Beyond the Hough transform: further properties of the $R\theta$ mapping and their applications [1]

M. Wright A. Fitzgibbon P. J. Giblin [2] R. B. Fisher
Department of Artificial Intelligence
University of Edinburgh
Edinburgh, Scotland EH1 2QL
markwr, andrewfg or rbf@aifh.ed.ac.uk

Abstract

The Hough transform is a standard technique for finding features such as lines in images. Typically, points or edgels are mapped into a partitioned parameter or Hough space as individual votes where peaks denote the feature of interest. The standard mapping used for line detection is the $R\theta$ mapping and the key property the Hough transform exploits is that lines in the image map to points in Hough space. In this paper we introduce and explore three further properties of the $R\theta$ mapping and suggest applications for them. Firstly, we show that points in Hough space with maximal R for any value of θ are on the convex hull of the object in image space. It is shown that approximate hulls of 2D and 3D hulls of objects can be constructed in linear time using this approach. Secondly, it is shown that a simple relationship exists between the occluding contour of an object and the $R\theta$ mapping and that this could in principle be used to generate approximate aspect graphs of objects whose geometry was known. Thirdly, it is shown that antipodal points on object boundaries, (which are optimal robot grasp points), can be found by translation and reflection of the $R\theta$ representation. In addition we show the relationship between the $R\theta$ mapping used in the Hough transform and the classical mathematical theory of duals. We use this analysis to prove formally stated properties of the $R\theta$ mapping from image space to Hough space and in particular the relationship to the convex hull.

[1] This work was funded by EPSRC grant number GR/J44018
[2] Department of Pure Mathematics, University of Liverpool

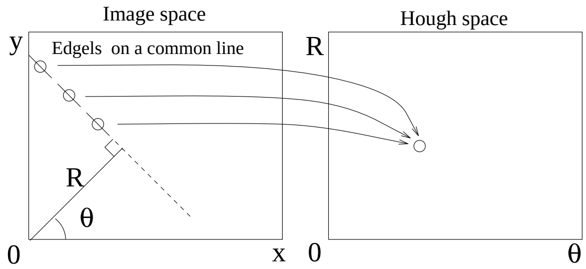


Figure 1: On the left is a set of edgels in image space with the same orientation and aligned such that they lie on the same line. On the right we can see that they map to a single point in Hough space and so the line they create can be detected as a peak at this point.

1 Introduction

The Hough transform [5] is a standard technique in computer vision having many applications for finding lines, circles and other features in images [6]. Typically edgels or other features are mapped into a partitioned parameter or Hough space as individual votes. The target image features are detected as peaks in the Hough space.

One particular variant of the Hough transform for line detection is shown in Figure 1. This uses the $R\theta$ parameterization and makes use of edgel orientation to reduce computation [7]. For each edgel a perpendicular is dropped from the origin in image space to a line passing through the edgel having the same orientation. This is then mapped to a space with two parameters R and θ where R is the length of perpendicular and θ is the angle the line makes with the x axis. As orientation information is available then each edgel maps to a single point in Hough space. If the edgels lie on a line then all will map to a common point in the Hough space and this peak in the Hough space or accumulator signifies a linear feature in image space. If no orientation information is available then each edge point maps to a sine wave in Hough space corresponding to the set of all lines passing through that point. The intersection of these sine waves signifies the parameters of the line that passes through all the points.

The specific property of the R, θ mapping which is used in the Hough transform is that lines in images space map to points in Hough space. In this paper we explore further properties of the R, θ mapping and suggest applications for them. Previously [14], we have shown the relationship of the R, θ mapping to the convex hull of objects and the occluding contour. In this paper we provide further examples of convex hulls computed using our approach, we show timings and code listings to demonstrate the simplicity of the algorithm. We show how a variant of the R, θ mapping can be used to detect optimal grasps for a robot gripper. We show the relationship of the R, θ mapping and the classical theory of duals. We also provide a proof of our assertion that maximal R for any θ in Hough space corresponds to the convex hull in image space.

2 The R, θ mapping of the shape boundary

In the Hough transform algorithm described above individual edgels are mapped into the Hough space and the peaks detected. Those edgels not contributing to peaks in the accumulator are ignored. We will now consider what happens to these other edgels and what this can tell us about the geometry of the curve. Consider the smooth image curve in Figure 2a. We can sample the boundary of this curve and at each point determine the orientation of the tangent to the curve. As we traverse the curve in image space a curve is also traced out in R, θ space. The mapping to Hough space for our smooth curve is given in Figure 2b. This curve is remarkably structured and we will now describe how its geometric features relate to the image space curve.

The main features of the curve are self intersections, cusps, maxima, minima and inflexions. The self intersections of the R, θ curve at C and F in Figure 2b are due to two points on the curve

