Table of Contents

Robust Methods

Signal noise poses problems in a variety of algorithms and techniques in computer vision. Images are notoriously noisy and some operations amplify this problem. We examine a few of the better known robust computational models which are often used in computer vision.

The Random Sample and Consensus (RanSaC) Method

This iterative method is a form of parametric matching that can be applied to data sets containing outliers. The algorithm is non-deterministic as its behavior is not repeatable.

A particularly difficult numerical problem is that of fitting lines through points in the presence of outliers. A least-squares solution to this problem will generally be poor. Assuming that the data set contains both inliers and outliers. RanSaC can produce a model computed only from the inlier points, provided that the probability of choosing only inliers in the selection of data is sufficiently high. In a line fitting example we would start by generating lines from pairs of points. Since only two points are needed to define a line, we set $n=2$, the minimum number of data elements required to obtain a sample model. Suppose ω is the ratio of inliers to points in the data set, and k is the chosen number of iterations. Then, the probability of one sample being correct is given by:

$$1-\left(1-\omega^n\right)^k$$

Given a sample model for a line, we now must determine how good this sample is, by counting the number of points that are on the sample line, with tolerance ϵ . Finally we choose the sample line that has the largest number of points consonant with it. The RanSaC algorithm may be summarized as:

- Inputs:
  - n the smallest number of points required to estimate parameters
  - k the number of iterations required

  - ϵ the threshold used to identify a point that fits well
  - d the number of fitting points required to assert that a model fits well
- Until k iterations have occurred,
  - Draw a sample of n points from the data, uniformly and at random
  - Fit parameters to that set of n points
  - For each data point outside this sample, test the distance from the point to the line against ϵ, and count the point as an inlier if distance is smaller than ϵ
  - If there are d or more points that are inliers, then this is a good fit and refit the line this time using all the points in the data set
- Use the best fit from this collection, using the fitting error as a criterion

Parameters ϵ and d are determined from specific requirements related to the application. The number of iterations k however can be determined from a theoretical result. Let p be the probability that the algorithm in some iteration selects only inliers from the data set when it chooses the n points to estimate the model parameters. Hence, p gives the probability that the algorithm produces a useful result.

Assuming that the n points needed for estimating the model are selected independently, then ω^n is the probability that the n points used to estimate the model are inliers and $(1-\omega^n)$ is the probability that at least one the n points is an outlier implying that an incorrect model will be estimated from these points. This probability to the power of k is the probability that the algorithm never selects a set of n points which are all inliers and it thus must be equal to $1-p$. Thus we can write:

$$1-p = (1-\omega^n)^k$$

Solving for k yields:

$$k = \frac{\log(1-p)}{\log(1-\omega^n)}$$

The standard deviation of the number of iterations is given by:

$$\sigma_k = \frac{\sqrt{1-\omega^n}}{\omega^n}$$

An advantage of RanSaC is its ability to do robust estimation of the model parameters. However there is no upper bound on the time it takes to compute these parameters.

