Theoretical and Applied Informatics
ISSN 1896–5334
Vol.24 (2012), no. 2
pp. 151–158
DOI: 10.2478/v10179-012-0007-2

Adapting the RANSAC algorithm to detect 2nd-degree manifolds in 2D and 3D

LESZEK LUCHOWSKI

Institute of Theoretical and Applied Informatics
Polish Academy of Sciences
44-100 Gliwice, ul. Bałtycka 5, Poland
e-mail: leszek.luchowski@iitis.pl

Received 24 April 2012, Revised 18 June 2012, Accepted 26 June 2012.

Abstract: The RAndom SAmple Consensus algorithm is adapted to sets of points detected in 2D and 3D images in order to identify conic lines and quadric surfaces, respectively. A first-order approximation of point-to-conic (quadric) distance is used as compliance criterion. Experimental results are presented.

Keywords: point cloud segmentation, second-order surface detection

1. Introduction

The segmentation of a 2D image may result in an unstructured set of 2D points. Similar sets in 3D, known as point clouds, are also commonly obtained as raw data from 3D scanners. We shall now consider an algorithm to discover, in such a set, the presence of a subset included in a second-degree manifold, i.e. a conic (in 2D) or a quadric (in 3D).

The setting for which this algorithm is designed is such that:

- the inlier points (those that represent a conic or a quadric) may meet the equation of the manifold with some error, due to imperfect imaging/segmentation, and possibly to the imperfect shape of the physical object;
- there may be more than one conic (quadric) in the image;
- there may be many outlier points, possibly more than inliers, in a given image.

One generic technique for this kind of problem is the RANSAC (RANdom SAmple Consensus) algorithm [3].

2. A summary of the RANSAC algorithm

Starting with the following prerequisites:

- a set S of data points (such as observation vectors);
- a parametric model formulation;
- a procedure G to identify the model parameters from a smaller set of points;
- an error measure estimating how far a point diverges from the model

the algorithm proceeds as follows:

1. Draw a random sample s from S, consisting of a sufficient number of points to feed into G. These points become the initial ıhypothetical inliers.
2. determine $G(s)$ as an initial estimate of the model being sought;
3. For all other points in S, verify how well they fit $G(s)$. Add to s the ones which diverge from it by less than a preset threshold η_1, thus forming an enlarged sample s^*.
4. Reestimate the model as $G(s^*)$. If the number of points in s^* is at least equal to a target value N_{min}, and they fit the reestimated model with an error that is smaller than another preset threshold η_8, accept $G(s^*)$ as the final model.
5. Otherwise, restart from Step 1.

3. Application

The data set is a collection of 2D or 3D points represented by their Euclidean coordinates. The model to be identified will be represented by a matrix M or M ($3 * 3$ of $4 * 4$, respectively), which describes it via the manifold equation

$$Q(\vec{X}) = \vec{X}^T M \vec{X} = 0 \tag{1}$$

(for conics), or

$$Q(\vec{X}) = \vec{X}^T M \vec{X} = 0 \tag{2}$$

(for quadrics). For most of this paper, only the case of the quadrics will be explicitly discussed, as the formulas are the same except for dimensionality.

3.1. The point-to-model distance

The RANSAC algorithm needs a measure of disparity between a point and the model. Although the left-hand side of the manifold equation is supposed to be zero for a perfect fit, and will be nonzero otherwise, its absolute value (sometimes called the *algebraic distance*) is not a good measure of the Euclidean distance between the point and the quadric.

On the other hand, determining the Euclidean distance between a point and a quadric is computationally expensive; we have therefore chosen its approximation, namely, the algebraic distance divided by its gradient at the point under consideration, known as the Sampson error [1]:

$$d_E(\vec{X}, M) \approx d_S(\vec{X}, M) = \frac{Q_M(\vec{X})}{|\nabla Q_M(\vec{X})|}$$

3.2. the model identification algorithm

Following the same approximation of the point-to-quadric distance, given point sets were approximated by quadrics using an iterative, gradient-weighted eigenvector approach [2] based on a minimum of 10 points (5 points for conics):