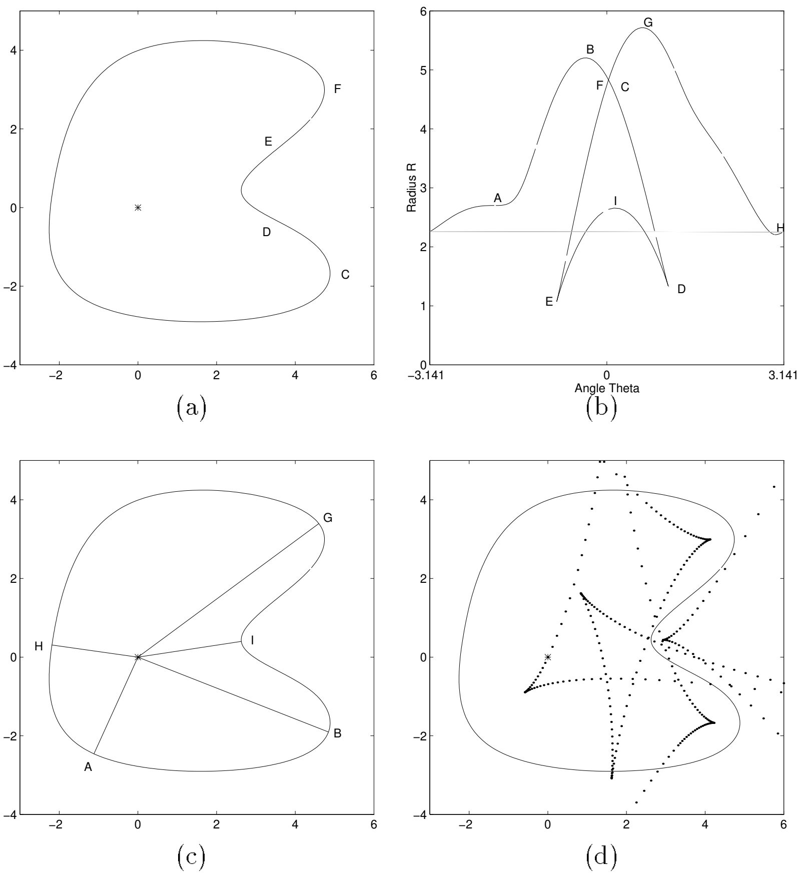
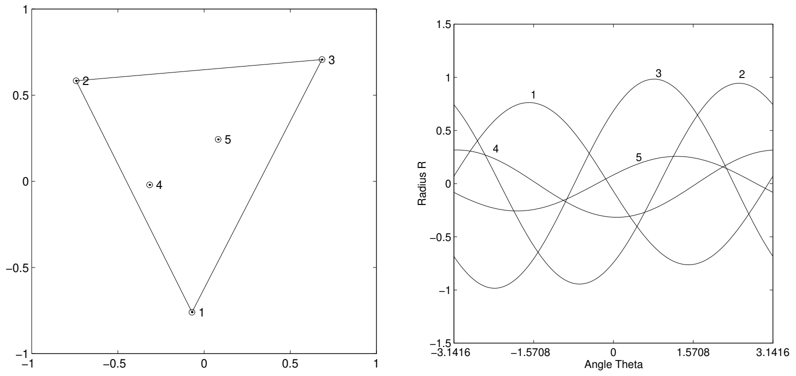


Figure 2a shows a smooth image curve. Figure 2b shows the mapping of the image curve to R, θ or Hough space. Features on the Hough space curve have been labeled to show corresponding features in image space depicted in Figures 2a and c (see text). Figure 2d shows the evolute of the image curve. The relationship between the evolute (dotted line) and origin (astrix) determine the type of turning points that exist on the Hough space curve (see text).

Figure 2: The mapping from image space to R, θ space

boundary at C and F in Figure 2a mapping to the same point in Hough space. This signifies where there is a bi-tangent line in the image *i.e.* a line tangent to the boundary at two distinct points. The loop formed by the self intersection corresponds in this case to the indentation in the curve boundary. The cusps within this loop at D and E in Figure 2b correspond to points of inflexion at D and E on the image space curve in Figure 2a. The maxima (B, G, I), minima (H) and inflexion (A) points on the Hough space curve in Figure 2a all correspond to points on the shape boundary where the normal at that point passes through the origin as shown in Figure 2c. It is a minima if the distance along the normal measured from the shape boundary to the origin is less than the radius of curvature of the curve at that point. It is a maxima if the normal length is greater than the radius of curvature and it is an inflexion if they are equal. The locus of centres of curvature defines a curve called the evolute which is depicted over the shape boundary as a dotted line in Figure 2d. The evolute consists of separate smooth sections running between cusp which correspond to curvature extrema on the image curve boundary. So the minima of the Hough space curve correspond to normals in image space which reach the origin before they cross their corresponding section of evolute. Conversely, maxima correspond to normals which do cross their



(a) (b)

Figure 3a shows a set of 5 image points and their convex hull. In Figure 3b we can see their corresponding sine waves in Hough space. Only those waves with maximal R for some value of θ contribute to the convex hull, in this case waves $1, 2$ and 3.

Figure 3: The relationship between the convex hull and the Hough transform for isolated points

section of evolute before reaching the origin. Finally, inflexions of the Hough space curve occur when the origin lies on the corresponding section of evolute. See §5 for proofs of the assertions made in this section.

3 The R, θ mapping and convex hull

The relationship between the R, θ mapping and the convex hull is as follows. **If we take for every value of θ that portion of the Hough space curve that has maximum R *i.e.* the upper envelope of the R, θ space graph, then the corresponding points in image space are those points on the convex hull.** (See §5 for proof).

3.1 Dealing with discontinuity

It would appear at first sight that this might only work for smooth curves because if the derivative of the boundary is discontinuous then there is a break in the Hough space curve. It is however a simple matter to join up these gaps. We remember that, for the original Hough transform, if there is no orientation information then points in image space map to sine waves in Hough space. At a corner we merely join the end of one edge to the beginning of the next with the appropriate sine wave. In fact this idea can be taken to its extreme if we wish to construct the convex hull of sets of isolated points. In Figure 3a we see a point set in image space. Each of these points maps to a sine wave in Hough space which are labeled in Figure 3b. The convex hull consists of all those points for which their corresponding Hough space sine wave is maximal in R for any particular value of θ. Furthermore their ordering around the hull is given simply by their adjacency in Hough space. Only sine waves $1, 2$ and 3 have a maximal R and are on the convex hull. Sine wave 5 never approaches a maximal position but wave 4 comes close at the intersection of wave 1 and 2. This is because point 4 in image space is nearly collinear with points 1 and 2. The sides of the convex hull correspond to the points in Hough space where the maximal sine waves intersect. In Figure 4 we see the convex hull of 7870 isolated points randomly distributed within a unit circle. Figure 5 shows examples of convex hulls computed for various objects using the Hough transform.