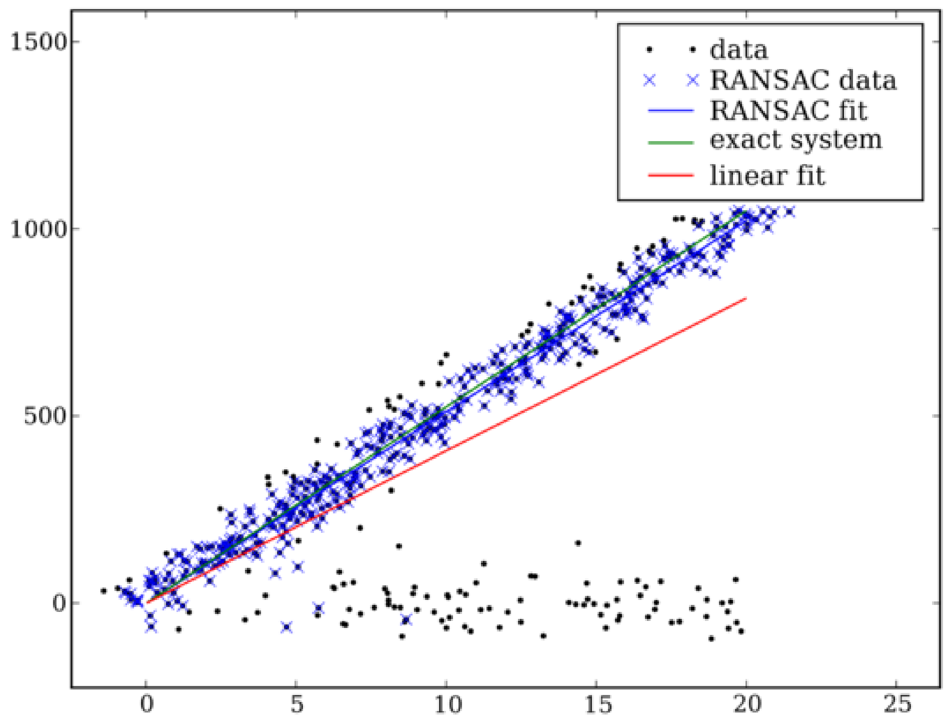

Illustration 1: RanSaC performance on noisy data

The Expectation Maximization (EM) Algorithm

EM is frequently used for data clustering in machine learning and computer vision. It is also employed in medical image reconstruction, especially in positron emission tomography.

Suppose we have multivariate observations that come from a set of probability density functions in the form $(\vec{x}_1, \ldots, \vec{x}_n)$ where $\vec{x}_i$ is a vector of p dimensions. Each $\vec{x}_i$ is to be viewed as arising from a super-population G which is a mixture of g populations $G_1, \ldots, G_g$ in some proportions $\pi_1, \ldots, \pi_g$ respectively, where:

$$\sum_{i=1}^{g} \pi_i = 1$$

with each $\pi_i \geq 0$. The probability density function (p.d.f.) of an observation $\vec{x}$ in

G can therefore be represented in the finite mixture form:

$$f(\vec{x};\vec{\theta}) = \sum_{i=1}^{g} \pi_i f_i(\vec{x};\vec{\theta})$$

where $f_i(\vec{x};\vec{\theta})$ is the p.d.f. of G_i , and $\vec{\theta}$ denotes the vector of all the unknown parameters associated with the parametric forms of the g component probability density functions.

In the particular case of multivariate normal component densities, $\vec{\theta}$ consists of the elements of the mean vectors $\vec{u}_i$ and the distinct elements of the covariance matrices Σ_i for $i=1,\ldots,g$. The vector $\vec{\phi}=(\vec{\pi}^T,\vec{\theta}^T)^T$ of all the unknown parameters belongs to some parameter space $\vec{\Omega}$.

The likelihood equation for $\vec{\phi}$:

$$\frac{\partial L(\vec{\phi})}{\partial \vec{\phi}} = \vec{0}$$

can be so manipulated that the likelihood estimate $\hat{\phi}$ satisfies:

$$\hat{\pi}_i = \sum_{j=1}^{n} \frac{\hat{\tau}_{ij}}{n}$$

and

$$\sum_{i=1}^{g} \sum_{j=1}^{n} \hat{\tau}_{ij} \frac{\partial \log f_i(\vec{x}_j;\hat{\theta}_i)}{\partial \hat{\theta}} = \vec{0}$$

These equations suggest an iterative computation for their solution and can be solved by a direct application of an Expectation-Maximization algorithm.

Let the vector of indicator variables $\vec{z}_j=(z_{1j},\ldots,z_{gj})^T$ be defined by:

$$z_{ij}=\begin{cases} 1 & \text{if} \quad \vec{x}_j \in G_i \\ 0 & \text{if} \quad \vec{x}_j \notin G_i \end{cases}$$

where $\vec{z}_1,\ldots,\vec{z}_n$ are independently and identically distributed. It is further assumed that $\vec{x}_1,\ldots,\vec{x}_n$, given $\vec{z}_1,\ldots,\vec{z}_n$ are conditionally independent and $\vec{x}_j$ given $\vec{z}_j$ has log density:

$$\sum_{i=1}^{g} z_{ij} \log f_i(\vec{x}_j;\theta) \qquad j=1,\ldots,n$$

Hence, the likelihood for the complete data $\vec{X}=(\vec{x}_1{}^T,\ldots,\vec{x}_n{}^T)^T$ and $\vec{Z}=(\vec{z}_1{}^T,\ldots,\vec{z}_n{}^T)^T$ is given by:

$$L_C(\vec{\phi})=\sum_{i=1}^{g}\sum_{j=1}^{n} z_{ij}\{\log\pi_i+\log f_i(\vec{x}_j;\theta)\}$$

The EM algorithm is applied to the mixture model by treating $\vec{Z}$ as missing data.