1. Create an N-element weight vector W filled with ones (N is the number of points);
2. Form matrix A :

$$A = \begin{bmatrix} W_1[1 & x_1 & y_1 & z_1 & x_1^2 & y_1^2 & z_1^2 & x_1y_1 & x_1z_1 & y_1z_1] \\ ... & & ... & & ... & & ... & & & ... \\ W_N[1 & x_N & y_N & z_N & x_N^2 & y_N^2 & z_N^2 & x_Ny_N & x_Nz_N & y_Nz_N] \end{bmatrix},$$

   where x_i, y_i, z_i are coordinates of the $i-th$ data point,
3. Decompose A into singular vectors:

$$A = USV^* \tag{3}$$

   Where

   - U is a matrix whose columns are left singular vectors of A,
   - S is a diagonal matrix containing, on the diagonal, the singular values of A,
   - V is a matrix whose columns are right singular vectors of A,
   - V^* is its conjugate transpose.

4. Take the right singular vector (column of V) corresponding to the smallest (in absolute value) singular value; use this vector as an approximation of the polynomial coefficients P of the quadric:

$$\begin{aligned} Q \approx [\ 1 \quad x \quad y \quad z \quad x^2 \quad y^2 \quad z^2 \quad xy \quad xz \quad yz\ ]P = \\ = P_1 + P_2x + P_3y + P_4z + P_5x^2 + P_6y^2 + P_7z^2 + \\ + P_8xy + P_9xz + P_{10}yz \end{aligned} \tag{4}$$

5. Update the weights in vector W to the reverse of the squared gradient of the approximated Q function at the corresponding sample points: $W_i = \frac{1}{\nabla(Q(\vec{X}_i))+\delta}$, where δ is an arbitrary small increment added to avoid division by zero.
6. Repeat from step 2 until convergence is achieved or a preset number of iterations is exhausted.
7. Rearrange the polynomial coefficients into matrix M:

$$\vec{M} = \begin{bmatrix} P_5 & P_8/2 & P_9/2 & P_2/2 \\ P_8/2 & P_6 & P_{10}/2 & P_3/2 \\ P_9/2 & P_{10}/2 & P_7 & P_4/2 \\ P_2/2 & P_3/2 & P_4/2 & P_1 \end{bmatrix}.$$

In the case of conics in a 2D space, the rows of A have 6 elements each:

$$W[i] * [\ 1 \quad x_i \quad y_i \quad x_i^2 \quad y_i^2 \quad x_iy_i\ ]\ ,$$

and the corresponding polynomial formula is

$$Q \approx [\ 1 \quad x \quad y \quad x^2 \quad y^2 \quad xy\ ]P = P_1 + P_2x + P_3y + P_4x^2 + P_5y^2 + P_6xy,$$

with

$$\vec{M} = \begin{bmatrix} P_4 & P_6/2 & P_2/2 \\ P_6/2 & P_5 & P_3/2 \\ P_2/2 & P_3/2 & P_1 \end{bmatrix}$$

4. Finding multiple quadrics

For the case where the cloud of points may contain more than one quadric, three approaches were tested:

1. Removing the inliers of a found quadric from the cloud before applying the algorithm again;

2. Re-applying the algorithm to the same full point cloud, but requiring new quadrics to be sufficiently different from those already found;
3. As above, but requiring the *points* of each successive quadric to be sufficiently far from the already identified quadrics.

The first approach yielded multiple similar quadrics approximating the same surface, with minor differences which, locally, amounted to a small shift perpendicular to the surface. This was due to noise in the scan data, which gave the surfaces an apparent nonzero "thickness". Increasing η_1 to allow for this led to another problem – the detection of suprious quadrics intersecting the point cloud along approximately its greatest dimensions, as the $2\eta_1$ wide belt along the line of intersection provided them with a sufficient number of points to meet the N_{min} target. Often, the quadrics were planes or such low-curvature forms that could be considered planes at the scale of the cloud.

The second approach required a measure of how different two quadrics are. As the quadrics in the experiments were represented by homogenous vectors of polynomial coefficients, a distance measure was chosen based on the absolute value of their correlation:

$$D(P_1, P_2) = 1 - \left| \frac{\Sigma_{i=1}^{10} P_{1,i} P_{2,i}}{\sqrt{\Sigma_{i=1}^{10} P_{1,i}^2 P_{2,i}^2}} \right| \tag{5}$$

The criterion was added to Step 2 of the RANSAC algorithm: if $G(s)$ was closer to any previously found quadric than a preset minimal distance, it was rejected and the algorithm returned to Step 1 to draw a new sample. The formula proved to be inadequate for our purposes – quadrics approximating very similar sets of points were sometimes more "distant" (under this measure) than quadrics representing clearly distinct objects. Search for other options ran into the difficulty of defining "distance" in the projective hyperspace of quadrics in a way that would be meaningful to the limited scene being imaged in the Euclidean 3D space.

