8) we set the partial derivatives to zero :

$$\frac{\partial E^2}{\partial v_j} = 2 \sum_{i=1}^n \left( \frac{\partial Q(\mathbf{x}_i^M)}{\partial v_j} \right)^T \mathbf{W}_i^I (\mathbf{x}_i^I - Q(\mathbf{x}_i^M)) = 0, \quad \begin{cases} (1 \leq j \leq 4) \\ v_1 = c, v_2 = d, \\ v_3 = t_x, v_4 = t_y \end{cases} \quad (9)$$

The partial derivatives of  $Q$  are given by :

$$\frac{\partial Q(\mathbf{x}_i^M)}{\partial v_1} = \mathbf{x}_i^M, \quad \frac{\partial Q(\mathbf{x}_i^M)}{\partial v_2} = \begin{bmatrix} -x_{i,y}^M \\ x_{i,x}^M \end{bmatrix}, \quad \frac{\partial Q(\mathbf{x}_i^M)}{\partial v_3} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \frac{\partial Q(\mathbf{x}_i^M)}{\partial v_4} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (10)$$

We can express equation(9) as follows :

$$\sum_{i=1}^n \left( \frac{\partial Q(\mathbf{x}_i^M)}{\partial v_j} \right)^T \mathbf{W}_i^T \sum_{k=1}^4 \frac{\partial Q(\mathbf{x}_i^M)}{\partial v_k} v_k = \sum_{i=1}^n \left( \frac{\partial Q(\mathbf{x}_i^M)}{\partial v_j} \right)^T \mathbf{W}_i^T \mathbf{x}_i^I \quad (1 \leq j \leq 4) \quad (11)$$

These 4 equations in the 4 unknowns  $\{c, d, t_x, t_y\}$  can be solved using standard methods. For scalar weights,  $\mathbf{W}_i^T = w_i$ , this solution can be shown to be the same as that described for the original ASM method [4].

#### 4.1.2 Alignment in 3D

To minimise equation(8) in 3D when using scalar weights,  $\mathbf{W}_i^T = w_i$ , we can use Horn's method of alignment [6] which uses unit quaternions to represent the rotation. Horn's method relies on being able to isolate the rotational element of the alignment problem. The scale and translation are computed once the rotation is known. By using quaternions the alignment of two 3D pointsets can be solved as a linear problem via an eigenvector analysis. Here, the coupling of the components of the error vectors,  $\mathbf{e}_i^T$ , via the weight matrices,  $\mathbf{W}_i^T$ , means that the rotation cannot be isolated in this fashion. We have used an alternative, iterative approach.

We represent the rotation using quaternions but allow the quaternion to also perform a scaling (i.e. we no longer consider *unit* quaternions). We will use Horn's notation for quaternions :  $\hat{\mathbf{q}} = q_0 + q_x i + q_y j + q_z k$ . The complex conjugate is defined by  $\hat{\mathbf{q}}^* = q_0 - q_x i - q_y j - q_z k$  and  $\hat{\mathbf{q}} \cdot \hat{\mathbf{q}}^* = q_0^2 + q_x^2 + q_y^2 + q_z^2 = \|\hat{\mathbf{q}}\|^2$ . For the definition of multiplication etc. see Horn[6]. A 3D point,  $\mathbf{p}$ , is represented by the purely imaginary quaternion  $\hat{\mathbf{p}} = 0 + p_x i + p_y j + p_z k$  and a rotation of  $\mathbf{p}$  is defined by  $\hat{\mathbf{q}} \hat{\mathbf{p}} \hat{\mathbf{q}}^*$  with  $\hat{\mathbf{q}} \cdot \hat{\mathbf{q}}^* = 1$ . To enable  $\hat{\mathbf{q}}$  to perform a scaling as well as a rotation we do not require a unit quaternion but set the scale using  $s = \hat{\mathbf{q}} \cdot \hat{\mathbf{q}}^*$ . We can write equation(8) as :

$$\text{Min } E^2 = \sum_{i=1}^n (\mathbf{x}_i^I - \Im\{\hat{\mathbf{q}} \hat{\mathbf{x}}_i^M \hat{\mathbf{q}}^*\} - \mathbf{t})^T \mathbf{W}_i^T (\mathbf{x}_i^I - \Im\{\hat{\mathbf{q}} \hat{\mathbf{x}}_i^M \hat{\mathbf{q}}^*\} - \mathbf{t}) \quad (12)$$

where  $\Im\{\hat{\mathbf{q}}\}$  is the imaginary part of the quaternion  $\hat{\mathbf{q}}$ .