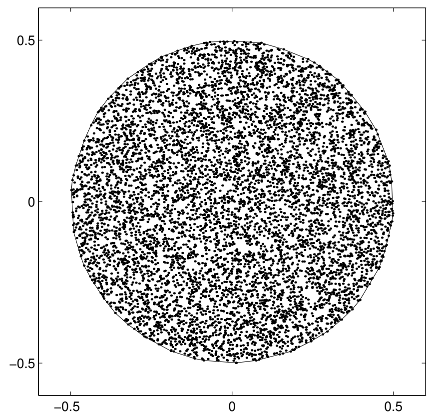


Figure 4: The convex hull of 7870 points generated using the Hough transform

3.2 Extending to 3D hulls

Extending the approach to 3D is straight forward. We now drop a perpendicular to a tangent plane rather than a tangent line. The parameters are the distance R of the plane from the origin and θ_1 and θ_2 which are the azimuth and elevation of the perpendicular. The Hough space now has three dimensions rather than two. (In the general case we drop the perpendicular to a hyperplane and the Hough space becomes n-dimensional.) Figure 6 shows the Hough space for a small three dimensional point set which is not shown. This is the upper envelope of the sinusoids generated by all points and each one corresponds to a point on the convex hull; compare this with Figure 3b.

Figure 7 is an example of a 3D convex hull generated using the R, θ mapping. The object on the left is Figure 7 is a Volkwagen Beetle body shell and the object on the right is its convex hull.

3.3 Algorithms

3.3.1 Algorithm 1: sampling

In Hough space the upper envelope of all sine waves defines the convex hull, in that each sine wave segment on this envelope corresponds to a single point in the hull. We now describe an algorithm to identify these segments. The array *hull* is workspace.

Step 1 Let $p_1(\theta), \ldots, p_n(\theta)$ be the sine waves in Hough space corresponding to the n points in the image space.

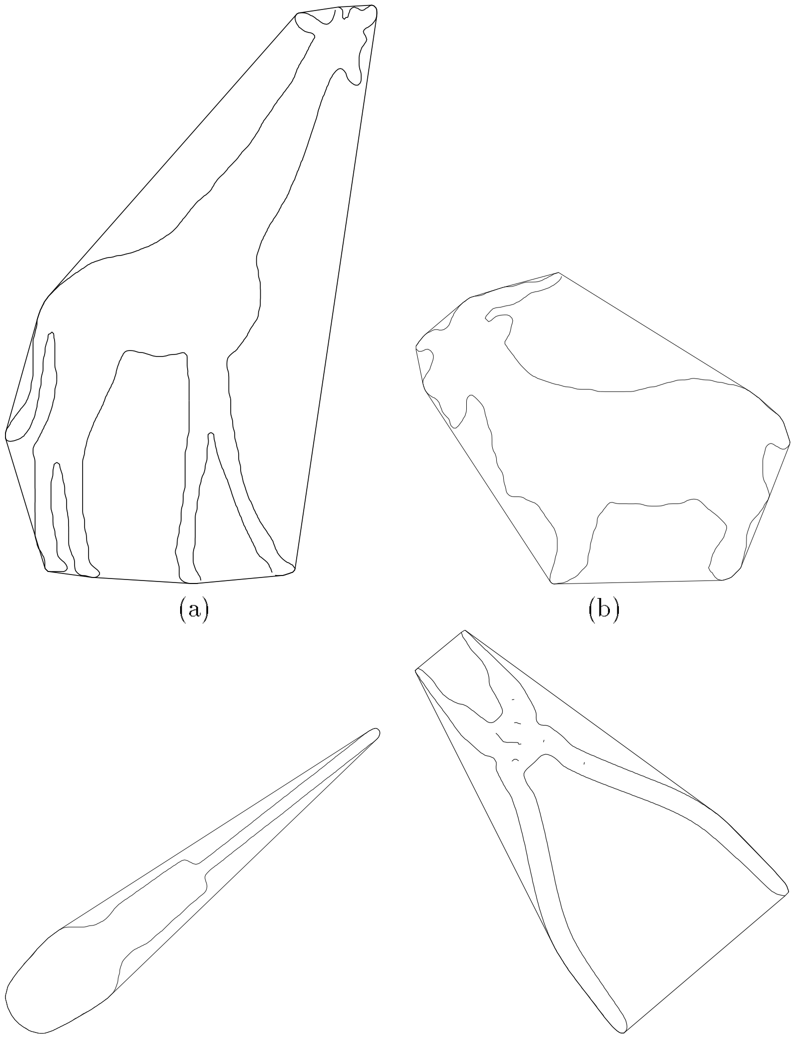
Step 2 Set index k to 1; for $\theta = 0$ to 2π in steps of Δ_θ (see discussion below) fill an array $hull[k]$ with indices as follows:

> **Step 2a** Find the greatest sine at this θ: Set $hull[k] := i$ where $p_i(\theta) \geq p_j(\theta)$ for all $1 \leq j \leq n$.
>
> **Step 2b** Increment index k by 1.

Step 3 The distinct indicies in *hull* are those points in the image plane in the convex hull.

(Termination is trivial since $\Delta_\theta > 0$.) An attractive aspect of the algorithm is its simplicity. Figure 8 shows the actual matlab code used to generate the 2D hulls, apart from these six lines, no other code is required.

An important issue is the granularity of the Hough space from the viewpoint of the complexity of computing the convex hull by this algorithm. Consider three collinear points a, b and c, with a



(a) (b) (c) (d)

Figure 5a shows the convex hull of a giraffe, Figure 5b is a goat, Figure 5c is a screwdriver and Figure 5d is a pair of pliers.

Figure 5: Examples of convex hulls computed using the Hough transform

and b in the convex hull, and c lying between them. Without loss of generality we assume that c lies at the origin of the image plane and a and b on either side of it along the x-axis, at $-x_a$ and x_b. We will consider the behaviour in Hough space as c is perturbed along the y-axis by δc, and derive from this the required sampling interval of the Hough space so as to detect that c is now part of the hull.

