WASP - Sensing, October 2016

SLAM, Minimal Solvers, RANSAC and systems of polynomial equations

Introduction

In the lecture I will talk about a few tools used to solve SLAM type problems, not only for vision, but also for other sensor modalities. In solving such problems there are many tools that one could use, e.g. mathematical statistics (to model and understand noise, parameter estimation, understanding the uncertainty in the estimated variables), optimization (non-linear optimization, convex optimization, l^2, l^1, l^∞ optimization, truncated l^2, l^1, l^∞, huber norm etc), algorithms, feature detection, geometry, algebra, algebraic geometry. In this two hour lecture I have picked a few topics,

- RANSAC
- Minimal Solvers
- Solving Systems of Polynomial Equations

These topics are chained together in the following way. RANSAC is an important part of SLAM system. In RANSAC you need to solve minimal problems. These problems can often be converted to systems of polynomial equations. Thus if you know how to solve systems of polynomial equations, you can make minimal solvers, which are used in RANSAC algorithms which are used for solving SLAM systems. Again this is only a part of the toolbox needed, but still a good starting point for further studies.

The idea is that with these tools one can produce simple SLAM systems. One goal for this lecture (and for the homework after the lecture) is that you should

- know a bit more about systems of polynomial equations,
- be able to convert a system of polynomial equations to an eigenvalue problem.
- be able to write solvers for a few minimal problems,
- learn about the RANSAC algorithm for parameter estimation in the presence of outliers. Have used the ransac algorithm for fitting a line to a set of points, fitting a circle to a set of points, fitting an essential matrix to a set of point correspondences, fitting three anchor positions to a set of TOA measurement triplets,
- be able to write a first version of a SLAM system for two images,
- be able to write a first version of a SLAM system for Time-Of-Arrival measurements to three anchors in a plane,

Extension to more views is outside the scope of this two-hour lecture, but is central in e.g. a course in computer vision, cf. http://www.maths.lth.se/course/datorseende/2016/.

Pre-requisited, not covered in lecture

Polynomials: Read http://en.wikipedia.org/wiki/Polynomial if you need to fresh up on what a polynomial is. At some point you need to realize that they are both a vector space ($p_1 + p_2$ is a polynomial and λp_1 is a polynomial) and a **ring** ($p_1 + p_2$ is a polynomial and $p_1 p_2$ is a polynomial). Read http://en.wikipedia.org/wiki/Polynomial_ring if needed.

Linear vector spaces: Fresh up on linear algebra. Read http:\\immersivemath.com if you need to refresh some of the topics from this. A linear vector space consists of things that you can add and that you can multiply with a scalar. If you understand that that the set of polynomials in one variable of degree 2 is a vector space, I thing you will do fine.

Ring, Ideals: We will not go into this here, but a basic knowledge of abstract algebra (rings, ideals, quotient rings) are needed to understand algebraic geometry, which is the tool to understand solutions to systems of polynomial equations. Further reading: https://en.wikipedia.org/wiki/Ring_(mathematics) and https://en.wikipedia.org/wiki/Ideal_(ring_theory).

Interest point detectors: Interest point detectors or feature detectors are algorithms that find interesting points in images. This is covered in Michael Felsbergs lectures. Read also http://en.wikipedia.org/wiki/Interest_point_detection and http://en.wikipedia.org/wiki/Corner_detection.

Exercise 1 - Roots in matlab

How are roots calculated in matlab?

Read `https://se.mathworks.com/help/matlab/ref/roots.html` .

Consider the polynomial equation

$$p(x) = x^3 - 7x^2 + 14x - 8 = 0$$

Clearly every solution x to the equation, also fulfills

$$\begin{cases} x^3 &= 7x^2 - 14x + 8 \\ x^2 &= x^2 \\ x &= x \end{cases}.$$

Rewrite this as a matrix equation

$$x \begin{pmatrix} x^2 \\ x \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}}_{\mathbf{M}} \begin{pmatrix} x^2 \\ x \\ 1 \end{pmatrix}$$

for some matrix $\mathbf{M}$. What is the matrix $\mathbf{M}$?

