Scale Descriptors through Phase Unwrapping

Alex Hughes and Richard C. Wilson

Department of Computer Science
University of York, York YO10 5DD, UK
{alex.hughes,wilson}@cs.york.ac.uk

Abstract. Shape matching typically relies on the candidate shapes being normalised for scale before the matching takes place. Many methods rely on finding the boundary or normalising the area of the shape. This is problematic when the object contains holes, or two objects have different underlying shapes. The advantage of our method is that the scale can be normalised separately from the shape.

1 Introduction

The shape of an object is typically defined as the configuration of pixels within the object, minus the effects of scale, rotation or translation. Shape matching techniques therefore need a way to cancel out, correct, or ignore these effects. In the case of scale, the objects can be resized to some nominal scale. However, this relies on generating a scale value for the object under examination, not always an easy task.

One technique is to detect the boundary of a shape and then use that information to determine a scale factor for the whole object. In the simplest case of a solid object with a clear unbroken outline (i.e. a silhouette), this is quite straightforward. However, there are a number of complications that may arise. For instance, the boundary of the object may not be clearly defined - it may be blurred or otherwise hard to determine. The object may also have multiple potential boundaries, due either to holes in the object, or to the object itself being composed of a number of separate parts. Even a single clear boundary may cause problems depending on its configuration, for example, if it doubles back on itself.

As such, it would appear that one of the major difficulties facing such feature-based derivations of scale is the resolution of ambiguities, a task which must largely be performed in the design phase, and by which they are later constrained. Additionally, there is the issue that, if only the boundary is being used, then arguably much potentially valuable information is being discarded. This may for instance make boundary-only based methods vulnerable to noise.

Another approach is to compute the area of the shape and then use that as a scale factor. While this works successfully for objects of precisely the same shape, if there is some perturbation of the shape, for example an internal hole, the scale will change.

We seek here to develop a technique that avoids the limitations of feature based techniques by finding a linear variate of scale based on the process of phase unwrapping.

2 Fourier Descriptors

We begin our analysis by assuming that we have an image of an object, centred at the origin, but with unknown scale and orientation. In addition, there is a degree of uncertainty as to whether any given point is inside the shape or not. As a result we may not be able to find an accurate boundary. We therefore define a shape function as a probability density

$$p((r, \theta) \in S) \quad (1)$$

defined in the polar coordinate system (r, θ) . In order to digitise this shape, we form a polar pixel lattice defined by $p(i, \gamma)$.

2.1 Radial Shape Slices

A radial slice is then defined as the vector

$$\mathbf{v}(\gamma) = [p(0, \gamma), p(1, \gamma), \dots, p(n, \gamma)]^T \quad (2)$$

The vectors $\mathbf{v}$ may be regarded as feature vectors for the shape, and statistical analysis of this feature space can provide a good characterisation of the overall scale of the shape. However, the unknown scale and orientation parameters modify the feature space in a very non-linear manner.

A change in the orientation of the shape will result in a change in the order of the vectors. We can accommodate this simply by using a statistic which does not depend on the order of the samples.

The effects of scale however, present a more challenging problem. A change in the scale of the shape will result in a change of the indices of the vectors themselves, a highly non-linear process, making analysis of the set of all vectors extremely difficult. We therefore need some method of converting this change in indices into a change in the values at each index.

2.2 Fourier Descriptors of Radial Shape Slices

We commence by applying the Discrete Fourier Transform directly to our radial slices.

$$(x_0, x_1, \dots, x_n) = \text{DFT}(p(0, \gamma), p(1, \gamma), \dots, p(n, \gamma)) \quad (3)$$

The quantities $x_0, \dots, x_n$ are complex numbers that can be represented in terms of a phase and a magnitude component.

This Fourier shape description is completely general, in the sense that the inverse DFT will recover the original data perfectly, for any possible shape. However, realistic shapes will span only a small subset of the possible shape space. For instance, rapid changes in either radial or angular directions are unlikely.

In other words, the amount of information present in a radial slice is typically much less than the size of the feature space would suggest.

Fourier descriptors provide an elegant means of discarding this excess information, in that we can simply discard high-frequency components. In addition, it has been previously shown that much of the information present in a signal is contained in the phase component alone [7], allowing us to discard all the magnitude information, further halving the size of our representation.

As a result, the original radial slice vector can ultimately be replaced by the first few components of the Fourier phase.

$$\mathbf{v}_f(\gamma) = (\phi_0, \phi_1, \dots, \phi_k)^T \quad (4)$$

By using a reduced representation, we can both reduce the storage space required, and boost the efficiency with which the representation can be manipulated.

