Ellipse detection using efficient grouping of arc segments [1)]

Michael Zillich (zillich@acin.tuwien.ac.at)
Institute of Automation and Control,Vienna Univ. of Technology
Jiri Matas (matas@cmp.felk.cvut.cz)
Center for Machine Perception, Czech Technical University

Abstract:
We present a fast and robust algorithm for ellipse detection. The algorithm works on edge segmented images and makes explicit use of edge connectivity information. It is based on a hypothesise and verify method. Ellipse hypotheses are formed of groups of arc segments. Efficient grouping of arc segments based on tangent intersections is used to significantly reduce the exponentially large number of possible groupings of arcs. The search for consistent groupings thus becomes a linear problem. Standard techniques are then used to fit ellipses into groups of arcs.

1 Introduction

Rotationally symmetric objects are common in many man made environments like home and office scenarios. Such objects with a circular or elliptical cross-section give rise to elliptical features in the image. Elliptical features are highly non-accidental and their detection in the image is therefore a strong, distinctive cue of the presence of a circular or elliptical structure in the 3D world. In contrast straight lines are weaker cues as they are generally more abundant in the 3D world and often rather stem from cast shadows or linear texture features than actual 3D structure. Furthermore ellipses are valuable features for 3D pose estimation.

Ellipse detection has received considerable interest since at least [5] but remains an open problem. The problem of robustly fitting an ellipse into a set of points can be considered solved [1, 8, 4]. Of course the problem remains how to get to these points from a general set of image edges.

The ellipse detection problem can be posed very generally as follows:
Given a set of 2D points, find subsets of points that support ellipses and estimate ellipse parameters from these subsets using some fitness function.

[1)] This work was supported by EU project IST-2001-32184 ActIPret.

In this formulation the input information available in the image is treated as a set of points. This situation is suited for parameter space clustering approaches like (randomised) Hough transform, although the high dimensionality of the parameter space in the case of ellipses poses a significant problem. Some methods try to reduce the dimension of the paramter space by first estimating ellipse centers or angles, leaving a lower dimensional parameter space.

[7] propose an algorithm which is explicitely general and independent of the domain (optical flow, intensity, 2D, 3D) and type of input data (sparse, dense) as well as type of model (straight line, ellipse, plane, sphere, cylinder) and fitting algorithm (least squares, robust) to fit parametric models. They employ a hypothesise and verify method. The first step, exploration, is based on a random sampling of points lying on boundary elements. In order to cope with the huge size of the search space exploration is biased by recency (recently used data are less likely to be selected). The problem of a stopping condition though remains: When do we have enough random samples? The risk of missing a hypothesis vs. accepting too many hypotheses (and thus long computation time) becomes apparent when the images contain significant amount of background clutter.

Treating boundary edgels as single points however neglects the fact that edgels are actually connected. This leads to an unnecessary increase of search space size as vital relation information is thrown away. If we are interested in processing perspective images of natural scenes, ellipses will be detected in the output produced by an edge detector and thus connectivity information is at hand.

[6] segment lists of connected edgels into straight lines and elliptical arcs. Their algorithm also joins adjacent segments to form possibly larger segments. They do not however group segments possibly belonging to the same ellipse if they are lying further apart. [3] deal with forming possible pairs of disconnected arcs. But between-list-clustering (merging of elliptical arcs from different edgel lists) suffers from combinatorial explosion.

Edge extraction can never be expected to be perfect, so we rarely get complete ellipses, but rather short arcs with gaps in between. Moreover part of an ellipse might be occluded and there will be many distracting edges from backgroud clutter.

So given a set of extracted image edges the problem is now to identify those groups of disconneted edges that together form an ellipse. Although using edge segments instead of single edge points reduces the number of groupable elements, the problem still is the huge number of possible combinations of edges. [2] use genetic algorithms to deal with this huge search space. However their selection of arcs from generic edges is rather ad hoc. Furthermore the search is still heuristic and makes no explicit use of the underlying structure of the problem, namely that arcs beloning to the same ellipse form a convex group and neighbouring arcs are usually close to each other.

In the approach proposed we aim at the very common situation, where connectivity information is available but but it is not sufficient to establish reliable ellipse hypotheses. Typically edges are broken due to poor contrast or occlusions leading to elliptic edge segments rather than whole ellipses. The approach too is based on a hypothesise and verify paradigm. Ellipse hypotheses are formed from groups of arcs and the hypotheses are verified based on support. Our approach seeks to overcome the problem of combinatorial explosion of arc grouping by explicitely forming neighbourhood relations between arcs in linear time and using these relations to form meaningful groups of arcs for further ellipse fitting. Robust standard techniques are then used to fit ellipses to selected groups of arcs.

2 Overview

Our method consists of three stages.

Arc selection In the first stage we take the output of a Canny edge detector and fit circular arcs into the edge segments. Fitting circles instead of ellipses is more stable and faster and locally approximates ellipses well. Arc selection is desribed in detail in section 3.

