A FAST APPROACH FOR PERCEPTUALLY-BASED FITTING STROKES INTO ELLIPTICAL ARCS

Pedro Company
Institute of New Imaging Technology, Universitat Jaume I
Av. Sos Baynat s/n, 12071-Castellon (Spain)
Tel.: +34-964 728 119
Fax: +34-964 728 106
E-mail: pcompany@uji.es

Raquel Plumed
Dept. of Mechanical Engineering and Construction, Universitat Jaume I

Peter A.C. Varley
Institute of New Imaging Technology, Universitat Jaume I

ABSTRACT

Fitting elliptical arcs to strokes of an input sketch is discussed. We describe an approach which automatically combines existing algorithms to get a balance of speed and precision. For measuring precision, we introduce fast metrics which are based on perceptual criteria and are tolerant of sketching imperfections. We return a likelihood estimate based on these metrics rather than deterministic yes/no result, in order that the approach can be used in higher-level collaborative-decision recognition flows.

KEYWORDS

Computer-Aided Sketching; Sketch strokes; fitting primitives to strokes; elliptical arcs; perceptual fit; fast fit.

1. INTRODUCTION

For sketch-based interfaces, fitting lines to strokes must be an efficient and reliable process. Dealing separately with overtraced [1,2], and grouped strokes [3] does not work: excessive grouping of overtraces strokes, or fitting lines before segmenting, results in macrolines (several true lines or curves are grouped together), but segmenting strokes each time a candidate corner is detected results in microlines, as the segmenter wrongly interprets undulations and oscillations as true corners. Instead, these processes should run in parallel, passing information to one another until they converge to a solution. In such an approach, ellipse-fitting will be invoked each time the segmentation process identifies a candidate ellipse, so fast ellipse-fitting is a necessity.

Most current fitting approaches are mainly concerned on geometrical accuracy or even ultra-accuracy, while recognition strategies in Sketch-Based Modelling (SBM) require equal consideration to perceptual criteria. “What you perceive is what you get” is the goal (WYPIWYG concept by [4]), instead of “what you measure is what you get”. In this sense, our new approach differs from current approaches in the field of image recognition, as it prioritises speed and perception over geometrical accuracy.

In this paper, we discuss fitting elliptical arcs. The input is a stroke, a time-ordered sequence of 2D points, generated by following hand movements as the user sketches the ellipse (some systems also time-stamp individual points). The output is an approximate rather than a precise ellipse fit, and to allow for multiple interpretations it also includes a merit figure which measures how likely the stroke is to represent an ellipse. The methods should be as fast as possible, both to allow for immediate feedback and because higher-level interpretation processes may use them repeatedly.

Although speed is more important than exact fitting, we nevertheless require an acceptable approximation. And the acceptability of the approximation depends on how the stroke is perceived by humans, not on preconceived geometrical measures. Thus, there are two stages to our approach: we first find a reasonably good fit, and then measure how well the original stroke fits into a set of three tolerance bands around the resulting elliptical arc.

Section 2 reviews the existing fitting algorithms and analyses their weaknesses and strengths. Section 3 briefly describes our experimental approach to determine how humans perceive shapes embedded in sketched strokes. Section 4 describes how we combine different algorithms for fitting elliptical arcs to strokes to find the best balance between speed and accuracy, and how we measure the likelihood of the fit. In Section 5, we test our approach to determine its “goodness”, measured as its ability to produce the same interpretation as humans. Section 6 presents our conclusions.

2. ASSESSMENT OF PREVIOUS WORK

The most common approaches for fitting an ellipse with unknown parameters to a set of stroke points are based on minimising the total error, which may be measured in different ways. An excellent discussion of the state of the art in ellipse fitting can be found in [5]. A distinction is made between Maximum Likelihood, geometric and algebraic approaches [6]. We do not consider Maximum Likelihood as a stand-alone approach, because “when the noise is large or the eccentricity of the ellipse is large, the algorithm breaks down” [6].

Since only five points are required to determine an ellipse, we can take advantage of the sequential ordering of points in a stroke by selecting a regularly-spaced five-tuple [7]. Although Rosin [7] described a technique for ellipse fitting based on accumulating many five-point ellipse fits, we note that each such five-point ellipse fits (5P) is itself a fast fit.

The Direct Ellipse Fit (DIR) of Fitzgibbon et al [8] is currently the most popular algebraic algorithm for fitting ellipses to scattered data. Halir and Flusser [9] describe a robust version of DIR, and Kotagiri [10] provides a C# implementation. DIR was first applied to sketch recognition by Shpitalni and Lipson [11]. DIR is computationally expensive, as solving the eigenvalues and eigenvectors problem is simple for symmetric matrices but becomes more complex for non-symmetric matrices. For 3x3 matrices, there is an exact solution, but its implementation is unstable. The other choice is to use a numerical approach: there are various libraries which include calculation of eigenvalues and eigenvectors [12-14].

One representative geometric approach is the Guaranteed Ellipse Fitting algorithm (GEF) by Szpak et al. [15]. It is supposed to improve on DIR when data points are sampled from only one portion of an ellipse, and it is claimed to give a fast and accurate approximation of the computationally more expensive orthogonal-distance-based ellipse fitting method.

We have found experimentally that 5P fails: (1) for arcs close to 360°, unless one of the endpoints is replaced by an extra intermediate midpoint; (2) for short (encompass a small angle, less than 180°) and flat arcs and for strokes with a change in the curvature, as they produce elongated fits, and (3) for flat and undulating strokes, as they do not return a valid ellipse (Figure 1).

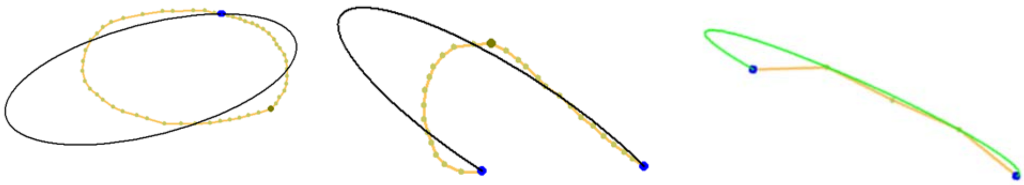


Figure 1: Elongated 5P fits.

Our experiments confirm that DIR fails for short and flat arcs, as it has a bias towards smaller ellipses when points are sampled from only a portion of the ellipse the estimated ellipse is often smaller than it should be [16]. It also fails occasionally through numerical instability.

