INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center Street • Suite 600 • Berkeley, California 94704 • 1-510-642-4274 • FAX 1-510-643-7684

Detecting Skewed Symmetries

Stefan Posch

TR-91-058

October 1991

Abstract

Many surfaces of objects in our world are bounded by planar bilaterally symmetric figures. When these figures are imaged under orthographic projection a skewed symmetric contour results. In this paper a new fast, local method to recover skewed symmetries from curve segments is proposed. It can be applied to complete as well as to occluded contours. Furthermore, the skewed symmetry property is employed to overcome fragmentation of a contour during segmentation.

1 Introduction

Many surfaces of objects in our world, man-made as well as natural, are bounded by planar bilaterally symmetric figures. When these figures are projected onto an image, this special property is generally lost. However, assuming orthographic projection (as an approximation to perspective) the image contour exhibits a skewed symmetry (see figure 1 for an example). This term was introduced by Kandade in [Kan81], and he showed that the parameters of a skewed symmetry constrain the possible orientations of the projected surface bounded by the figure. Work to resolve the remaining *gradient ambiguity* to get an unique solution for the surface orientation includes [UN88, Yue90, Kan81, BY84].

A prerequisite to exploit this constraint is, of course, the detection of skewed symmetries and their parameters. This is the topic of this work. There has been other recent work addressing this problem. Since I think it is important to permit occlusion, only local methods are appropriate. All proposed local methods are relatively slow, however. Furthermore, all of these methods assume a priori grouping of all visible edge elements constituting a symmetric contour. One aim is, to relax this assumption and allow segmentation into few curve segments. In this sense, the algorithm is to perform a perceptual grouping task using skewed symmetry as a non-accidental property ([Low85]) to group curve segments into meaningful entities. Thus the goal is to develop a fast, local method to detect all symmetry parameters of a skewed symmetric contour. The algorithm should be able to handle occlusion and fragmentation of the complete contour into a few curve segments.

In the rest of this section some notation is introduced and related work is described. The new algorithm is explained in section 2 and results are discussed in section 3.

1.1 Notation

Throughout this paper I refer to surface boundaries and surface markings as *(3D) figures* and to their projections in the image plane as *contours*. Edge detection and linking of connected edge elements (or edgels) results in *curve segments*, where one contour may be fragmented in more than one curve segment.

Consider the skewed symmetry shown in figure 1. The *symmetry axis* of the skewed symmetry is the projection of the symmetry axis of the bilaterally symmetric plane figure. This symmetry axis will be parametrized with the angle ϕ it makes with the x-axis and the distance d to the origin of the coordinate system. A *generator (line)* connects pairs of curve points, which are projections of reflected points in the 3D figure. Such curve points are also called *symmetric (curve) points* and said to *match* each other. The point bisecting the line connecting two symmetric points lies on the symmetry axis and is named *midpoint*. The symmetry axis and generator

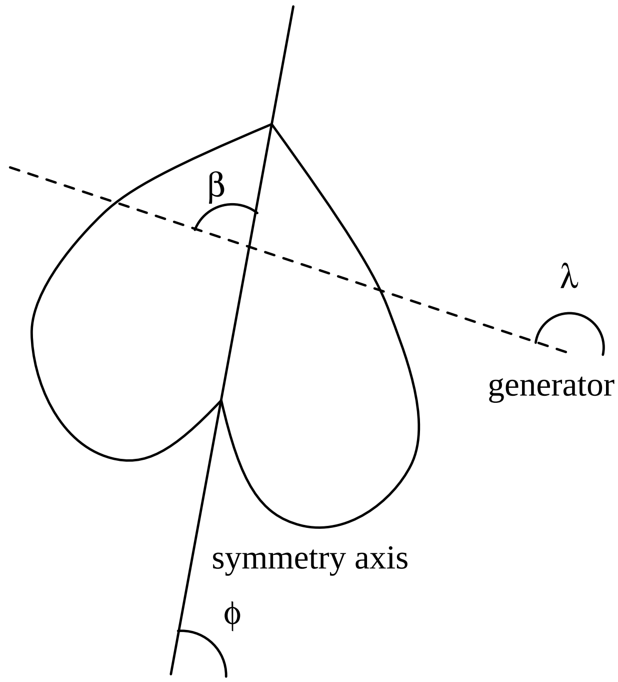


Figure 1: Notation for a skewed symmetry

angle make up a set of parameters completely determining a skewed symmetry. The generators have also been called axes of skew, lines of symmetry, traverse axis, and ribbon axes. All generators make the same fixed angle β, the *skew (angle)*, with the symmetry axis. The angle between a generator and the x-axis is named *generator angle*, thus $\lambda = \phi + \beta$. If a skewed symmetric contour has more than one symmetry axis, it is called *skew ambiguous*. It is well known, that the process of generating a skewed symmetry can also be described as a oblique transformation followed by a two-dimensional rotation of a bilaterally symmetric contour (see e.g. [Fri86]).

1.2 Related Work

Methods to find skewed symmetries are divided in local and global methods. Local methods consider pairs of curve points, while global methods work with global properties of the contour, generally moments.

Friedberg [Fri86] describes a global method based on the first and second order moments to constrain possible values of rotation ϕ and skew β. All valid $\phi - \beta$ pairs are then tested for a skewed symmetry using a heuristic measure of symmetry. Due to this heuristic he reports that some incorrect symmetries are found. Gross and

Boult use moments up to order n to detect symmetry axes of a contour with skewed ambiguity of order n or less ([GB91]). They report fast and reliable results on various contours. However, either the degree of skew symmetry has to be known in advance or higher order moments than necessary have to be used, which can be computed less reliably.

A local method was proposed in [Pon90]. It exploits a necessary condition for two curve points to be symmetric based on the curvature and normal of the points. The method of projection ([NB77]) is used to build symmetric curve points complying with the criterion, which are then grouped together. Since the method relies on curvature, it is subject to noise and can be applied only to contours with nonzero curvature. In [GB91] another local method is proposed based only on the tangent of curve points. They derive a condition yielding for any pair of curve points the parameters of the symmetry axis, assuming that they are indeed symmetric curve points. Their algorithm compares all pairs of curve points and for each discretized generator angle accumulates their votes for a skew angle (not however for the distance of the symmetry axis from the origin). A distribution is fit to these histograms and the generator angles yielding a standard deviation below a given threshold selected as resulting skewed symmetries. Another local method is reported in [SMM90] using a B-spline approximation for the contour. Employing the method of projection for each generator angle all conics of two B-splines are searched to find midpoints of elementary axis of symmetry. These elementary axes are then grouped to form a global symmetry axis. A straight line is fit through all resulting midpoints and the generator angle yielding the straightest line chosen. If the best axis is not straight enough, the contour is rejected as being not symmetric. They do not address the problem of multiple matches between conics for a given generator angle.

Two other local methods make use of the Hough transform. The method described in [OAT88] uses a polygonal approximation of a region boundary. First, for pairs of straight line segments a Hough transform in the (ϕ, d)–space of possible symmetry axes is computed. A second Hough transform in the (ϕ, λ)–space examines the candidates found in the first step. A verification of axes corresponding to prominent peaks in this Hough array determines the final decision on symmetry axes, taking occlusion into account. This method depends critically on the stable polygonal approximation, which is difficult to achieve. In [Yue90] Yuen proposes to perform a Hough transform for each generator angle. For each pair of curve points connected by a line of the given generator angle their midpoint is recorded and the Hough transform used to fit straight lines to these midpoints. No results are given and it is not obvious how well this algorithm performs.

2 A new method

As argued in [GB91] global methods have the chance of being fast, since not pairs of contour points have to be considered. However, they inherently require a priori knowledge of all contour points of a skewed symmetry and hence suffer from occlusion and fragmentation of contours. Therefore, I adopt a local approach.

The method is divided into two stages. In the first stage one or two curve segments are considered and a (possibly empty) set of possible symmetry parameters computed. In the second stage these hypotheses are exploited to handle fragmentation of contours adding more curve segments to the skewed symmetry. If no fragmentation is assumed the first stage can of course be employed alone and yields a number of symmetry axes along with a measure of reliability.