Grouping In the second stage we form ellipse hypotheses from groups of arcs. Here we have the problem of an exponentially large number of groupings. To determine which group of arcs could possibly form an ellipse we use neighbourship relations between pairs of arcs. We intersect the tangents at arc endpoints and only group arcs that are connected via an intersection. This results in a significant reduction of hypotheses. Section 4 explains details and shows the benefit of this method.

Ellipse selection In the third stage we verify the ellipse hypotheses by fitting an ellipse into the arcs of each group. Relative support (number of supporting pixels / circumference) is used to rank hypotheses and make a final selection. See section 5 for details.

Note that due to the reduction of possible arc groupings in stage 2 we don't need to draw (random) samples but can list all grouping hypotheses at once that could form an ellipse. This discriminates our approach from others who try to optimally sample a large search space. We use the structure of the underlying problem (how arcs forming an ellipse are related to each other) to pose a much simpler search problem. The resulting small search space then does not require sampling.

3 Arc Selection

The first step in our algorithm is a Canny edge segmentation of the image. The output is a set of edge segments. Quite often edges are broken due to shadows or poor contrast. So we cannot expect to get full ellipses here but only parts of ellipses. Of the many edge segments in the image we first have to identify those that can be part of an ellipse. This would require fitting of an ellipse into a possibly short arc segment. Fitting of an ellipse into a set of points covering only a small angle however is very unstable. We thus choose to fit a circle, which locally approximates an ellipse well and is much faster and more stable to fit. Note that fitting a circle might actually break up a perfect elliptic arc into several circular arcs. That does no harm as arcs are later combined anyway to form groups of arcs. One might argue that destroying information already at hand only to recover it again later is stupid. It shows however that trying to extract too much information locally will often lead to suboptimal decisions (like unstable ellipse fits). More global methods on the next higher level usually can make better decisions.

From now on we treat each edge segment as an arc with some radius and opening angle. We can discard arcs with too small radius (e.g. ≤ 5 pixels). Otherwise we would regard each speckle as an ellipse. A radius larger than the image diagonal too makes no sense and can be discarded. A radius of infinity would be a straight line.