Perturbing c in this way contributes $\delta c.\sin\theta$ to Hough space, and we see that the interval Δ_θ must be such that

$$\Delta_\theta < \tan^{-1}(\delta c/x_a) + \tan^{-1}(\delta c/x_b)$$

for c to be seen to be in the hull. This gives an approximate complexity $O(\frac{n}{\tan^{-1}(1/p)})$ where p measures the greatest distance between any two points a and b. Choosing p *a priori* to be the image diagonal renders it independent from (a bounded) n, and overall complexity is $O(n)$; however, constants are much greater than in the Graham's scan algorithm [4] [10].

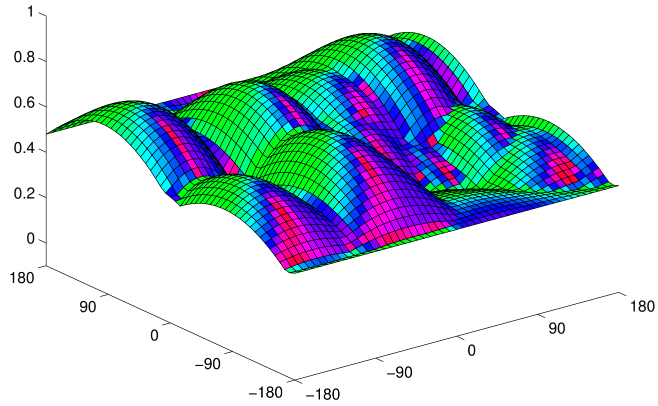


Figure 6: The Hough space for a small set of 3D points (not shown). The surface is of the upper sinusoids in Hough space corresponding to points on the convex hull.

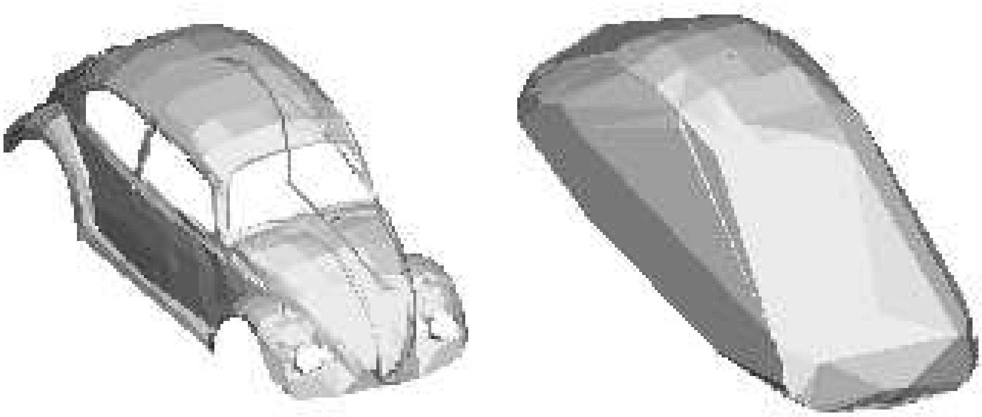


Figure 7: A Beetle body shell and its 3D convex hull computed using the R, θ mapping.

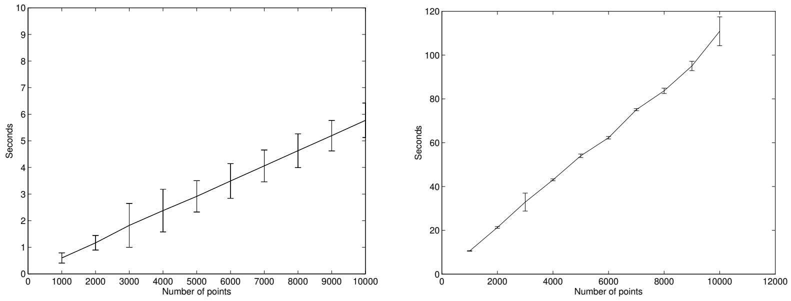
```
function hull_indices = chull2(P, max_error)
% 1. Calculate accumulator size from max_error
nsamples = floor(1/atan(max_error))

% 2. Generate unit circle (2D unit sphere)
theta = linspace(0,2*pi, nsamples)';
Normals = [cos(theta) sin(theta)];

% 3. Calculate hull
[Rmax, hull_indices] = max(P * Normals');

% 4. Remove duplicate indices
hull_indices = hull_indices(find(diff([hull_indices Inf])));
```

Figure 8: Demonstration of algorithm simplicity: Matlab code for 2D Hull



(a) (b)

Figure 9a shows timing trials for the 2D case and Figure 9b shows the 3D case. In both cases the solid line is mean for 100 runs and it can be seen that it is linear. Error bars indicate +/- the standard deviation of each run.

Figure 9: Timing trials for computation of the convex hull using the Hough transform

For example, for a 512×512 image we have $512.4\pi n$ for n points with a twice-over sampling interval.

3.3.2 Algorithm 2: non-sampling

We can make a significant (see below) average-case improvement in speed of the algorithm, with the following approach, which has complexity $O(n^2)$. (We adopt the same notation as above.)

Step 1 Determine i such that $p_i(0) > p_j(0)$ for all $1 \leq j \leq n$. We use this hull point to start.

Step 2 Record that the point i is on the hull. If $i = 2\pi$ then we have completed the hull, so stop.

Step 3 Compute the next point i' known to be on the hull as follows. Determine the smallest $\theta_{i'}$ such that $p_i(\theta_{i'}) = p_{i'}(\theta_{i'})$ subject to $\theta_{i'} > \theta_i$. i' is the 'nearest' sine wave which intersects with since wave i.

Set i to $'i$ and go to Step 2.

(We omit proofs of termination.)

This algorithm has quadratic complexity since there are n computations in Step 3 (finding the next intersection point), and we do this for each point in the hull, which is $O(n)$ (think of all n points arranged as as circle). Hence $O(n^2)$. However, we anticipate much better that this worst-case performance in practice, since typically considerably fewer points are in the hull in comparison with n.