To solve equation(12) we use an iterative approach as follows: Assume we have some estimate of  $\hat{\mathbf{q}}$  available at the  $r^{\text{th}}$  iteration,  $\hat{\mathbf{q}}_{[r]}$  say. We generate a new estimate at the  $(r+1)^{\text{th}}$  iteration using  $\hat{\mathbf{q}}_{[r+1]} = \hat{\mathbf{q}}_{[r]} + \hat{\delta}$ . We can now write :

$$\Im\{\hat{\mathbf{q}}_{[r+1]} \hat{\mathbf{x}}_i^M \hat{\mathbf{q}}_{[r+1]}^*\} = \Im\{\hat{\mathbf{q}}_{[r]} \hat{\mathbf{x}}_i^M \hat{\mathbf{q}}_{[r]}^* + \hat{\delta} \hat{\mathbf{x}}_i^M \hat{\mathbf{q}}_{[r]}^* + \hat{\mathbf{q}}_{[r]} \hat{\mathbf{x}}_i^M \hat{\delta}^* + \hat{\delta} \hat{\mathbf{x}}_i^M \hat{\delta}^*\}$$

We assume that  $\hat{\delta}$  is small with respect to  $\hat{\mathbf{q}}_{[r]}$  and approximate the previous expression using :

$$\Im\{\hat{\mathbf{q}}_{[r]} \hat{\mathbf{x}}_i^M \hat{\mathbf{q}}_{[r]}^* + \hat{\delta} \hat{\mathbf{x}}_i^M \hat{\mathbf{q}}_{[r]}^* + \hat{\mathbf{q}}_{[r]} \hat{\mathbf{x}}_i^M \hat{\delta}^*\} = s_{[r]} \mathbf{R}_{[r]} \mathbf{x}_i^M + \mathbf{G}_i \hat{\delta} \quad (13)$$

$$\mathbf{G}_i = \begin{bmatrix} g_{i,x} & g_{i,0} & g_{i,z} & -g_{i,y} \\ g_{i,y} & -g_{i,z} & g_{i,0} & g_{i,x} \\ g_{i,z} & g_{i,y} & -g_{i,x} & g_{i,0} \end{bmatrix}, \quad \hat{\mathbf{g}}_i = 2 \hat{\mathbf{x}}_i^M \hat{\mathbf{q}}_{[r]}^* \quad (14)$$

where we have used  $\Im\{\mathring{\mathbf{a}}\mathring{\mathbf{x}}\mathring{\mathbf{b}}^*\} = \Im\{\mathring{\mathbf{b}}\mathring{\mathbf{x}}\mathring{\mathbf{a}}^*\}$  for the purely imaginary quaternion  $\mathring{\mathbf{x}}$ . In equation(13)  $\mathring{\delta}$  is regarded as a 4-element vector in the term  $\mathbf{G}_i\mathring{\delta}$ . Equation(12) now becomes :

$$\text{Min } E^2 = \sum_{i=1}^n (\mathbf{z}_i - \mathbf{G}_i\mathring{\delta} - \mathbf{t})^T \mathbf{W}_i^I (\mathbf{z}_i - \mathbf{G}_i\mathring{\delta} - \mathbf{t}), \quad \mathbf{z}_i = \mathbf{x}_i^I - s_{[r]}\mathbf{R}_{[r]}\mathbf{x}_i^M \quad (15)$$

Equation(15) is linear with respect to  $\mathring{\delta}, \mathbf{t}$  and we can write :

$$\text{Min } E^2 = \sum_{i=1}^n \left( \mathbf{z}_i - \sum_{k=1}^7 \frac{\partial \mathbf{e}_i^I}{\partial v_k} v_k \right)^T \mathbf{W}_i^I \left( \mathbf{z}_i - \sum_{k=1}^7 \frac{\partial \mathbf{e}_i^I}{\partial v_k} v_k \right) \quad (16)$$

with  $v_1 = \delta_0, v_2 = \delta_x, v_3 = \delta_y, v_4 = \delta_z, v_5 = t_x, v_6 = t_y, v_7 = t_z$  and the partial derivatives defined by :

$$\begin{aligned} \frac{\partial \mathbf{e}_i^I}{\partial v_1} &= \mathbf{g}_{i,1}, \quad \frac{\partial \mathbf{e}_i^I}{\partial v_2} = \mathbf{g}_{i,2}, \quad \frac{\partial \mathbf{e}_i^I}{\partial v_3} = \mathbf{g}_{i,3}, \quad \frac{\partial \mathbf{e}_i^I}{\partial v_4} = \mathbf{g}_{i,4} \\ \frac{\partial \mathbf{e}_i^I}{\partial v_5} &= [1, 0, 0]^T, \quad \frac{\partial \mathbf{e}_i^I}{\partial v_6} = [0, 1, 0]^T, \quad \frac{\partial \mathbf{e}_i^I}{\partial v_7} = [0, 0, 1]^T \end{aligned}$$