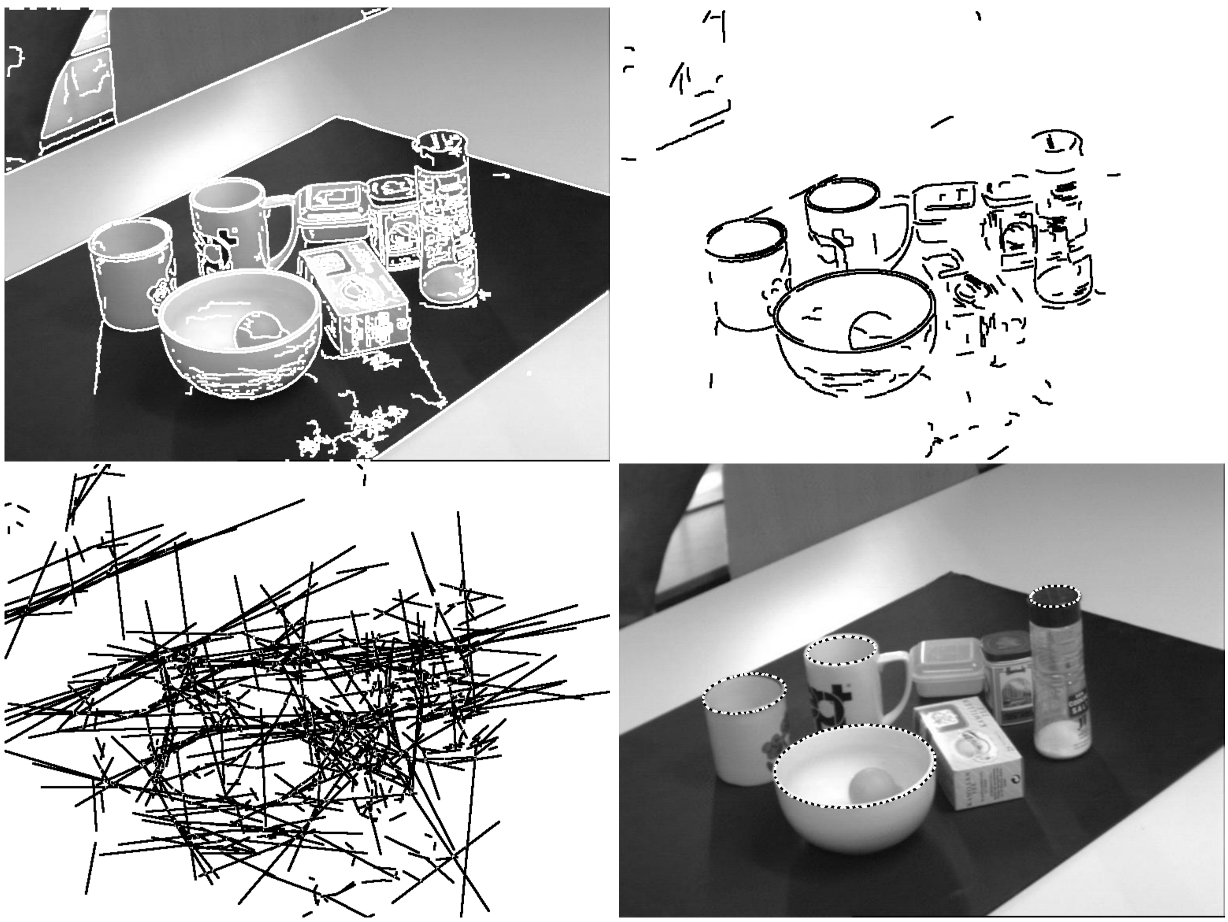

Figure 1: Kitchen scene: Canny edges, extracted arcs, arc tangents and intersections, final ellipses

4 Grouping

Now given a set of single arcs the next step is to form groups of arcs into which we can finally fit ellipses. We require that the arcs in a group together span at least π as the ellipse fit would be unstable otherwise. We further require that all arcs in a group are pairwise convex, i.e. all points of one arc must lie on the inner half plane of the other arc. All groups not fulfilling these requirements are rejected.

We use intersections of tangents at both arc ends to efficiently find all other arcs that one arc can group with. In a separate tangent image we draw the tangents at both arc end points. This requires only very simple functions per pixel: Check if at current pixel there is already a tangent pixel and if so record an intersection. Let l be the image diagonal and n the number of arcs. Then there are at most $2ln$ pixel operations required to draw all tangents and thus find all intersections. This is linear in the number of arcs as $2l$ is a constant.
We actually draw each tangent only up to a length of r, the radius of the arc. This avoids distant arcs being grouped. And it shows that in the output of the edge detector gaps between arcs belonging to the same ellipse are usually much smaller than r.

For an average image (see figure 1) with $n = 317$ arcs each tangent intersects on average with 2 other tangents. Thus one arc can only group with 4 other arcs as opposed to originally $n - 1 = 316$. Thus the originally quadratic problem of grouping pairs of arcs becomes on average a linear problem. The gain in speed becomes even more apparent when grouping not only pairs but triples or generally k-tuples. The number of possible groupings of k arcs out of n is reduced from $O(n^k)$ to $O(n4^{k-1})$ and is still linear in the total number of arcs in the image n.

5 Ellipse Selection

Once groups of arcs are formed, a standard ellipse fitting algorithm [1] implemented in the Intel OpenCV library is applied to all pixels of these arcs. The resulting ellipses are ranked according to their relative support (number of supporting pixels / circumference).
Ellipse hypotheses are then pruned. Each lower ranked ellipse sharing an arc with a higher ranked ellipse is discarded as each arc can only be part of one ellipse. This reduces the number of ellipse hypotheses from e.g. 115 to 28 in the kitchen example.
From the remaining set of ellipse hypotheses only those with a support significantly higher than average are selected as final "good" ellipses.

6 Results and Outlook

Figure 1 shows the main processing steps on a 768×576 image of a kitchen scene. Arc selection takes about $10ms$. All consistent groupings of arcs are found in about $20ms$. Fitting ellipses takes another $300ms$ and selection of good ellipses another $150ms$. All ellipses are found in a total processing time of $480ms$. Tests were perfomed on an Athlon XP1880+ @1.53 GHz.

In our work we proposed an ellipse detection algorithm designed to work on edge-segmented images of real world scences. It is based on a hypothesise and verify paradigm where our main contribution is the use of connectivity information of edgels as well as neighbourhood relations bewteen edge segments to significantly reduce the number of possible ellipse hypotheses. This enables a complete instead of a random search for hypotheses and still works in real time.

Further work will go into exploring whether similar techniques can be used to find other image primitives like rectangles or even more general perceptual groupings.

References

[1] Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.

[2] Tsuyoshi Kawaguchi and Ryo-Ichi Nagata. Ellipse detection using a genetic algorithm. In *Fourteenth International Conference on Pattern Recognition*, volume 1, pages 141–145, 1998.

[3] Ji Qiang and R.M. Haralick. A statistically efficient method for ellipse detection. In *1999 International Conference on Image Processing (ICIP 99)*, volume 2, pages 730–734, 1999.

[4] Paul L. Rosin. Assessing error of fit functions for ellipses. *Graphical models and image processing: GMIP*, 58(5):494–502, 1996.

[5] Paul L. Rosin and G.A.W. West. Segmentation of edges into arcs and lines. *Image and Vision Computing*, 7:109–114, 1989.

[6] Paul L. Rosin and G.A.W. West. Non-parametric segmentation of curves into variuos representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1140–1153, 1995.

[7] Markus Stricker and Ales Leonardis. Exsel++: A general framework to extract parametric models. In *Computer Analysis of Images and Patterns*, pages 90–97, 1995.

[8] Zhengyou Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. Technical Report RR-2676, INRIA, 1997.