

# Extending the Point Distribution Model using Polar Coordinates

Tony Heap and David Hogg

School of Computer Studies, University of Leeds, Leeds LS2 9JT, UK  
email: `ajh@scs.leeds.ac.uk`

**Abstract.** The Point Distribution Model (PDM) has already proved useful for many tasks involving the location or tracking of deformable objects. A principal limitation lies in the fact that non-linear variation must be approximated by a combination of linear variations, resulting in a non-optimal model which can produce implausible object shapes. The Polynomial Regression PDM improves on the PDM by allowing polynomial deformation. However, computational complexity is greatly increased, and the model still fails for objects in which bending or pivoting occurs. We propose an extension to the PDM which selectively uses polar coordinates at little computational cost, and give examples to show that models produced are both more compact and less likely to generate implausible shapes than either of the above methods. We also give an algorithm which automatically classifies model landmark points into the Cartesian or polar domain, based on training set analysis.

## 1 Introduction

Models are used widely in computer vision; image features can be located, tracked or classified using *a priori* knowledge of object shape. Many objects are non-rigid, and thus require a *deformable* model to capture shape variability.

One such model is the Point Distribution Model (PDM) [1], which has already been used as the basis for several vision applications [2, 3, 4, 5, 6]. An object is defined in terms of landmark points positioned strategically on various object features, and at regular intervals in between. By labelling such points on a set of training examples of the object, a statistical approach can be used to discover the mean object shape, and the major modes of shape variation.

The standard PDM is based purely on linear statistics; for any particular mode of variation, landmark points can move only along a straight line. Non-linear movement is achieved by combining two or more modes. This situation is not ideal, firstly because the most compact representation of shape variability might not be achieved, and secondly because implausible shapes can be generated, due to the incorrect assumption that the variation modes are independent. The effect is particularly bad when the object being modelled can bend or pivot.

A non-linear improvement to the model has been suggested by Sozou *et al* [7]. The Polynomial Regression PDM removes the assumption that the modes of variation are independent; minor modes are described in terms of polynomial functions of more major modes. This allows for non-linear variation and produces a more compact model. However, this method still fails for objects which can bend or pivot through large angles, and is also much more computationally intensive than the standard approach.

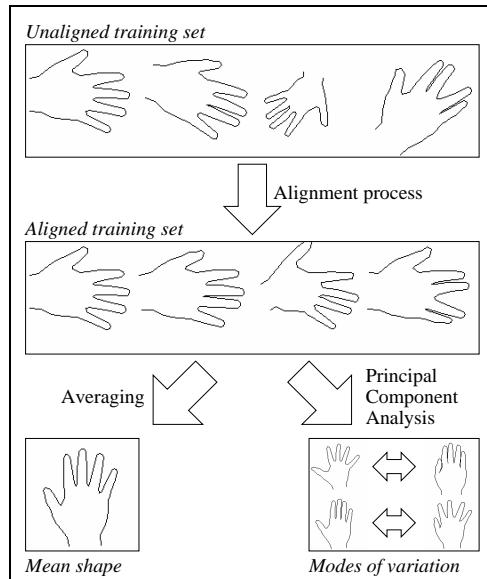
Since bending and pivoting are such major features of so many classes of object, it is suggested that these actions should be modelled directly. This can be achieved within the bounds of a PDM by the use of polar coordinates. Objects with both linear and angular deformation are best served by a hybrid model in which each landmark point can choose to exist either in Cartesian or polar space.

The remainder of this paper gives more details on these ideas; the standard PDM and Polynomial Regression PDM are briefly described. The construction of a Cartesian-Polar Hybrid PDM is explained, along with an algorithm for automatically determining the classification (Cartesian versus polar) of landmark points. Finally, a comparison of the three models is made both graphically and numerically, and some conclusions are drawn.

## 2 The Standard PDM

A PDM is built purely from the statistical analysis of a number of examples of the object to be modelled [1]. Given a collection of training images of an object, the Cartesian coordinates of  $N$  strategically-chosen landmark points are recorded for each image. Training example  $e$  is represented by a vector  $\mathbf{x}_e = (x_{e1}, y_{e1}, \dots, x_{eN}, y_{eN})$  (for a 2D model).

Figure 1 summarizes the training process; the examples are aligned (translated, rotated and scaled) using a weighted least squares algorithm, and the mean shape  $\bar{\mathbf{x}}$  is calculated by finding the mean position of each landmark. Modes of variation are found using Principal Component Analysis (PCA) on examples' deviations from the mean, and are represented by  $N$  'variation vectors'  $\mathbf{v}_j$ .