GEF does not fail (provided that 5P or DIR feeds it with a valid seed). However, GEF execution times are unacceptable for an on-line application

To test the five-point algorithm, we selected the two *endpoints* plus three equally spaced *intermediate* points as input for the Davis implementation [17]. We also tested the Kotagiri implementation of the Direct Ellipse Fit (DIR) [8], [10], and our own C++ implementation of Guaranteed Ellipse Fitting with Sampson Distance (GEF) [16].

We compared the times taken by four algorithms: 5P, DIR, GEF seeded by 5P, and GEF seeded by DIR. Since these algorithms behave differently for arcs encompassing small angles and large angles, we treat these cases separately. Figure 2 plots running time against number of points for all of the examples of Section 5. Additionally, we resampled some examples to widen the range of number of points.

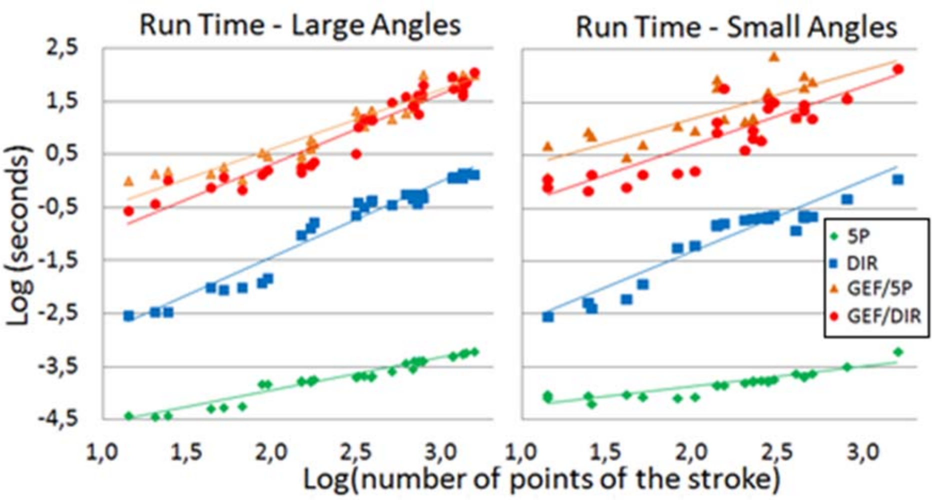


Figure 2. Execution times when strokes encompass large angles (left) and small angles (right).

It is evident that 5P is fastest—roughly 1000 times faster than DIR and 100000 times faster than GEF—and that this advantage increases somewhat with increasing the number of points. 5P should always be tried first if speed is a criterion.

DIR execution times might be acceptable for strokes of up to a few hundred points, and DIR combined with resampling to fewer than 100 points is a credible fast-fit alternative.

Both GEF/5P and GEF/DIR execution times are unacceptable for an interactive application, even when combined with resampling to 20 points. GEF/5P is not worth considering, as it is almost always slower than GEF/DIR.

3. PERCEPTION OF ELLIPSE FITTING

Design engineers are accustomed to seeing ellipses in engineering sketches. But what criteria do they take into account when assessing whether and how to fit an ellipse to a curve? For example, where does perception leave off and inductive reasoning begin? We wish to investigate psychological behaviour, without accidentally misinterpreting learnt behaviour of particular subjects as general perceptual behaviour.

To determine the limits of human perception, we must ask humans what they perceive, and the only scientific strategy which has proved at all useful in determining what humans perceive is performing experiments with groups of humans who are then interviewed to make their perceptions explicit (e.g. [18]).

In Experiment #1, subjects compared a set of strokes (see Figure 6) against given arcs. In Experiment #2 they compared strokes against their own mind's-eye arcs. The conclusion from these two experiments support our first hypothesis: humans perceive that strokes depict *good*, *average* or *poor* arcs, regardless of whether or not they are given a pattern to compare the stroke with.

Experiments #3 and #4 showed that subjects are stricter with large strokes and less confident with short strokes, as shorter strokes convey less perceptual information. Hence, relaxing evaluation criteria for short strokes mimics human perception.

More details on the studies can be found at [19].

3.1 Oscillations and undulations

During the experiments, we observed that some interviewed subjects evaluated oscillating strokes as *poor*, while other group of subjects seemed to mentally filter out oscillations and evaluate the underlying stroke. In other words, some subjects distinguish oscillations from undulations (respectively higher or lower frequency waves around the theoretical line). This interpretation was reinforced by queries (which we did not answer) from some subjects asking the interviewers whether or not they should ignore the lack of "smoothness", "flatness" or "horizontality" of some lines.

We hypothesise that subjects distinguish curvature changes from high frequency oscillations and low frequency undulation, and that oscillations are less distracting (as they are usually perceived as unintentional) than undulations.

4. FAST AND PERCEPTUALLY-BASED ALGORITHM FOR FITTING ELLIPTICAL ARCS

In fitting elliptical arcs to strokes, we have two concerns. Firstly, we must find a balance between speed and accuracy: although speed is more important than exact fitting, we nevertheless require an acceptable approximation. Secondly, the acceptability of the approximation depends on how the stroke is perceived by humans, not on preconceived geometrical measures. Thus, there are two stages to our approach: we first find a reasonably good fit (possibly to a reduced point set to reduce calculation time), and then measure how well the original stroke fits into a set of three tolerance bands around the resulting elliptical arc (Section 4.7).

We suggest two complementary strategies to reduce the point set. Firstly, we can find the complex hull (as described in Section 4.3) since this is what humans often do when oscillations or undulations mask the underlying shape. Secondly, we resample the point set (as described in Section 4.4) at equal intervals to obtain a reduced but representative subset, thus speeding up DIR and GEF.

5P applied on the convex hull stroke is calculated first (5P/CH). However, it does not always work. A certain subset of short strokes causes systematic failures with 5P. We can detect short strokes before we try to fit them (as explained in Section 4.5). A human observer easily detects those short strokes with undulations and oscillations that are going to be problematic, but fitting algorithms are blind to oscillations and undulations. If 5P fails, this may be due to numerical instabilities (for very flat and short arcs the difference between elliptic and hyperbolic fits is very small) so we next try 5P applied to the original stroke. If both fail, then we have a clue that the stroke may belong to the problematic subset. At this point, we switch: from fast but uncertain 5P, to slower but more accurate, and less prone to fail, DIR/R. If all of them fail, the merit is returned as zero.

