University of Toulon Master 2ème année Ingénierie des Systmes Complexes, robotique et objets connectés Visual SLAM

Andrew Comport

CNRS - I3S/UNSA

January 5, 2022

Lecture Outline

Outline

Andrew Comport

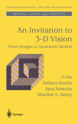
A bit about me: Andrew Comport http://www.i3s.unice.fr/∼comport

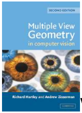
- 1997 - Bachelor of Science (BSc), Computer Science, Monash University, Australia.
- 2000 - Bachelor of Engineering (BE - hons), Electrical and Computer Systems Engineering with Honours, Monash.
- 2005 - (PhD) on "Robust real-time 3D tracking of rigid and articulated objects", IRISA/INRIA in Rennes.
- 2005-2007 - Postdoc, INRIA Sophia-Antipolis.
- 2007-2009 - CNRS permanent researcher, LASMEA, University of Blaise Pascal, Clermont Ferrand.
- 2005-2007 - Co-founder and CSO of PIXMAP.
- 2009-present - CNRS permanent researcher, I3S, Univerisity of Nice Sophia-Antipolis.

3D Vision

- Mainstream Technologies
  - Recognition,
  - Motion analysis,
  - Scene reconstruction,
  - Image restoration,
- Autonomous navigation : real-time localisation and mapping with one or more cameras,
- Core problem for localisation in unknown or dynamic environments.
- **Localisation**: Estimate the 6dof trajectory (3D position and orientation) of a mobile camera using visual information,
- **Mapping**: Estimate a map of the 3D environment visible to the camera(s).
- In this course we will focus on real-time i.e. < 30fps,
- Using only vision, i.e. no gyros, accelerometers, etc.,

Textbooks - Computer Vision

- An Invitation to 3D Vision. By Yi Ma, Stefano Soatto, Jana Kosecka , S. Shankar Sastry. Springer Verlag, 2005.


website: http://vision.ucla.edu/MASKS/
- Multiple View Geometry in Computer Vision, Second Edition, Richard Hartley and Andrew Zisserman, Cambridge University Press, March 2004.


website: http://www.robots.ox.ac.uk/∼vgg/hzbook/
- Trucco, Emanuele; Verri, Alessandro (1998). Introductory techniques for 3-D computer vision. Upper Saddle River, NJ [u.a.]: Prentice Hall.

Software tools:

- Robot Operating System : http://www.ros.org/
- Open SLAM database : https://openslam-org.github.io/
- OpenCV (Open Source Computer Vision) is a library of programming functions for real time computer vision. https://opencv.org/ Intel, then Willow garagem then Itseez (acquired by Intel).
- Augmented Reality Toolkit (ARToolkit). http://www.hitl.washington.edu/artoolkit/
- MATLAB and Octave Functions for Computer Vision and Image Processing - Peter Kovesi. http://www.csse.uwa.edu.au/~pk/research/matlabfns/
- Open source SLAM software: Orbslam2, Elastic fusion, DVOSLAM, LSDSLAM,...

Recommended reading:

- CVonline Bob Fisher's Compendium of Computer Vision : http://homepages.inf.ed.ac.uk/rbf/CVonline/
- Hugh Durrant-Whyte and Tim Bailey, Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. Robotics and Automation Magazine, 2006.
- Tim Bailey and Hugh Durrant-Whyte, Simultaneous Localisation and Mapping (SLAM): Part II State of the Art. Robotics and Automation Magazine, 2006.
- On unifying key-frame and voxel-based dense visual SLAM at large scales, Maxime Meilland, Andrew I. Comport, International Conference on Intelligent Robots and Systems, 2013, Tokyo, Japan. (Best paper award).
- Comport, A. I., Malis, E. and Rives, P. (2010). Real-time Quadrifocal Visual Odometry. International Journal of Robotics Research, Special issue on Robot Vision, 29(2-3), 245-266.

Online Courses

- Multiple View Geometry, Marc Pollefeys: http://www.cs.unc.edu/∼marc/mvg/slides.html
- Introduction to Computer Vision, James Hays: http://www.cs.brown.edu/courses/cs143/
- Theoretical, Conceptual and Experimental Vision: http://www.eecs.berkeley.edu/∼yang/courses/cs294-6/index.html
- Advanced topics in computer vision, Rene Vidal: http://www.vision.jhu.edu/teaching/vision/vision14/
- Computer Vision, Ahmed Elgammal: http://www.cs.rutgers.edu/∼elgammal/classes/cs534/cs534.h

Companies

- Microsoft - HoloLens, Roomalive, Kinectfusion, etc.
- Apple - ARkit, Metaio, etc.
- Occulus - Facebook Reality Labs, 13th labs, Surreal vision.
- Qualcom - mobile augmented reality, Autriche
- Google - Cartographer, Perceptionio, Robotics (Alphabet),...
- Magic leap
- ECA Robotics France - Sponsored this course with Asus Xtion Pro Live devices.
- Artisens - Munich, https://www.artisense.ai/
- Slamcore - London, https://www.slamcore.com
- Previous companies : Total Immersion (Paris), Pixmap (Sophia-Antipolis).

Course: Practical Work

- Planar direct template tracking.
  - Implement the warping function $\mathbf{p}_2 = \mathbf{H}(\mathbf{x})\mathbf{p}_1$.
  - Track planar patches in sequences of images.
  - Analyse accuracy and precision by comparing Homographies $\Delta\mathbf{H} = \mathbf{H}_1^{-1}\mathbf{H}_2$.
  - Augmented Reality.
  - Download :

Download from: http://gofile.me/2LJ9p/ttcyByoHh pw: utoulon2022

Outline

Applications

- Applications requiring a low cost, portable positioning and 3D reconstruction device:
  - Augmented reality (real-time interfaces)
  - Robotic control / Visual servoing (industrial and domestic)
  - Virtual map building (Google earth)
  - Remote computing such as PDAs or wearable devices (location awareness)
  - Navigation in unknown environments (i.e. Planetary landing)
- Application domains include *medical, military, industrial, edutainment, human computer interaction, etc*

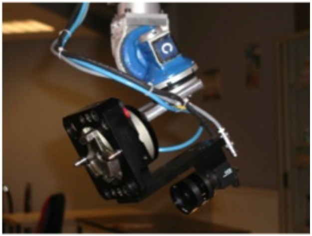
Augmented Reality

- Augmented reality (AR):
  - Coherent insertion of virtual objects within real image streams.
  - Augmented reality is handled as a 2D-3D registration issue:
- Post production AR
  - Full knowledge of the video sequence,
  - Localisation of the camera and mapping the structure of the scene,
  - Commercial techniques (Realvis - Autodesk Matchmover, 2D3)
- Online AR
  - Real-time requirements,
  - No knowledge of the future,

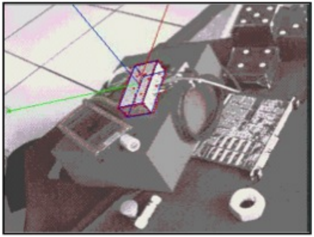
Navab [Lepetit-Fua][Berger][Kutulakos],...

Visual Servoing

- Control the movement of a robot using visual information.
- Real-world visual servoing requires:
  - Extraction of complex features in "real" images,
  - Sensitivity of the control law to aberrant data,
- 3D tracking is a key issue in vision-based control
- Useful to know the robot's position wrt its environment.



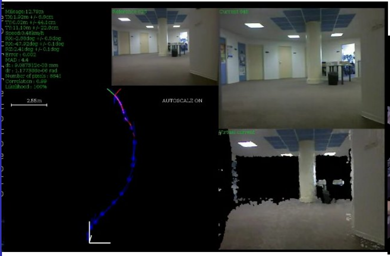
6 degrees of freedom gantry robot (Afma6).



2.5D visual servoing 3D model-based tracking.

3D Dense Mapping in real-time

- Map densely and track an unknown environment in real-time.

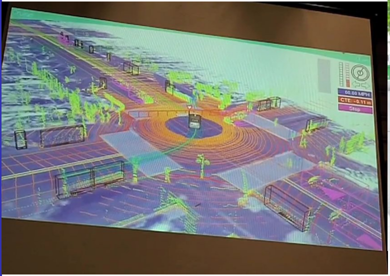


I3S large scale dense mapping [Audas11].

Imperial college [Newcomb11].

Smart Cars

- Autonomous Navigation .