**Fig. 1.** Training a Point Distribution Model

An object shape  $\mathbf{x}$  is generated by adding linear combinations of the  $t$  most significant variation vectors to the mean shape:

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_{j=1}^t b_j \mathbf{v}_j \quad (1)$$

where  $b_j$  is the weighting for the  $j^{\text{th}}$  variation vector.

By ensuring,  $t \ll N$ , we extract only the important deformations, discarding training set noise, and thus we can compactly capture object shape and variation. In most cases, deformation is restricted to plausible shapes, making the PDM an ideal tool for robust location or tracking of non-rigid objects. However, the linear nature of the PDM can limit its usefulness; models are not as compact as they could be, and implausible shapes can occur (see [7] for examples).

### 3 The Polynomial Regression PDM

In the formulation of the Polynomial Regression PDM (PRPDM) [7], Sozou *et al* observe that modes of variation are, in general, not independent. It is suggested that these dependencies can be modelled by expressing the second and subsequent modes of variation in terms of a polynomial function of the first:

$$b_j = a_{j0} + a_{j1}b_1 + a_{j2}b_1^2 + \dots + a_{jd}b_1^d \quad (2 \leq j \leq t) \quad (2)$$

where  $d$  is the degree of the polynomial being used. The values of the  $a_{jk}$  coefficients are determined by finding polynomials of best fit through the training data. Because, in general, not all legitimate variation can be captured with only one shape parameter, it is necessary to perform this process iteratively on residual deviation from the polynomial curves (hence the term ‘regression’), until the only remaining variation is due to noise in the training data.

The PRPDM generally performs better than the standard PDM; as expected, the model becomes more accurate as the polynomial degree  $d$  is increased.

However, there are two major drawbacks with the PRPDM. The first is one of computational complexity – as well as the overhead incurred due to polynomial calculations, every regressive iteration requires an entire coordinate transform, so the algorithm is likely to be almost  $t$  times as expensive as for the standard PDM (where  $t$  is the number of variation modes). Secondly, bending and pivoting deformations are still not modelled accurately; the problem becomes accentuated when large angles of pivot are involved.

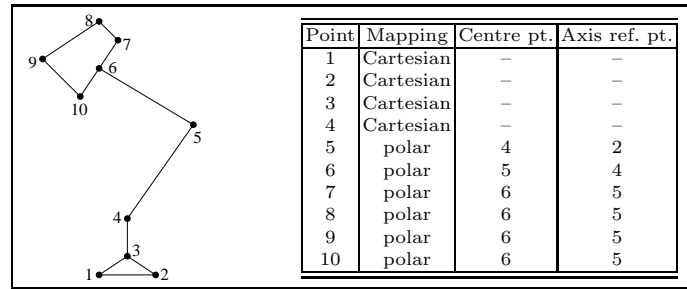
### 4 The Cartesian-Polar Hybrid PDM

Our Cartesian-Polar Hybrid PDM attempts to overcome limitations of both the standard PDM and the PRPDM by allowing angular movement to be modelled directly. This is achieved by a reparameterization of coordinates according to some predefined mapping function. Landmark points which appear to pivot about some other point in the model are transformed into polar coordinates (over all training examples), with the suspected pivot point as origin and some other point chosen sensibly to provide an axis reference. Landmark points which have no such angular behaviour remain in Cartesian form.

A training example  $\mathbf{x} = (x_1, y_e, \dots, x_N, y_N)$  is reparameterized by a mapping  $P$  into a parameter vector  $\mathbf{p} = (p_1, p_2, \dots, p_{2N-1}, p_{2N})$  whereby each  $(x, y)$  point in  $\mathbf{x}$  is either mapped in a Cartesian or polar fashion as follows:

$$\begin{array}{lcl}
 (x_i, y_i) & \xrightarrow{\text{Cartesian map}} & p_{2i-1} = x_i \\
 & & p_{2i} = y_i \\
 \\
 (x_i, y_i) & \xrightarrow{\text{polar map}} & p_{2i-1} = r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \\
 & & p_{2i} = \theta_i = \phi_i + \phi_a \\
 & & \text{where } \phi_p = \tan^{-1} \frac{y_p - y_c}{x_p - x_c} \\
 & \text{centre point} = c & \\
 & \text{axis point} = a & 
 \end{array}$$

The axis reference point  $a$  for a polar point  $p$  should be chosen with care. If  $p$  is pivoting off some locally rigid structure within the object (about a centre point  $c$ ) then  $a$  should be a point within this structure. For maximum stability,  $a$  should be as distant from  $c$  as possible (averaged over the training set). Figure 2 gives an example labelling.