where  $\mathbf{g}_{i,j}$  is the  $j^{\text{th}}$  column of  $\mathbf{G}_i$ . We can set the partial differentials of  $E^2$  to zero and obtain :

$$\sum_{i=1}^n \left( \frac{\partial \mathbf{e}_i^I}{\partial v_j} \right)^T \mathbf{W}_i^I \sum_{k=1}^7 \frac{\partial \mathbf{e}_i^I}{\partial v_k} v_k = \sum_{i=1}^n \left( \frac{\partial \mathbf{e}_i^I}{\partial v_j} \right)^T \mathbf{W}_i^I \mathbf{z}_i, \quad (1 \leq j \leq 7) \quad (17)$$

which gives 7 equations in the 7 unknowns. Once  $\mathring{\delta}$  has been determined the new estimate of the quaternion defining the rotation and scaling is given by  $\mathring{\mathbf{q}}_{[r+1]} = \mathring{\mathbf{q}}_{[r]} + \mathring{\delta}$ . The iterative scheme is deemed to have converged when  $\|\mathring{\delta}\|/\sqrt{s} < \epsilon$ . We have used  $\epsilon = 10^{-6}$ .

## 4.2 Solving for $\mathbf{b}$ given $Q$

Here, we substitute equation(2) into equation(7) to give :

$$E^2 = \sum_{i=1}^n (\mathbf{x}_i^I - Q(\bar{\mathbf{x}}_i + \mathbf{P}_i \mathbf{b}))^T \mathbf{W}_i^I (\mathbf{x}_i^I - Q(\bar{\mathbf{x}}_i + \mathbf{P}_i \mathbf{b}))$$

which can be written as :

$$E^2 = \sum_{i=1}^n (\mathbf{x}_i^I - s\mathbf{R}\bar{\mathbf{x}}_i - \mathbf{t} - s\mathbf{R}\mathbf{P}_i \mathbf{b})^T \mathbf{W}_i^I (\mathbf{x}_i^I - s\mathbf{R}\bar{\mathbf{x}}_i - \mathbf{t} - s\mathbf{R}\mathbf{P}_i \mathbf{b})$$

We minimise as before, by setting the partial derivatives to zero :

$$\frac{\partial E^2}{\partial b_j} = 2 \sum_{i=1}^n \left( -s\mathbf{R}\mathbf{P}_i \frac{\partial \mathbf{b}}{\partial b_j} \right)^T \mathbf{W}_i^I (\mathbf{x}_i^I - s\mathbf{R}\bar{\mathbf{x}}_i - \mathbf{t} - s\mathbf{R}\mathbf{P}_i \mathbf{b}) = 0, \quad (1 \leq j \leq t)$$

This expression is rearranged to give :

$$\frac{\partial \mathbf{b}}{\partial b_j}{}^T \sum_{i=1}^n (\mathbf{P}_i^T \mathbf{R}^T \mathbf{W}_i^T \mathbf{R} \mathbf{P}_i) \mathbf{b} = \frac{\partial \mathbf{b}}{\partial b_j}{}^T \sum_{i=1}^n \mathbf{P}_i^T \mathbf{R}^T \mathbf{W}_i^T \frac{1}{s} (\mathbf{x}_i^t - s \mathbf{R} \bar{\mathbf{x}}_i - \mathbf{t})$$

We can use the relationship  $\mathbf{W}_i^t = \mathbf{R} \mathbf{W}_i^M \mathbf{R}^T$  to simplify the previous equation :

$$\sum_{i=1}^n (\mathbf{P}_i^T \mathbf{W}_i^M \mathbf{P}_i) \mathbf{b} = \sum_{i=1}^n \mathbf{P}_i^T \mathbf{W}_i^M \left( \frac{1}{s} \mathbf{R}^T (\mathbf{x}_i^t - \mathbf{t}) - \bar{\mathbf{x}}_i \right)$$

This equation can be expressed simply as :

$$(\mathbf{P}^T \mathbf{W}^M \mathbf{P}) \mathbf{b} = \mathbf{P}^T \mathbf{W}^M (Q^{-1}(\mathbf{x}^t) - \bar{\mathbf{x}}) \quad (18)$$

where  $\mathbf{W}^M = \text{diag}\{\mathbf{W}_i^M\}$ . For scalar weights,  $\mathbf{W}_i^M = w_i$ ,  $\mathbf{W}^M$  simplifies to a diagonal matrix and equation(18) represents the same solution as that used in [3].