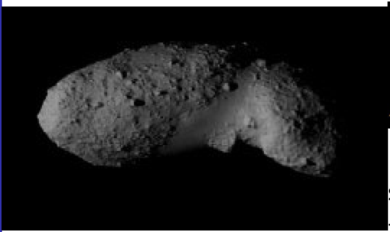


Google Car [IROS].

Autonomous vision-based navigation [INRIA Cycab, IROS].

Visual SLAM in Space

- Autonomous Navigation and mapping.
- Long communication delays require vehicules to be autonomous.



Thales Alenia Space/CNRS - Itokawa Asteroid [IROS11].

Mars Challenge [Tykkala11] - Vision systems (JPL) on Spirir Rover used for several tasks: Visual Odometry, panorama stitching, 3D terrain modelling, obstacle detection.

Aerial mobile robots

- Inspection of structures - bridges, dam walls.
- Suveillance - defense applications.

Infotron Drone IT180-5.

Ground control station.

Intelligent navigation in dangerous environments

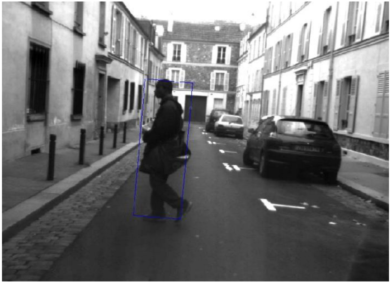
- DGA-Rapid, Fraudo,
- 2012-Present I3S/CNRS-UNS,
- Search and rescue,
- Disaster and emergency scenarios.

Augmented and assistive driving

- Augmented reality (AR):
  - Coherent insertion of virtual objects within real image streams.
  - Augmented reality is handled as a 2D-3D registration issue:
- Intelligent transport applications:
  - Networks of communicating vehicules,
  - Danger-detection,
  - Pedestrian tracking,
  - Navigation assistance,

Pedestrian tracking

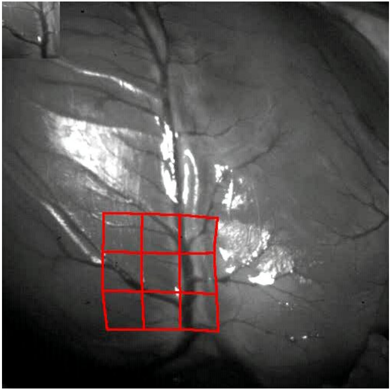
- Intelligent transport applications:
  - Occuded obstacles :
  - Networks of communicating vehicules,
  - Danger-detection,
  - Pedestrian tracking,
  - Navigation assistance,



Medical robotics

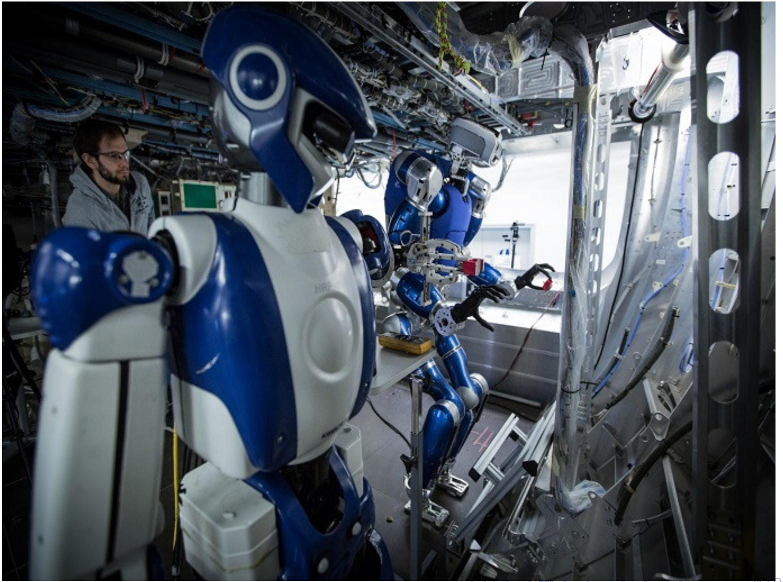
- Tracking of deformable surfaces.
- Stabilisation of surgical instruments.

High speed planar tracking.



Multi-planar deformable heart tracking.

Manufacturing



Why Vision

Darpa challenge.

Drone fail.

Various Approaches

- Computer vision community:
  - Structure from motion,
  - Bundle adjustment,
  - Post production (not real-time).
- Robotic community:
  - Probabilistic approaches have been studied extensively using various different sensor including gps, laser range scanners, sonars, etc..
  - For example: work of Durrant-Whyte and Thrun.
  - Primarily aimed at autonomous robots, often slow moving with additional control input, e.g. odometry

Recent 3D vision approaches

- Various implementations differing with
  - feature extraction,
  - matching,
  - direct intensity based approaches,
  - stereo motion from structure,
  - visual odometry etc...
- Here we consider 3D vision for both single and stereo cameras, operating in real-time.

Outline

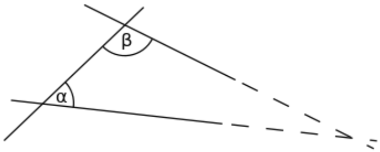
Outline

- Revision - Lecture 1.
- Euclidian Space.
- Points and Vectors.
- 3D Rotations
- SE(3) Transformations and Lie Groups.
- 3D Angular Velocity.
- se(3) twist motion and Lie Algebra.
- Twist transformations.
- The Exponential Map.

Euclid's Axioms

Let $\mathbb{E}^3$ represent 3 dimensional Euclidean space defined by Euclid's 5 axioms:

- "To draw a straight line from any point to any point."
- "To produce [extend] a finite straight line continuously in a straight line."
- "To describe a circle with any centre and distance [radius]."
- "That all right angles are equal to one another."
- The parallel postulate:



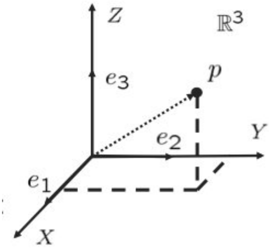
Cartesian Coordinate Frame

- Standard base vectors

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- Coordinates of a point **P** in space:

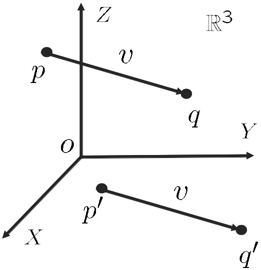
$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \mathbb{R}^3$$



Vectors

A free vector is defined by a pair of points $(\mathbf{p}, \mathbf{q})$:

$$^{p}\mathbf{P} = \begin{bmatrix} X_p \\ Y_p \\ X_p \end{bmatrix} \in \mathbb{R}^3, \quad ^{q}\mathbf{P} = \begin{bmatrix} X_q \\ Y_q \\ X_q \end{bmatrix}$$



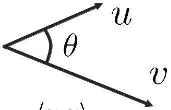
The coordinates of the vector are:

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} X_q - X_p \\ Y_q - Y_p \\ Z_q - Z_p \end{bmatrix} \in \mathbb{R}^3,$$

Inner Product and Cross Product

The inner product between two vectors:

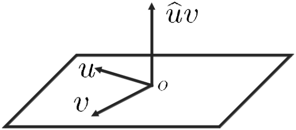
$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$



$$< \mathbf{u}, \mathbf{v} > \doteq \mathbf{u}^\top \mathbf{v} = u_1 v_1 + u_2 v_2 + u_3 v_3$$

$$||\mathbf{u}|| \doteq \sqrt{\mathbf{u}^\top \mathbf{u}} = \sqrt{u_1^2 + u_2^2 + u_3^2}$$

$$\cos(\theta) = \frac{\langle u, v \rangle}{||u|| \, ||v||}$$



The cross product between two vectors:

$$\mathbf{u} \times \mathbf{v} = [\mathbf{u}]_\times \mathbf{v} \quad \mathbf{u}, \mathbf{v} \in \mathbb{R}^3,$$

$$[\mathbf{u}]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

Rigid Body Motion

The position of a point in 3D Euclidean space is defined to be relative to an inertial Cartesian coordinate frame so that a point $\mathbf{P} \in \mathbb{E}^3$ is identified with a point in $\mathbb{R}^3$. The set of three orthonormal axes, representing a reference frame, form the basis for defining the position of a point in 3D using three coordinates as:

$$\mathbf{P} = (X, Y, Z) \in \mathbb{R}^3. \tag{1}$$

The motion trajectory of a point can be represented as a parameterized curve:

$$\mathbf{P}(t) = (X(t), Y(t), Z(t)) \in \mathbb{R}^3. \tag{2}$$

Special Euclidian Group $\mathbb{SE}(3)$

Definition

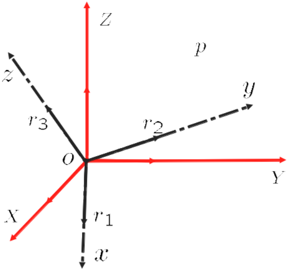
Special Euclidean Transformation and *Rigid Body Motion*.