**Fig. 2.** Landmark point mappings used for an anglepoise lamp

By allowing angles to be measured relative to an axis reference point, it is possible to more accurately model objects which have series of pivot points (such as in an anglepoise lamp), or continuous bending regions (such as the tadpoles and chromosomes described by Sozou *et al* [7]). For obvious reasons, it is important that cyclic dependencies of polar points are avoided.

The mean shape in parametric form  $\bar{\mathbf{p}}$  is found simply by averaging each parameter over the training set. PCA is used on the training parameter vectors in order to find the major modes of variation. Prior to this, it is necessary to remove any bias created by mapping to polar coordinates; polarized points which are close to their pivot point give rise to large angular movement for relatively small displacements. This is avoided by multiplying the  $\theta$  parameter for each polarized point by the mean (over the training set) of the corresponding magnitude parameters  $r$ . This results in  $\theta$  representing a true *distance* (along an arc) rather than an angle.

For practical use of the model, angle values are more desirable than arc distance values, so after PCA, all  $\theta$  parameters in the variation vectors are rescaled back to angles by redividing by the mean of the corresponding  $r$  parameters.

Generating a shape from shape parameters  $b_1 \dots b_t$  is as for the standard PDM, using (1), but of course the polar points must be reparameterized back into Cartesian coordinates afterwards:

$$\begin{array}{lcl}
p_{2i-1} & \xRightarrow{\text{Cartesian map}} & x_i = p_{2i-1} \\
p_{2i} & & y_i = p_{2i} \\
p_{2i-1} (= r_i) & \xRightarrow{\text{polar map}} & x_i = x_c + r_i \cos(\theta_i + \psi) \\
p_{2i} (= \theta_i) & & y_i = y_c + r_i \sin(\theta_i + \psi) \\
& & \text{centre point} = c \\
& & \text{axis point} = a \\
& & \text{where } \psi = \tan^{-1} \frac{y_a - y_c}{x_a - x_c}
\end{array}$$

Because the position of point  $i$  is dependent on the positions of points  $c$  and  $a$ , these points must be remapped *before* point  $i$ .

#### 4.1 Automatic Pivot Location

Of chief importance to the hybrid model is the choice of a suitable mapping. This can be done by hand when the pivot points are obvious and the model has relatively few landmark points. However, it is possible to discover pivot points automatically by analysis of the training set.

The most ideal mapping would provide the most *compact* model. Compactness can be measured in terms of the amount of variance existing in the model. It is mentioned above that, when PCA is performed, the resulting eigenvalues provide a measure of the variance captured by each mode of variation. The sum of the magnitudes of these eigenvalues provides a measure of the *overall* variance present in the model. The smaller this value, the more compact the model.

The best mapping, then, is the one which minimizes this eigenvalue sum. Trying every possible combination of mappings takes exponential time and is thus not feasible for large models. A heuristic method is suggested below.

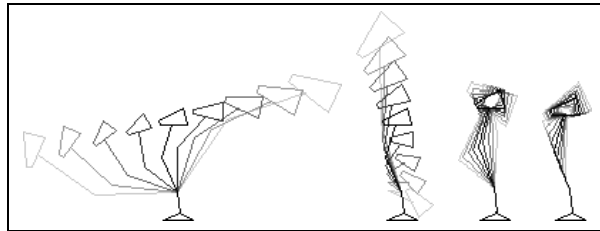
1. Following training set alignment, the mean *Cartesian* shape  $\bar{x}$  is found.
2. For each training example, the distance of each point from its mean position is found, and an average taken.
3. A point whose average distance from its mean position is less than a specified threshold value (we have used 0.05 times the total model size) is included in a *base set* of points, being a relatively stable set of points from which potential polar points can be pivoted. Since variation is small in the base set, a Cartesian mapping is assigned to each point. Each base set point is also assigned an axis reference partner, being its most distant base set point.
4. Choose the next unassigned point. Call this point  $a$ .
5. Assign a Cartesian mapping to point  $a$  and perform PCA on the base set combined with point  $a$ . The sum of the eigenvalues produced gives a measure of the total variance of the model when  $a$  is mapped in the Cartesian domain.
6. Take each base set point in turn, and call it point  $b$ . Calculate the distance between  $a$  and  $b$  for every training example. Find the variance of these distances. If it is small then it is possible that  $a$  pivots about  $b$ . To test this hypothesis, assign a polar mapping to point  $a$ , with point  $b$  as centre point and point  $b$ 's axis reference partner as axis reference. Perform PCA on the base set plus point  $a$ , and record the sum of the eigenvalues produced.
7. Once all base set points have been tested against point  $a$ , find the lowest recorded eigenvalue sum. Point  $a$  is assigned the corresponding mapping (be it Cartesian or polar), and is added to the base set. However, if *no* base set points were tested against point  $a$  then it is *not* added to the base set. This allows for cases where  $a$  is best pivoted around an as-yet unassigned point.