Using some initial value $\vec{\phi}^{(0)}$ for $\vec{\phi}$, the Expectation step requires the computation of the expectation of the complete data log likelihood $L_C(\vec{\phi})$, conditional on the observed data and the initial fit $\vec{\phi}^{(0)}$ for $\vec{\phi}$:

$$Q(\vec{\phi},\vec{\phi}^{(0)}) \;=\; E\{L_C(\vec{\phi})|\vec{X};\vec{\phi}^{(0)}\}$$

This step is performed by replacing each indicator variable z_{ij} by its expectation conditional on $\vec{x}_j$, given by:

$$E\{z_{ij}|\vec{x}_j;\vec{\phi}^{(0)}\} \;=\; \tau_i(\vec{x}_j;\vec{\phi}^{(0)}) \qquad i=1,\ldots,g$$

That is, z_{ij} is replaced by the initial estimate of the posterior probability that $\vec{x}_j$ belongs to component distribution G_i .

The Maximization step is to choose the value of $\vec{\phi}^{(1)}$ that maximizes and therefore leads to replacing $\hat{\tau}_{ij}$ with $\tau_i(\vec{x}_j;\vec{\phi}^{(0)})$. The solution to the M step often exists in closed form, as is the case with mixtures of normal component distributions. The E and M steps are alternated repeatedly. Since the log likelihood satisfies $L(\vec{\phi}^{(k+1)})\geq L(\vec{\phi}^{(k)})$, then convergence is assured.

Mixture of Normals

In the case when the component densities are normal:

$$\vec{x}_j\sim N(\vec{u}_i,\Sigma_i) \quad\in\quad G_i$$

with probability π_i , the vector θ of parameters associated with the component densities contains the elements of the mean vectors $\vec{u}_{1,}\ldots,\vec{u}_g$ and the distinct elements of the covariance matrices $\Sigma_{1,}\ldots,\Sigma_g$. It follows from under the normality assumption that the likelihood estimates of component distribution parameters satisfy:

$$\hat{\pi}_i \;=\; \sum_{j=1}^{n}\frac{\hat{\tau}_{ij}}{n}$$

$$\hat{u}_i \;=\; \sum_{j=1}^{n}\frac{\hat{\tau}_{ij}\vec{x}_j}{n\hat{\pi}_i}$$

$$\hat{\Sigma}_i \;=\; \sum_{j=1}^{n}\frac{\hat{\tau}_{ij}(\vec{x}_j-\hat{u}_i)(\vec{x}_j-\hat{u}_i)^T}{n\hat{\pi}_i}$$

where the posterior probability that $\vec{x}_j$ belongs to G_i is given by:

$$\hat{\tau}_{ij} = \frac{\pi_i |\Sigma_i|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\vec{x}_j - \vec{u}_i)^T \Sigma_i^{-1} (\vec{x}_j - \vec{u}_i)\right\}}{\sum_t \pi_t |\Sigma_t|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\vec{x}_j - \vec{u}_t)^T \Sigma_t^{-1} (\vec{x}_j - \vec{u}_t)\right\}}$$

Normal Homoscedastic Components

In the case when the component densities have equal covariance matrices (homoscedasticity):

$$\vec{x}_j \sim N(\vec{u}_i, \Sigma)$$

the maximum likelihood estimator of the matrix specializes to:

$$\hat{\Sigma} = \sum_{i=1}^{g} \sum_{j=1}^{n} \frac{\hat{\tau}_{ij}(\vec{x}_j - \hat{u}_i)(\vec{x}_j - \hat{u}_i)^T}{n}$$

and

$$\hat{\tau}_{ij} = \frac{\pi_i |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\vec{x}_j - \vec{u}_i)^T \Sigma^{-1} (\vec{x}_j - \vec{u}_i)\right\}}{\sum_t \pi_t |\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\vec{x}_j - \vec{u}_t)^T \Sigma^{-1} (\vec{x}_j - \vec{u}_t)\right\}}$$

The problem of estimating how many component distributions are present in a super-population is difficult and still open. Therefore, the applicability of the algorithm is limited to cases when g is known a-priori.