2.1 Stage I (Finding a partial skewed symmetry)

The basic idea for stage I is illustrated in figure 2. The method of projection is used for a given generator angle λ to slice the image plane in strips of a specified width. Within each strip connected curve points are combined to form a *bucket* and the connectivity among the buckets is retained. For each bucket the center of gravity is computed, shown as dots in the figure 2. In this example the bucketing results in a pair of buckets for each strip, where each of these pairs gives rise to a midpoint located on the symmetry axis[1]. Of course the correct generator angle or angles are not known in advance. Therefore, the projection is iterated for different generator angles and a straight line is fit to the resulting midpoints. For a correct generator angle this line fitting will result in a small approximation error.

So far, the basic idea is similar to the method described in [SMM90], except that no approximation of the contour is necessary. However, I will also deal with ambiguities within one strip (see next paragraph) and skew ambiguity. Also a possible fragmentation of the contour is handled with the second stage of the method.

Figure 3 (a) shows the contour of a heart along with the buckets for a given generator angle. For some strips no unique match of buckets exists and the simple minded algorithm outlined above would fail. However, for a part of the contour the pairing is still unambiguous (see figure 3 (b)). For virtually any skewed symmetry this will be the case for a correct generator angle. A counter example is a – possibly skewed – sine wave, which of course is not a surface boundary. Such a unique pairing of a set of connected buckets is called a *seed*.

Now the connectivity within a curve segment (and therefore within the buckets) is

[1]If two curve segments are considered they are assumed to be located on different sides of the symmetry axis as indicated in the figure.

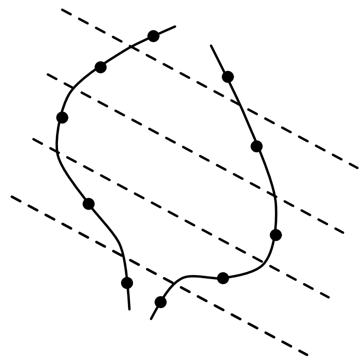


Figure 2: Strips and buckets for a simple skewed symmetry. The center of gravities of the buckets are represented by the dots.

exploited to extend initial seeds. For the bucket pair at each end of a seed the neighboring buckets (not already in the seed) are found. If they lie in the same strip they are considered as a new unique bucket pair and the expansion continues. Figure 3 (c) illustrates some steps of the expansion at the bottom end of the seed.

At this point another complication is demonstrated. The neighboring buckets at the end of the expanded seed belong to different strips and hence can not be symmetric buckets. This may be due either to the discretization of λ or positioning errors of some edgels, but of course also to a wrong generator angle. To cope with these effects, the curve is tentatively followed at both buckets, until a bucket in the correct strip is reached. In the example, for the left bucket six buckets have to be skipped, only two for the right bucket. The shorter path is assumed to be correct and the expansion continues (see figure 3 (d)). Of course, due to fragmentation or nonexistence of a skewed symmetry no appropriate bucket may be found during skipping and the expansion stops at this point. The expansion also stops at the end of one curve or whenever another seed is reached. After all seeds have been expanded a line is fit through the midpoints yielding a possible symmetry axis.

Now the quality of the skewed symmetry for the various generator angles has to be evaluated. For a perfect skewed symmetry a straight line can be fit without error through the midpoints of symmetric buckets. To account for noise and discretization a certain approximation error has to be tolerated. Applying a threshold to the mean approximation error bears two problems. First, this measure is not scale independent. Using different resolutions or imaging the same object from different distances will result in different approximation errors for the same 3D figure. Therefore, the approximation error is normalized by the mean distance between matched buckets:

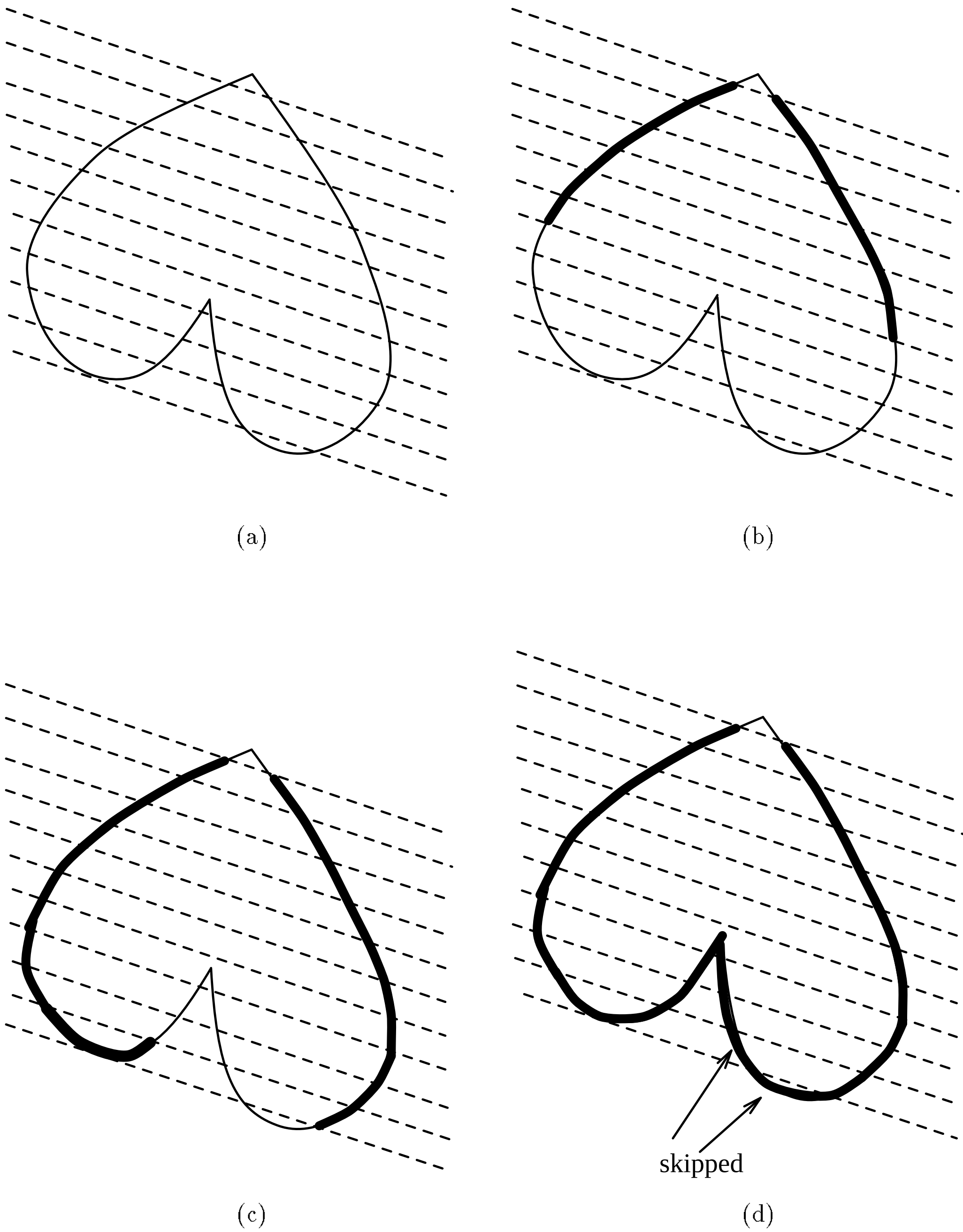


Figure 3: Illustration of the algorithm to find bucket pairs. Matched buckets are shown bold. (a) Skewed symmetric contour of a heart with strips as defined for a given generator angle. (b) Bucket pairs for strips with a unique match constituting the initial seed. (c) Some steps of the expansion are performed until neighboring buckets are in different strips. (d) Two buckets in the right half are skipped and the expansion completed.

$$\mathrm{error}_{\mathrm{norm}} = \frac{\mathrm{error}_{\mathrm{approx}}}{\mathrm{mean\_distance}} \leq \vartheta_{\mathrm{error}} \quad (1)$$

with	$\mathrm{error}_{\mathrm{norm}}$	normalized approximation error
	$\mathrm{error}_{\mathrm{approx}}$	approximation error fitting a straight line
	mean_distance	mean distance between symmetric buckets

Second, this error often increases only slowly with deviation from the correct generator angle (see figure 4 for an example). Therefore a tight threshold would have to be employed to avoid these false multiple symmetry axes. This on the other hand has the potential for missing correct symmetry axes due to the low threshold. For this reason only generator angles of local minima of the normalized approximation error are considered, since I am concerned with contours exhibiting a moderate degree of skewed ambiguity. In other words, I assume that generator angles corresponding to different symmetries of a contour are separated by at least twice the amount of the chosen spacing for the generator angles[2]. Moreover, I will take advantage of the smoothly varying approximation error to reduce computation. High resolution testing the generator angles is only necessary in the vicinity of a correct generator angle, but not for the whole range of possible angles. Therefore, in a first pass the space of generator angles is searched using a coarse resolution. In a second pass a finer resolution is used in the vicinity of local minima of the approximation error to reduce the normalized approximation error.