As we are searching for a balance between speed and precision, we have implemented two other operating modes (in addition to the *fast* mode already described). In the *balanced* mode, we try 5P/CH, DIR/R and DIR/R/CH In the *accurate* mode, we try DIR (only for long arcs), followed (if necessary) by GEF/R.

The variables of our approach are the parameters that define all conics (ellipses, parabolas and hyperbolas), which can be defined by a general second degree equation,

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \qquad / \, \Phi = (a, b, c, d, e, f) \qquad (1)$$

In all cases, failures occur when the six parameters returned by the algorithm—which should define the equation of the conic in general form (Equation 1)—do not define a valid ellipse.

Valid fits may differ from each other (Figure 3), and may be poor (differ greatly from the stroke). To know if an algorithm has returned a poor fit we have two choices: a very slow point-by-point comparison, or fast metrics based on perceptual criteria. The approach uses the second way, based on the metrics defined in Section 4.5.

Figure 3: Nearly polygonal stroke fitted by 5P (left), DIR (middle) and GEF (right).

Thus, our approach emulates human perception, as, from results obtained in Section 3, we know that distinguishing between *good*, *average* and *poor* is perceptual behaviour. It is also similar to human perception in behaving different for *short* and *long* arcs, and by choosing the fastest approach instead of the more geometrically accurate.

Once a fit is obtained, evaluation of the merit of the fit—by distinguishing between *good*, *average* and *poor* strokes, as humans do—defines three tolerance bands and checks whether the points of the stroke are located inside them. This is described in Section 4.7.

The full code is available at [20], and approach may be summarised as follows:

1. Calculate stroke metrics (*BoxLength, Sagitta, ShortStroke, FlatStroke, SmallAndFlat*). (see Sect. 4.5)
2. Fit an elliptical arc to the stroke.
    1. If operating mode is *fast* or *balanced*, calculate Φ= (a, b, c, d, e, f) by 5P/CH (Sect. 4.3).
        i. Calculate main axes, centre, foci and endpoints from Φ.
        ii. Φ is valid if Δ ≠0 and J>0 and Δ *(a+c)<0 (see Section 4.1)
        iii. Calculate arc metrics (*SmallArc, ArcOverflow*). (see Section 4.5)
        iv. Set Φ as invalid if:
            1. (*SmallAndFlat* and not *SmallArc*), or
            2. (Not *ShortStroke* and *SmallArc*) or
            3. (Not *SmallArc* and *ArcOverflow*)
    2. If Φ is not valid, and operating mode is *fast*, then calculate Φ by 5P applied to the original stroke.
        i. Same steps i to iv as in 2.1.
    3. If Φ is not valid, and (operating mode is *fast* or *balanced*) or (operating mode is *accurate* and not *ShortStroke),* resample to 20 points and calculate Φ by DIR/R.
        i. Calculate main axes, centre, foci and endpoints from Φ.
        ii. Φ is valid if Δ ≠0 and J>0 and Δ *(a+c)<0 (see Sect. 4.1)
        iii. Calculate arc metrics (*SmallArc, ArcOverflow, GapEnds, ArcUnderflow*). (see Sect. 4.5)

iv. Set Φ as invalid if:

1. (*SmallAndFlat* and not *SmallArc*), or
2. (Not *ShortStroke* and *SmallArc*) or
3. (*GapEnds* is set) or
4. (*SmallAndFlat* and *ArcUnderflow*)
5. (Not *SmallArc* and *ArcOverflow*)

4. If Φ is not valid, and operating mode is *balanced*, resample to 20 points and calculate Φ by DIR/R/CH (Sect. 4.3 and 4.4).

   i. Same steps i to iv as in 2.3.

5. If Φ is not valid and operating mode is *accurate,* then calculate Φ by GEF/R (after resampling the original stroke to 20 points).

   i. Calculate main axes, centre, foci and endpoints from Φ.

   ii. Φ is valid if Δ ≠0 and J>0 and Δ *(a+c)<0 (see Sect. 4.1)

6. If Φ is not valid, then **return** zero merit

3. Evaluate the merit of the fit

   1. Calculate *Tol* for the original stroke.
   2. If *Tol* is less than minimum tolerance, then

      i. Return the highest figure of merit "1".

   3. Smooth the original stroke, as in Section 4.6.
   4. Calculate *Tol* for the *smoothed* stroke (Sect. 4.7).
   5. Assign merit 1 for *Tol*≤*TolMin*, null for *Tol*>*TolMax*, and linearly decreasing from 1 to 0 inside this range.
   6. Reduce merit for oscillating strokes:

      Merit= max(Merit - Penalty*NSS, 0), see Sect. 4.7.

4.1 Valid ellipse

All conics (ellipses, parabolas and hyperbolas) can be defined by a general second degree equation (Equation 1), where one parameter can be fixed without loss of generality. Defining Δ as the determinant of the matrix of coefficients (Equation 2), and *J* as the Jacobian (Equation 3), Φ describes a non-degenerate real ellipse if Δ≠0 (otherwise it is a degenerate conic), *J*>0 (otherwise it is not an ellipse) and Δ (a+c)<0 (otherwise it is an imaginary ellipse).

$$\Delta = acf - \frac{b^2 f}{4} + \frac{bde}{4} - \frac{cd^2}{4} - \frac{ae^2}{4} \qquad (2)$$

$$J = ac - \frac{b^2}{4} \qquad (3)$$

4.2 Endpoints detection

Once we get the conic in general form (Equation 1), we can easily calculate the canonical parameters: main axes (defined by the length of both semi-axes R_a and R_b, plus the orientation of the major axis α), the centre (c_x, c_y) and the foci (F1, F2).

$$(c_x, c_y) = \left(\frac{be - 2cd}{4J}, \frac{bd - 2ae}{4J}\right) \qquad (4)$$

$$aa1 = \frac{ae^2 + cd^2 + fb^2 - bde}{2} - 2acf \qquad (5)$$

$$aa3 = \sqrt{(a - c)^2 + b^2} \qquad (6)$$

$$R_a = \sqrt{\frac{aa1}{J(a + c - aa3)}}, \qquad R_b = \sqrt{\frac{aa1}{J(a + c + aa3)}} \qquad (7)$$

$$\begin{cases} \alpha = 0, & if\ b = 0\ and\ a < c \\ \alpha = \frac{\pi}{2}, & if\ b = 0\ and\ a \geq c \\ \alpha = \frac{\tan^{-1}\left(\frac{b}{a - c}\right)}{2}, & if\ b \neq 0\ and\ a < c \\ \alpha = \frac{\pi}{2} + \frac{\tan^{-1}\left(\frac{b}{a - c}\right)}{2}, & if\ b \neq 0\ and\ a \geq c \end{cases} \qquad (8)$$

$$F1 = c_x + \sqrt{{R_a}^2 - {R_b}^2}\cos(\alpha) \qquad (9)$$

$$F2 = c_y + \sqrt{{R_a}^2 - {R_b}^2}\sin(\alpha) \qquad (10)$$

However, since we fit a full ellipse while we need an elliptical arc, we also calculate the points of the ellipse that are closest to the endpoints of the stroke. For this, we need a function to calculate the point of an ellipse which is closest to a generic point P. This is the point T where the normal to the tangent to the ellipse passes through P. Since this results in a transcendental equation, we use a root-finding technique explained and implemented by Eberly [21]. We also calculate the point on the ellipse closest to the midpoint of the stroke, to choose from the two arcs defined by the endpoints the one which better fits the stroke.

