A Global Polynomial Approximation Algorithm for Image Rectification

Christopher O.Ward
Department of Mathematics and Computer Science
University of the West Indies (St. Augustine)
cward@uwi.tt

Abstract

In this paper we report on the development of a global function approximation algorithm for image rectification and registration. The approximation algorithm fits a linear combination of two-dimensional Chebyshev polynomials to a set of sample points obtained from a test image. The algorithm was successfully used to map pixel positions in unrectified images to object locations on a planar surface. The mapping was used to guide a robot arm to objects placed on the surface, thus demonstrating the practical utility of the algorithm.

Polynomial approximation of functions

In the one-dimensional case, the problem that we address here is that of finding a polynomial $P(x)$ that fits a set of m sample points (x_i,y_i); where i=1,2,…,m and the x_i are distinct; such that

$$\max_{i=1}^{m}\left|y_i - P(x_i)\right| \le \varepsilon, \text{ for } \varepsilon \ge 0.$$

That is, we are trying to identify a polynomial, $P(x)$, that fits a finite set of points to a degree of precision determined by ε.

The *Weierstrass approximation theorem* assures us that this is a solvable problem for $\varepsilon > 0$ (Pinkus 2005):

If f is any continuous function on the finite closed interval $[a,b]$, then for every $\varepsilon > 0$ there exists a polynomial $p_n(x)$ of degree n (where n depends on ε) such that $\max_{x\in[a,b]}\left|f(x) - p_n(x)\right| < \varepsilon$.

We do not require an exact fit to the sample points in our application, so the restriction, $\varepsilon > 0$, is not a problem.

Furthermore, our set of points is not generated by a known function that we are trying to approximate, so we will be looking instead for an algorithm that exhibits good interpolation properties. By this we mean that the algorithm should generate approximating polynomials that do not "curve" unnecessarily between sample points. The nature of the application leads us to expect such smooth interpolations to be an appropriate approximation for the unknown points.

The approximating algorithm

We address the approximation problem by finding a linear combination of the first n Chebyshev polynomials of the first kind that exhibits the desired properties of $P(x)$. Thus,

$$P(x) = \sum_{j=0}^{n-1} a_j T_j(x);$$

where $T_j(x)$ is defined as follows:

$T_0(x) = 1$
$T_1(x) = x$
$T_j(x) = 2x\,T_{j-1}(x) - T_{j-2}(x)$; for $j > 1$;

$x\in[-1,1]$; and a_j ($j = 0,1,\ldots,n-1$) is the set of coefficients that establish the fit. In what follows, we will assume for convenience that x falls within the interval [-1,1].

We use m sample points to reformulate the problem as one involving vectors in a Cartesian m-space. We break with tradition (Geddes and Mason, 1975), by using Cartesian spaces instead of function spaces. The problem is restated as follows (we will use italicized Greek letters to denote vectors):

Given the set of m-dimensional vectors, τ_j, where $j = 0,1,\ldots,n-1$, such that the i'th component of τ_j is equal to $T_j(x_i)$, and a vector ψ such that the i'th component of ψ is equal to y_i, for i=1,2,…,m, find values for a set of scalar quantities, a_j, such that

$$\max_{i=1}^{m}\left|y_i - P(x_i)\right| \le \varepsilon, \text{for } \varepsilon \ge 0;$$

where

$$P(x_i) = \sum_{j=0}^{n-1} a_j T_j(x_i).$$

We will refer to the vectors, τ_j, as *term vectors* and define a vector, ρ, as

$$\rho = \sum_{j=0}^{n-1} a_j \tau_j.$$

Figure 1 shows the *approximating algorithm*. This algorithm derives a vector ρ that is close to ψ. We refer to