For the resulting generator angles yielding small errors one last issue has to be discussed. This is related to *non matched buckets*, i.e. buckets for which no symmetric bucket has been found. There are three possible reasons for non-matchable parts of a curve segment, illustrated in figure 5: Nonexistence of a skewed symmetry for the given generator angle (a) and (b), an occlusion of a part of the contour (c), and the fragmentation of the contour (d).

If the second stage of the algorithm is not to be employed, i.e. no fragmentation of the contour is assumed, only the first two possibilities have to be considered. It is not possible to distinguish between these two cases, unless higher level evidence is used, e.g. looking for surfaces which occlude a certain area of the image. Since such information is not available at this stage of processing, it is not considered here. Therefore, I restrict the amount of occlusion which is tolerated. This is a reasonable assumption, since otherwise virtually every contour could be interpreted as being skewed symmetric (e.g. figure 5 (b)). Assuming that the number of occluded and non matched buckets is the same, we get the requirement

[2]In the case of a higher degree of skewed ambiguity a certain amount of symmetry axes still would be found.

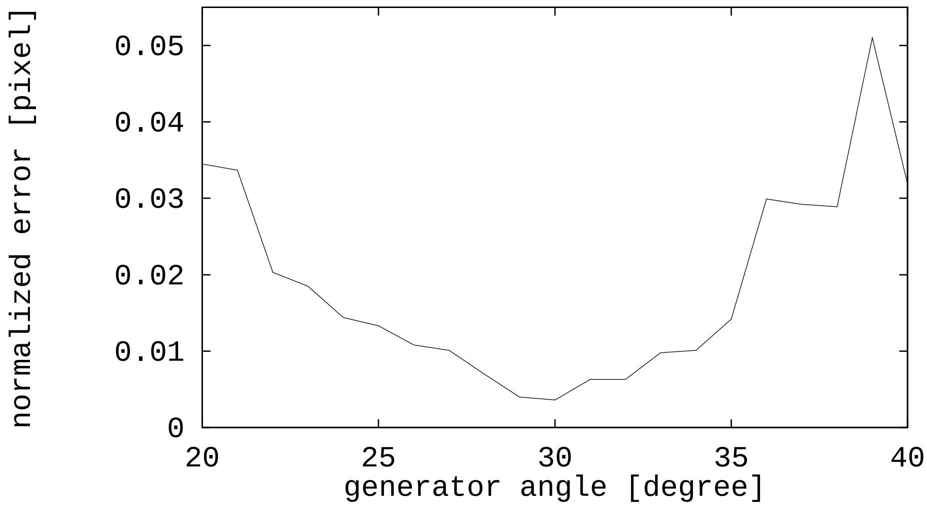


Figure 4: Normalized approximation error for the heart in figure 12.

$$\frac{n}{2n+m} \leq \vartheta_{\text{nonmatched}} \tag{2}$$

with	n	number of non matched buckets
	m	number of matched buckets

If the second stage is conducted to handle fragmentation, no decision regarding the non matched buckets is taken at this point. In this case, all hypotheses for symmetry axes with error below the threshold ϑ_{error} are passed and the threshold for non matched parts of the contour is applied to the results of the second stage.

To summarize the first stage, the curve points are projected according to a given generator angle, seeds of unique bucket pairs computed, and these seeds expanded utilizing the connectivity along the curve segments. Then a line is fit through the midpoints yielding a mean approximation error, which is normalized by the mean distance between matched buckets. This is iterated for various generator angles according to the chosen discretization and the symmetry axes yielding local minima of the normalized approximation error are thresholded. If the second stage is employed subsequently, all the resulting hypotheses are retained. Otherwise (assuming no fragmentation of the contour), a threshold is applied to the amount of non matched buckets to discard the hypotheses, which would require too much occlusion. To enhance the resolution with respect to the generator angle, the projection is repeated for neighboring angles of resulting solutions and the angle resulting in the minimal

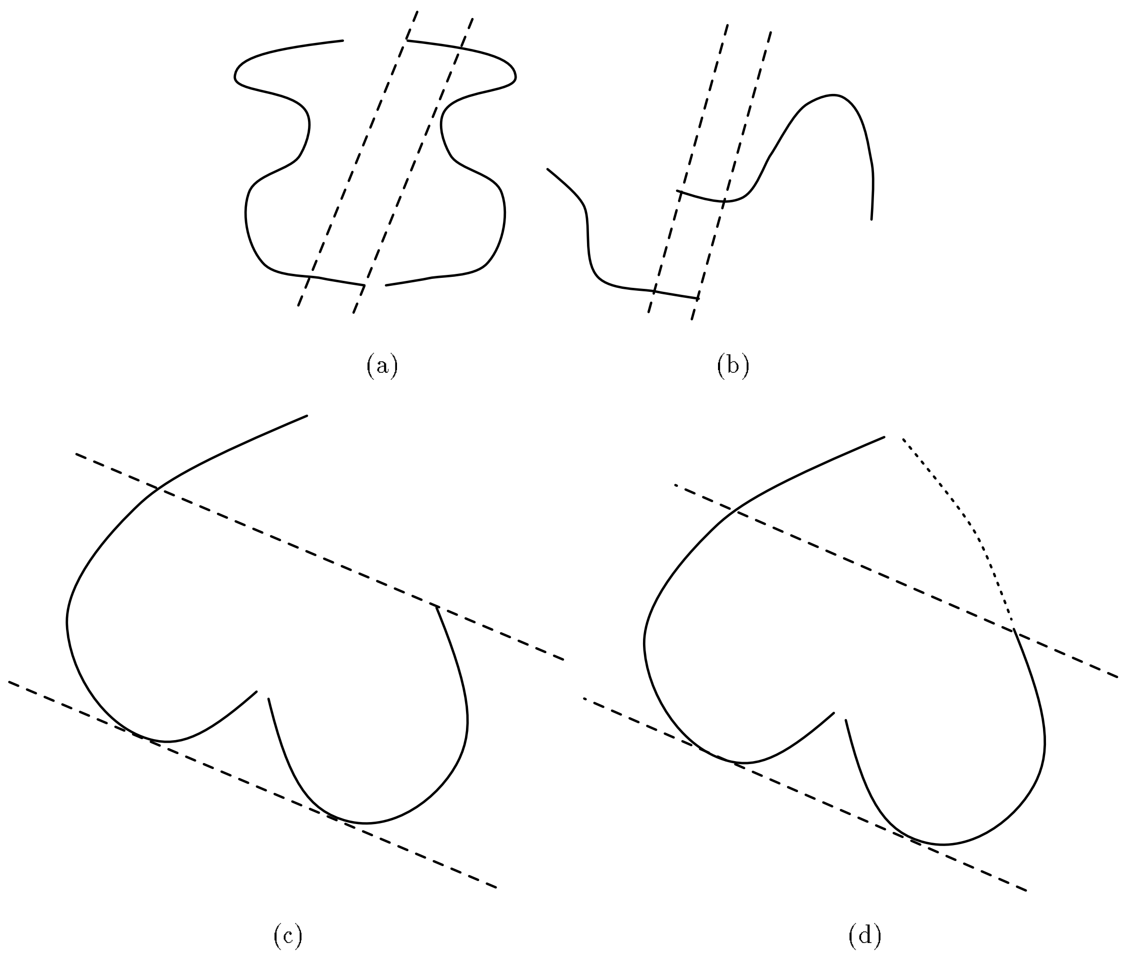


Figure 5: Non matched parts of a contour due to nonexistence of a skewed symmetry for the given generator angle (a) and (b), occlusion (c), and fragmentation (d). The matched parts of the curve segments are enclosed in dashed lines.

error is retained. For a discussion of thresholds, parameters, and results see section 3. Figure 6 gives an outline of the algorithm.