The third approach, while computationally intensive, proved successful and correctly identified quadric shapes in a physical scene.

5. Test results

The algorithm was applied to a cloud of points representing the 3D scan of three artefacts: two ceramic mugs and a 3D-printed model of a saddle surface (hyperbolic paraboloid). The raw set of points is shown in Figure 1.

Based on experiments, the parameters were set to:

$\eta_1 = \eta_8 = 0.3$ to accomodate the imperfections of both the physical objects and the scanning process;

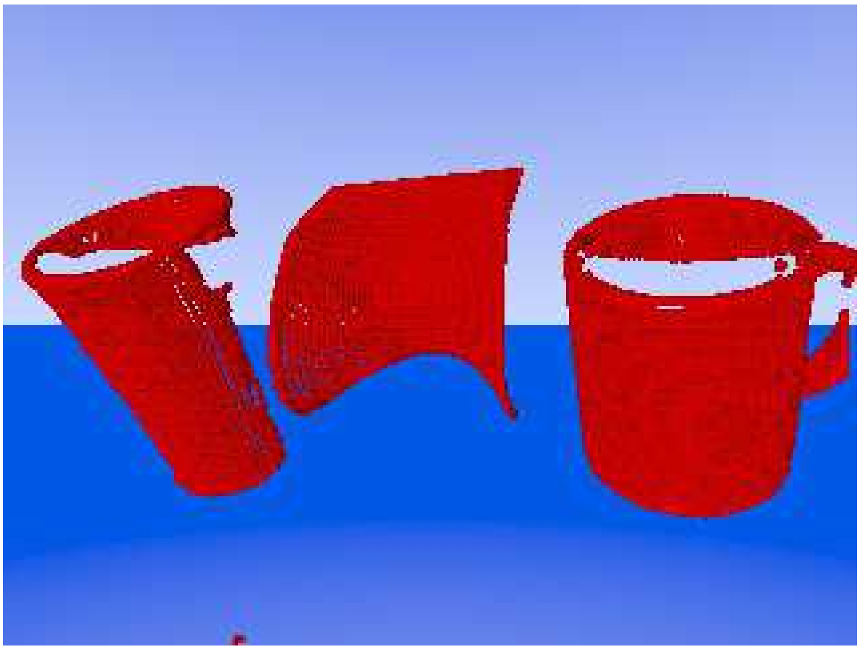

Fig. 1. Scanned set of points

$N_{min} = 1200$ to reject incidental alignments of points, especially cases where a narrow streak of points on a physical surface happens to lie on the intersection of that surface with some otherwise unrelated quadric;

$D_{min} = 10$ was chosen as a value comparable to the physical distance between the objects.

As can be seen in Figures 2 and 3, the result was a set of three quadrics approximating the outer surfaces of the mugs as well as the model quadric.

Acknowledgment

This work was supported in part by the Polish National Science Centre (NCN) under project N516 369136.

References

1. Paul D. Sampson: *Fitting conic sections to a very scattered data: An iterative refinement of the bookstein algorithm*, Computer Graphics and Image Processing **18** (1982), no. 1, 97–108.