A mapping $\mathbf{m} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ *satisfies the following properties:*

1. *Length is preserved:* $||\mathbf{m}(\mathbf{P}_1) - \mathbf{m}(\mathbf{P}_2)|| = ||\mathbf{P}_1 - \mathbf{P}_2||$ *for all points* $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^3$.
2. *The cross product is preserved:* $\mathbf{m}_*(\mathbf{v} \times \mathbf{w}) = \mathbf{m}_*(\mathbf{v}) \times \mathbf{m}_*(\mathbf{w})$ *for all vectors* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^3$, *where vectors transform according to* $\mathbf{m}_*(\mathbf{v}) = \mathbf{m}(\mathbf{P}_1) - \mathbf{m}(\mathbf{P}_2)$.

The set of all such transformations is denoted as the special Euclidean group SE(3).

The rotation matrix has columns mutually orthonormal, $\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}$. The coordinate frame is chosen to be right handed such that $det(\mathbf{R}) = +1$. The space of rotation matrices is defined by:



$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}\mathbf{R}^\top = \mathbf{I}, det(\mathbf{R}) = +1 \right\}. \qquad (3)$$

$SO(3) \subset \mathbb{R}^{3\times 3}$ is called the special orthogonal group under the operation of matrix multiplication and satisfies the axioms of closure, identity, inverse and associativity. (4x2)x3 = 4x(2x3)

if you have a scalar (a real number) c and a matrix A that is size m×n, then when you multiply A by c the resulting matrix is also of size m×n.

Euler's Theorem

Theorem

Any rotation $\mathbf{R}$ *can be represented as a single rotation* $\theta \in [0, 2\pi)$ *around a fixed axis* $\mathbf{u} \in \mathbb{R}^3$.

Let θ be the angle of rotation and the unit vector $\mathbf{u} = (u_1, u_2, u_3)^\top$ define the axis of rotation with the constraint $|\mathbf{u}| = 1$. The rotation matrix is then given by:

$$\mathbf{R} = \mathbb{I}\cos\theta + (1-\cos\theta)\begin{bmatrix} u_1^2 & u_1 u_2 & u_1 u_3 \\ u_2 u_1 & u_2^2 & u_2 u_3 \\ u_3 u_1 & u_3 u_2 & u_3^2 \end{bmatrix} + \sin\theta \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \tag{4}$$

A complete derivation for Euler's parametrization can be found in many textbooks on rigid-body motion and is closely related to Rodrigues' formula.

Euler Angles (1/2)

A second method is to consider rotations parameterized by α, β, γ around the principle x, y and z axes respectively. Each elementary rotation is defined by the following matrices:

$$\mathbf{R}_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$\mathbf{R}_Y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$\mathbf{R}_Z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

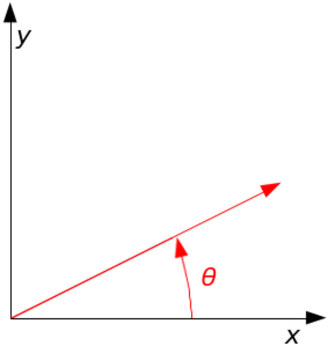
Euler Angles (2/2)

The matrix describing the overall rotation being composed of the product of these elementary matrices:

$$\mathbf{R} = \mathbf{R}_X \mathbf{R}_Y \mathbf{R}_Z = \begin{bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & -\sin\alpha\cos\beta \\ -\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\alpha\sin\beta\sin\gamma + \sin\alpha\cos\gamma & \cos\alpha\cos\beta \end{bmatrix}$$

Note that the order of multiplication of elementary rotation matrices is not commutative.

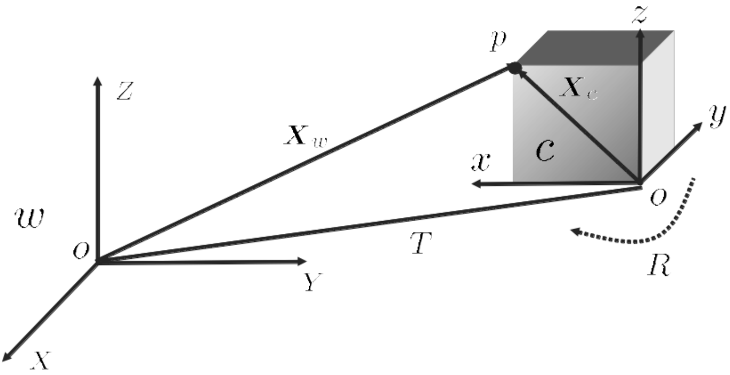
Simple 2D Rotation Example



Assume a rotation of the axes, $\theta = 45$ degrees counter-clockwise. A point $\mathbf{P} = [1 \;\; 2]^\top$.

- Question 1. Calculate the new value of the point.
- Question 2. Write the rotation in matrix form.

Rotation and Translation



Co

are related by :

$$^c\mathbf{P} = \mathbf{R}\,^w\mathbf{P} + \mathbf{t}$$

Homogeneous Coordinates

3D Coordinates are related by $\mathbf{P}_c = \mathbf{R}\mathbf{P}_w + \mathbf{t}$ The homogeneous coordinates of a point :

$$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \overline{\mathbf{P}} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \in \mathbb{R}^4$$

Homogeneous coordinates are related by:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Pose and Location

Theorem

Every rigid body motion can be realized by a rotation about an axis combined with a translation parallel to that axis.

This parametrization of rigid motion is commonly referred to as a screw motion which can be defined by the space of Special Euclidean transformations:

$$SE(3) = \left\{ \mathbf{m} = (\mathbf{R}, \mathbf{t}) : \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}. \quad (5)$$

The homogeneous representation of $\mathbf{m}$ is then obtained in matrix form as:

$$^{a}\mathbf{T}_b = \begin{bmatrix} ^{a}\mathbf{R}_b & ^{a}\mathbf{t}_b \\ \mathbf{0}_3 & 1 \end{bmatrix} \in \mathbb{SE}(3), \quad (6)$$

where $\mathbf{T}$ is an equivalent representation of the minimal vector form $\mathbf{x} \in se(3)$.

Homogeneous Transformation

The transformation of a homogeneous point between two reference frames can therefore be written as:

$$^{a}\overline{\mathbf{P}} = {}^{a}\mathbf{T}_{b}\,{}^{b}\overline{\mathbf{P}}$$

Composition and Inverse Transformations

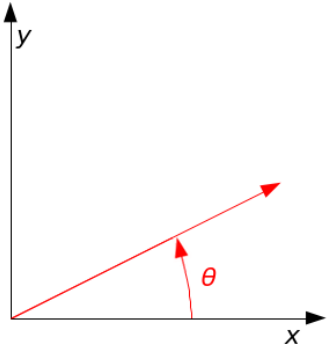
Furthermore, a pose between frame a and frame c can be expressed as a composition of homogeneous transformation matrices as:

$$ {}^{a}\mathbf{T}_{c} = {}^{a}\mathbf{T}_{b}\,{}^{b}\mathbf{T}_{c}. \tag{7} $$

${}^{b}\mathbf{T}_{a} = {}^{a}\mathbf{T}_{b}^{-1}$ where the inverse transformation is:

$$ {}^{a}\mathbf{T}_{b}^{-1} = \begin{bmatrix} {}^{a}\mathbf{R}_{b}^{\top} & -{}^{a}\mathbf{R}_{b}^{\top}\,{}^{a}\mathbf{t}_{b} \\ \mathbf{0}_{3} & 1 \end{bmatrix}. \tag{8} $$

Simple 2D Rotation and Translation Example



Assume a rotation of the axes, $\theta = 45$ degrees counter-clockwise. A point $\mathbf{P} = [1 \;\; 2]^\top$. Now add a translation to the axes by $\mathbf{t} = [2 \;\; 3]^\top$.