## 5 A Closed Solution for the 2D Problem

The non-linear relationship between  $\{Q, \mathbf{b}\}$  in equation(7) can be handled using the 2-stage scheme described above. To obtain a closed solution we linearise the expression by measuring the errors in the model frame rather than the image frame. In section 6 we will see that, for a real image interpretation problem, the increased errors resulting from measuring the errors in the model frame are statistically insignificant. We measure the errors in the model frame as follows :

$$\mathbf{e}_i^M = Q^{-1}(\mathbf{x}_i^t) - (\bar{\mathbf{x}}_i + \mathbf{P}_i \mathbf{b}) \quad (19)$$

The error term we wish to minimise becomes :

$$\begin{aligned} E^2 &= \sum_{i=1}^n (Q^{-1}(\mathbf{x}_i^t) - (\bar{\mathbf{x}}_i + \mathbf{P}_i \mathbf{b}))^T \mathbf{W}_i^M (Q^{-1}(\mathbf{x}_i^t) - (\bar{\mathbf{x}}_i + \mathbf{P}_i \mathbf{b})) \\ &= \sum_{i=1}^n \left( \frac{1}{s} \mathbf{R}^T (\mathbf{x}_i^t - \mathbf{t}) - \bar{\mathbf{x}}_i - \mathbf{P}_i \mathbf{b} \right)^T \mathbf{W}_i^M \left( \frac{1}{s} \mathbf{R}^T (\mathbf{x}_i^t - \mathbf{t}) - \bar{\mathbf{x}}_i - \mathbf{P}_i \mathbf{b} \right) \quad (20) \\ &= \sum_{i=1}^n \left( \begin{bmatrix} c & -d \\ d & c \end{bmatrix} \mathbf{x}_i^t + \mathbf{t}' - \bar{\mathbf{x}}_i - \mathbf{P}_i \mathbf{b} \right)^T \mathbf{W}_i^M \left( \begin{bmatrix} c & -d \\ d & c \end{bmatrix} \mathbf{x}_i^t + \mathbf{t}' - \bar{\mathbf{x}}_i - \mathbf{P}_i \mathbf{b} \right) \end{aligned}$$

where  $c = \cos(-\theta)/s$ ,  $d = \sin(-\theta)/s$ ,  $\mathbf{t}' = -\mathbf{R}^T \mathbf{t}/s$ . Equation(20) can be written as :

$$E^2 = \sum_{i=1}^n \left( \sum_{j=1}^{t+4} \frac{\partial \mathbf{e}_i^M}{\partial v_j} v_j - \bar{\mathbf{x}}_i \right)^T \mathbf{W}_i^M \left( \sum_{j=1}^{t+4} \frac{\partial \mathbf{e}_i^M}{\partial v_j} v_j - \bar{\mathbf{x}}_i \right) \quad (21)$$

where  $v_1 = c, v_2 = d, v_3 = t'_x, v_4 = t'_y, v_{j+4} = b_j$  ( $1 \leq j \leq t$ ). The partial derivatives in equation(21) are given by :

$$\frac{\partial \mathbf{e}_i^M}{\partial v_1} = \mathbf{x}_i^t, \quad \frac{\partial \mathbf{e}_i^M}{\partial v_2} = \begin{bmatrix} -x_{i,y}^t \\ x_{i,x}^t \end{bmatrix}, \quad \frac{\partial \mathbf{e}_i^M}{\partial v_3} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \frac{\partial \mathbf{e}_i^M}{\partial v_4} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \frac{\partial \mathbf{e}_i^M}{\partial v_{j+4}} = -\mathbf{P}_{i,j} \quad (22)$$

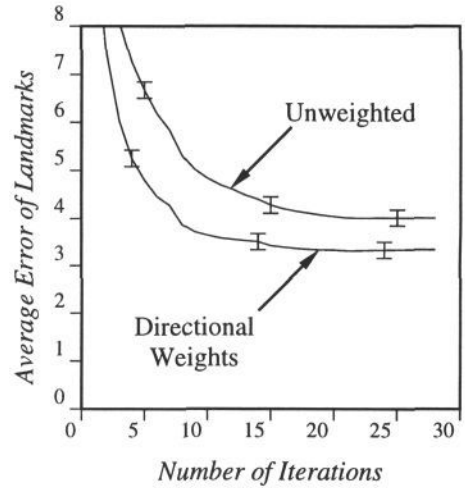
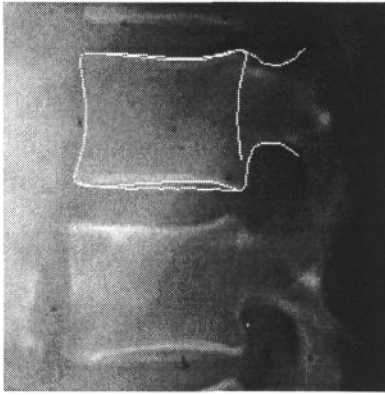