Simple timing trials for algorithm 1 were carried out for both the 2D and 3D case. Algorithms were implemented in matlab [13] and run on a Sparc 10 workstation. The algorithms were tested on points randomly distributed within the unit square for 2D and unit cube for the 3D case. The number of input points was varied from 1000 to 10,000 in increments of 1000. For any fixed number of points the algorithms were run 100 times and then mean run times and standard deviations were calculated. Figure 9a shows the results for the 2D case and Figure 9b. In both cases the solid line indicates the mean run time and the error bars are +/- the standard deviation for each trial.

4 The R, θ mapping and aspect graph

In the last section we computed the convex hull of 2D and 3D objects by using the R, θ mapping of object boundaries with a stationary origin in image space. In this section we consider what happens to the same mapping as we move the origin in image space.

If the origin is outside the shape then we can consider it as a vantage point from which we look at the object. Using this perspective the first striking observation is that the occluding contour of the object, *i.e.* those point where the local tangent line or plane lies along the line of sight, is simply those points where $R = 0$. Furthermore those points with $R > 0$ have normals pointing toward the origin or vantage point and those with $R < 0$ have normals points away. We suggest that these observations can form the basis of an algorithm to build aspect graphs [9] of arbitrary 2D and 3D shapes. We can move the origin about the viewing sphere and keep track of the self intersections and cusps of the Hough space curve. As these features cross the θ axis *i.e.* $R = 0$ these constitute event boundaries in the aspect graph. To illustrate this principle we have devised a graphics program which animates in real time the mapping from image space to Hough space as the origin is moved or the shape boundary is deformed. Figure 10 shows a few different frames of the animation. As the vantage point (cross) is moved so the concavity comes into view the self intersection and cusps of the curve in Hough space move below the θ axis. We suggest that tracking of these events could be used to trigger the building of new branches and nodes in the aspect graph.

5 Detection of robot grasp points

In this section we show how a variant of the R, θ mapping can be used to detect optimal robot grasp points. Blake [1], has developed a theory of planar grasp where optimal grasps are defined using the criterion of least friction required to obtain force closure. Antipodal grasps, *i.e.* where the boundary normals precisely oppose one another, are identified as being particularly good as they requiring nominal friction for force closure. Normally, these grasp points would be detected using an n^2 search. We now show how these antipodal grasps can be detected using a variant of the R, θ mapping.

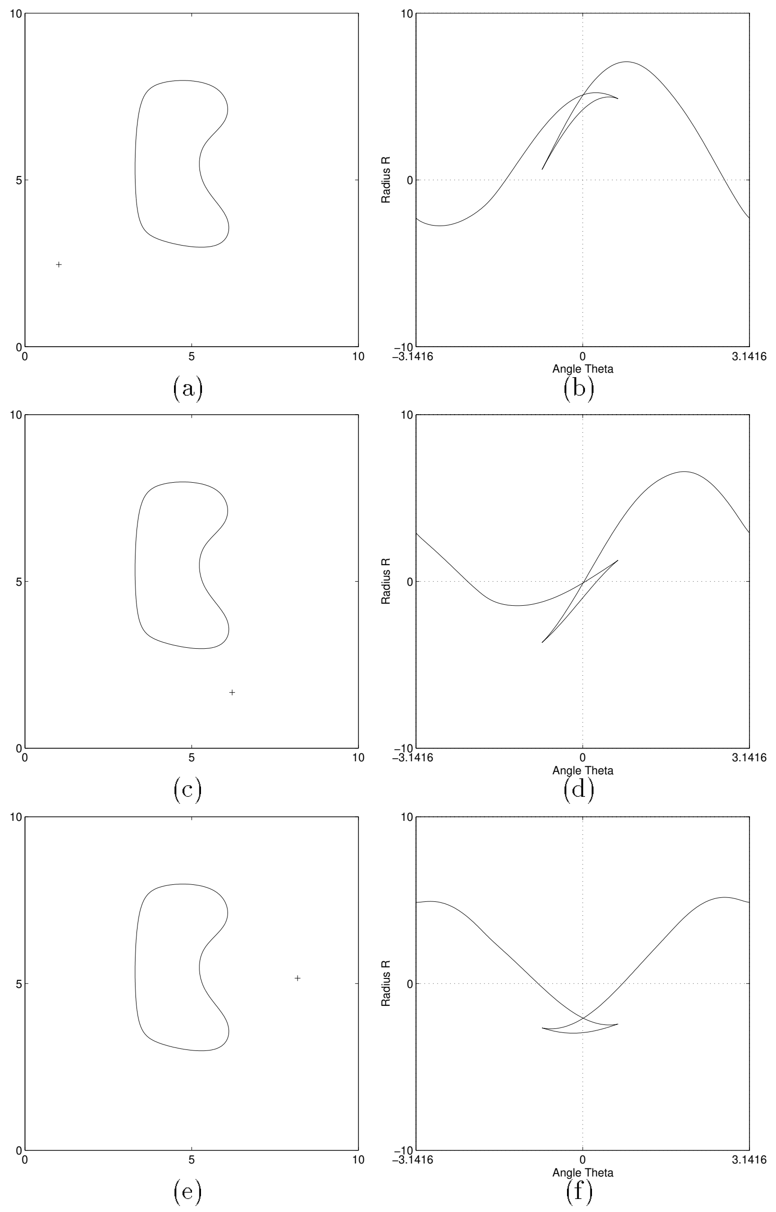
Consider the ellipse in figure 11. We compute an R, θ curve for the boundary with respect to the arbitrary origin at O. However, instead of the using the tangent to construct the curve we use the normal, *i.e.* we drop a perpendicular from the origin to the inward normal *not* to the tangent line.

The solid curve of figure 12 is the R, θ curve for the ellipse constructed using the normals instead of the tangents. Antipodal points can be found by a simple manipulation of the R, θ curve. Firstly, the curve is phase shifted with respect to itself by an angle of π; this can be visualised as a translation along the θ axis by π. Secondly, R values of the phase shifted curve are negated; this can be visualised as a reflection in the θ axis. The intersections thus resulting with the original R, θ curve correspond to the antipodal grasp points. The dotted curve in figure 12 is the result of applying the translation and reflection operations to the original solid curve. The intersection points A and C in figure 12 correspond to antipodal grasp points A and C on the ellipse in figure 11. Similarly, the intersection points B and D in figure 12 correspond to antipodal grasp points B and D on the ellipse in figure 11. Notice in figure 12 that antipodal pairs are separated by exactly π radians along the θ axis.