The Hough Transform

The principle behind the Hough transform is to map a problem into a simple extremum in the parameter space of a feature (a line, a circle, etc).

The Hough transform is best exemplified by specializing it to the detection of lines in images. The stages of the transform may be summarized as follows:

- Any line $y = mx + n$ is identified by a unique pair of parameters (m, n) , which is a point in the parameter space.
- Any point $\vec{p} = (x, y)^T$ in the image corresponds to a line $n = x(-m) + y$ in parameter space. As the elements of (m, n) vary, we obtain the set of all lines passing through $\vec{p}$.

- We divide the parameter space (m,n) into a grid of cells, and set a counter $c(m,n)$ to zero.
- Assume the image contains a single line (m',n') formed by points $\vec{p}_{1,}\cdots,\vec{p}_n$.
- For each image point $\vec{p}_i$, we increment the counters on the corresponding line in parameter space.
- All the parameter space lines $L_{1,}\ldots,L_n$ associated with points $\vec{p}_{1,}\ldots,\vec{p}_n$ go through $(m'n')$ which in turns implies that $c(m',n')=n$ whereas any other counter on $L_{1,}\ldots,L_n$ is 1.
- Hence, the line is identified by the peak in the counter function $c(m,n)$ in parameter space.

For the detection of multiple lines, we simply find the local maxima of the counter function $c(m,n)$. Non-linear contours yield spreads of low random counter values throughout the parameter space. They can be eliminated by an experimentally determined threshold on $c(m,n)$.

It is worthwhile to consider the fact that a finite parameter space is computationally easier to work with. Towards this end, we may represent the space in polar coordinates $\rho=x\cos\theta+y\sin\theta$. The variation of ρ and θ are therefore finite.

Hough Transform Algorithm for Lines

- Consider E an $m\times n$ binary image in which each element $E(i,j)=1$ if it is an edge pixel, and 0 otherwise (in other words, E is the output from Canny's edge detector)
- Let ρ_d and θ_d be arrays containing the discretized intervals of the (ρ,θ) parameter space $(\rho\in[0,\sqrt{m^2+n^2}],\ \theta\in[0,\pi])$ with (r,t) their respective number of elements (these are the discretization parameters).
- Discretize the parameter spaces of ρ and θ using sampling steps $(\delta\rho,\delta\theta)$
- Let $A(r,t)$ be an array of integer counters initialized to zero.
- For each pixel $E(i,j)=1$ and for $h=1,\ldots,t$
    - Let $\rho=i\cos\theta_d(h)+j\sin\theta_d(h)$.
    - Find k , the index of the element of ρ_d closest to ρ .
    - Increment $A(k,h)$ by one.

- Find all local maxima (k_p, h_p) such that $A(k,h) > \tau$, a predetermined threshold

The output of this algorithm is the set of pairs $(\rho_d(k_p), \theta_d(h_p))$ describing the lines detected in the binary image E in polar coordinates.

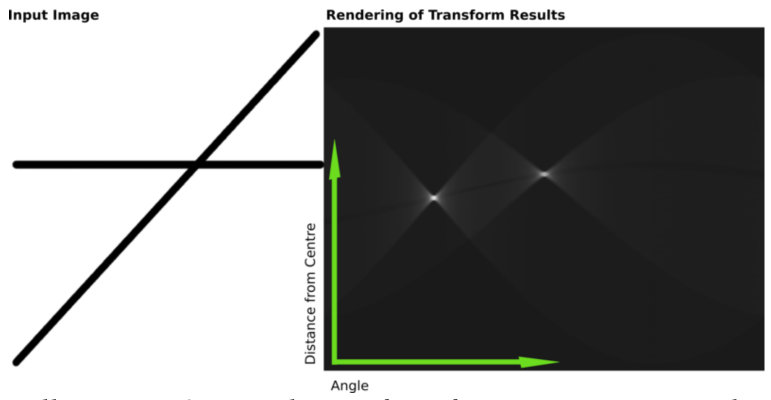

Illustration 2: Hough transform for an input image with two lines