Imperial College London

# Monocular SLAM

## Robot mapping and location using a single camera

Maurice Yap Alessandro Bonardi
Paul Larsen Abbie Howell

Tuesday 22nd March 2016

mauriceyap.co.uk/slam

# What are we talking about?



Made using **Metropolis** LaTeX theme

# What is monocular SLAM?

**SLAM = Simultaneous Localisation and Mapping (1)**

- **Mobile robot** - independent machine with sensors that moves through an environment
- **Mapping** - creating and storing a model of its environment
- **Localisation** - where the robot is in its environment

# Why is SLAM important?

# Automated Cars

A prototype of Google's self-driving car (2)

# Household Appliances

The Dyson 360 Eye, an autonomous robotic vacuum cleaner which uses SLAM techniques (3)

# Augmented Reality


A demonstration of GlorAR, an AR engine which uses SLAM (4)

# Why is monocular SLAM important?

# Why use one camera?

- Cost
- Size
- Simplicity (5)
- Mobile devices (6)
- ”Infinite” distance (7)

**Disadvantages:**

- Complex algorithms

# Algorithms and SLAM Methods

- Two different methods:
  - Direct
  - Feature based
- Multiple different algorithms
  - MonoSLAM
  - LSD-SLAM
  - PTAM

# Feature based methods

- Start by processing each image to find ”features” - usually corners
- Faster than direct methods since only uses definitely relevant data (8)
- Dense maps can be reconstructed if necessary, but requires estimation of camera location, so can be error prone (9)
- Not ideal in human environments which contain many straight lines
- Storing features has to be done in a matrix, can be very costly as features increase.

# Direct methods

- Compare entire images to each other
- More data
- Allows creation of semi dense 3D maps in real time on a typical smartphone
- 3D maps makes it more interesting for robotics and AR
- Better for larger environments
- However: doesn't handle erroneous data very well (8)
- Typically slower than feature based variants (8)

# LSD-SLAM

- Stands for Large Scale Direct Monocular SLAM (9)
- The use of direct methods make it good in large environments (10)
- Use of Direct methods also makes it good for use in urban environments (9)
- No need for specialised hardware

# Depth Map Estimation

- New keyframes made when camera out of range of existing keyframes (9)
- New keyframe's depth map initialised; points projected, spatial regularisation and outlier removal as per Engel, Sterm and Cramer's algorithm (11)
- Depth map scaled
- New keyframe used to track subsequent new frames

# Differences between LSD-SLAM and other SLAM types

- Use of Direct methods
- At its creation, LSD-SLAM was faster than other monocular SLAM algorithms, but as of 2015 it has been overtaken by new feature based methods (8)

# PTAM: Parallel Tracking and Mapping

- Tracking and mapping run in different processor threads
- Tracking thread uses motion model to estimate position from previous frame
- Projects map points onto image using estimate
- Looks for matches between projected points, starting with coarse features then refining with finer features (12)

# PTAM: Mapping

- Map consists of keyframes – snapshots taken at various points in time
- Each keyframe stores a four level pyramid of greyscale images for depth
- Initialisation from user input (keypress, smooth movement, keypress)
- Uses two keyframes to triangulate initial position
- Each point stored with reference to a keyframe, pyramid level and pixel location
- New map points created using two closest keyframes to determine pyramid level (12)

# PTAM: Mapping

- New keyframes added when following criteria are met (12):
  - Good tracking quality
  - Last keyframe >20 frames ago
  - Current frame not too close to any other keyframe (avoids corruption problem common in other SLAM algorithms)
- Mapping only performed when background processing thread has free resources – ensures constant frame rate for tracking
- In well explored environments, system corrects outliers and checks for new features (12)

# PTAM: Issues and Advantages

- Fairly powerful hardware needed; not all phones can run it
- Struggles to deal with self-occlusion (13)
- Feature based methods
- Requires initialisation by the user – issues with accuracy

- Map is not corrupted by camera being kept stationary
- Map is constantly updated and improved
- Doesn't require an existing map or any features to be inputted (12)

# What is MonoSLAM?

- First online monocular SLAM algorithm, developed by Davison in 2003 (14)
- Based on a probabilistic feature-based map
- The map keeps track of both the estimates and the uncertainty of the state of the camera and the feature (15)
- To compute them, the algorithm uses the Extended Kalman Filter, a recursive process which uses the last estimated map and the current measurement to compute the current state
- The map, representing the state of the world, is modelled as a single multivariate Gaussian distribution $\mathrm{X} \sim \mathcal{N}(\mathrm{x}, \Sigma)$ (16)

# EKF – State vector and Covariance Matrix

- The camera state is represented by (17):

$$x_c = \begin{bmatrix} \mathrm{r}^W \\ \mathrm{q}^{WR} \\ \mathrm{v}^W \\ \omega^R \end{bmatrix}$$