2.2 Stage II (Completing a partial skewed symmetry)

In the first stage, either one or two curve segments are considered to form hypotheses of skewed symmetries. Now, each hypothesis is further investigated to possibly find more curve segments belonging to the same symmetric contour. This stage is necessary, whenever contours may be fragmented and the segmentation of the resulting

```
find_partial_skewed_symmetries
    for λ = 0 step Δλ to π do
        test_generator_angle(λ)
    endfor

    retain λ, with
        i)   local minimum of normalized approximation error
        ii)  normalized approximation error below ϑ_error
        iii) percentage of non matched buckets below ϑ_nonmatched
        (condition iii) is dropped, if stage II is to be performed)

    for remaining λ do
        λ = min_{μ∈{λ+Δμ}} { normalized approximation error of test_generator_angle(μ)}
    endfor
```

```
test_generator_angle(λ)
    project curve segments according to λ; connected curve points within a strip
    generate a bucket; retain connectivity between buckets

    find strips with exactly two buckets and create for each set of such neighboring strips
    a seed

    for each seed do
        expand_seed at both ends
    endfor

    fit straight line through all midpoints and compute normalized approximation error
```

```
expand_seed
    until next seed is reached or one curve ends do
        let b1, b2 be the buckets of the last midpoint in the seed

        let b̂1, b̂2 be the connected buckets to b1, b2 outside the seed

        if b̂1, b̂2 are in the same strip then
            create midpoint for b̂1, b̂2 and add it to seed
        else
            find transitively the buckets b̃1, b̃2 connected to b̂1, b̂2,
            such that b̃1 is in the same strip as b̂2 and b̃2 is in the same strip as b̂1

            if b̃1 or b̃2 exists then
                create midpoint for pair with fewer skipped buckets and add it to seed
            endif
        endif
    enduntil
```

Figure 6: Outline of stage I, finding a partial skewed symmetry.

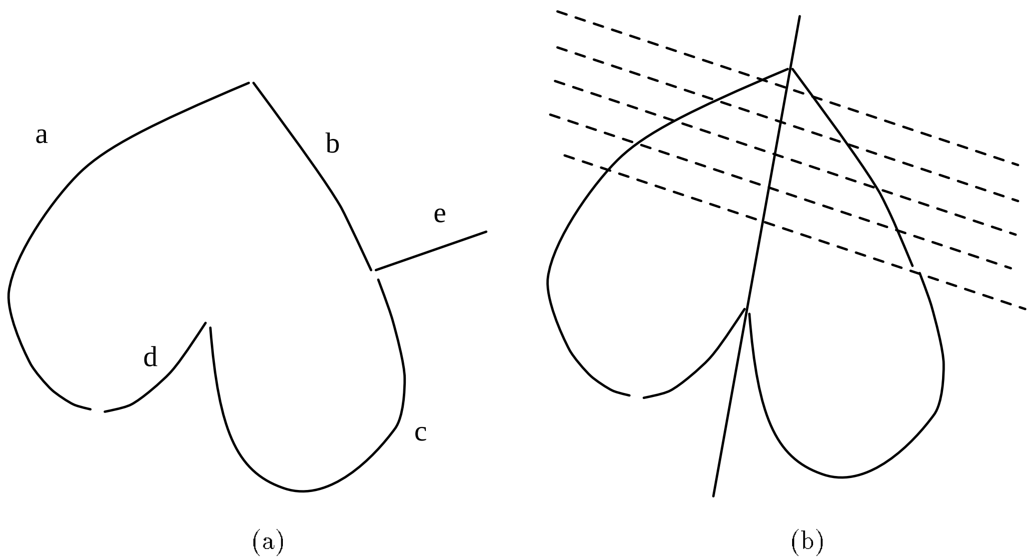


Figure 7: (a) Heart contour fragmented into four curve segments a to d and an additional curve segment e. (b) Skewed symmetry hypotheses in stage I considering a and b.

curve segments into a complete contour is not assumed to be known in advance.

To avoid the necessity to bucket all curve segments of the image, this second stage operates on the curve points directly. Therefore, the first step is to compute the symmetric curve points induced by matched buckets. For each curve point the matched bucket is searched for the curve point yielding the midpoint closest to the symmetry axis and all these midpoints are collected.

Figure 7 (a) shows the heart contour, used in the last section, fragmented into four curve segments plus an additional one. One hypothesis found applying stage I to the curve segments a and b is shown in figure 7 (b). Due to the fragmentation some curve points have not been matched. Since a hypothesis of the symmetry parameters is available from stage I, it is possible to search for partners of these curve points, the curve segment is said to be *completed*. This is simply done by reflecting the coordinates at the symmetry axis in direction of the generator angle and looking for the nearest curve point in a neighborhood of this reflected point. If it exists, a new pair of symmetric curve points is added to the symmetry.

Beside adding new midpoints, this completion also yields a set of new curve segments possibly belonging to the symmetric contour. These are the curve segments of the new curve points found by completing an old curve segment (curve segments c and e in the example). In order to decide which of theses curve segments belong to the symmetric contour, the curve points of each potential segment are tentatively completed in the same way as the original curve segments yielding a set of potential midpoints.

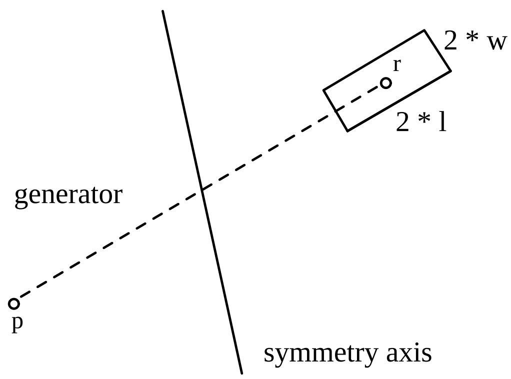


Figure 8: Neighborhood around the reflected point to search for matching curve points.

The curve segment is considered to be part of the symmetry if these midpoints lie approximately on the hypothesized symmetry axis and if a certain fraction of the curve points have found a partner. To check the first condition, a line is fit to the union of the midpoints already in the symmetry and the new potential midpoints. The resulting normalized approximation error has to be below the threshold ϑ_{error}. The second condition excludes a curve segment from being considered a part of the symmetry, which accidentally has a small number of curve points in the vicinity of the symmetric contour (like curve segment e in the example). Again condition (2) is employed, where curve points instead of buckets are considered.

Whenever a new curve segment of the symmetry is found, this may give rise to new potential curve segments, since some of the symmetric curve points may belong to curve segments not considered so far (curve segment d in the example). So the process is iterated recursively until no more new potential curve segments are encountered.

The shape of the neighborhood used to find symmetric curve points is shown in figure 8. A width of $w = 1$ pixels at each side of the generator is used to handle discretization effects. A certain length l has to be provided to account for an inaccurate initial estimation of the symmetry axis. If a small value is used, a symmetric curve point can be missed due to an inaccurately hypothesized location of the reflected point. On the other hand, a large value might result in wrong matches, in case no symmetry exists or part of the contour is occluded. However, since the curve segment is not included in the symmetry if the resulting approximation error gets to big, this is not very critical. A length $l = 8$ is used, the results are however quite robust with regard to this choice.