- Question 1. Calculate the new value of the point.

Rigid Body Velocity (1/2)

The infinitesimal version of rigid motion is called a *twist* where a twist represents the instantaneous velocity of a rigid body in terms of linear and angular components. A rigid body velocity is defined as a 6-dimensional twist vector $\mathbf{x} = (\boldsymbol{\upsilon}, \boldsymbol{\omega})$ where $\boldsymbol{\upsilon} = (\upsilon_x, \upsilon_y, \upsilon_z)$ is the linear component of the velocity vector and $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ the angular velocity. A twist vector is the tangent vector to an element $\mathbf{T}(t)$ of $SE(3)$.
The velocity of a point:

$$ {}^{a}\dot{\mathbf{P}}(t) = \frac{d}{dt}\,{}^{a}\mathbf{P}(t). \tag{9} $$

Rigid Body Velocity (2/2)

Differentiating the point transformation:

$$ {}^{a}\dot{\bar{\mathbf{P}}} = {}^{a}\dot{\mathbf{T}}_{b}{}^{b}\bar{\mathbf{P}}, \tag{10} $$

The velocity of the point with respect to the spatial reference frame a:

$$ {}^{a}\dot{\bar{\mathbf{P}}} = {}^{a}\dot{\mathbf{T}}_{b}{}^{a}\mathbf{T}_{b}^{-1}{}^{a}\bar{\mathbf{P}} = {}^{a}\dot{\mathbf{T}}_{b}{}^{b}\mathbf{T}_{a}{}^{a}\bar{\mathbf{P}}, \tag{11} $$

remembering that ${}^{a}\mathbf{T}_{b}^{-1} = {}^{b}\mathbf{T}_{a}$.
The 4×4 velocity mapping in equation (11) is called a twist and is then easily obtained by using equation (6) as:

$$ \begin{aligned} {}^{a}\dot{\mathbf{T}}_{b}{}^{a}\mathbf{T}_{b}^{-1} &= \begin{bmatrix} {}^{a}\dot{\mathbf{R}}_{b} & {}^{a}\dot{\mathbf{t}}_{b} \\ \mathbf{0}_3 & 0 \end{bmatrix} \begin{bmatrix} {}^{a}\mathbf{R}_{b}{}^{\top} & -{}^{a}\mathbf{R}_{b}{}^{\top}{}^{a}\mathbf{t}_{b} \\ \mathbf{0}_3 & 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^{a}\dot{\mathbf{R}}_{b}{}^{a}\mathbf{R}_{b}{}^{\top} & -{}^{a}\dot{\mathbf{R}}_{b}{}^{a}\mathbf{R}_{b}{}^{\top}{}^{a}\mathbf{t}_{b} + {}^{a}\dot{\mathbf{t}}_{b} \\ \mathbf{0}_3 & 0 \end{bmatrix}, \end{aligned} \tag{12} $$

where ${}^{a}\dot{\mathbf{t}}_{b}$ is the translational velocity of a point in frame a with respect to a point in frame b.

Rotational Velocity

The matrix $({}^a\dot{\mathbf{R}}_b {}^a\mathbf{R}_b^{-1})^\top = -{}^a\dot{\mathbf{R}}_b {}^a\mathbf{R}_b^{-1}$ is a 3×3 skew symmetric matrix mapping from $\mathbb{R}^3$ to $\mathbb{R}^3$ as:

$$[\mathbf{u}]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}, \tag{13}$$

where the operator, $[.]_\times$, is chosen to represent a skew symmetric matrix.
The space of all skew symmetric matrices is denoted by $so(3)$:

$$so(3) = \left\{[\mathbf{u}]_\times \in \mathbb{R}^{3\times3} : \mathbf{u} \in \mathbb{R}^3\right\}. \tag{14}$$

Lie Algebra: $se(3)$

Therefore, there exists an operator, based on the skew-symmetric operator $[.]_\times$ for rotations, which transforms a full twist matrix to its minimal vector form defined as:

$$[\mathbf{x}]_\wedge = \begin{bmatrix} [\omega]_\times & \boldsymbol{v} \\ 0 & 0 \end{bmatrix}. \tag{15}$$

The space of velocity twists can therefore be written as a 4×4 homogeneous twist matrix $[\mathbf{x}]_\wedge$ as:

$$se(3) = \left\{ [\mathbf{x}]_\wedge \in \mathbb{R}^{4\times4} : [\omega]_\times \in so(3), \boldsymbol{v} \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{4\times4}, \tag{16}$$

where $se(3)$ is the Lie Algebra of the Lie Group $SE(3)$.

Velocity Transformation: The adjoint matrix

The relationship between an instantaneous twist $[{}^a\mathbf{x}]_\wedge$, in the spatial reference frame and an instantaneous twist $[{}^b\mathbf{x}]_\wedge$ in the body reference frame is obtained by homogeneous composition as:

$$\text{4x4} \longrightarrow [{}^a\mathbf{x}]_\wedge \begin{aligned} &= {}^a\dot{\mathbf{T}}_b\, {}^a\mathbf{T}_b^{-1} = {}^a\mathbf{T}_b({}^a\mathbf{T}_b{}^{-1}\,{}^a\dot{\mathbf{T}}_b)\,{}^a\mathbf{T}_b{}^{-1} \\ &= {}^a\mathbf{T}_b[{}^b\mathbf{x}]_\wedge\, {}^a\mathbf{T}_b{}^{-1}, \end{aligned} \tag{17}$$

which gives the twist transformation matrix as:

$${}^a\mathbf{V}_b = \begin{bmatrix} {}^a\mathbf{R}_b & [{}^a\mathbf{t}_b]_\times {}^a\mathbf{R}_b \\ \mathbf{0}_3 & {}^a\mathbf{R}_b \end{bmatrix}, \tag{18}$$

which is a 6×6 matrix that transforms twists from reference frame a to reference frame b. This is commonly known as the *adjoint* map.

Exponential Map

The relationship between the velocity twist of a moving body and its pose:

$$\mathbf{T} = e^{[\mathbf{x}]_\wedge}, \tag{19}$$

The rotational component is found using Rodrigues formula:

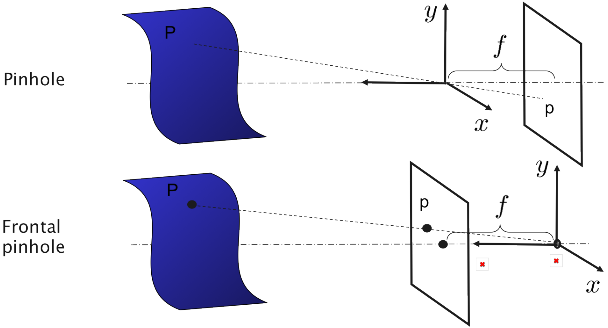
$$e^{[\omega]_\times} = \mathbb{I} + \frac{[\omega]_\times}{||\omega||} \sin ||\omega|| + \frac{[\omega]_\times^2}{||\omega||^2}(1 - \cos(||\omega||)), \tag{20}$$

The full transformation is given by:

$$e^{[\mathbf{x}]_\wedge} = \begin{bmatrix} e^{[\omega]_\times} & \frac{(I - e^{[\omega]_\times})([\omega]_\times \upsilon) + \omega\omega^\top \upsilon}{||[\omega]_\times||} \\ 0 & 1 \end{bmatrix}, \quad \text{if} \quad \omega \neq 0 \ . \tag{21}$$

Outline

Pinhole Camera Model



$$\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}$$

Image coordinates

Homogeneous coordinates:

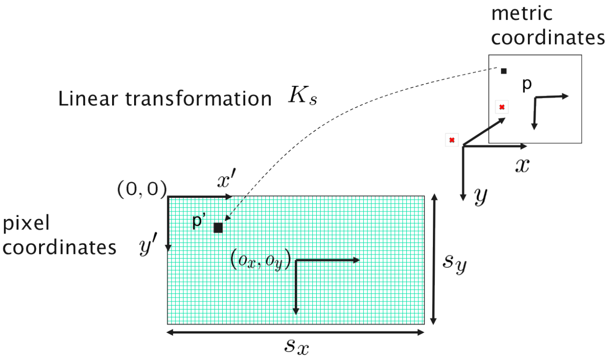
$$\overline{\mathbf{p}} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}, \overline{\mathbf{P}} \rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

$$Z\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \mathbf{K}_f \mathbf{\Pi}_0$$