3 Scaling and Fourier Descriptors

One important property of the Fourier Transform is that translations in the original data correspond to changes in the phase values of the DFT. Thus, we have converted the effects of scaling from a change in the indices of the values to a change in the values themselves. The change in phase values can be described mathematically by use of the Fourier Shift Theorem:

$$f(t - t_0) = e^{j\omega t_0} F(\omega) \quad (5)$$

This demonstrates that as the values in the radial slices change index values (the ‘shift’ in the above equation), then the phase values will be incremented or decremented accordingly. Notice however, that this assumes that all index values in the radial slice will be shifted by a constant amount, something that will not be the case as the shape is resized. Scaling operations, by their very definition, will stretch or contract the waveform represented by the radial slice.

It is easy to obtain a fixed shift for a fixed change in scale by simply sampling the radial slices logarithmically. This is equivalent to sampling the original shape function exponentially. In other words, we re-define eq. 2 as

$$\mathbf{v}(\gamma) = [p(e^{\beta i} \gamma)]^T \quad (6)$$

The phase representation can then be constructed as before.

3.1 Phase Wrapping and Unwrapping

The problem with adding an offset to each of the phase values is that phase values are constrained to lie between 0 and 2π . Adding values that would take them beyond these limits merely causes them to wrap around. In other words, there are multiple solutions to the phase shift of a particular frequency component. For a measured phase shift of ϕ , the possible solutions are $\phi, \phi + 2\pi, \dots, \phi + n\pi$

(where n can also be negative). For this reason, it is not possible to directly infer the shift from the phase change.

The problem of determining the number of times the phase ‘wraps’ is known, unsurprisingly, as phase unwrapping. Phase unwrapping procedures are perhaps most commonly employed in decoding data supplied by Synthetic Aperture Radar (SAR), or other techniques that represent distances in terms of phases. If the total range of distance to be measured is greater than the wavelength used by the measuring device, then the data will be returned in a wrapped state. Under the assumption that the true data varies smoothly, fringes or jumps in the phase values can be interpreted as wrapping events. However, noise and discontinuities in distance confound this process. There are a number of algorithms for tackling the unwrapping of wrapped-phase-only data [3, 8, 1, 10, 9, 11, 6].

Fig. 1. Phase Unwrapping

In contrast to the problem faced in decoding SAR data however, we have the original data which gives rise to the phase information, making the unwrapping problem more tractable. There are a few techniques that make use of this extra data. Perhaps the most famous was published by Tribolet [12] and functions by re-integrating the derivative of the phase values, on the assumption that the general gradient of the curve formed by the phase values will not change significantly at wrapping points.

Later McGowan and Kuc published an ‘exact’ method [5], which was subsequently refined by Long [4]. These algorithms purport to calculate an exact solution numerically, but have the drawback that they can be very computationally demanding, both in processing time and storage space.

Most recently of all, Krajník has published a method [2] that claims to be both swift and accurate. It functions in an iterative manner, determining the unwrapped phase for each phase value from the previous phase value, working through the phases in order of their corresponding frequencies. Specifically, he defines the phase increment between adjacent frequencies to be:

$$\Delta\Phi_{n+1} = \text{ARG} \left(\frac{X_{n+1}}{X_n} \right) \quad (7)$$

In essence, this complex division results in a third complex number whose phase is the difference in phase between the first two values. For this reason, it is important that the (wrapped) phase does not change by more than π between adjacent phase values, something that can be achieved by padding the original data with zeros, thus interpolating the data in the frequency domain.

One issue that plagues all the aforementioned unwrapping techniques is the case of zeros on the complex plane (i.e.: where the magnitude is zero). At these points, it becomes impossible to tell whether a wrap has occurred or not. Such zeros are most likely to occur with synthetic data however, rather than real-world input. Zeros could potentially be eliminated by adding a small amount of noise to the signal.

4 Shape Scale Representation

4.1 Basic Method

The method we used was to generate 360 radial slices. These slices were sampled from the perimeter inwards (the last sample corresponded to the centre of the shape), making a total of 64 samples per slice. The test shapes were white (value=255) against a black background (value=0). Each of the slices were then extended from 64 to 4096 values by padding with zeros. This was to ensure that the phases of adjacent frequencies would not differ by more than π , as per Krajník's technique.

The Discrete Fourier Transform of each slice was calculated and Krajník's iterative method used to find the first 10 unwrapped phases (this proved to be a good balance between efficiency and accuracy) from the complex output of the DFT. All other phase and magnitude information was discarded.

Finally, all 360 sets of unwrapped phases were averaged together to yield a single set of unwrapped phases whose values represented the scale of the shape in question.

4.2 Measurement of Scale for a Single Object

Our first example consists of a simple image of a car, at 3 scales. The 3 images represent successive doublings in scale.

As can be seen from the graph, the doublings in scale are represented by a constant change in the gradient of the curve formed by the unwrapped values. This verifies that we can measure the scale of an object through this technique. Note that we can obtain the scale by using just a single frequency component, eg: the 10th component.