After this recursion stops, the threshold $\vartheta_{\text{nonmatched}}$ is applied to the non matched curve points. The condition for the straight line fit of the symmetry axis is automat-

```
complete_partial_skewed_symmetry
    for each bucket pair of the skewed symmetry do
        for each curve point in bucket without match do
            find the curve point in matched bucket which yields the midpoint nearest to the symmetry axis
        endfor
    endfor

    for each curve segment considered in stage I do
        complete_curve_segment
    endfor

    fit straight line through new midpoints

    for each potential curve segment do
        complete_potential_curve_segment
    endfor

    discard skewed symmetry if percentage of non matched curve points above ϑ_nonmatched
```

```
complete_curve_segment
    for each curve point of curve segment without match do
        search for nearest curve point in the neighborhood of the reflect curve point

        if curve point exists then
            add resulting midpoint to symmetry
            add curve segment to potential curve segment if not already in symmetry
        endif
    endfor
```

```
complete_potential_curve_segment
    for each curve point of potential curve segment without match do
        search for nearest curve point in the neighborhood of the reflect curve point

        if curve point exists then
            add resulting midpoint to potential midpoints
        endif
    endfor

    fit straight line through midpoints and potential midpoints

    if potential curve segment belongs to symmetry then
        add potential curve segment to curve segments of symmetry
        add potential midpoints to symmetry
        all curve segments involved in potential midpoints are new potential curve
            segments, if not already in symmetry
    endif
```

Figure 9: Outline of stage II, completing a partial skewed symmetry.

ically met due to the construction of the algorithm. Figure 9 summarizes this second stage.

3 Results

In this section the choices of all parameters and thresholds are discussed, and results of the proposed algorithm for both synthetic and real contours are presented.

3.1 Parameters and complete contours

The following parameters have to be determined: The width of strips for bucketing, the discretization of the generator angle, and the thresholds for the normalized approximation error and the percentage of non matched buckets.

To achieve good accuracy, the strip width w should be as small as possible, while retaining the connectivity of the resulting buckets within a curve segment. To guarantee this condition, the strip width has to be at least the maximal distance between connected curve points. The width is set to $w = 2$ pixel to allow for eight connected edgels plus a small additional margin.

The discretization of the generator angle should be fine for accurate results, but coarse to reduce the amount of computation. Since a two pass approach is adopted, the resolution in the first pass only needs to be fine enough, to find a local minimum of the approximation error below the threshold in the vicinity of each correct generator angle. If the discretization is too coarse, this minimum could be missed. For all examined contours, a resolution of three degrees is sufficient to find these minima. In the second pass, generator angles

$$\mu \in \{\lambda + \Delta\mu \mid \Delta\mu = \pm 0.5°, \pm 1.0°, \pm 1.5°\}$$

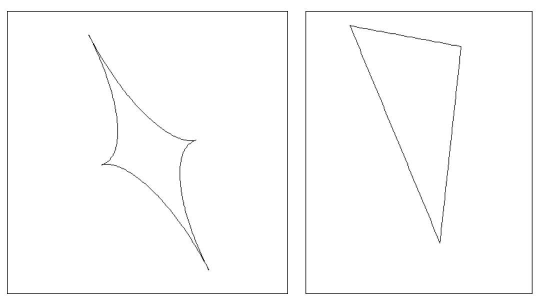
are tested, resulting in a final resolution of half a degree.

The choice of $\vartheta_{\text{nonmatched}}$ determines the amount of occlusion tolerated. The threshold $\vartheta_{\text{nonmatched}} = 0.25$ is used throughout the experiments, allowing for an occlusion of a quarter of the contour. A much larger value does not seem to be reasonable, since assuming a very large amount of occlusion almost any contour can be interpreted as being skewed symmetric (see last section). If desired, a lower value is of course possible. It should however tolerate a small amount of non matched buckets or curve points, which may occur due to noise.

The most important decision is of course the value of ϑ_{error}, since it is the main factor to distinguish between symmetric and non symmetric contours. The threshold was determined evaluating the results of stage I, setting $\vartheta_{\text{error}} = \infty$. For the experiments two synthetic (figure 10) and six real contours (figure 12) were used. The curve segments are detected with a Canny operator ([Can83]) for a single scale ($\sigma = 2$) followed by a simple local linking procedure. Table 1 lists the normalized approximation error for the correct generator angles and the minimal error for the wrong generator angles.

contour	normalized approximation error		time [s]
	correct λ	wrong λ	
ellipses	0.0007 0.0008 0.0010 0.0018	——	2.9
triangle	0.0004 0.0006 0.0015	≥ 0.1216	1.5
heart	0.0036	≥ 0.0218	1.7
club	0.0049	≥ 0.0491	1.9
letter A	0.0032	≥ 0.0745	1.6
spade	0.0044	≥ 0.0316	1.8
letter K	——	≥ 0.0370	1.8
letter J	——	≥ 0.0379	1.5

Table 1: Normalized approximation error for correct generator angles and minimal error for wrong generator angles applying stage I using $\vartheta_{\text{error}} = \infty$; the last column lists the computation time for $\vartheta_{\text{error}} = 0.01$ on a Sparc station 2 without I/O.



(a) (b)

Figure 10: Two synthetic contours exhibiting skewed symmetry.

These results demonstrate that the normalized approximation errors of correct and wrong angles are well separated and this error can be employed as a robust criterion. The maximal error for a correct symmetry is 0.0049, while the minimal error for a wrong angle is 0.0218. For the remaining experiments $\vartheta_{\text{error}} = 0.01$ is used as the threshold. For a contour with a mean distance of 40 pixels between symmetric curve points (typical for the examples), this allows a mean distance of 0.4 pixels between midpoints and the symmetry axis.

In figure 13 and 14 some of the resulting skewed symmetries are depicted. All four

Figure 11: Die image, originally published in [Pon90].

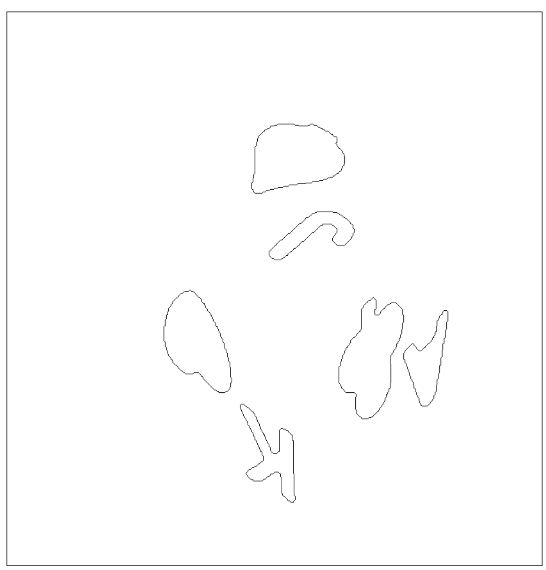


Figure 12: The six contours of the die used for the experiments.

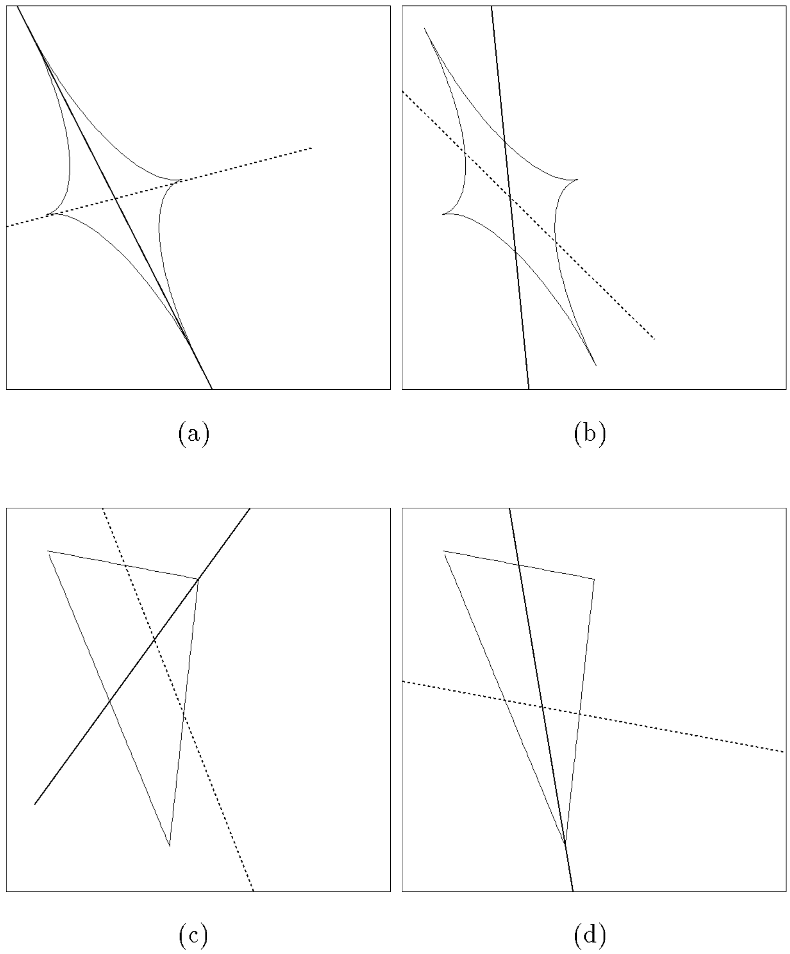


Figure 13: Symmetry axes and generators found by the stage I for the complete contours. Two of the four solutions for the ellipses (a) and (b), two of the three solutions for the triangle (c) and (d).