6 Duals and the Hough transform

In this section, we relate the Hough transform to the classical mathematical theory of duals, and prove the assertions made in §2 above.

The *dual* of a smooth curve in the plane is a model of its tangent lines (and, for a surface, a model of its tangent planes). There is an extensive theory of duals, going back to the work of Plücker [11, 12] in the early 19th century; some modern references are [2, 3]. Consider a plane curve C given parametrically by $\mathbf{r}(t)$, where t lies in some interval of real values, which could be all of the real numbers. The curve comes then with an *orientation*, given by t going from 'left to right', i.e. increasing. We usually take the orientation of a *closed* curve C without self-crossings to be such that the interior of the curve lies on the *left* as we traverse the curve.



Figures 10a,c and e show a kidney bean shape in image space where different view points are marked with a cross. In Figure 10a no part of the concavity can be seen and the loop in Hough space in Figure 10b is above the x axis. In Figure 10c the viewpoint is precisely along the bi-tangent line, this is signified by the self intersection in Figure 10d crossing the x axis. In Figure 10e the concavity is in plain view and the loop in Figure 10f is below the x axis.

Figure 10: The transitions in Hough space relating to changing view point.

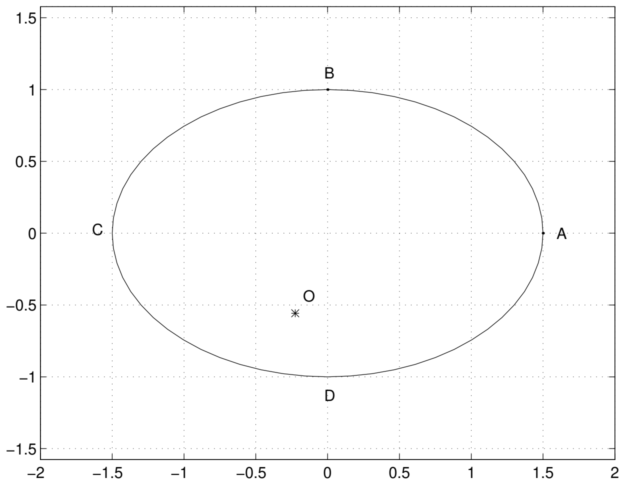
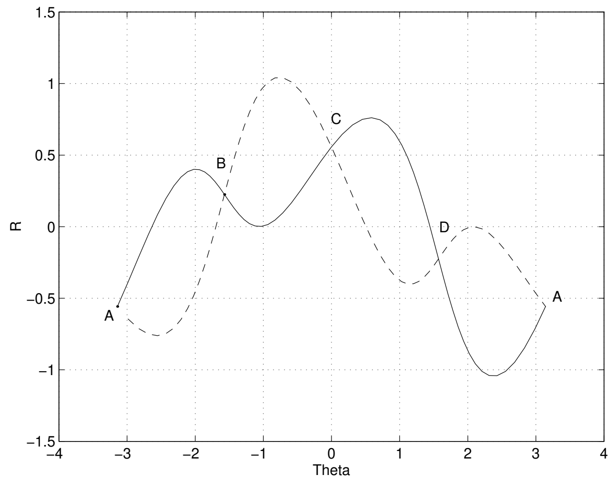


Figure 11: An ellipse with arbitrary origin O and antipodal grasp pairs A, C and B, D



The smooth curve is the R, θ mapping of the ellipse in figure 11 obtained using the normals rather than the tangent lines. The dotted line is obtained by translating the original curve by π and reflecting it in the θ axis. Intersections thus formed correspond to antipodal grasps.

Figure 12: Antipodal grasps identified using transformations of the R, θ mapping

Assuming that the velocity vector $\mathbf{r}'(t)$ is never zero, there is a well-defined (oriented) tangent line at each point: it passes through $\mathbf{r}(t)$ and has the direction of the velocity vector $\mathbf{r}'(t)$. The *oriented dual* of C is a model for the set of these tangent lines, and there are many ways of setting up such a model. If we forget the orientation and regard the tangent simply as a line in the plane, then a model for the set of these tangent lines is the *dual* of C. The oriented dual is easier to describe and we concentrate on that here. (The reason is that the set of all oriented lines in the plane is conveniently parameterized by the points of a cylinder, as we show below, but the unoriented lines are parameterized by the points of a Möbius band, which is a little harder to handle.)

Given an oriented line l in the plane (a line with an arrow on it) we can describe l completely by an angle θ and a real number R. Let $\mathbf{v}$ be a unit vector obtained as follows: start with a unit vector along the line l, i.e. pointing with the arrow, and turn this clockwise through $\frac{1}{2}\pi$. (This will later be along the outward normal to our curve C.) Let $\mathbf{x}_0$ be any point of l. Then R, θ are defined by

$$\mathbf{v} = (\cos\theta, \sin\theta), \qquad R = \mathbf{x}_0.\mathbf{v},$$

the dot being scalar product. If a different point $\mathbf{x}_1$ of l is selected then the scalar product is unaltered since $\mathbf{x}_1 - \mathbf{x}_0$ is perpendicular to $\mathbf{v}$. Geometrically, R is, up to sign, the distance from the origin to the line l and θ is the anticlockwise angle turned from the positive x-axis in the plane to the vector $\mathbf{v}$. Thus l determines R, θ; conversely R, θ determine a line with equation

$$\mathbf{x}.\mathbf{v} = R$$

and this line, which is perpendicular to $\mathbf{v}$, is oriented by turning $\mathbf{v}$ anticlockwise through $\frac{1}{2}\pi$. (This line will later be the oriented tangent line to C.) In this way, all oriented lines in the plane are parameterized by an angle and a real number, that is by the points of a cylinder, which we conventionally represent by a rectangle in R, θ space with $\theta = \pi$ identified with $\theta = -\pi$.