- The state vector (mean) $\mathrm{x}$ and the covariance matrix $\Sigma$ are defined as (17):

$$x = \begin{bmatrix} \mathrm{x}_c \\ \mathrm{y}_1 \\ \vdots \\ \mathrm{y}_N \end{bmatrix} \quad , \quad \Sigma = \begin{bmatrix} \Sigma_{\mathrm{x}_c\mathrm{x}_c} & \Sigma_{\mathrm{x}_c\mathrm{y}_1} & \dots & \Sigma_{\mathrm{x}_c\mathrm{y}_N} \\ \Sigma_{\mathrm{y}_1\mathrm{x}_c} & \Sigma_{\mathrm{y}_1\mathrm{y}_1} & \dots & \Sigma_{\mathrm{y}_1\mathrm{y}_N} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{\mathrm{y}_N\mathrm{x}_c} & \Sigma_{\mathrm{y}_N\mathrm{y}_1} & \dots & \Sigma_{\mathrm{y}_N\mathrm{y}_N} \end{bmatrix}$$

- The algorithm has $O(n^2)$ complexity
- EKF operates two iterative phases: Prediction and Update (15)

# Initialisation

- When first started the algorithm already knows the position of a few features: the scale of the map would otherwise be unknown (15)
- New features are tracked using the Shi-Tomasi algorithm, locating the best candidate in a 100x50 pixel box chosen randomly on the image (15)
- They have unknown depth, are assumed to be lying on a line between the real position and the camera, and also on a locally planar surface (15)

# Prediction Step

1. $\mathrm{x}_{pred} = f(\mathrm{u}_t, \mathrm{x}_{t-1})$
2. $\Sigma_t^{pred} = F_t \Sigma_{t-1} F_t^{\mathrm{T}} + Q_t$

- This step estimates in which state the system should be transferred to from the previous estimate if actions $u_t$ are executed (17)
  - The state is updated using $u_t$ and $\mathrm{x}_{t-1}$ through the state transition function $f$
  - The covariance matrix is also updated accordingly

# Map Update

- The expected position of a feature is computed before the measurement using the pinhole camera model as a function *h* of the predicted camera pose and $\mathrm{y}_i$ (18)
- The size of the search region is given by the innovation covariance matrix $\mathrm{S}_t = H_t \Sigma_t H_t^T + Q_t$
- The Kalman Gain is computed (18):

$$\textbf{3}. \quad K_t = \Sigma_t H_t^T S^{-1}$$

- The state vector and the covariance matrix are updated and used as the old estimation in the next EKF cycle (18):

$$\textbf{4}. \quad \mathrm{x}_t = \mathrm{x}_{pred} + K_t(z_t - h(x_{pred}))$$

$$\textbf{5}. \quad \Sigma_t = (I - K_t H_t)\Sigma_{pred}$$

# Corner detection

In computer vision, corner are the most preferred type of feature to track in an image

- It has to be as recognisable as possible
- View from different angles would result in significant changes
- Shifting a window over a corner would result in significant changes

One of the most used corner detector is Harris's, published in 1988 (19), based on the existing Moravec's algorithm (18, 19)

# Moravec’s Corner Detector

- Tests every pixel in an image to find a corner by considering how similar a patch (window) centred on the pixel is to the surrounding patches (20)
- The sum of squared differences (SSD) measures this similarity
- PROBLEM: the directions considered are only multiples of 45°
- SOLUTION: Harris-Stephens Corner Detector

Given a patch P, the SSD over it compare to a patch shifted by (x,y) along the (u,v) axis of the image is (20):

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

*w is the window function, I is the gray-scale value at a given position*

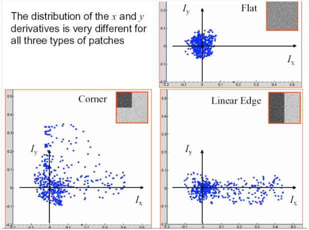
# Harris Corner Detector

- To achieve an isotropic detector (18-20):
  - The window function delimits a circular window and has a Gaussian profile
  - The algorithm uses partial derivatives of $I$ over the directione of the shifting
- Substituting $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$

$$E(u, v) \approx (u, v) \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\Rightarrow (u, v)\mathcal{M} \begin{pmatrix} u \\ v \end{pmatrix}$$

# Harris Corner Detector




Derivative graphs for three different types of patches (21)

- A corner will have a large variation of E in the direction of the vector (x,y)
- An interest point will therefore have two large eigenvalues
  - If $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$ the pixel has no features
  - If $\lambda_1 \approx 0$ and $\lambda_2 >> 0$, it is an edge
  - If $\lambda_1$ and $\lambda_2 >> 0$, it is a corner (18)