4.3 Shape Matching

We can also make use of this technique to align two shapes in rotation and scale. The technique is as follows:

Given two images to match, we will refer to one of them as the base image, and the other as the test image. We start by generating a set of 360 radial slices for the base image, generating phase information from these slices and unwrapping the phases, as described previously.

We then rotate the test image relative to the base image in a number of increments. At each step of the rotation, we generate the set of unwrapped phases for the test image.

Fig. 2. Unwrapped Phases for Car Images

Finally, we calculate the difference between the unwrapped phase values at a particular frequency (eg: the 10th frequency component) for the corresponding slices of the base and test images. These differences are used to increment an accumulator array, indexed by rotation angle and difference in unwrapped phase.

If the two images are rotationally aligned then we would expect to find the same difference between the unwrapped values for the slices of the test image and the corresponding slices of the base image, for every slice, because the scaling difference is the same for every corresponding pair of slices (this of course assumes isotropic scaling). Hence, in our case, we would expect to see 360 counts of the same difference value, with that difference value representing the overall difference in scale between the two objects. As the two images drift out of rotational alignment, corresponding slices will not represent the same parts of the objects, and instead a broad range of difference values will be obtained.

Ultimately then, we can expect to find a spike in the accumulator array corresponding to the relative rotation and scale factors. This is indeed the case in fig. 4, where we can see a clear spike in the accumulator data indicating that the third step of the test image rotation sequence is that which most closely matches the base image, confirming what we can see from a visual inspection of the data.

The peaks near the bottom of the grid are caused by single-pixel discrepancies due to the pixellated nature of the images, exaggerated by the exponential sampling.

An important point to note is that it is not necessary to actually rotate the test shape relative to the base shape. The same effect can be obtained merely by re-ordering the original radial slices (or, more accurately, the sets of unwrapped phases generated from them) of the initial test shape image. Thus it is only necessary to generate one set of unwrapped phase slices for the test image.

Fig. 3. Base Shape and Series of Rotated Test Shapes

Fig. 4. Accumulator Array

Also, it would be possible to perform the alignment iteratively, starting with 10 coarse radial slices, using these to align the shapes roughly, and then repeating the process at a finer resolution for only part of the images. This would mean that only a few sets of unwrapped data would need to be generated.

4.4 Partial Matches

It is also possible to use this technique to make partial matches. When the shapes shown in fig. 5 were processed, they produced the accumulator graph shown. The two spikes on this graph indicate that there were two possible matching scales. This could potentially be useful when trying to match partially occluded objects.

5 Conclusion

We have presented here a new technique for obtaining a scale factor for unknown shapes that avoids many of the limitations of feature-based scale-finders, while

Fig. 5. Shapes That Match at Two Possible Scales

also being more robust and precise than area-based methods. The technique can also be adapted to form the basis of an efficient shape-matching system.

References

1. Kazuyoshi Itoh. Analysis of the phase unwrapping algorithm. *Applied Optics*, 21(14):2470, 1982.
2. Eduard Krajinik. A simple and reliable phase unwrapping algorithm. In J. Vandewalle et al., editor, *Signal Processing VI: Theories and Applications*, pages 917–919. Elsevier, Amsterdam, 1992.
3. José M. N. Leitão and Mário A. T. Figueiredo. Absolute phase image reconstruction: A stochastic nonlinear filtering approach. *IEEE Trans. Image Processing*, 7(6):868–881, 1998.
4. David G. Long. Exact computation of the unwrapped phase of a finite-length time series. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 36(11):1787–1790, 1988.
5. Richard McGowan and Roman Kuc. A direct relation between a signal time series and its unwrapped phase. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-30(5):719–726, 1982.
6. Oleg Michailovich and Dan Adam. Phase unwrapping for 2-D blind deconvolution of ultra-sound images. Submitted to *IEEE Trans. Medical Imaging*, 2002.
7. A. V. Oppenheim and J. S. Lim. The importance of phase in signals. In *Proc. of the IEEE*, number 69, pages 529–541, 1981.
8. A. V. Oppenheim and R. W. Schaffer. *Digital Signal Processing*. Prentice Hall, 1975.
9. Mark D. Pritt and Jerome S. Shipman. Least-squares two-dimensional phase unwrapping using FFT's. *IEEE Trans Geoscience and Remote Sensing*, 32(3):706–708, 1994.
10. Kenneth Steiglitz and Bradley Dickinson. Phase unwrapping by factorization. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-30(6):984–991, 1982.
11. Torfinn Taxt and Galina V. Frolova. Noise robust one-dimensional blind deconvolution of medical ultrasound images. *IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control*, 46(2):291–299, 1999.
12. José M. Tribolet. A new phase unwrapping algorithm. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-25(2):170–177, 1977.