symmetry axes of the synthetic contour composed of ellipses, and all three of the triangle have been found. Two examples for each contour are presented in figure 13. The axes of the symmetric contours of the die are shown in figure 14. Thus, all skewed symmetries are detected with good accuracy with stage I of the algorithm for the complete unfragmented contours, while no false symmetries are found for the non symmetric contours (letter K and J).

Table 1 additionally lists the computation times for the implementation in Sather, an object oriented language derived from Eiffel, on a Sparc station 2. They are very favorable compared to other local methods. Ponce reports runtimes of about 30

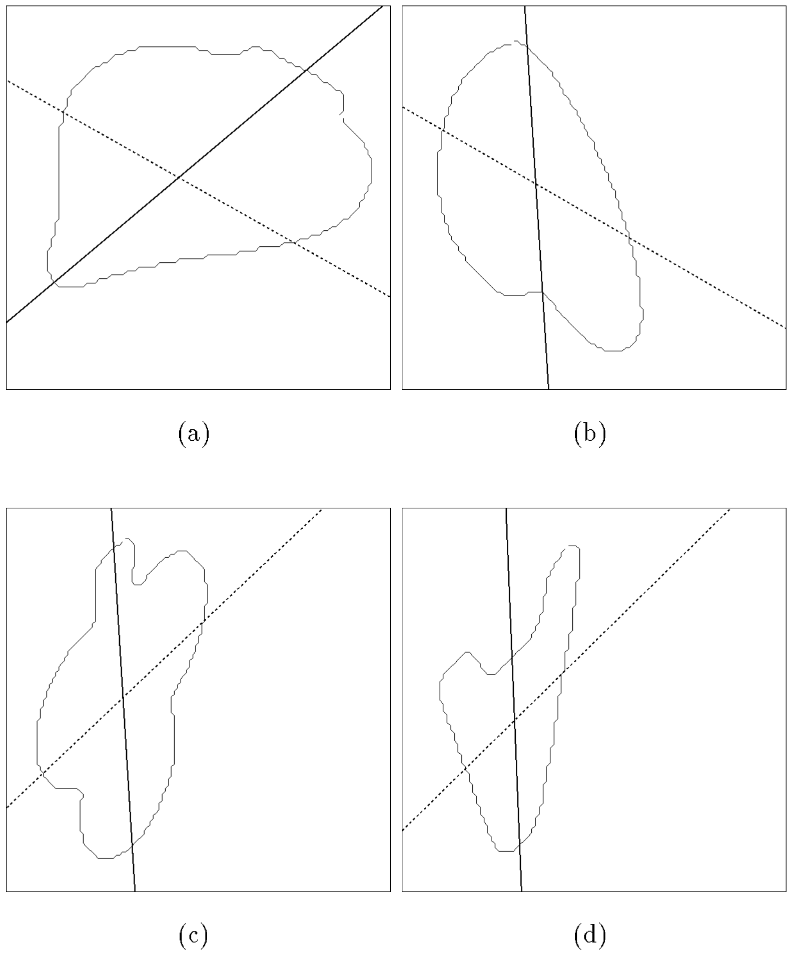


Figure 14: Symmetry axis and generators found by the stage I for the complete skewed symmetric contours of the die.

seconds in [Pon90], Gross and Bouldt in [GB91] a computation time between 30 and 60 seconds for the same contours of the die on a Sun 4 Sparc station. These results are hence a magnitude faster, even taking a performance difference of a factor 2 or 2.5 into account. As a local method is still slower then the global method in [GB91], where the runtimes, again for the same contours, are between 0.44 and 0.95 seconds.

3.2 Occlusion

In this section occlusion of varying degree is introduced to the same six contours. A randomly selected part of the contour covering 5, 10, 15, and 20 percent of the total length is removed from the curve segments and the algorithm applied. Table 2 lists the number of correctly found skewed symmetry axes for the partially occluded contours.

	Percentage of occlusion			
	5 %	10 %	15 %	20 %
ellipses	4	3	2	1
triangle	3	2	2	2
heart	1	1	1	1
club	1	1	1	0
letter A	1	1	1	0
spade	1	1	1	1

Table 2: Number of correct skewed symmetries found for each contour with varying degree of occlusion.

When there is little occlusion all symmetries are detected. If the amount of occlusion increases, the number of missed symmetries also rises. Altogether, three quarter of the symmetries are detected, half of the missing symmetries occur for an occlusion of 20 percent. Some of the detected symmetries are shown in figure 15.

Figure 16 depicts three typical situations where the symmetry could not be recovered. In figure 16 (a) due to the occlusion wrong symmetric curve points are found and although the symmetry axis is found, the normalized approximation error is above the threshold and the symmetry therefore discarded.

Figure 16 (b) shows the situation in which the correct skewed symmetry could not be detected for the letter A. Due to the occlusion wrong unique bucket pairs are found in stage I in the upper part of the letter and therefore the symmetry cannot be recovered. The same effect is responsible for most of the symmetries not detected for the ellipses.

The situation for the missing symmetry of the club is more complicated. Here, the symmetry almost has been reported, but the normalized approximation error of 0.01013 is slightly above the threshold. Two effects are responsible for this result. First, a part of the contour was occluded, which could contribute to a reliable approximation of the symmetry axis due to the large distance between symmetric curve points. On the other hand, due to noise in the image the upper part of the remaining curve segment deviates from an ideal skewed symmetry. This increases the