4.3 5P Variants

Since any selection of the five-tuples for 5P is arbitrary, we tested three variants of 5P in which we took different approaches to five-tuple selection. There is no loss of generality in such arbitrary selection: any resulting lack of precision during the fitting stage does not prevent assigning the right merit to the fit during the second stage, where we measure the tolerance between the elliptical arc and the *original* points. Firstly, we implemented a "standard" 5P, for which the five points are the two endpoints of the stroke and three intermediate points are at equal intervals in the stroke (effectively the same as resampling to five points). For nearly-closed arcs where the two endpoints are close, we chose instead the midpoint of the two endpoints and four intermediate points (effectively the same as resampling to six points, of which the first and the last are the same). All these 5P approaches trivially guarantee that the endpoints of the stroke belong to the ellipse (meeting the closure principle of the Gestalt criteria [22]).

The second and third variants are designed to avoid the problem of locally concave strokes. These are based on the perceptual assumption that humans, when forced to make sense of a very irregular shape, ignore its details and try to make sense of its global contour ("where possible, interpret a curve in an image as the rim of a surface in 3D" [23]). The second variant, 5P/CH, discards those points of the stroke not on the convex hull and runs 5P (as above) on the reduced list of points which remains. A similar strategy is used to create a DIR/CH variant of DIR.

To obtain the convex hull, we recursively remove those points that define a concave connection with their neighbours. We use the Shoelace Formula to determine if the stroke is clockwise or anticlockwise (the output is unpredictable for self-intersecting strokes, but those strokes are not important here, as they cannot depict an elliptic arc). For clockwise strokes, if the current point lies in the right semiplane of the two semiplanes defined by the line connecting the preceding and following points, this means that current point is concave and must be removed. For anticlockwise strokes, points in the left semiplane are removed. In both cases, the semiplane is determined by the sign of the cross product of the oriented lines (previous-current) and (current-next).

Again, for nearly-closed arcs we chose instead the midpoint of the two endpoints and four intermediate points from the convex hull.

The third variant, 5P/DQ, starts with the two endpoints (A and B), adds a third point C which is the stroke point furthest geometrically from the line AB, and fourth and fifth points D and E which are, respectively, the point in the sequence between A and C which is furthest geometrically from the line AC, and the point in the sequence between C and B which is furthest geometrically from the line CB. We did not test any alternative to 5P/DQ for nearly-closed arcs.

Figure 4 allows for a visual comparison between 5P, 5P/CH, and 5P/DQ. In the first row (an average quality stroke) all merits are average. In the second row (a poor quality stroke) all merits are zero.

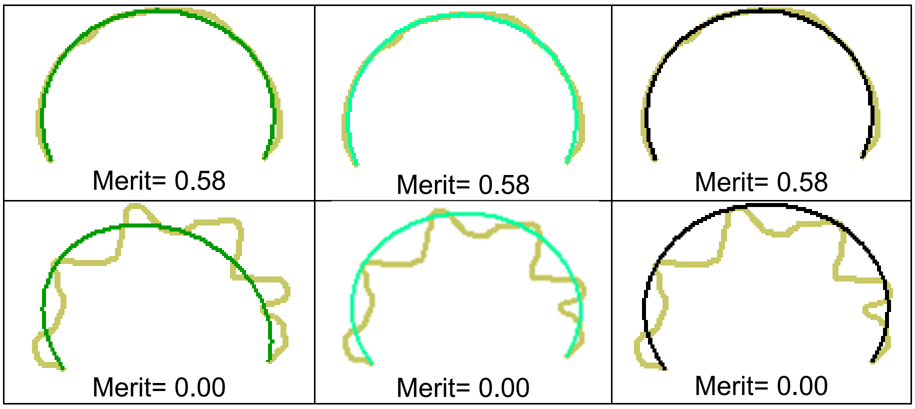


Figure 4: Two strokes (upper and lower rows), fitted with 5P (left), 5P/CH (middle) and 5P/DQ (right)

4.4 Resampling

We can reduce the time taken by DIR or GEF, without significantly reducing the quality of the fit, by reducing the number of input points. DIR/R and GEF/R are, respectively, DIR and GEF where the input is limited to 20 points, extracted at equal intervals from the sequentially-ordered input, plus the two endpoints. Where the interval is fractional, we interpolate between the appropriate two points in the input.

Kumar et al. [24] is similar to our approach as it also proposes resampling the input data before fitting. But their goal is improving accuracy (so they increase the number of points of the stroke), while ours is reducing the time.

4.5 Fast metrics for measuring fit quality

As well as finding our fit, we must assess how good it is. We could measure differences between the input stroke and the resulting arc (for instance, by comparing differences between their respective perimeters), but this would require too much calculation time for an interactive application. Instead, our approach is based on fast metrics that measure reasonably well the discrepancies between the stroke and the arc.

According to our analysis (see Section 2) 5P and DIR failures appear mainly for arcs covering small angles. We set a flag *ShortStroke* when the angle encompassed by the stroke is smaller than 180 degrees. The approach can indirectly measure the angle before fitting the arc since values *GapF* and *GapL* (Figure 5) are null for such short strokes. Both gaps can be calculated as the maximum distances between the endpoint and its closest side of the bounding box. To allow for sketching errors, we set the flag if the larger of (*GapF*, *GapL*) is smaller than a tolerance threshold set to 10% of the larger of (*Chord*/2, *Sagitta*).