# Harris vs. Shi-Tomasi

- The computation of the eigenvalues is expensive
- A score R is given to each window (18, 20)

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2) = det(\mathcal{M}) - ktrace(\mathcal{M})^2$$

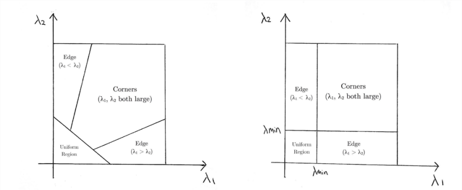
*Where k is determined empirically (0.04-0.15)*

# Harris vs. Shi-Tomasi

In 1994 Shi and Tomasi modified the scoring function R

$$R = min(\lambda_1, \lambda_2)$$

This means that only when both eigenvalues are greater than a threshold $\lambda_{min}$ the feature is considered a corner (20)




(original image)

# Applications of Monocular SLAM

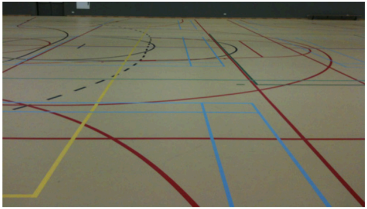
Examples:

- In drones:
    - Mapping with a Parrot AR drone (22)
    - Autonomous Search and Rescue Drones (23)
- In robotics:
    - Dyson 360 Eye (3)
    - Suggestions at uses in humanoid household robots (15, 16)

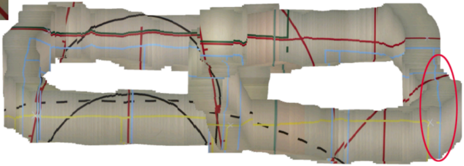
# In drones: Mapping with the Parrot AR drone (22)

- Mainly proof of concept uses
- Show accurate mapping can be done with very low res cameras
- Very dependent on the features of the environment mapped: texture Poor vs texture Rich

# Parrot AR drone in a feature poor environment

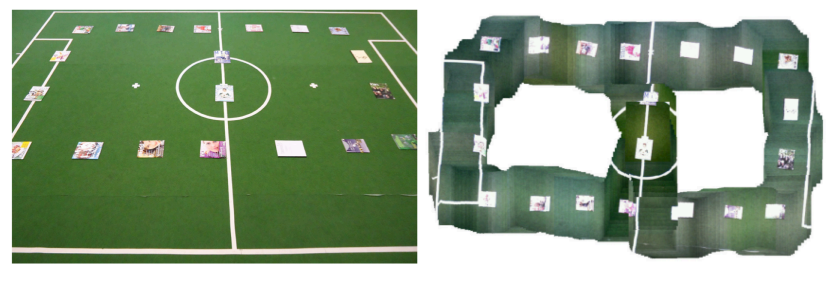


Note: the red ellipse shows an area where the drone had issues determining its velocity.



(22)

# Parrot AR drone in a feature rich environment



(22)

# In drones: Autonomous Search and Rescue (23)

- A similar use of drones as the Parrot AR approach, but with a specific usage in mind, and with certain different sensors
- Use of an RGBD camera, as opposed to RGB

This is a good example of monocular SLAM being advantageous to stereo SLAM

# Robotics: Dyson 360 Eye (3)

- Announced in 2014
- Autonomous vacuum cleaner
- Dyson is the first company to use SLAM commercially
- One of the few consumer products with SLAM available
- Uses a 360 degree panoramic camera to locate and map its environment with SLAM algorithms

# Future of SLAM

- Optimise/ create new algorithms
- New tracking methods
- Dedicated SLAM hardware
- Similar Path to the one Dyson has taken
- Combine Artificial intelligence and SLAM for environment aware intelligence