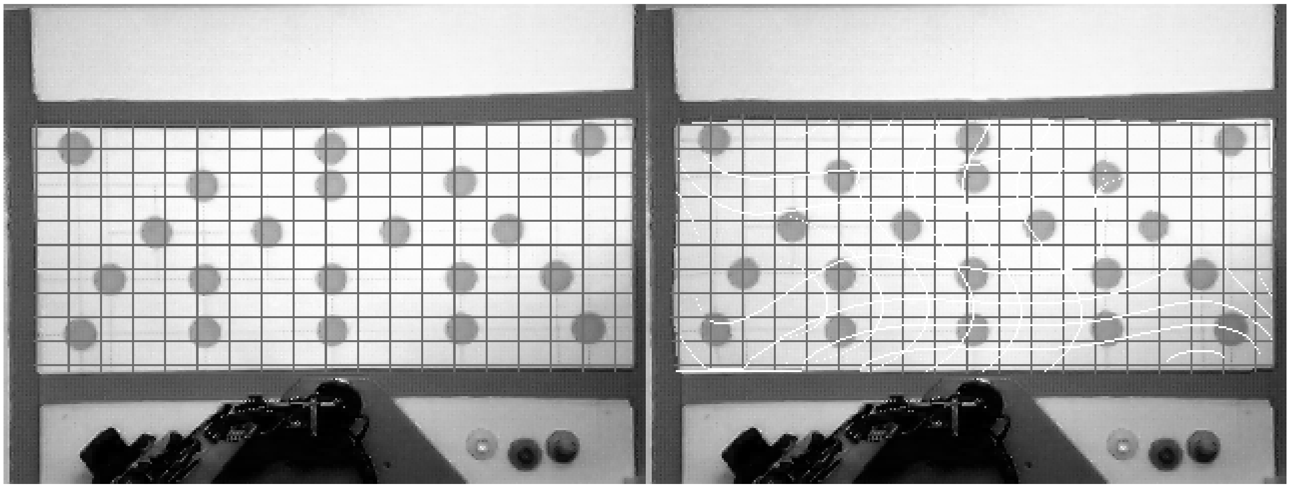

Figure 2: An unrectified image (left) and the corresponding rectified image (right). Only the region depicting the rectangular workspace was rectified and a grid was superimposed on this region to aid in comparison. The white streaks in the rectified image are “stretch marks”.

$\psi - \rho$ as the *error vector* and denote it by δ. The algorithm progressively reduces the Euclidean norm, $\|\delta\|$, towards zero. It does this by progressively subtracting from δ, projections of δ on term vectors. The subtractions are accumulated in the coefficients of the terms in the summation that defines ρ. The geometry of the subtractions is such that $\|\delta\|$ is progressively reduced.

1. Initialize all the coefficients, a_j (j = 0,1,…,n), to zero. The vector, ρ, is therefore initially the m-dimensional zero vector (where m is the number of sample points as stated above).
2. Set j to 0
3. Do
   - 2.1 Set `Converged` to 1.
   - 2.2 If $\max_{i=1}^{m} \left| y_i - P(x_i) \right| > \varepsilon$ then
     - 2.2.1 Set `Converged` to 0 .
     - 2.2.2 Set `inc` to $\dfrac{\tau_j \bullet \delta}{\tau_j \bullet \tau_j}$.
     - 2.2.3 If |`inc`| > α (where α is a small positive number.) then
     - 2.2.4 Add `inc` to a_j .
     - 2.2.5 For k = (j-1) downto 0 do
       - 2.2.5.1 Set `inc` to $\dfrac{\tau_k \bullet \delta}{\tau_k \bullet \tau_k}$.
       - 2.2.5.2 Add `inc` to a_k .
     - Endfor
   - Endif
   - 2.3 Increment j by 1.
   - Loop While `Converged` = 0.
4. Stop.

Figure 1: The approximating algorithm.

The approximating algorithm gives preference to using the term vectors associated with lower order Chebyshev terms, thus favouring simpler approximations. To produce this effect we subtract projections of δ on term vectors in a fixed order that effectively “fits” lower order Chebyshev terms first and maintains this fit as higher order terms are used to eliminate the remaining error.

Figure 2 shows the results of using a 2D version of the approximation algorithm to rectify an image from our robotic application. A generalised version of the Weiestrass approximation theorem applies in the case of multivariate function approximation (Pinkus 2005).

In practice, we used an upper limit on the number of terms to be considered that exceeded the number of terms required to achieve the desired degree of accuracy.

In addition to a pincushion distortion in the unrectified image (figure 1, left image), this image is also slightly rotated in a counterclockwise direction. In comparison, the rectified image shows a more rectangular border to the workspace and the rows of dots no longer tilt upwards.

References

Pinkus, A. 2005. *Density in Approximation Theory*, **1**. [Online article at] http://www.math.technion.ac.il/sat

Forsythe, G. E.; Malcolm, M. A.; and Moler, C.B. 1977. *Computer Methods for Mathematical Computations*. Prentice-Hall Series in Automatic Computation.

Brown, Lisa Gottesfeld 1992. *A Survey of Image Registration Techniques*. ACM Computing Surveys.