Pixel Coordinates

$$\bar{\mathbf{p}}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{w}{s_x} & s_\theta & o_x \\ 0 & \frac{h}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix},$$
$$= \mathbf{K_s}\bar{\mathbf{p}}$$



Calibration Matrix and Camera Model

Pinhole camera: $\lambda\overline{\mathbf{p}} = \mathbf{K}_f \Pi_0 \overline{\mathbf{P}}$

Pixel coordinates: $\overline{\mathbf{p}}' = \mathbf{K}_s \overline{\mathbf{p}}$

$$\lambda\overline{\mathbf{p}}' = \mathbf{K}_s \mathbf{K}_f \Pi_0 \overline{\mathbf{P}} =$$

$$\begin{bmatrix} f\frac{w}{s_x} & s_\theta & o_x \\ 0 & f\frac{h}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

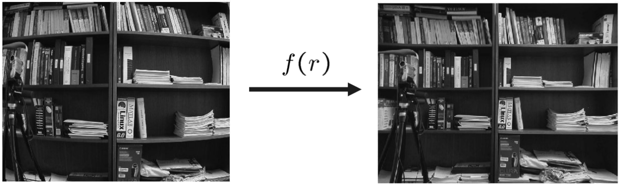
Calibration matrix (intrinsic parameters: $\mathbf{K} = \mathbf{K}_s \mathbf{K}_f$

Projection matrix: $\Pi = [\mathbf{K}, 0] \in \mathbb{R}^{3\times 4}$

Camera model: $\lambda\overline{\mathbf{p}}' = \mathbf{K}\Pi_0 \overline{\mathbf{P}} = \Pi\overline{\mathbf{P}}$

Radial Distortion

Non-linear transformation along the radial direction.



$$\mathbf{p} = c + f(r)(\mathbf{p}_d - c), r = ||\mathbf{p}_d - c||$$

$$f(r) = 1 + a_1 r + a_2 r^2 + a_3 r^3 + a_4 r^4 + \dots$$

Distortion correction: make lines straight.

Image of a point

Homogeneous coordinates of a 3D point:

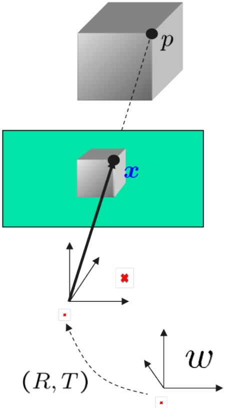
$$\overline{\mathbf{P}} = [X, Y, Z, W] \in \mathbb{R}^4, (W = 1)$$

Homogeneous coordinates of its 2D image:

$$\overline{\mathbf{p}} = [x, y, z] \in \mathbb{R}^3, (z = 1)$$

Projection of a 3D point to an image plane:

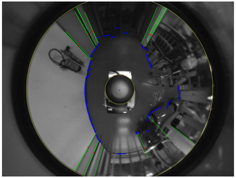
$$\lambda\overline{\mathbf{p}} = \Pi\overline{\mathbf{P}}, \lambda \in \mathbb{R}, \Pi = [\mathbf{R}, \mathbf{t}] \in \mathbb{R}^{3\times 4}$$
$$\lambda\overline{\mathbf{p}}' = \Pi\overline{\mathbf{P}}, \lambda \in \mathbb{R}, \Pi = [\mathbf{KR}, \mathbf{Kt}] \in \mathbb{R}^{3\times 4}$$



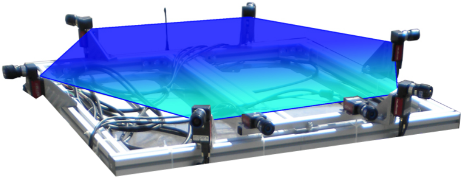
Various camera models

Two main categories [Sturm05]:

- Central projection: Monocular, Central catadioptric, ...
- Non-central: Multi-camera, X-slit cameras, Non-central catadioptric, Omnivergent imaging system, Spheric mirror,...



Omni-directional

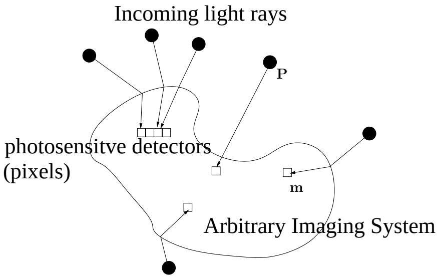


Multi-baseline camera system

Generalised camera model

Define a model for an arbitrary physical imaging system [Grossberg01]

- The incoming light rays are projected onto a photosensitive detector (pixel) .
- A point $\mathbf{P} \in \mathbb{R}^3$ is imaged at a pixel $\mathbf{m} = (u, v)$ that is found by following the ray through the physical system.



3D viewing rays

Plücker lines

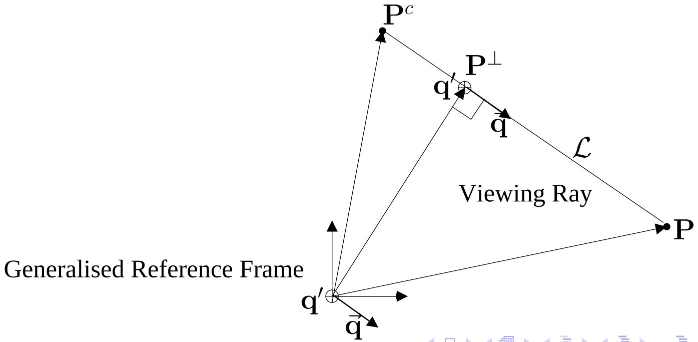
- Parametrised as Plücker vectors [Pless03].
- $\overline{\mathbf{P}}_1 = (X_1, Y_1, Z_1, W_1)^\top$ and $\overline{\mathbf{P}}_2 = (X_2, Y_2, Z_2, W_2)^\top \in \mathbb{RP}^3$ are two 3D points.
- Plücker matrix $\mathbf{L} = \overline{\mathbf{P}}_1\overline{\mathbf{P}}_2^\top - \overline{\mathbf{P}}_2\overline{\mathbf{P}}_1^\top$ - line joining the points.
- $\mathcal{L} = (\mathbf{q}, \mathbf{q}')^\top$ are the Plücker coordinates. The 6 non-zero independent elements of :

$$\mathcal{L} = \begin{pmatrix} \mathbf{L}_{41} \\ \mathbf{L}_{42} \\ \mathbf{L}_{43} \\ \mathbf{L}_{32} \\ -\mathbf{L}_{31} \\ \mathbf{L}_{21} \end{pmatrix} = \begin{pmatrix} W_1X_2 - X_1W_2 \\ W_1Y_2 - Y_1W_2 \\ W_1Z_2 - Z_1W_2 \\ Z_1Y_2 - Y_1Z_2 \\ X_1Z_2 - Z_1X_2 \\ Y_1X_2 - X_1Y_2 \end{pmatrix} \in \mathbb{RP}^5,$$

3D viewing rays (6 parameters - 4 dof.)

Plücker Line $\mathcal{L} = (\mathbf{q}, \mathbf{q}')^\top$

- Normalised direction vector $\tilde{\mathbf{q}} = \frac{\mathbf{q}}{|\mathbf{q}|}, \tilde{\mathbf{q}} \in S^2$.
- The moment vector $\mathbf{q}' = \mathbf{q} \times \mathbf{P}$.
- Point on the line closest the origin $\mathbf{P}^\perp = \tilde{\mathbf{q}} \times \mathbf{q}'$.