In the reference page for `roots` in matlab, it is written 'The algorithm simply involves computing the eigenvalues of the companion matrix:'

```
A = diag(ones(n-1,1),-1);
A(1,:) = -c(2:n+1)./c(1);
eig(A)
```

Explain what c is and why the algorithm works

Try

```
c = [1 -7 14 -8];
n = 3;
A = diag(ones(n-1,1),-1);
A(1,:) = -c(2:n+1)./c(1);
[v,d] = eig(A)
```

Study v and d. Explain what they should contain? How can one find the solutions from d? How can one find the solutions from v? Why is the last row of v not all equal to one? Interpretation?

Exercise 2 - Planar trilateration

Time-of-Arrival measurements gives (after multiplication with the speed of sound or speed of light) distance measurements. Assume that there are a set of $m = 2$ anchors $\mathbf{r}_i$,

$i = 1, \ldots, m$, with $\mathbf{r}_i \in \mathbb{R}^2$. Assume that time-of-arrival measurements provide distance measurements

$$d_i = \|\mathbf{r}_i - \mathbf{s}\|_2, \tag{1}$$

from an unknown position $\mathbf{s} \in \mathbb{R}^2$. Write a minimal solver that given the two anchor positions $\mathbf{r}_1$ and $\mathbf{r}_2$ and given the two distance measurements d_1 and d_2 calculates all solutions $\mathbf{s}$ so that

$$\begin{cases} d_1 = \|\mathbf{r}_1 - \mathbf{s}\|_2, \\ d_2 = \|\mathbf{r}_2 - \mathbf{s}\|_2, \end{cases}$$

Exercise 3 - Planar trilateration 2

Here we will show how to convert the planar trilateration problem to an eigenvalue problem. This is not the most efficient way to solve this particular problem. However, it is a useful example to illustrate that a system polynomial equation can be solved in a similar technique as for `roots`, i.e.

- Mulitply the original equations with a number of monomials.
- Express the system of equations as $Cm = 0$, where C is a matrix of coefficients and m is a vector of monomials.
- Use numerical linear algebra to simplify the equations.
- Rearrange the equations to form an eigenvalue problem $\lambda \mathbf{v} = M\mathbf{v}$.
- Calcualte eigenvalues and eigenvectors.
- Extract the solutions from the eigenvalues/eigenvectors.

Read through the example and make sure you understand the different steps. What is similar to roots? Are there any differences?

Assume that the measurements are

$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \mathbf{r}_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, d_1 = 3, d_2 = 3.$$

Parametrize the unknown vector as

$$\mathbf{s} = \begin{pmatrix} x \\ y \end{pmatrix}.$$

By squaring the measurements equations we get two polynomial equations in two unknowns.

$$\begin{cases} x^2 - 2x + y^2 - 4y - 4 = 0 \\ x^2 - 6x + y^2 - 8y + 16 = 0 \end{cases}.$$

Multiplying the first equation with 1, x and y and similarily for the second equation gives

$$\begin{cases} x^2 - 2x + y^2 - 4y - 4 = 0 \\ x^3 - 2x^2 + xy^2 - 4xy - 4x = 0 \\ x^2y - 2xy + y^3 - 4y^2 - 4y = 0 \\ x^2 - 6x + y^2 - 8y + 16 = 0 \\ x^3 - 6x^2 + xy^2 - 8xy + 16x = 0 \\ x^2y - 6xy + y^3 - 8y^2 + 16y = 0 \end{cases}.$$

This can be written as

$$Cm = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & -2 & -4 & -4 \\ 1 & 0 & 1 & 0 & -2 & -4 & 0 & -4 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -2 & -4 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -6 & -8 & 16 \\ 1 & 0 & 1 & 0 & -6 & -8 & 0 & 16 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -6 & -8 & 0 & 16 & 0 \end{pmatrix}}_{C} \underbrace{\begin{pmatrix} x^3 \\ x^2y \\ xy^2 \\ y^3 \\ x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{pmatrix}}_{m} = 0$$