8. Iterate steps 4 through 7 until no more mapping assignments are possible. Assign a Cartesian mapping to all remaining unassigned points.

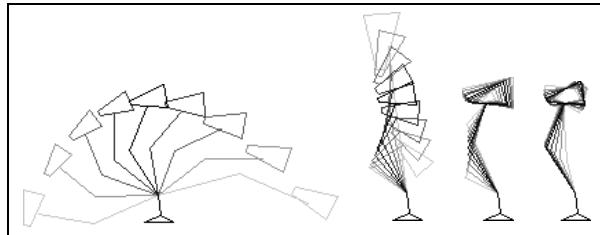
This process is computationally intensive, taking several minutes to run even for a relatively small model (60 points), but it is performed off-line.

## 5 Results

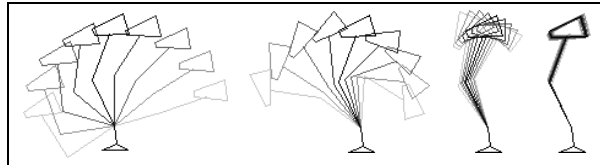
Thirty training images were captured of an anglepoise lamp (chosen for its multiple jointed pivots) in various positions, and the locations of 10 landmark points (as shown in Fig. 2) recorded for each one. A model was generated using each of four methods; the standard PDM, a quadratic PRPDM, a cubic PRPDM and the hybrid PDM. Figures 3, 4 and 5 show the first four modes of variation for three of these models. A range of  $\pm 2sd$  is shown, with the darkest central figure being the mean in each case.



**Fig. 3.** The first four modes of variation under the standard PDM



**Fig. 4.** The first four modes of variation under the quadratic PRPDM



**Fig. 5.** The first four modes of variation under the hybrid PDM

Table 1 shows how the models compare statistically. The values given are percentages of the the total standard deviation present in the model. In the case of the hybrid model, the *total* standard deviation is lower than for the other models. The difference between the two totals (58.49%) has been ‘explained’ by the reparameterization. The *cum. %* columns give the cumulative percentages.

**Table 1.** How variation is accounted for under the various models

| Feature            | Standard PDM |        | Quadratic PRPDM |        | Cubic PRPDM |        | Hybrid PDM |        |
|--------------------|--------------|--------|-----------------|--------|-------------|--------|------------|--------|
|                    | %            | cum. % | %               | cum. % | %           | com. % | %          | cum. % |
| reparameterization | 0            | 0      | 0               | 0      | 0           | 0      | 58.49      | 58.49  |
| mode 1             | 50.92        | 50.92  | 60.91           | 60.91  | 62.96       | 62.96  | 18.48      | 76.97  |
| mode 2             | 29.78        | 80.70  | 23.54           | 84.45  | 23.67       | 86.63  | 12.28      | 89.25  |
| mode 3             | 6.54         | 87.24  | 5.82            | 90.27  | 5.41        | 92.04  | 5.36       | 94.61  |
| mode 4             | 5.74         | 92.98  | 4.27            | 94.54  | 3.65        | 95.69  | 1.43       | 96.04  |
| mode 5             | 2.76         | 95.74  | 2.32            | 96.86  | 1.87        | 97.56  | 1.32       | 97.36  |
| mode 6             | 2.40         | 98.14  | 1.44            | 98.30  | 0.95        | 98.51  | 0.73       | 98.09  |
| mode 7             | 0.56         | 98.70  | 0.54            | 98.84  | 0.51        | 99.02  | 0.50       | 98.59  |

As can clearly be seen from Fig. 3, the standard PDM fails to capture the pivotal nature of the object; points move in straight lines for each variation mode, the lamp head changes size and the pivotal arms are stretched and compressed, so implausible shapes can be generated. The anglepoise has only three pivots, so any variation seen in the fourth mode must be acting to compensate for inaccuracies in earlier modes. Statistically, this model is the least compact, the cumulative deviation being the lowest at every stage.