Outline

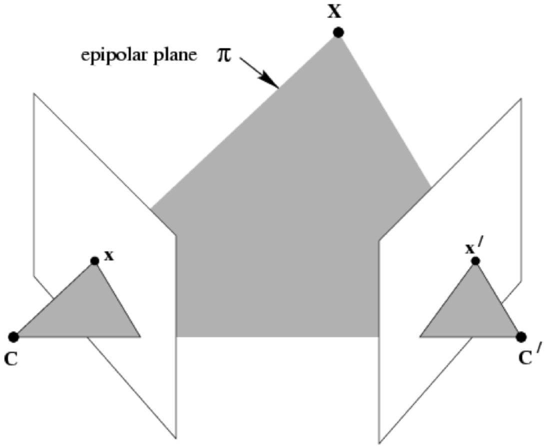
Andrew Comport

Two-view geometry: epipolar geometry

Three questions:

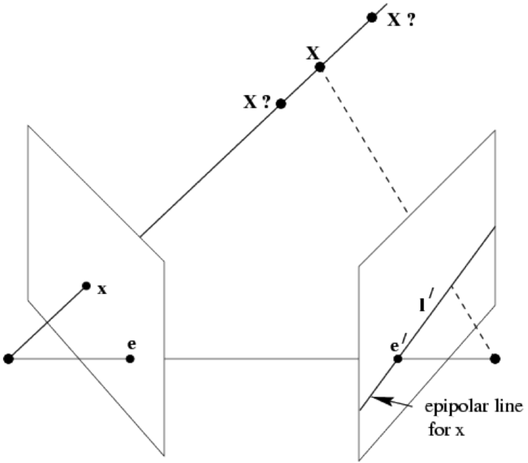
1. Correspondence geometry: Given an image point $\mathbf{p}$ in the first image, how does this constrain the position of the corresponding point $\mathbf{p}$ in the second image?
2. Camera geometry (motion): Given a set of corresponding image points $\{\mathbf{p}_i \ldots \mathbf{p}_i\}, i = 1, , n$, what are the camera matrices $\mathbf{T}$ and $\mathbf{T}$ for the two views?
3. Scene geometry (structure): Given corresponding image points $\mathbf{p}_i \rightarrow \mathbf{p}_i$ and camera matrices $\mathbf{T}$, $\mathbf{T}$, what is the position of their 3D point $\mathbf{P}$ in space?

The epipolar geometry (1/5)



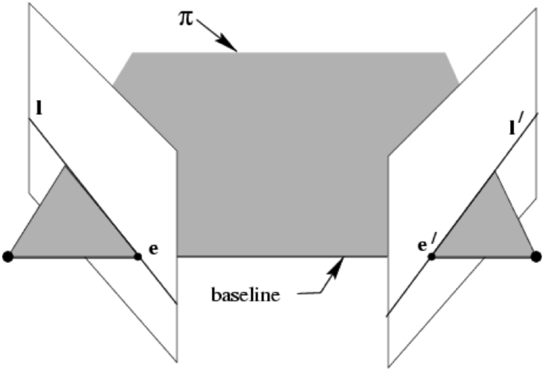
$\mathbf{C}$, $\mathbf{C}'$, $\mathbf{x}$, $\mathbf{x}'$ and $\mathbf{X}$ are coplanar.

The epipolar geometry (2/5)



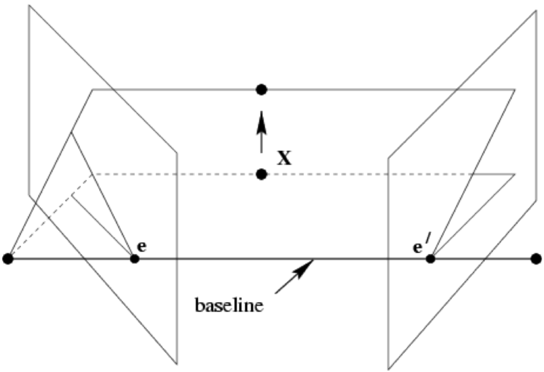
What if only **C**, **C**′, **x**, are unknown?

The epipolar geometry (3/5)



All points on π project on $\mathbf{l}$ and $\mathbf{l}'$.

The epipolar geometry (4/5)

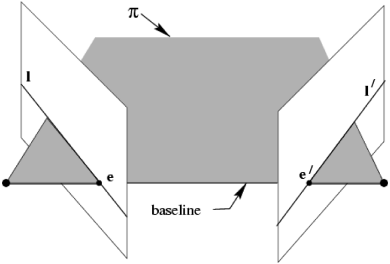


Family of planes π and lines $\mathbf{l}$ and $\mathbf{l}'$ intersect in $\mathbf{e}$ and $\mathbf{e}'$.

The epipolar geometry (5/5)

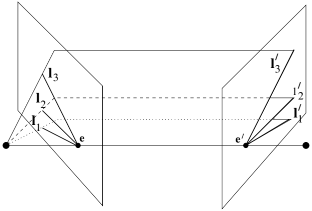
Epipoles $\mathbf{e}$, $\mathbf{e}'$

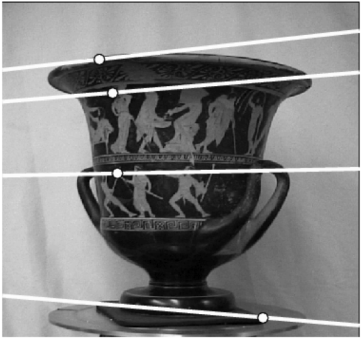
- Intersection of baseline with image plane.
- Projection of projection center in other image.
- Vanishing point of camera motion direction



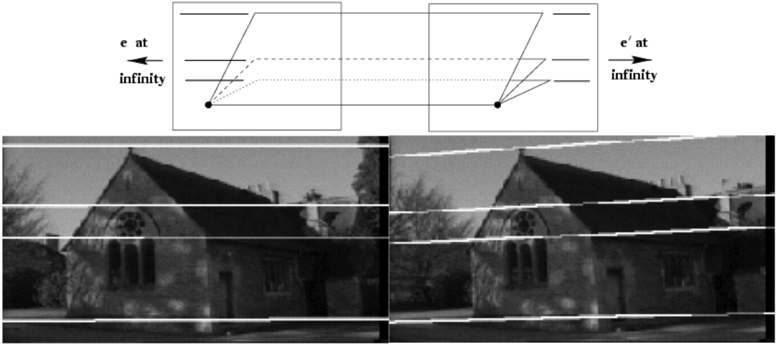
An epipolar plane = plane containing baseline (1-D family).
An epipolar line = intersection of epipolar plane with image.

Example: converging cameras



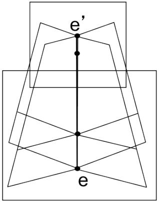


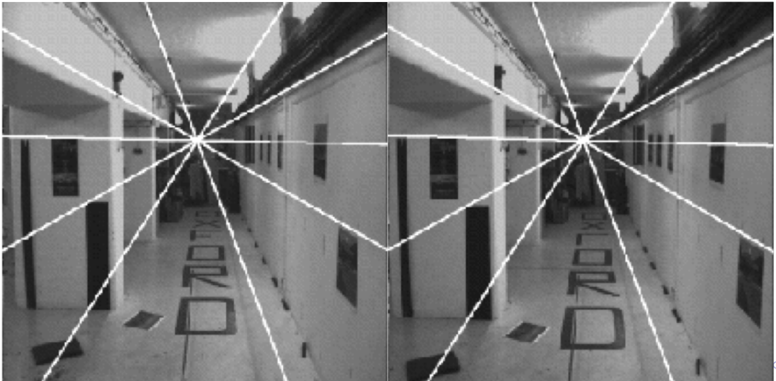
Example: motion parallel with image plane



(simple for stereo → rectification)

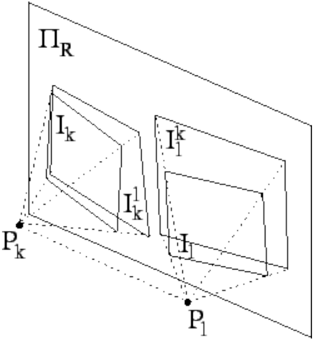
Example: forward motion





Planar Rectification

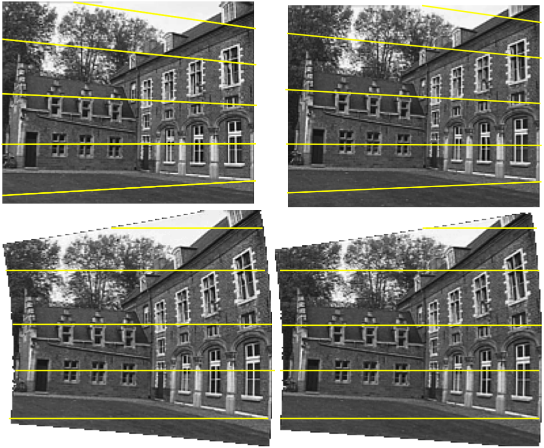
Select a plane parallel with the baseline and reproject the two images onto this plane (moves epipole to ∞.