Gaussian Elimination gives six equations

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 16 & -81 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -26 & 11 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 4 & -20 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -5.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -6 & 5.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -5 \end{pmatrix} \underbrace{\begin{pmatrix} x^3 \\ x^2y \\ xy^2 \\ y^3 \\ x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{pmatrix}}_{m} = 0.$$

Two of these size equations are

$$\begin{cases} xy + y - 5.5 = 0 \\ x + y - 5 = 0 \end{cases}.$$

Rewriting these as

$$\begin{cases} xy = -y + 5.5 \\ x = -y + 5 \end{cases}.$$

gives the following eigenvalue problem

$$\begin{pmatrix} xy \\ x \end{pmatrix} = x \begin{pmatrix} y \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} -1 & 5.5 \\ -1 & 5 \end{pmatrix}}_{\mathbf{M}} \begin{pmatrix} y \\ 1 \end{pmatrix}.$$

The first eigenvector-eigenvalue pair is $\lambda_1 = 0.13$ and $v_1 = \begin{pmatrix} 4.87 \\ 1 \end{pmatrix}$. Thus

$$\mathbf{s} = \begin{pmatrix} 0.13 \\ 4.87 \end{pmatrix}$$

is one solution to the problem. The second eigenvector-eigenvalue pair is $\lambda_2 = 3.87$ and $v_2 = \begin{pmatrix} 1.13 \\ 1 \end{pmatrix}$. Thus

$$\mathbf{s} = \begin{pmatrix} 3.87 \\ 1.13 \end{pmatrix}$$

is one solution to the problem.

Exercise 4 - TOA SLAM (challenging exercise!)

Time-of-Arrival measurements gives (after multiplication with the speed of sound or speed of light) distance measurements. Assume that there are a set of $m = 3$ anchors $\mathbf{r}_i$, $i = 1, \ldots, m$, with $\mathbf{r}_i \in \mathbb{R}^2$. Assume that time-of-arrival measurements provide distance measurements

$$d_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|_2, \tag{2}$$

from $n = 3$ unknown positions $\mathbf{s}_j$, $j = 1, \ldots, n$, with $\mathbf{s}_j \in \mathbb{R}^2$.

Write a minimal solver that given 9 distance measurements $d_{ij}, i = 1, \ldots, m, j = 1, \ldots, n$ calculates all solutions $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$ so that

$$d_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|_2, i = 1, \ldots, m, j = 1, \ldots, n.$$

Is there enough information to solve this problem? How many solutions are there?

Exercise 5 - RANSAC - Fitting a line to a set of points

Download the github project `https://github.com/kalleastrom/ImageAnalysisExamples`

Among the files is an unfinished script `ransac_line_stub2` for fitting a line to a set of points. It loads a dataset and displays the data. There are also pre-coded functions for

- Fitting a line to a minimal set of two points.
- Calculating inliers to a line with a given threshold.
- plotting the data and the solution.

Implement the RANSAC algorithm and study how it works.

Exercise 6 - RANSAC - Fitting a circle to a set of points

The RANSAC algorithm is quite general.

Among the files is an unfinished script `ransac_circle_stub2` for fitting a circle (with known radius) to a set of points. It loads a dataset and displays the data. There are also pre-coded functions for

- Fitting a circle to a minimal set of two points.
- Calculating inliers to a circle with a given threshold.
- plotting the data and the solution.

Use your implementation of the RANSAC algorithm and make it work for the circle case to. For the circle case the minimal solver gives two solutions (not a unique solution as for the line case). How does this change the code?

Try to write this part of the code so that the identical piece of code works for both the line and the circle case.

Demo - Two view visual slam

Study the script `relative_motion_demo_short`. In the script we use RANSAC to fit an essential matrix to a set of point correspondences obtained from two views using an interest point detector.