# Questions?

# Reference list I


1. Riisgaard S, Blas MR. SLAM for dummies: A Tutorial approach to simultaneous localization and mapping by the ‘dummies’. [Online] Massachusetts Institute of Technology, 2005 [Accessed: 31st January 2016]. Available from: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf[Accessed: 31st January 2016]
2. Google. On the Road. [Online] Google Self-Driving Car Project. Google; Available from: https://www.google.com/selfdrivingcar/where [Accessed: 18th March 2016]
3. Dyson Ltd. Dyson 360 EyeTM robot. [Online] Dyson. Dyson 360 EyeTM robot; Available from: https://www.dyson360eye.com [Accessed: 4th March 2016]
4. Goncharuk V. Augmented reality: Forget about markers!. [Online] Augmented Pixels. Augmented Pixels; Available from: http://augmentedpixels.com/augmented-reality-forget-about-markers/ [Accessed: 18th March 2016]
5. Royer E, Lhuillier M, Dhome M, Lavest J-M. Monocular vision for mobile robot localization and autonomous navigation. International Journal of Computer Vision. [Online] 2007;74(3): 237–260. Available from: doi:10.1007/s11263-006-0023-y [Accessed: 31st January 2016]
6. Prisacariu VA, Kähler O, Murray DW, Reid I. Simultaneous 3D tracking and reconstruction on a mobile phone. In: IEEE (ed.) 2013 IEEE International Symposium on Mixed and Augmented Reality. Adelaide, Australia: IEEE; 2013. p. 89–98.
7. Leutenegger S. An explanation of the Extended Kalman Filter. 2016.
8. Malisiewicz T. The Future of Real-Time SLAM and ‘Deep Learning vs SLAM’. [Online] Tombone’s computer vision Blog. Available from: http://www.computervisionblog.com/2016/01/why-slam-matters-future-of-real-time.html [Accessed: 4th March 2016]
9. Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct Monocular SLAM. [Online] [Accessed: 1st March 2016]. Available from: https://vision.in.tum.de/_media/spezial/bib/engel14eccv.pdf [Accessed: 1st March 2016]
10. Jan S. LSD-SLAM: Large-scale direct Monocular SLAM. [Online] Computer Vision Group, Technische Universität München. Available from: http://vision.in.tum.de/research/vslam/lsdslam [Accessed: 29th January 2016]
11. Engel J, Sturm J, Cremers D. Semi-Dense visual Odometry for a Monocular camera *. [Online] 2013 Dec [Accessed: 4th March 2016]. Available from: http://vision.in.tum.de/_media/spezial/bib/engel2013iccv.pdf [Accessed: 4th March 2016]

# Reference list II


12. Klein G, Murray D. Parallel tracking and mapping for small AR Workspaces. [Online] 2007 Nov [Accessed: 27th February 2016]. Available from: http://www.robots.ox.ac.uk/ gk/publications/KleinMurray2007ISMAR.pdf [Accessed: 27th February 2016]
13. Virtual Worldlets Network. VWN virtual dictionary: Self-occlusion. [Online] Available from: http://www.virtualworldlets.net/Resources/Dictionary.php?Term=Self-Occlusion [Accessed: 29th February 2016]
14. Davison A, Reid I. Monocular Visual Slam (monoSLAM). [Online] University of Oxford Active Vision Laboratory. Available from: http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2003ajd_monoslam/project.html [Accessed: 31st January 2016]
15. Davison AJ, Reid ID, Molton ND, Stasse O. MonoSLAM: Real-time single camera SLAM. 2007; Available from: doi:10.1109/TPAMI.2007.1049 [Accessed: 4th March 2016]
16. Pretto A. Visual-SLAM for Humanoid Robots. [Online] [PhD Thesis] Available from: http://www.dis.uniroma1.it/ pretto/papers/phd_thesis_pretto.pdf [Accessed: 7th March 2016]
17. Chli M. SLAM: Simultaneous localization and mapping. [Online] [Accessed: 9th March 2016]. Available from: http://www.roboticsschool.ethz.ch/sfly-summerschool/programme/5.2-Chli-SLAM.pdf [Accessed: 9th March 2016]
18. Albrecht S. An analysis of VISUAL MONO-SLAM. [Online] 2009 Oct [Accessed: 11th March 2016]. Available from: http://www-lehre.inf.uos.de/ svalbrec/documents/master_thesis.pdf [Accessed: 11th March 2016]
19. Harris C, Stephens M. A COMBINED CORNER AND EDGE DETECTOR. [Online] 1988 [Accessed: 13th March 2016]. Available from: https://www.cis.rit.edu/ cnspci/references/dip/feature_extraction/harris1988.pdf [Accessed: 13th March 2016]
20. OpenCV. Harris corner detector — OpenCV 2.4.12.0 documentation. [Online] Available from: http://docs.opencv.org/2.4/doc/tutorials/features2d/trackingmotion/harris_detector/harris_detector.html [Accessed: 13th March 2016]
21. Collins R. Lecture 06: Harris Corner detector. [Online] [Accessed: 14th March 2016]. Available from: http://www.cse.psu.edu/ rtc12/CSE486/lecture06.pdf [Accessed: 14th March 2016]
22. Dijkshoorn N. Simultaneous localization and mapping with the AR.Drone. [Online] 2012 [Accessed: 4th March 2016]. Available from: http://dare.uva.nl/cgi/arno/show.cgi?fid=451715 [Accessed: 4th March 2016]
23. Heukels F. Simultaneous localization and mapping (SLAM) towards an autonomous search and rescue aiding drone. [Online] 2015 [Accessed: 4th March 2016]. Available from: http://essay.utwente.nl/66615/1/ThesisFrankHeukels.pdf [Accessed: 4th March 2016]