Planar rectification: $(\mathbf{I}_k^l, \mathbf{I}_l^k)$ are the rectified images for the pair $(\mathbf{I}_k, \mathbf{I}_l)$ (the plane Π_R should be parallel to the baseline $(\mathbf{P}_k, \mathbf{P}_l)$).

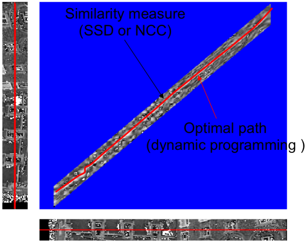
Choose remaining dof. so that no pixels are compressed.

Stereo rectifiction



Matlab code: http://profs.sci.univr.it/~fusiello/demo/rect/

Stereo matching



- Real-time on GPU - sum-of-square differences using pixelshader.
- Pixel disimilarity - Absolute intensity difference, SSD, NCC.
- Constraints: epipolar, ordering, uniqueness, disparity limit.
- Trade-off: matching cost (data), discontinuities (prior).
- Many approaches: AI-interval matching [Birchfield], Optical flow, Graph cuts, SGBM, DP,...

Disparity map

Image $\mathbf{I}(x, y)$

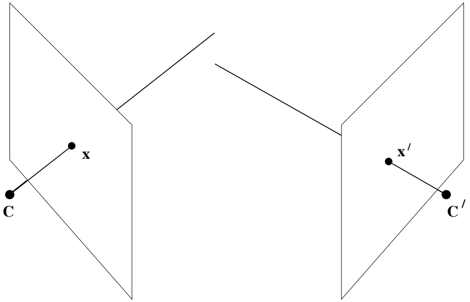
Disparity Map $\mathbf{D}(x, y)$

Image $\mathbf{I}'(x', y')$

$$(x', y') = (x + \mathbf{D}(x, y), y)$$

Point reconstruction

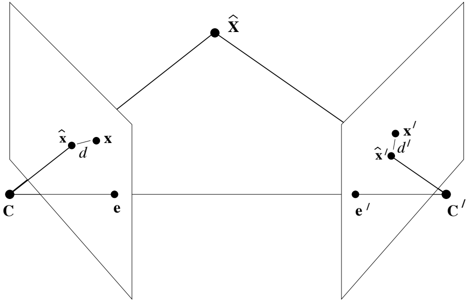
In general configuration measured points $\mathbf{x}$ and $\mathbf{x}'$ are skew in 3-space and dont satisfy the epipolar constraints.



Triangulate while considering the epipolar constraint: Linear method, geometric non-linear method minimising the distance between the point at the corresponding epipolar line and the polynomial method.

Point reconstruction

If the points have been matched using epipolar geometry and rectified images then the triangulation problem is much simpler.



$$Z(depth) = \frac{ft_x}{d}$$

where Z is depth, f focal length, t_x baseline and d disparity.

Two-view Geometry Induced by a Plane

Current frame 3D point $\mathbf{P} = [X, Y, Z]^\top$ and reference frame point $\mathbf{P}^*$ are related as:

$$\mathbf{P} = \mathbf{R}\mathbf{P}^* + \mathbf{t}.$$

If $\mathcal{P}$ lies on a plane Π gives an additional constraint:

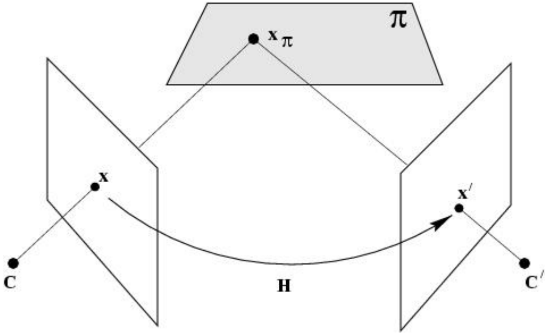
$$\mathbf{n}^\top \mathbf{P}^* = d^* \Leftrightarrow \frac{1}{d^*}\mathbf{n}\mathbf{P}^* = 1,$$

where $\mathbf{n} \in \mathbb{R}^3$ the unit normal vector to the plane Π and $d^* \in \mathbb{R}^+$ the distance from the plane to the origin.
The overall transformation is:

$$\mathbf{P} = \left(\mathbf{R} + \frac{1}{d^*}\mathbf{t}\mathbf{n}^\top\right)\mathbf{P}^* = \mathbf{H}\mathbf{P}^*,$$

where $\mathbf{H} \in \mathbb{R}^{3\times 3}$ is the so-called Euclidean homography matrix.

Scene planes and Homographies



Using the calibration matrix $\mathbf{K}$, the transformation between the pixel coordinates is obtained as follows:

$$\mathbf{p} \propto \mathbf{K}\mathbf{H}\mathbf{K}^{-1}\mathbf{p}^* = \mathbf{G}\mathbf{p}^*,$$

where $\mathbf{G} \in \mathbb{R}^{3\times 3}$ is the projective homography matrix and $\propto$ denotes proportionality.

Homographic Warping

The geometric transformation defines a warping function as:

$$\mathbf{w}(\cdot;\mathbf{G}) : \mathbb{P}^2 \mapsto \mathbb{P}^2, \qquad \mathbf{p}^* \mapsto \mathbf{p} = \mathbf{w}(\mathbf{p}^*;\mathbf{G}) \propto \mathbf{G}\mathbf{p}^*.$$

The warping function has important properties:

- The identity element $\mathbf{I}$: $\mathbf{w}(\mathbf{p};\mathbf{I}) = \mathbf{p},\ \forall \mathbf{p} \in \mathbb{P}^2$;
- The composition of two actions corresponds to the action of the composition:

  $$\mathbf{w}(\mathbf{w}(\mathbf{p};\mathbf{G}_1);\mathbf{G}_2) = \mathbf{w}(\mathbf{p};\mathbf{G}_1\mathbf{G}_2),\ \forall \mathbf{p} \in \mathbb{P}^2 \text{ and } \forall \mathbf{G}_1, \mathbf{G}_2 \in \mathbb{SL}(3)$$

- The inverse of an action corresponds to the action of the inverse:

  $$\mathbf{w}(\cdot;\mathbf{G})^{-1} = \mathbf{w}(\cdot;\mathbf{G}^{-1}),\ \forall \mathbf{G} \in \mathbb{SL}(3)$$

Homographic paramterisation: Special Linear Group $\mathbb{SL}(3)$

- Constrain $det(\mathbf{H} = 1)$ so that $\mathbf{H}$ belongs the *Special Linear* $\mathbb{SL}(3)$ group.
- The $\mathbb{SL}(3)$ is a Lie group and $\mathfrak{sl}(3)$ its Lie algebra.
- The elements of $\mathfrak{sl}(3)$ are 3×3 matrices of null trace.

$\{\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_8\}$ is a basis of $\mathfrak{sl}(3)$. Any matrix $\mathbf{A}(\mathbf{x}) \in \mathfrak{sl}(3)$ is a linear combination of the basis:

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^{8} x_i \mathbf{A}_i, \tag{22}$$

with $\mathbf{x} = [x_1, x_2, ..., x_8]^\top \in \mathbb{R}^8$ the minimal vector.
The exponential map is determined as:

$$\mathbf{G}(\mathbf{x}) = \exp\left(\mathbf{A}(\mathbf{x})\right) = \sum_{i=0}^{\infty} \frac{1}{i!} \left(\mathbf{A}_i(\mathbf{x})\right)^i . \tag{23}$$

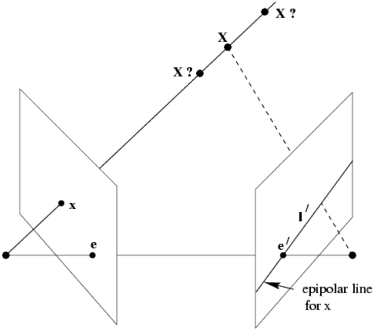
Homographic Transformations

Consider now a region-of-interest $\mathcal{R}^* \subseteq \mathcal{I}^*$ of a planar region of an image.
If the true homography $\overline{\mathbf{H}}$ is known then we have the following relation:

$$\overline{\mathbf{p}}^* = w(\mathbf{H}\overline{\mathbf{p}}^*) = [u, v, 1]^\top \qquad \forall \mathbf{p}^* \in \mathcal{R}^*$$

- Note that the warping function serves to normalise the homogeneous coordinate to 1.
- The set of transformed points is in general *scattered* after warping meaning that the transformed points dont correspond to a regular image grid. We will re-visit this when tranforming images.