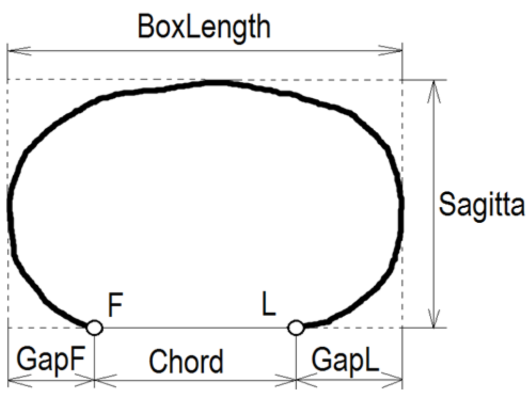


Figure 5: Parameters to determine whether the stroke depicts an arc that covers a small angle of an ellipse.

This metric assumes that the stroke roughly depicts an elliptical arc. It would fail—and does—for very poor quality strokes (e.g. Stroke 12 in Figure 6), but this is not a problem, as our goal for such bad strokes is simply detecting that they do not depict elliptical arcs.

Some long strokes are wrongly classified as short by our fast metric. A tight threshold solves this problem, but does not allow for sketching mistakes. However, short strokes are not a problem for 5P *unless* they are also “flat”: we set a second flag, *FlatStroke*, if the stroke depicts a nearly straight line. We first calculate the bounding box of the stroke, where *BoxLength* is parallel to *Chord* (the chord connecting the two endpoints F and L) and *Sagitta* is perpendicular (Figure 5). For small-curvature strokes, *Sagitta* is much smaller than *Chord*. Currently, *FlatStroke* is set if *Sagitta* is less than 20% of *Chord*.

If the stroke depicts an arc that covers a small angle and is nearly flat (i.e. if both *ShortStroke* and *FlatStroke* are set), we set a flag *SmallAndFlat*. Also, the angle covered by the stroke is compared with the angle covered by the arc: we set a flag *SmallArc* if the arc encompasses less than 180 degrees.

For long strokes, 5P has a bias towards excessively big arcs: we set a flag *ArcOverflow* if the arc is much bigger than the stroke. The approach uses the orientation of the stroke, measured as the angle of *Chord*, and the angle of the major axis of the arc (α), where $0 \leq angle < \pi/2$. If the absolute difference between both angles is less than $\pi/8$, the major axis is assumed to be nearly parallel to *BoxLength*. In this case, *ArcOverflow* is set if $2 \cdot R_a$ is greater than *BoxLength*, or $2 \cdot R_b$ is greater than *Sagitta*. If the absolute difference between both angles is greater than $3 \cdot \pi/8$, major axis is assumed to be nearly perpendicular to *BoxLength*. In this case, *ArcOverflow* is set if $2 \cdot R_b > BoxLength$ or $2 \cdot R_a > Sagitta$. In other cases, *ArcOverflow* is set if $R_a > \sqrt{(BoxLength^2 + Sagitta^2)}$.

Thus, there are three cases where 5P is prone to produce bad fits: (1) If *SmallAndFlat* and not *SmallArc*; (2) if not *ShortStroke* and *SmallArc*, and (3) If *ArcOverflow* and not *SmallArc*.

In evaluating DIR fits, we pay attention to its typical failure: the bias toward smaller ellipses. This failure is linked to short and flat strokes, but it also results in failures to meet the Gestalt principle of closure. Hence, we set flag *GapEnds* if the distance from the elliptical arc to at least one of the two endpoints (see next section) is greater than 5% of the *reduced* length of the stroke; where reduced length of the stroke is calculated as the sum of the lengths of *numIntervals* chords. A small value of *numIntervals* (typically 10) is used to find the approximate length of the undulating arc that should result after removing high frequency oscillations from the original arc. In this way, the approach distinguishes oscillations from undulations.

Since DIR has a bias towards excessively small arcs (but only for arcs covering large angles), we set flag *ArcUnderflow* if the arc is much smaller than the stroke. This occurs when *BoxLength* is greater (typically 15% more) than $2 \cdot R_a$.

Thus, DIR is assumed to be prone to produce a poor fit: (1) if *SmallAndFlat* and not *SmallArc*, (2) if *GapEnds* is set, (3) if *ArcOverflow* is set.

4.6 Smoothing

There are well-known techniques for removing high frequency oscillations (for instance [25]), of which moving average (or rolling average) is perhaps the most popular. However, it is also well-known that estimating the right parameters for distinguishing noise from signal is critical [26]. Since we are interested in what humans perceive, we opted for a very simple method which can be easily controlled by those parameters which seem to be the most important for humans: corners and width. Thus, our approach removes "micro-corners", taking advantage of the fact that strokes are sequences of points (not clouds of points) to sequentially remove alternate points of the stroke as long as the smoothed stroke still contains corners, and while the width of the resulting stroke is still similar to the width of the original stroke. The metric for *similarity* is the maximum permitted variation in the stroke width while smoothing it, defined as threshold *TolSmooth*—we only remove alternate points whenever their mutual distance is lower than *2*TolSmooth*. The number of smoothing steps (*NSS*) required for every stroke is recorded.

The approach uses one of three segmentation methods to find corners: IStraw [3], Shortstraw [27] and Sliding strips [28]. The initial choice is made by the user, with IStraw as the default. Since IStraw requires timing information, if it is selected but no timing information is available, Shortstraw is used instead.

Note that, although corners are calculated, strokes are not actually segmented, since we attempt to fit the line before deciding whether the stroke must be segmented. Here, we only use segmentation information at that end.

4.7 Evaluation of the merit of the fit

We evaluate the merit of the fit to distinguish between *good*, *average* and *poor* strokes (as in Experiments #1, #2 and #3). To determine the merit of the fit, we define tolerance bands and check whether the points of the stroke are located inside them. First, we check the original stroke against the minimum tolerance band (*TolMin*). Strokes passing this filter are nearly perfect elliptical arcs, and are acknowledged as such. Strokes not passing the minimum tolerance filter are smoothed (smoothing, as explained in Section 4.6, is a process for filtering out oscillations, i.e. high frequency irregularities), and checked against the maximum tolerance band (*TolMax*).

The resulting figure of merit is reduced depending on the number of smoothing stages (*NSS*) as follows: Merit= Merit - *Penalty***NSS*, where negative merits are not allowed.

Measuring the distance from every stroke point to the ellipse to determine whether the stroke is inside tolerance bands is clearly time consuming, as it requires solving one transcendental system using root finding techniques—described in Section 4.2 "Endpoints detection"—for every point in the stroke.