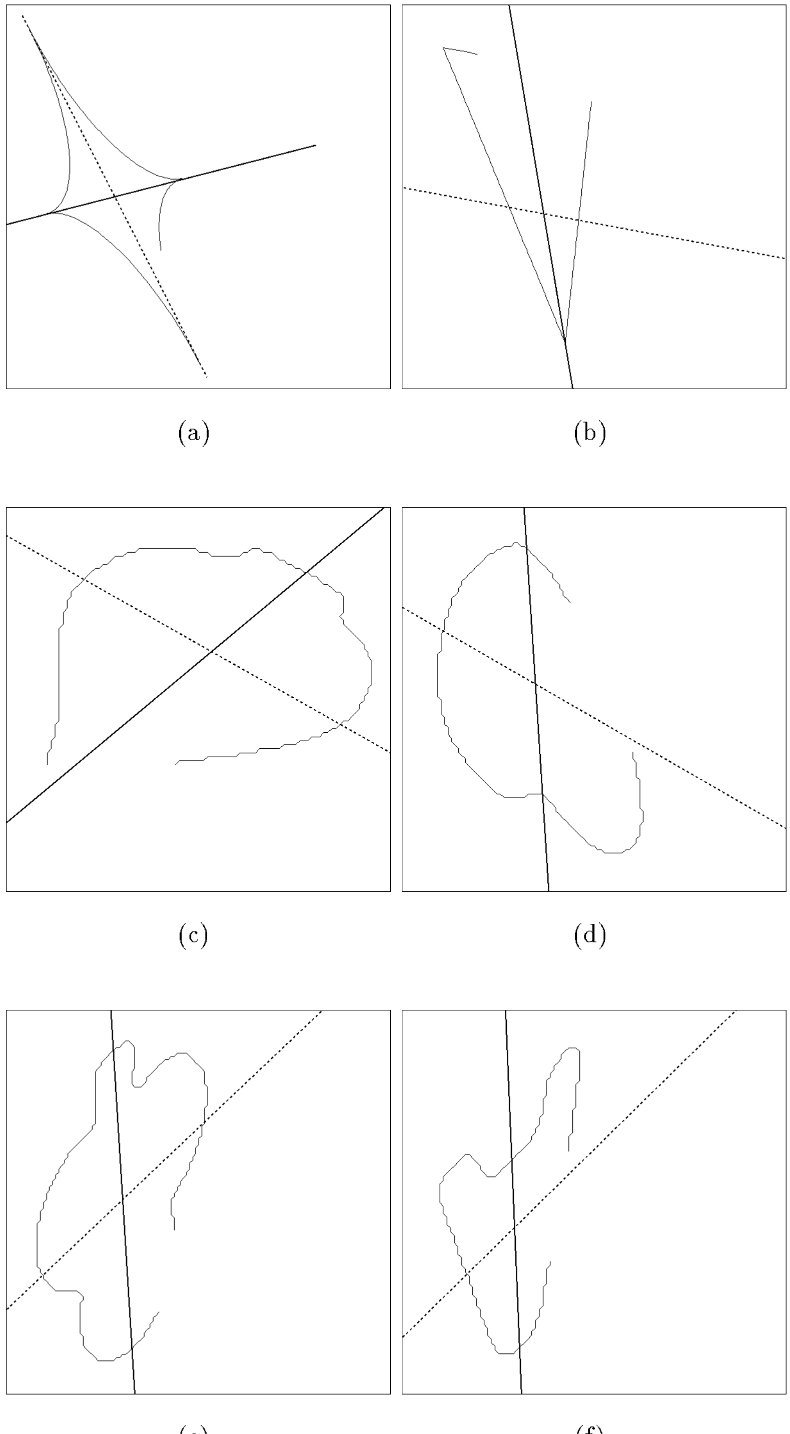
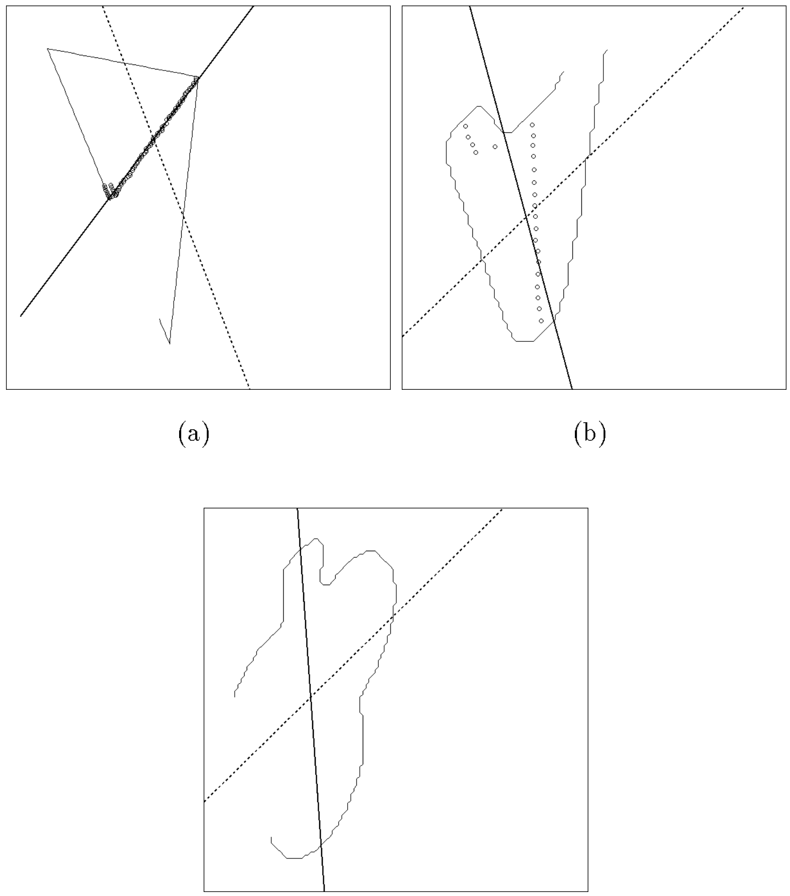


Figure 15: One correctly detected skewed symmetry for each of the six contours with different degree of occlusion.

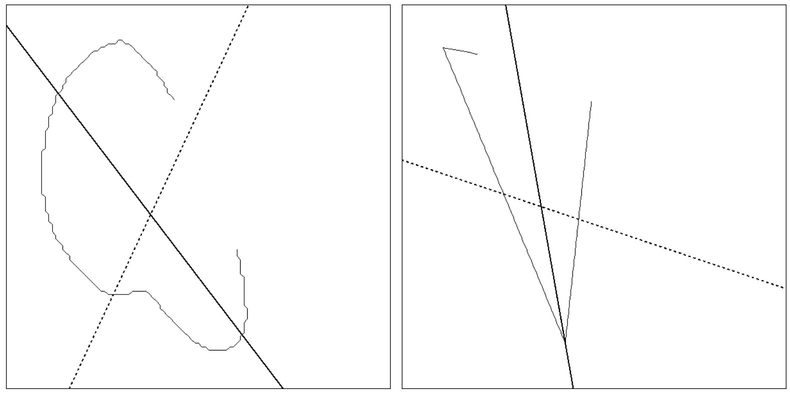


(a) (b) (c)

Figure 16: Missed skewed symmetries for occluded contours: (a) Wrong curve points have been matched where the correctly matching curve points are occluded. The midpoints are shown as circles. (b) Due to occlusion wrong unique bucket pairs where found, again the midpoints are shown as circles. (c) Non ideal curve points and occluded curve points with large distance to symmetric curve point result in a normalized approximation error slightly above threshold.

normalized approximation error particularly severe due to the small distance between matching curve points.

In two cases, not included in table 2, additional symmetry axes are found. However, figure 17 shows that these are actually reasonable symmetries for the given occlusion. In each case, a completion of the partial contour yielding a very accurate skewed



(a) (b)

Figure 17: Additionally found skewed symmetries. The partial occluded contours can be completed to yield skewed symmetries.

symmetry is possible.

In summary, the algorithm exhibits good ability to tolerate partial occlusion when detecting skewed symmetries. In some cases, especially for a large amount of occlusion, the occluded part of the contour results in wrong matches of curve points. This in turn yields either wrong symmetry axes or approximation errors above the threshold. In only two cases an additional symmetry was detected. In both cases, however, these skewed symmetries are totally reasonable for the remaining, not occluded contour.

3.3 Fragmented contours

Finally, the capability of stage II to cope with fragmented contours is investigated. For this, the same symmetric contours used in former experiments are randomly split into three or four curve segments. Then for each contour the algorithm is applied, starting stage I with all possible combinations of one or two curve segments of that contour. In practice, once a set of curve segments forming a symmetry is detected, curve segments of this set would not be further tested, so not all combinations would be processed. The mean runtime for each combination of curve segments is 3.1 seconds, again using a Sparc station 2.

To evaluate the performance, three aspects have to be discussed: First and most important, are the fragmented contours grouped together and a complete skewed symmetry detected? For all six contours and both fragmentations this was the case.

Figure 18 depicts some examples of these grouped contours with the detected axes. In the case of fragmentation into four curve segments, one of the skew ambiguous axes have not been found for the triangle and the ellipses. The reason for this is, that the correct symmetry axis could not be hypothesized from the fragmented curve segments with sufficient precision to enable the completion.

Second, the accuracy of the resulting symmetry parameters has to be considered. The generator angles of the detected symmetries deviate in some cases up to two degrees from the previous results. The reason for this is, that the decision for the generator angle is made at the end of stage I before the completion. Since only a part of the contour is known at this point, some inaccuracy may be introduced. The main objective for stage II is its ability to complete fragmented contours. If in addition a higher accuracy of the symmetry parameters is desired, some final adjustments of the generator angle, and therefore also the symmetry axis, may be done.