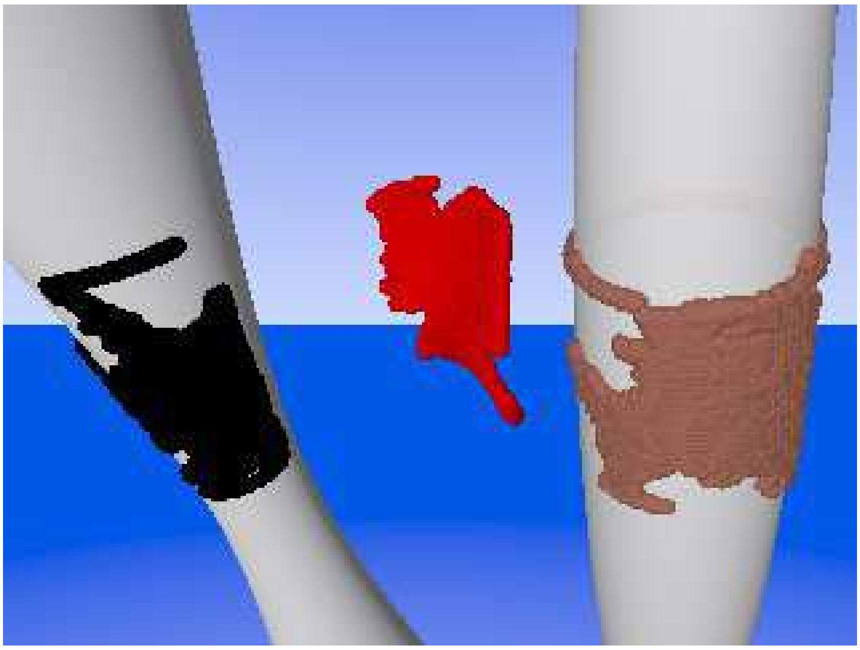

Fig. 2. Quadrics approximating the mugs

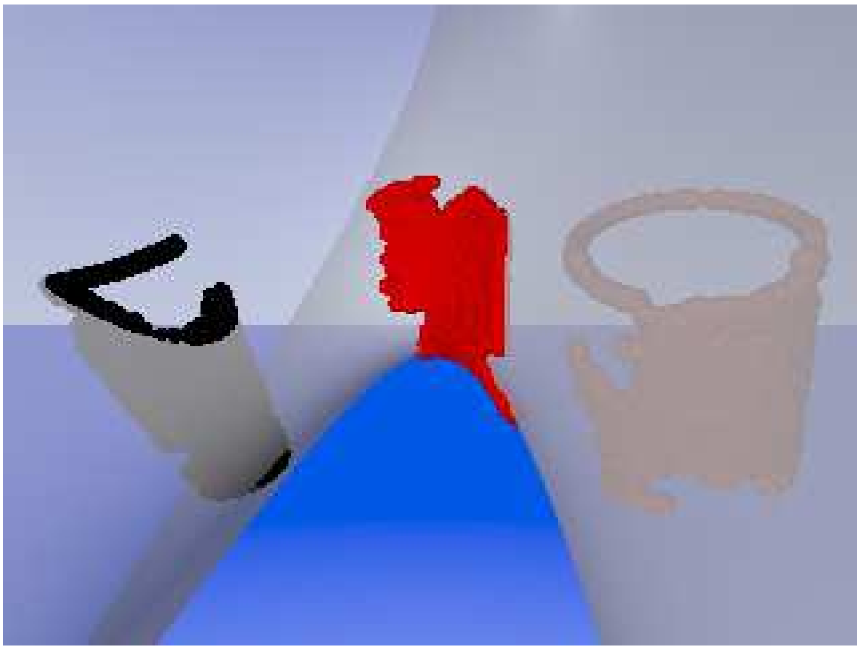

Fig. 3. Quadric approximating the saddle model

2. Zhengyou Zhang: *Parameter estimation techniques: a tutorial with application to conic fitting*, Image Vision Comput. **15** (1997), no. 1, 59–76.

3. Martin A. Fischler and Robert C. Bolles: *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Comm. ACM 24 (1981), no. 6, 381-395.

Przystosowanie algorytmu RANSAC do wykrywania rozmaitości drugiego rzędu w 2D i 3D

Streszczenie

Algorytm RANdom SAmple Consensus, który pozwala wykrywać regularne struktury w heterogenicznych zbiorach danych, zastosowano do zbiorów punktów w obrazach 2D i 3D, aby zidentyfikować, odpowiednio, krzywe stożkowe lub powierzchnie drugiego rzędu (kwadryki). Jako kryterium zgodności poszczególnego punktu z równaniem opisującym krzywą bądź powierzchnię wykorzystano przybliżenie pierwszego rzędu odległości od stożkowej (kwadryki), zaproponowane przez Sampsona. Dla przypadku obrazu zawierającego wiele krzywych bądź kwadryk zastosowano warunek wystarczającej średniej odległości punktów kolejnych wykrywanych struktur od struktur już wykrytych. Przedstawiono wyniki doświadczalne.