Instead, we assume that an ellipse is a circle that has been stretched in one direction. Hence, we rotate the stroke to align its major axis with the x-axis and rescale the stroke to convert the elliptical arc fit into a unit-radius circle. The points of the transformed stroke most distant from the unit radius are considered to be the most distant from the ellipse to the original stroke, and their distances to the elliptical arc are calculated (respectively *d_In* and *d_Out*). Finally, *Tol* is the sum (*Tol*= *d_In*+*d_Out*).

5. ANALYSIS

This section analyses the results of testing the ideas implemented in Section 4. The suggested set of tuning parameters is described in Section 5.1. Since merits obtained using our approach are close to those obtained by the most accurate approach (GEF), we can conclude that our new approach provides fits of an acceptable quality. The analysis also includes a brief description of pathological cases (Sect. 5.4).

5.1 Tuning parameters

By testing a large set of strokes—included but not limited to the strokes of the experiments in Section 3—we found that the approach is fast and returns figures of merit comparable with the results of human perception when: *Tolmin* equals 1.0% of the length of the stroke; *TolMax* is 10% of the length of the stroke; *TolSmooth* is 1.0% of the stroke length, and the smoothing *Penalty* is 5%, for every smoothing step.

Tuning the approach in this way, it returns the merits shown in Figures 6, 7 and 8. Comparing these figures with the assessments made by our experimental subjects, we can conclude that our approach nearly always matches human perception. Comparison shows that strokes with a figure of merit equal or greater than 0.5 correspond with strokes clearly perceived *good* or average by humans, and this is the criterion we suggest for those interested in converting the merit into a binary decision (*yes*/*no*).

5.2 General behaviour

5P/CH can fit all of the strokes of Experiments #1 and #2 (light green arcs in Figure 6). Poor strokes (10 and 12, and arguably 11) give fits whose only virtue is that they include the two endpoints.

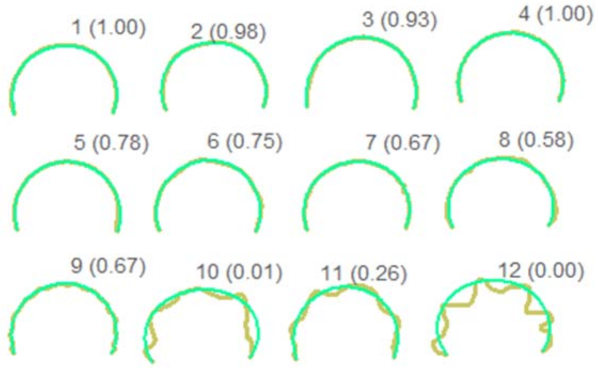


Figure 6: 5P fitting of the 12 strokes of the set “a” used in Experiment #1

5P/CH fits six of the strokes of Experiment #3, shown in light green in Figure 7. 5P fits one more (stroke 4, drawn in dark green), and DIR/R fits another four (3, 5, 10 and 11 in dark blue). None of the three methods fits a valid elliptical arc to the stroke 12, which is only fitted by DIR/CH in the balanced mode (light blue), although with a zero merit.

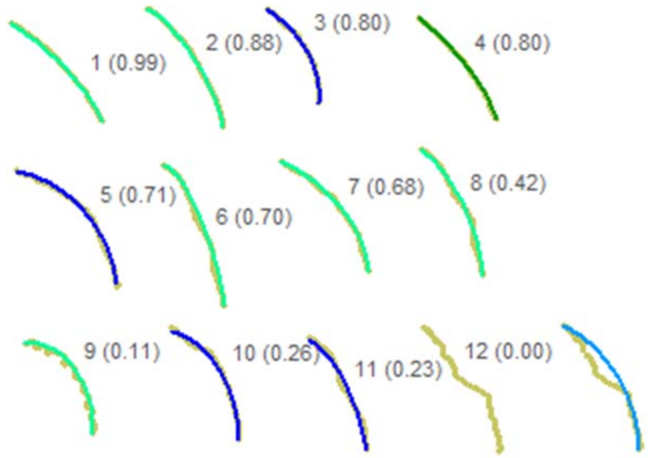


Figure 7: Fitting of the 12 strokes of the set “b” used in Experiment #3.

5P/CH fits all of the strokes of Experiment #4 (Figure 8).

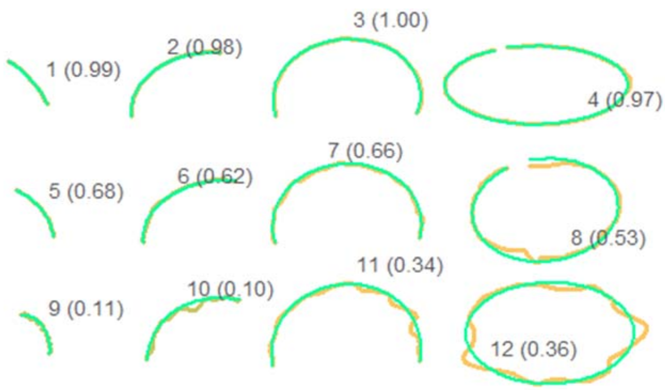


Figure 8: Fitting of the 12 strokes of the set “c” used in the Experiment #4.

It is worth noting that the first stage of our approach returned valid fits for all of the test examples except b.12. The fastest methods (5P/CH and 5P) were sufficient for all large arcs and 13 of 18 small arcs. In further tests (not illustrated in the paper), the fastest methods were sufficient for 68 of 69 long arcs and for 42 of 71 short arcs.

In the case of example b.12, we note that DIR/R/CH and GEF/R produce valid fits, so *balanced* or *accurate* mode would return a valid elliptical arc. The second stage of our approach assigns low merit to such poor fits (in this case, the merit is zero), meeting the requirement that our approach should emulate human perception (example b.12 is considered to represent an ellipse by only 1 in 7 humans).

5.3 Figures of merit

In Table 1, we tabulate the merit values of the fits which would be returned by each of the eight methods we evaluated. All three of the 5P variants sometimes returned hyperbolas (labelled as "Hyp" in the table), and 5P/DQ fails for closed loops (labelled as "Closed" in the table). Grey backgrounds indicate fits which were labelled as invalid by sub-stage 2.1.iv.

Table 1. Merits of fits to the examples of Section 5.2.