Figure 3: Improved Search with Directional Constraints

where  $\mathbf{p}_{i,j}$  is the  $j^{th}$  column of  $\mathbf{P}_i$ . To minimise  $E^2$  in equation(21) we set the partial derivatives to zero,  $\partial E^2 / \partial v_k = 0$ , ( $1 \leq k \leq t+4$ ), and obtain :

$$\sum_{i=1}^n \left( \frac{\partial \mathbf{e}_i^M}{\partial v_k} \right)^T \mathbf{W}_i^M \sum_{j=1}^{t+4} \frac{\partial \mathbf{e}_i^M}{\partial v_j} v_j = \sum_{i=1}^n \left( \frac{\partial \mathbf{e}_i^M}{\partial v_k} \right)^T \mathbf{W}_i^M \bar{\mathbf{x}}_i \quad (1 \leq k \leq t+4) \quad (23)$$

which form  $(t+4)$  equations in the  $(t+4)$  unknowns. In the special case of unit weights,  $\mathbf{W}_i^M = \mathbf{I}$ , we note that the system of equations(23) takes the form :

$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{B}_2^T & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_Q \\ \mathbf{y}_b \end{bmatrix}, \quad \mathbf{q} = [c, d, t'_x, t'_y]^T \quad (24)$$

which we can solve as :

$$\begin{aligned} (\mathbf{B}_1 - \mathbf{B}_2 \mathbf{B}_2^T) \mathbf{q} &= \mathbf{y}_Q - \mathbf{B}_2 \mathbf{y}_b \\ \mathbf{b} &= \mathbf{y}_b - \mathbf{B}_2^T \mathbf{q} \end{aligned}$$

## 6 Results

To see how using directional constraints improves the performance of the ASM we used a set of images and accompanying PDM of lumbar vertebrae [5] - see figure 3. We selected a subset of the training images on which to test the method, the correct labelling of the landmarks being known for each image. The ASM was applied to each of the 25 selected images 8 times. The initial set of shape parameters was set to  $\mathbf{b} = \mathbf{0}$  (i.e. the mean shape). The initial pose was a random perturbation of the pose required to align the mean shape with the known landmarks for the given image. At each iteration of the ASM the average distance of each landmark from its annotated position was computed for all landmarks and all images. The consolidated results are shown in figure 3 for the ASM both with and without

directional weights. For these experiments we used  $\alpha_i = 1, \beta_i = 0.09 \forall i$ . It can be seen clearly that directional weighting improves both the rate of convergence and the accuracy of the final interpretation.

In order to test how the modification of the error term used in section 5 to achieve a closed form solution for  $\{Q, \mathbf{b}\}$  affects the interpretation generated by the ASM we have performed a similar set of experiments. In this case the ASM using the standard 2-stage iterative algorithm for the errors defined in the image frame was compared with the ASM using the closed solution for errors defined in the model frame. For the vertebra location problem described above we found the increased errors resulting from calculating the errors in the model frame were statistically insignificant.

## 7 Conclusions

We have presented a detailed and extended analysis of the shape approximation problem associated with the ASM algorithm. We have shown how to establish error terms relevant to both the image and model frames for the general case of symmetric weight matrices. This extension is necessary to support a new approach to the shape approximation problem which uses directional constraints to allow the points more freedom of movement in specified directions during ASM search. We have presented iterative solutions for the most general case in both 2D and 3D. We have shown how directional weighting results in an ASM which requires fewer image iterations than an un-weighted ASM and also generates more accurate interpretations. We have also shown that it is possible to obtain a closed solution to the shape approximation problem in 2D which, for a real image interpretation problem, generates interpretations which are statistically no less accurate than those generated by the corresponding 2-stage iterative solution.

## References

- [1] A. Blake, R. Curwen, and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision*, 11(2):127–145, 1993.
- [2] T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham. A trainable method of parametric shape description. *Image and Vision Computing*, 10(5):289–294, June 1992.
- [3] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):276–285, July 1994.
- [4] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, Jan. 1995.
- [5] A. Hill, T. F. Cootes, C. J. Taylor, and K. Lindley. Medical image interpretation: A generic approach using deformable templates. *Journal of Medical Informatics*, 19(1):47–59, 1994.
- [6] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, Apr. 1987.