This 'R, θ representation' is one of many classical ways of describing lines in the plane, and is sometimes referred to as the 'Hesse normal form' (1861); for this form and some others, see [3, pp.123-127]. Returning now to our curve C with parameterization $\mathbf{r}(t)$, the oriented tangent is given by the velocity vector $\mathbf{r}'(t)$, and conventionally the oriented normal $\mathbf{N}(\mathrm{t})$ is given by turning $\mathbf{r}'(t)$ anticlockwise through $\frac{1}{2}\pi$. Thus in the above discussion we take $\mathbf{v} = -\mathbf{N}(t)$. If $\mathbf{r}(t) = (X(t), Y(t))$ then, making $\mathbf{N}$ a unit vector, we have (using $'$ for d/dt)

$$\mathbf{N}(t) = \frac{(-Y', X')}{\sqrt{X'^2 + Y'^2}},$$

and associated with the oriented tangent at $\mathbf{r}(t)$ we have the R, θ values given by

$$\mathbf{N}(t) = -(\cos\theta, \sin\theta), \qquad R = -\mathbf{r}(t).\mathbf{N}(t). \tag{1}$$

As t varies, the point R, θ describes a parameterization of the oriented dual of C. It is a model for the set of oriented tangent lines to C, and is the Hough transform as used in the text of this paper. As it is a model for the oriented tangent lines, it is evident that self-crossings of the R, θ curve correspond to oriented lines tangent to C twice, that is, to *bitangent lines* to C.

We can use the above parameterization to deduce geometrical properties of the Hough transform (or oriented dual). Here is a brief indication of how this is done. The curve parameterized $\theta(t), R(t)$ can be considered as a 'graph', that is as R a function of θ, provided the tangent line is not 'vertical', that is provided $\theta' \neq 0$. We assume that t, the parameter on C, is arclength. This does not affect the results but shortens the calculations, since the Serret formulae then take the following simple form (see [8, p.170], where $\tau = 0$ since our curves are planar)

$$\mathbf{r}' = \mathbf{T}, \quad \mathbf{T}' = \kappa\mathbf{N}, \quad \mathbf{N}' = -\kappa\mathbf{T}. \tag{2}$$

Here $\mathbf{T}$ is the unit tangent and κ is curvature. It follows from (1) and (2) that

$$\begin{aligned} \mathbf{T} &= (-\sin\theta, \cos\theta), \\ -\kappa\mathbf{T} &= -\theta'(-\sin\theta, \cos\theta) = -\theta'\mathbf{T}, \quad \text{so} \quad \theta' = \kappa \qquad (3) \\ R' &= -\mathbf{T}.\mathbf{N} + \mathbf{r}.\kappa\mathbf{T} = \kappa\mathbf{r}.\mathbf{T} \qquad (4) \end{aligned}$$

The following are now easy deductions.

1. $\theta' = 0$ if and only if $\kappa = 0$, that is the curve C has an *inflexion*. When $\kappa = 0$ we also have $R' = 0$, which means that the R, θ curve is actually *singular* (generally has a cusp). Thus inflexions of C correspond to singular points (cusps) of the R, θ curve, and, between cusps, the R, θ curve is a graph, that is R is a function of θ. For a given θ, there may be several values of R (as in the example of Figure 2b in §2). On a section between two cusps there are no vertical tangents, so locally R is a single valued function of θ; we denote the slope of this curve by $dR/d\theta$.

2. Suppose $\kappa \neq 0$. We have $dR/d\theta = 0$ if and only if $R' = 0$, which from (4) means $\mathbf{r}.\mathbf{T} = 0$, that is to say the normal to C passes through the origin (the equation of the normal is $(\mathbf{x} - \mathbf{r}(t)).\mathbf{T}(t) = 0$). So the R, θ curve has a maximum, minimum or inflexion if and only if the normal to C passes through the origin.

3. From (4) we can find the second derivative:

$$R'' = \kappa' \mathbf{r}.\mathbf{T} + \kappa \mathbf{T}.\mathbf{T} + \kappa \mathbf{r}.\kappa \mathbf{N} = \kappa^2 \left(\frac{1}{\kappa} - R \right)$$

   when $R' = 0$, that is when $\mathbf{r}.\mathbf{T} = 0$. From this it follows that, when $\mathbf{r}.\mathbf{T} = 0$, R has, as a function of t or of θ,

   - a maximum if $R > \rho$, where $\rho = 1/\kappa$ is the radius of curvature of C,
   - a minimum if $R < \rho$,
   - generally an inflexion with horizontal tangent if $R = \rho$. (The condition for a non-horizontal inflexion of the R, θ curve, namely $R'' = 0$ without $R' = 0$, does not appear to have a simple geometrical interpretation.)

   (*Remark* The process of taking duals is in a sense 'involutory', that is the dual of the dual of C is the curve C again. In that case, as inflexions of C give cusps on the dual, it might be expected that inflexions on the dual could only come from cusps on C. But whether or not a plane curve, such as the dual, has inflexions, depends on the choice of coordinate system used to describe it. In another standard representation of the dual of a nonsingular curve C (not the Hough transform, see [3, p.126]) there are no inflexions. Cusps, on the other hand, are present irrespective of the coordinate system, so in any representation of the dual, inflexions of C (using the ordinary coordinate system in the plane) give cusps on the dual.)