	5P/ CH	5P	5P/ DQ	DIR/ R	DIR/R /CH	DIR	GEF/ R	GEF	New app.	New-GEF
a.1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
a.2	0.98	0.98	0.96	0.97	0.97	0.97	0.97	0.97	0.98	0.01
a.3	0.93	0.93	0.92	0.94	0.94	0.95	0.94	0.95	0.93	-0.02
a.4	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
a.5	0.78	0.78	0.78	0.81	0.81	0.79	0.81	0.79	0.78	-0.01
a.6	0.75	0.71	0.74	0.80	0.79	0.80	0.80	0.80	0.75	-0.05
a.7	0.67	0.64	0.62	0.64	0.64	0.64	0.64	0.64	0.67	0.03
a.8	0.58	0.58	0.58	0.66	0.66	0.67	0.66	0.67	0.58	-0.09
a.9	0.67	0.65	0.65	0.67	0.66	0.68	0.67	0.68	0.67	-0.01
a.10	0.01	0.02	0.00	0.13	0.10	0.17	0.10	0.16	0.01	-0.15
a.11	0.26	0.18	0.28	0.27	0.27	0.28	0.26	0.27	0.26	-0.01
a.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
b.1	0.99	0.97	0.95	1.00	0.96	1.00	1.00	1.00	0.99	-0.01
b.2	0.88	0.85	0.87	0.88	0.86	0.89	0.88	0.88	0.88	0.00
b.3	Hyp	Hyp	0.75	0.80	0.78	0.79	0.85	0.87	0.80	-0.07
b.4	Hyp	0.80	Hyp	0.85	0.87	0.85	0.85	0.88	0.80	-0.08
b.5	Hyp	Hyp	0.13	0.71	0.51	0.64	0.74	0.71	0.71	0.00
b.6	0.70	Hyp	Hyp	0.68	0.59	0.68	0.67	0.67	0.70	0.03
b.7	0.68	0.51	Hyp	0.66	0.71	0.67	0.65	0.68	0.68	0.00
b.8	0.42	Hyp	Hyp	0.43	0.00	0.42	0.44	0.46	0.42	-0.04
b.9	0.11	Hyp	0.00	0.16	0.00	0.16	0.18	0.19	0.11	-0.08
b.10	Hyp	Hyp	0.14	0.26	0.32	0.15	0.37	0.38	0.26	-0.12
b.11	0.00	Hyp	0.34	0.23	0.00	0.13	0.36	0.38	0.23	-0.15
b.12	Hyp	Hyp	Hyp	0.00	0.00	0.00	0.00	0.00	0.00	0.00
c.1	0.99	0.97	0.95	1.00	0.96	1.00	1.00	1.00	0.99	-0.01
c.2	0.98	0.98	0.97	0.99	0.98	1.00	0.99	1.00	0.98	-0.02
c.3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
c.4	0.97	0.95	Clos	0.99	0.99	0.98	0.99	1.00	0.97	-0.03
c.5	0.68	0.51	Hyp	0.66	0.71	0.67	0.65	0.68	0.68	0.00
c.6	0.62	0.65	0.41	0.65	0.63	0.65	0.68	0.68	0.62	-0.06
c.7	0.66	0.64	0.65	0.70	0.68	0.70	0.70	0.70	0.66	-0.04
c.8	0.53	0.56	Clos	0.64	0.62	0.65	0.65	0.64	0.53	-0.11
c.9	0.11	Hyp	0.00	0.16	0.00	0.16	0.18	0.19	0.11	-0.08
c.10	0.10	0.06	0.12	0.20	0.11	0.04	0.20	0.20	0.10	-0.10
c.11	0.34	0.33	0.33	0.33	0.34	0.33	0.32	0.32	0.34	0.02
c.12	0.36	0.13	Clos	0.31	0.38	0.32	0.39	0.40	0.36	-0.04

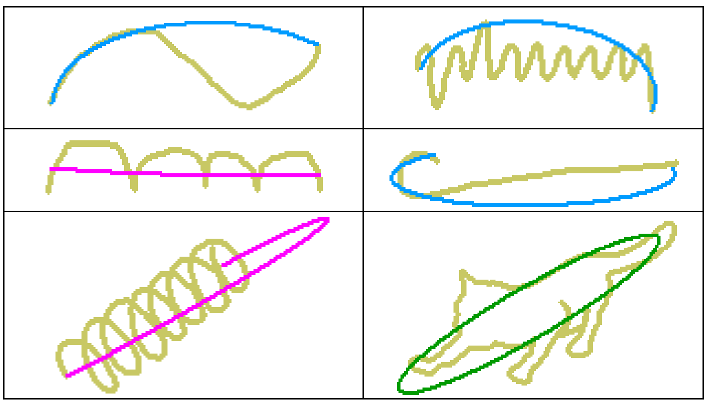
These figures allow us to choose between the eight variants. We prefer 5P/CH to 5P as 5P/CH is sometimes much the better of the two, and hardly ever worse (and even then, only by one or two points). We prefer 5P to 5P/DQ as, although they are roughly comparable for small arcs, 5P/DQ fails for closed loops. We prefer DIR/R to DIR/R/CH as DIR/R is always as good as, and sometimes better than, DIR/R/CH.

For *balanced* mode, GEF/R is usually better than any of the 5P methods, although not always by very much. GEF/R is also always as good as, and occasionally better than, DIR/R, but DIR/R is faster than GEF/R. This small gain in accuracy is not enough to justify the higher execution time, so we only use GEF/R as a last resort.

It can be seen that, in general, our approach produces fits which are almost as good as those produced by GEF (where "almost" means that the difference in measured merit is 0.10 or less). Of the four cases where our approach produces fits which are clearly inferior to GEF, three (a.10, b.10 and b.11) are poor strokes which should probably not be interpreted as ellipses, and while the fourth, c.8, is geometrically a good ellipse, it is not perceived as such by humans. In this final case, GEF is "too good", and our new approach corresponds better to human perception.

5.4 Pathological cases

Perceptually, it is as important to reject strokes which do not represent ellipses as it is to fit elliptical arcs to those which do. Figure 10 shows several strokes which, to humans, clearly do not represent ellipses. Our approach finds tentative fits for all of them but returns zero merit for these fits—there are no false positives.



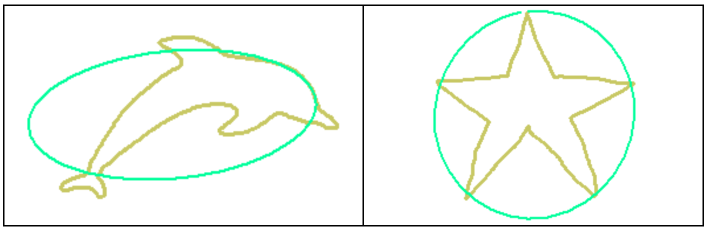


Figure 10: These are not ellipses.