Third, additional skewed symmetries are found. In all cases, these symmetries are made up of only some of the curve segments of the complete contour. In figure 19 a typical example is depicted. It illustrates, that such a skewed symmetry is in fact due to symmetry of a part of the contour, so this is a perfectly reasonable result. However, since the algorithm also finds a skewed symmetry for the whole contour, it is easy to detect that it is only a local symmetry. It is accomplished by checking whether there exists a superset of curve segments also forming a skewed symmetry. If this is the case, the former symmetry is considered to be a local symmetry of a part of the contour.

4 Conclusion

We presented a new local method to detect skewed symmetries based on curve segments. No contour approximation, tangent or curvature estimation is necessary. Key elements are the detection of unique matches for a given generator angle and the exploitation of connectivity among curve points. Hypotheses of symmetry axes for fragmented contours are completed in a second stage of the algorithm, thus performing a perceptual organization task.

Results for synthetic and real contours were presented. For complete contours the algorithm finds all skewed symmetries, as well as in the case of a small amount of occlusion. If the amount of occlusion increases, some symmetry axes were missed and two additional ones were found, the later however being consistent with the occluded contours. The method was able to group all fragmented curve segments, utilizing the symmetry property. Additionally, local symmetries of the contours were found. The runtime of the algorithm is an order of magnitude shorter than other reported methods.

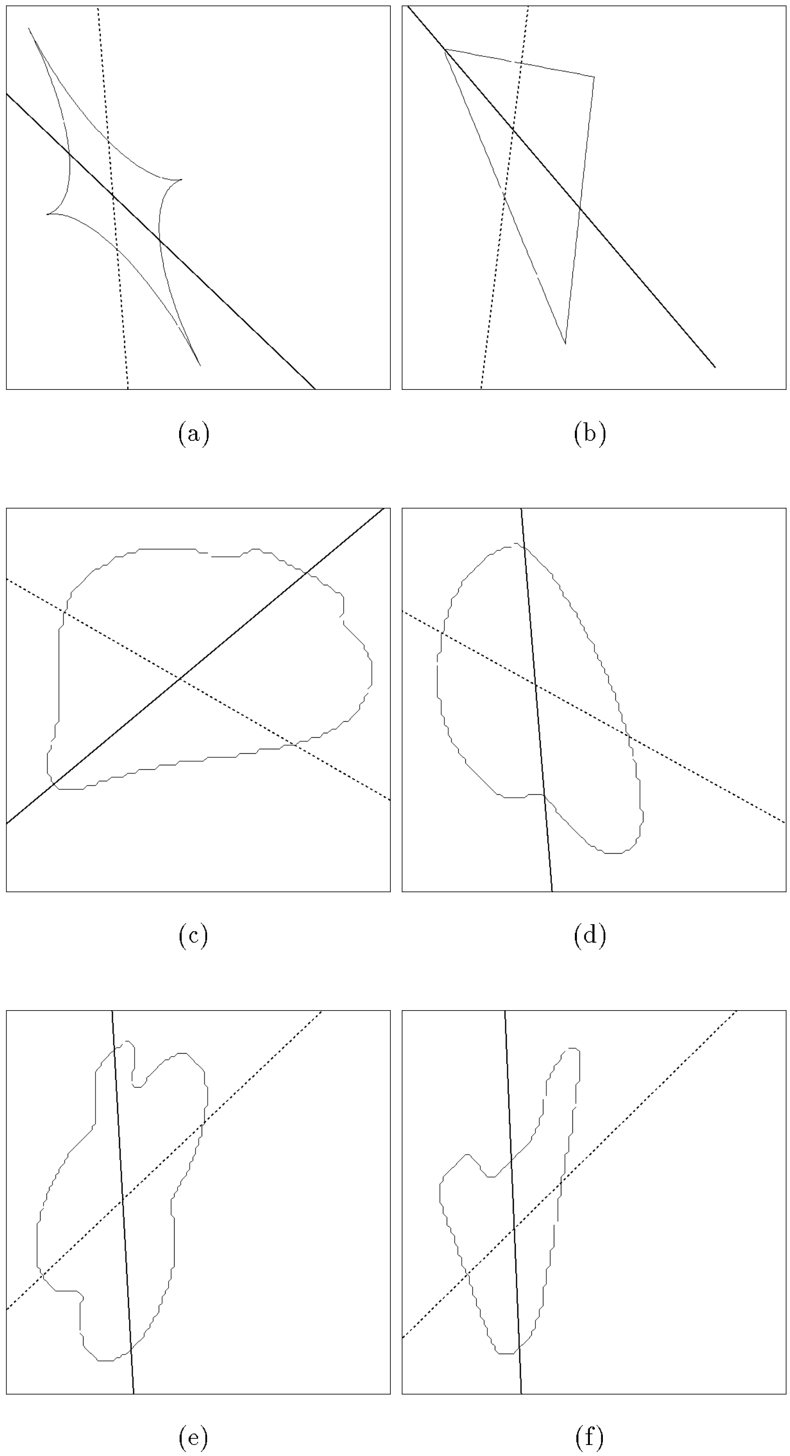


Figure 18: Examples of correctly detected skewed symmetries for each of the six contours.

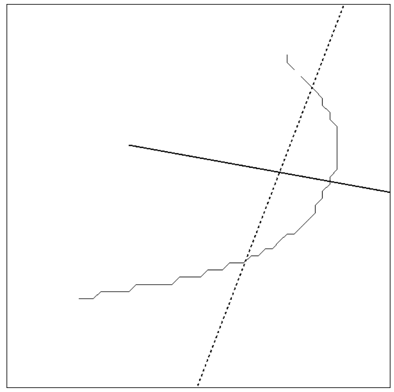


Figure 19: Local symmetry in the spade contour.

For these results, the algorithm was supplied with pairs of curve segments to start the processing. Clearly an automated method is needed, which should avoid blind search of all pairs of curve segments, for example utilizing the distance between curve segments. In this work, only the symmetry property was employed to group curve segments. For a more general system for perceptual grouping, further properties, like proximity, curvilinearity, closure, parallelism, have to be incorporated.

Acknowledgments

I would like to thank the "Förderverein" and ICSI for making my stay at the ICSI possible. Also, I thank Steve Omohundro and Michael Gilge for their comments on drafts of this paper and Ari Gross for supplying me with the die image.

References

[BY84] Michael Brady and Alan Yuille. An extremum principle for shape from contour. *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 6(3):288–301, 1984.

[Can83] J.F. Canny. Finding edges and lines in images. Dissertation, MIT Artificial Intelligence Laboratory, Cambridge, 1983.

[Fri86] Stuart A. Friedberg. Finding axes of skewed symmetry. *Computer Vision, Graphics and Image Processing (CVGIP)*, 34:138–155, 1986.

[GB90] Ari D. Gross and Terrance Boult. Analyzing skewd symmetries. Technical report, TR CUCS-64-90, Columbia University, Department of Computer Science, 1990.

[GB91] Ari D. Gross and Terrance Boult. Syman: a symmetry analyzer. In *Proc. Computer Vision and Pattern Recognition*, 744–746, Lahaina, Maui, 1991.

[Kan81] T. Kanade. Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence*, 17:409–460, 1981.

[Low85] David G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Press, Boston, 1985.

[NB77] Ramakant Nevatia and Thomas O. Binford. Description and recognition of curved objects. *Artificial Intelligence*, 8:77–98, 1977.

[OAT88] Weon Geun Oh, Minoru Asada, and Saburo Tsuji. Model-based matching using skewed symmetry information. *Proc. 9. Int. Conf. on Pattern Recognition (ICPR)*, 1043–1045, 1988.

[Pon90] Jean Ponce. On charaterizing ribbons and finding skewed symmetries. *Computer Vision, Graphics and Image Processing (CVGIP)*, 52:328–340, 1990.

[SMM90] Phillipe Saint-Marc and Gerard Medioni. B-spline contour representation and symmetry detection. IRIS Report 262, Institut for Robotics and Intelligent Systems, Univ. of Southern California, 1990.

[UN88] Fatih Ulupinar and Ramakant Nevatia. Using symmetries for analysis of shape from contour. In *Proc. 2. Int. Conf. on Computer Vision*, 414–426, Tampa, 1988.

[Yue90] S.Y.K. Yuen. Shape from contour using symmetries. *Proc. First European Conf. on Computer Vision*, 437–450, 1990.