4. Consider a particular value of θ, corresponding to a unit vector $\mathbf{v} = (\cos\theta, \sin\theta)$. Turning $\mathbf{v}$ anticlockwise through $\frac{1}{2}\pi$ to give $\mathbf{T}$, the curve C may have several oriented tangents parallel to $\mathbf{T}$. Using the convention on orienting C mentioned above, the interior of C is, near the point of contact, always to the left of such an oriented tangent. Going in the direction of $\mathbf{v}$, the *last* such oriented tangent encountered must have the whole curve to its left, that is it is a 'support line' of C. Wherever the origin O is situated, this tangent will have the *largest* value of R for that θ (this R is > 0 if the oriented tangent is 'anticlockwise about O' and < 0 otherwise). So the support lines of C, for each θ, correspond to the maximum value of R for that θ. Since the support lines are tangents to the convex hull, this verifies the connexion with convex hulls in §3.

Finally, a few words about how these ideas extend to 3 dimensions. For a surface M in 3-space, it is possible to parameterize the oriented tangent planes by pairs $(\mathbf{v}, R)$ where $\mathbf{v}$ is a unit vector in 3-space, hence a point of the unit sphere S^2, and R is a real number. In fact $\mathbf{v}$ can be chosen as the oriented (outward) normal of M and $R = \mathbf{r}.\mathbf{v}$ where $\mathbf{r}$ is the point of contact of the tangent plane. Unlike the circle, there is no overall parameterization of the sphere S^2; for example azimuth and elevation break down at the poles, where azimuth is undefined. Away from the poles, we can locally use a 'Hough space' given by azimuth, elevation and R, or we can do the job globally by using the space $S^2 \times \mathcal{R}$ where $\mathcal{R}$ is the real numbers. The oriented dual of M is a locus in this space, and it has many properties analogous to those for a curve above. For example a parabolic point of M (Gauss curvature zero) corresponds to a singular point (generally an 'edge of cusps') on the dual. See [2].

7 Discussion

The convex hull algorithms presented above has similarities to Graham's algorithm [4] where points are sorted according to angle. The complexity of Graham's algorithm is $O(n \ln(n))$. However our

Hough based approach (algorithm 1) has a high constant term due to the sampling of Hough space. Algorithm 2 has much poorer worse case complexity but in the average case we anticipate better performance than algorithm 1. Neither algorithm can compete with Graham's algorithm in performance as a practical convex hull algorithm. At first sight it might appear that algorithm 1 breaks the lower bound for convex hull computation but we must remember that we are computing an *approximate* convex hull in *discrete* space. It is interesting to note that an equivalence has been proved between convex hull computation and the operation of sorting and therefor and interesting comparison can be made between convex hull algorithms and sorting algorithms. There are *linear* time algorithms for sorting and perhaps one of the simplest is bucket sort. Both bucket sort and the R, θ approach to convex hull computation work in linear time, both discretize space and both work approximately for arbitrary input, giving exact correct output for restricted input. (Bucket sort requires that the minimum distance between numbers is greater than the bucket interval, and the convex hull algorithm presented here requires that the angle between facets on the hull is greater than sampling interval in Hough space.

We believe that sampling may also be the basis for producing approximate convex hulls (and aspect graphs) in a principled way which has been identified as an important problem. Unlike Graham's algorithm there is a global representation of the hull. We could in principle use this to assess the effect of leaving out a particular point. This could be on the principle of the area of the sine wave for which it is maximal in R.

The Hough approach to building the aspect graph has the problem that although we can tell if a face points towards the viewer we cannot tell if it is occluded by another part of the object. It is as if we are considering the singularities of a transparent object. We could solve this by filtering the Hough space output with some kind of visibility check. One such check would be to compute a different transform, say d, α where d is the length of a line from the boundary point to the origin and α is the angle this line makes with the x axis. The points in d, α space with minimal d for a particular α are those points which are unoccluded.

8 Conclusions

The Hough transform uses a particular property of the R, θ, *i.e* that lines in image space map to points in Hough space, to detect linear features in images. In this paper we have presented and explored three further properties of the R, θ mapping and suggest useful applications for them. We have shown the relationship of the mapping to the convex hull and computed 2D and 3D hulls of objects. We have seen the relationship of the mapping to the occluding contour and pointed to the possiblity of constructing aspect graphs of objects with known geometry. We have also shown how translations and reflections of the mapping can be used to detect optimal robot grasp points. We have shown the relationship of the Hough transform and R, θ mapping to the classical mathematical theory of duals. We hope that we have challenged the traditional view of the the R, θ mapping and Hough transform as purely for feature detection. The R, θ mapping has many interesting properties and uses and many more may yet be waiting to be discovered.

9 Acknowledgement

Mark Wright would like to thank Ian Green for helpful discussions concerning algorithm complexity. Part of this work has been submitted for review to the journal Image and Vision Computing.

References

[1] A. Blake. A theory of planar grasp. OUEL Report 1958/92, Oxford University Engineering Department, October 1992.

[2] J. W. Bruce. The duals of generic hypersurfaces. *Math. Scand*, 49:36–69, 1981.

[3] J.W Bruce and P.J. Giblin. *Curves and singularities.* Cambridge University Press, second edition, 1992.

[4] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information processing letters*, 1:132–133, 1972.

[5] P. V. C. Hough. *Method and means for recognising complex patterns.* US Patent 3069654, 1962.

[6] J. Illingworth and J. Kittler. A survey of the Hough transform. *CVGIP*, 44:87–116, 1988.

[7] C. Kimme, D. Ballard, and J. Sklansky. Finding circles by an array of acculators. *Comm ACM*, 18:120–122, 1975.

[8] J. J. Koenderink. *Solid shape.* M.I.T Press, 1990.

[9] J. J. Koenderink and A. J. van Doorn. The internal represention of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[10] J. O'Rourke. *Computational geometry in C.* Cambridge University press, 1992.

[11] J. Plucker. *System der analytischen geometrie.* Berlin, 1835.

[12] J. Plucker. *Theorie der algebraischen kurven.* Bonn, 1839.

[13] Math Works. *MATLAB user guide*, 1992.

[14] M. W. Wright, A. Fitzgibbon, P. J. Giblin, and R. B. Fisher. Convex hulls, occluding contours, aspect graphs and the Hough transform. In *British Machine Vision Conference*, pages 493–502, 1995.