Our approach currently fails when the stroke depicts an elliptical arc of more than 360 degrees. Future versions should detect and deal with this during endpoint detection.

6. CONCLUSIONS

We propose a strategy based on selecting algorithms for their speed rather than their geometrical accuracy, and emulating human perception by returning a merit instead of a binary classification. The contribution here is twofold.

Firstly, we have performed experiments to identify which strokes humans perceive as depicting elliptical arcs, and which they consider cannot be elliptical arcs. We then used the sketches from these experiments to test whether or not our approach replicates human behaviour.

Secondly, we developed a systematic approach for fitting elliptical arcs to strokes. Our approach is designed to return valid fits for every stroke in a sketch in the shortest overall time. If speed is the priority, we start with five-points applied to the convex hull (5P/CH), followed if necessary by 5P and DIR/R. If precision is the priority, we choose DIR/R (if the stroke is not short) followed by GEF/R. The metrics we use to determine when a fitting algorithm has returned a bad fit are another original contribution. In future we may try other merit functions to see how well they match human perception.

ACKNOWLEDGEMENTS

This work was partially funded by financial support from the Ramon y Cajal Scholarship Programme and by the "Pla de Promoció de la Investigació de la Universitat Jaume I", project P1 1B2010-01. We wish to thank Salvador Mondragón, who collected many questionnaires from his students, and Margarita Vergara, for her contribution to statistical data treatment.

REFERENCES

[1] Wang S., Qin S., Gao M. (2014). New grouping and fitting methods for interactive overtraced sketches. The Visual Computer 30, 285–297.

[2] Pusch R., Samavati F., Nasri A., Wyvill B. (2007). Improving the sketch-based interface. The Visual Computer 23, 955–962.

[3] Xiong Y, and LaViola J. (2010). A shortstraw-based algorithm for corner finding in sketch-based interfaces. Computers & Graphics 34 (5), 513-527.

[4] Saund, E., & Moran, T.P. (1994). A perceptually-supported sketch editor. Proceedings UIST, 175-184.

[5] Chernov N, Huang Q, Ma H. Fitting Quadratic Curves to Data Points. British Journal of Mathematics & Computer Science. 2014; 4: 33-60.

[6] Yu J, Kulkarni SR, Poor HV. Robust ellipse and spheroid fitting. Pattern Recognition Letters. 2012; 33: 492–9.

[7] Rosin PL. Ellipse fitting by accumulating five-point fits. Pattern Recognition Letters. 1993; 14: 661-9.

[8] Fitzgibbon A, Pilu M, Fisher RB. Direct least square fitting of ellipses. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1999; 21: 476-80.

[9] Halir R, Flusser J. Numerically Stable Direct Least Squares Fitting Of Ellipses. Proc. 6th Int. Conf. in Central Europe on Computer Graphics and Visualization WSCG98. 1998; 1: 125–32.

[10] Kotagiri S. (2010). C# Implementation for Fitting an Ellipse for a Set of Points. Srikanth Kotagiri's Blog. Retrieved from <http://skotagiri.wordpress.com/2010/06/19/c-implementation-for-fitting-an-ellipse-for-a-set-of-points/> Last Access November 2013.

[11] Shpitalni M, Lipson H. Classification of sketch strokes and corner detection using conic sections and adaptive clustering. Journal of Mechanical Design, Transactions of the ASME. 1997; 119: 131-5.

[12] GSL, GNU Scientific Library. Retrieved from <http://www.gnu.org/software/gsl/> Last Access November 2013.

[13] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical Recipes. The art of Scientific Computing. Cambridge Press. Third edition. 2007

[14] LAPACK — Linear Algebra PACKage. Retrieved from <http://www.netlib.org/lapack/> Last Access November 2013.

[15] Szpak Z, Chojnacki W, van den Hengel A. Guaranteed Ellipse Fitting with the Sampson Distance. Proc. 12th European Conference on Computer Vision. (ECCV). 2012; LNCS 7576: 87-100.

[16] Szpak Z.L. Research Themes: Ellipse fitting. Retrieved from: https://sites.google.com/site/szpakz/research/ellipse-fitting. Last Access May 2014.

[17] Davis T. (1996) Code to find the equation of a conic. Retrieved from <http://mathforum.org/library/drmath/view/51735.html> Last Access November 2013.

[18] Elsen C, Demaret J-N, Yang MC, Leclercq P. Document Sketch-based interfaces for modeling and users' needs: Redefining connections. Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM. 2012; 26: 281-301.

[19] Company, P., Plumed, R., Varley, P.: Study on perceptually-based fitting elliptic arcs. Technical Report Ref.09/2015, Regeo (2015), Geometric Reconstruction Group, http://www.regeo.uji.es.

[20] Company, P., Plumed, R., Varley, P.: Source code for Fitting Ellipses. <http://www.regeo.uji.es/FitEllipse.htm>, Regeo (2015), Geometric Reconstruction Group, http://www.regeo.uji.es.

[21] Eberly D. Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid. 2004. Internet publication: "Distance from a point to an ellipse in 2D". Geometric Tools, LLC, www.geometrictools.com. Book publication: "3D Game Engine Design", 2nd edition. Morgan Kaufmann Publishers, San Francisco, CA, 2007. (see Section 14.13.1)

[22] Boyer KL. Perceptual Organization in Computer Vision: Status, Challenges, and Potential. Computer Vision and Image Understanding. 1999; 76: 1–5

[23] Hoffmann D. Visual Intelligence. How we create what we see. New York: WW Norton & Company, 1998.

[24] Kumar P, Cai J, Miklavcic S. Improved Ellipse Fitting by Considering the Eccentricity of Data Point Sets. Proceedings of ICIP. 2013; 815-9.

[25] Vaseghi, S.V. (2008). Advanced Digital Signal Processing and Noise Reduction. Wiley, 2008 (fouth edition).

[26] Durbin, J. (1959). Efficient Estimation of Parameters in Moving Average Models. *Biometrica*, 46, 306–316.

[27] Wolin, A., Eoff, B., & Hammond, T. (2008). Shortstraw: a simple and effective corner finder for polylines. *SBIM 2008*, 33-40.

[28] Masood, A., Sarfraz, M. (2007). Corner detection by sliding rectangles along planar curves. *Computers & Graphics* 31, 440–448.