The Quadratic PRPDM improves on the standard PDM; the major mode of variation captures *some* bending, but the circular arc is approximated by a parabola. Again, some stretching is seen, but not as much as in the standard PDM. There is still a fair amount of compensatory variation in the fourth mode. The Cubic PRPDM performs better statistically than the Quadratic version, and in general, performance increases with polynomial degree.

The Cartesian-Polar Hybrid PDM captures the pivotal nature of the object precisely. The modes of variation show no changes in lamp head size or stretching of arms. It is interesting to note, too, that the major mode captures the fact that the lamp head generally remains facing in the same direction as the lamp body is moved. The three angular movements do not appear entirely independently as three separate variation modes; this is due to the limited number of training examples used. Statistically, the mere reparameterization of the model explains over 50% of the standard deviation. The Hybrid model is most compact up until mode 5, by which time the remaining variation is due purely to noise.

**Table 2.** Machine cycles used to generate a model instance under each technique

| Model           | Cycles  | Time /ms |
|-----------------|---------|----------|
| Standard PDM    | 215621  | 8.6      |
| Quadratic PRPDM | 1326126 | 53.0     |
| Cubic PRPDM     | 1417455 | 56.7     |
| Hybrid PDM      | 221105  | 8.8      |

Table 2 shows how the techniques compare in terms of computational complexity. The values shown are numbers of machine cycles required to generate a model instance (this affects the speed of feature location/tracking). The tests were performed on a MIPS R4400 processor, with a MIPS R4010 FPU. A machine speed of 25MHz is assumed; statistics were captured using the code-

profiler, *pixie*. As expected, the standard PDM is the cheapest, but note that the hybrid model is not much more expensive. Also, and most importantly, the two PRPDMs are notably slower (about 6 times).

## 6 Discussion

In this paper we have shown how, in some cases, the use of a standard or polynomial regression PDM is non-optimal, and how the Cartesian-Polar Hybrid PDM can improve the compactness and accuracy of a model.

Whilst the hybrid model was designed with pivotal or bending deformation in mind, it is possible that it can more closely approximate other types of deformation too. More importantly, the concept of a hybrid model can be extended to incorporate other types of reparameterization, such as polynomial, elliptical or sinusoidal, simply by providing suitable mapping functions.

Research into the hybrid model is ongoing. An important issue concerns dealing with objects which rotate uniformly through a full 360°; is it ‘correct’ to have an arbitrary mean position with 180° variation on either side? Also, the automatic pivot location algorithm is heuristic, so we cannot be sure it is finding the best mapping. The system is also limited by the fact that all pivots *must* be marked with landmark points. We hope to develop a system which breaches this constraint, inventing its own pivot points and annotating them onto training examples as appropriate.

Initial experiments using the hybrid PDM for tracking have proved promising; the only problems being involved with finding suitable relative weights for linear and angular movement. Experiments were performed with hand-tracking, for which the standard PDM has already proved successful [6]; however, because the hybrid model requires fewer modes of variation, the system runs faster. A more challenging task is required to prove that the hybrid model can succeed where the standard PDM fails. For this reason, future work is to include the construction of a 3D model of the human hand.

## References

1. T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. BMVC*, pages 9–18, Leeds, UK, 1992. Springer-Verlag.
2. A. Hill, T.F. Cootes, and C.J. Taylor. A generic system for image interpretation using flexible templates. In *Proc. BMVC*, pages 276–285, Guildford, UK, 1993. BMVA Press.
3. A. Hill, A. Thornham, and C.J. Taylor. Model-based interpretation of 3D medical images. In *Proc. BMVC*, pages 339–348, Leeds, UK, 1992. Springer-Verlag.
4. A. Lanitis, C.J. Taylor, and T.F. Cootes. A generic system for classifying variable objects using flexible template matching. In *Proc. BMVC*, pages 329–338, Guildford, UK, 1993. BMVA Press.
5. A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *Proc. 3rd ECCV*, pages 299–308, Stockholm, Sweden, 1993. Springer-Verlag.
6. A.J. Heap. Real-time hand tracking and gesture recognition using Smart Snakes. In *Proc. Interface to Human and Virtual Worlds*, Montpellier, France, June 1995.
7. P.D. Sozou, T.F. Cootes, C.J. Taylor, and E.C. Di-Mauro. A non-linear generalisation of PDMs using polynomial regression. In *Proc. BMVC*, volume II, pages 397–406, York, UK, 1994. BMVA Press.