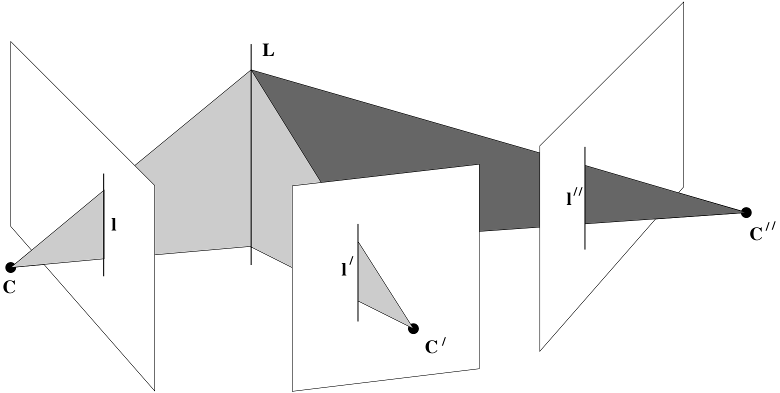
General Two View Geometry - The Fundamental Matrix



- $\mathbf{F}$ relates $\mathbf{x}$ to its epipolar line $\mathbf{l}' = \mathbf{F}\mathbf{x}$
- Since $\mathbf{x}'$ must be on $\mathbf{l}'$ we have $\mathbf{x}'^\top \mathbf{l}' = 0$
- Thus $\mathbf{x}'\mathbf{F}\mathbf{x} = 0$.
- $\mathbf{F}$ is singular, of rank 2, $det(\mathbf{F}) = 0$ and $\mathbf{F}$ has 7 d.o.f.

Three View Geometry - Trifocal tensor

The geometric relation between 3 views can be derived from the incidence relations for lines (3 planes intersection)



The three lines $\mathbf{l} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$ with $ax + bx + c = 0$ and

$\mathbf{l} = (a, b, c)^\top$ back project to planes: $\pi = \mathbf{P}^\top \mathbf{l} = \begin{pmatrix} \mathbf{l} \\ 0 \end{pmatrix}$

Three View Geometry - Trifocal tensor

Consider a line in each images back projecting onto 3 planes:

$$\pi = \mathbf{P}^\top \mathbf{l} = \begin{pmatrix} \mathbf{l} \\ 0 \end{pmatrix}$$

$$\pi' = \mathbf{P}'^\top \mathbf{l}' = \begin{pmatrix} \mathbf{A}^\top \mathbf{l}' \\ \mathbf{a}_4^\top \mathbf{l}' \end{pmatrix}$$

$$\pi'' = \mathbf{P}''^\top \mathbf{l}'' = \begin{pmatrix} \mathbf{B}^\top \mathbf{l}'' \\ \mathbf{b}_4^\top \mathbf{l}'' \end{pmatrix}$$

remembering that:

- $\mathbf{P} = \begin{bmatrix} \mathbf{A} & \mathbf{a} \end{bmatrix}$ bec. they are up to transformations
- **A** and **B** are the infinite homographies from the 1st to the 2nd and 3rd cameras resp.
- **a** and **b** are the epipoles arising from projecting 1st and 2nd camera centers resp.

The intersection constraint among image lines is obtained by analysing the points that lie on the line of intersection which are linearly dependent:

$$\mathbf{X} = \alpha \mathbf{X}' + \beta \mathbf{X}''$$

Which means that the $det(\mathbf{M}) = 0$ and $\mathbf{M}$ must be rank 2:

$$\mathbf{M} = [\pi_1, \pi', \pi''] = \begin{bmatrix} \mathbf{l} & \mathbf{A}^\top \mathbf{l}' & \mathbf{B}^\top \mathbf{l}'' \\ \mathbf{0} & \mathbf{a}_4^\top \mathbf{l}' & \mathbf{b}_4^\top \mathbf{l}'' \end{bmatrix}$$

The linear dependece means that: grouping the terms

$$\pi = \alpha\pi + \beta\pi$$

and since the bottom left hand element is 0 it follows that $\alpha = k(\mathbf{b}_4^\top \mathbf{l}'')$ and $\beta = -k(\mathbf{a}_4^\top \mathbf{l}')$. solving the equations for alpha and beta

Three View Geometry - Trifocal tensor

Applying this to the top row of $\mathbf{M}$ gives:

$$\begin{aligned} \mathbf{l} &= (\mathbf{b}^\top \mathbf{l}'')\mathbf{A}^\top \mathbf{l}' - (\mathbf{a}_4^\top \mathbf{l}')\mathbf{B}^\top \mathbf{l}'' \\ &= (\mathbf{l}''^\top \mathbf{b}_4)\mathbf{A}^\top \mathbf{l}' - (\mathbf{l}'^\top \mathbf{a}_4)\mathbf{B}^\top \mathbf{l}'' \end{aligned}$$

For each row $\mathbf{a}_i$ and $\mathbf{b}_i$ gives the matrix i of the trifocal tensor:

i: tensor notation

$$\mathbf{T}_i = \mathbf{a}_i \mathbf{b}_4^\top - \mathbf{a}_4 \mathbf{b}_i^\top$$

and the incidence relation can be written:

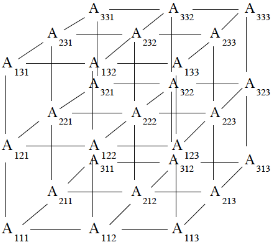
$$\mathbf{l}_i = \mathbf{l}'^\top \mathbf{T}_i \mathbf{l}''$$

and now this gives the relation between the other two lines and the first line which is in the world's frame

The set of three matrices $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$ constitue the trifocal tensor in matrix notation.

Three View Geometry - Tensor Notation

A tensor of rank 3 looks like this:



The trifocal constraint in tensor notation given by:

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k$$

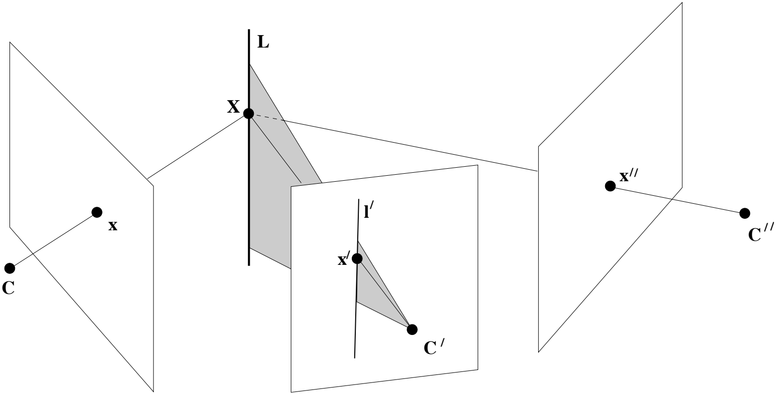
and the basic incidence relation becomes:

$$\mathbf{l}_i = \mathbf{l}'_j \mathbf{l}''_k \mathcal{T}_i^{jk}$$

Note the indicies are important but not the order.

Three View Geometry

Point transfer using the trifocal tensor:



$$\mathbf{x}''^k = \mathbf{x}^i \mathbf{l}'_j \mathcal{T}_i^{jk}$$

Will be used later for direct stereo SLAM.

Outline

Homographic Intensity Warping

Consider now a region-of-interest $\mathcal{R}^* \subseteq \mathcal{I}^*$ in the reference image.
If the true homography $\overline{\mathbf{H}}$ is known then we have the following relation:

$$\mathcal{I}^* = \mathcal{I}(w(\mathbf{p}^*, \mathbf{H})) \qquad \forall \mathbf{p}^* \in \mathcal{R}^*$$

where $\mathcal{I}(\mathbf{p})$ refers to an interpolation (bi-linear or cubic) of the current image at $\mathbf{p} = \mathbf{w}(\mathbf{p}^*; \mathbf{H}) = [u, v, 1]^\top$ since there is rarely a one-to-one pixel correspondence between pixels in each image.
Note that the intensity warping function also inherits the geometric properties.

Image Resampling - Nearest Neighbour

Every pixel value in the output image is set to the nearest input pixel value.

Pros	**Cons**
Very simple, fast	Some pixels get lost and others are duplicated
No new values are calculated by interpolation	Loss of sharpness
Fast, compared to Cubic Convolution resampling	

Following figure demonstrates the calculation of the new pixel value.

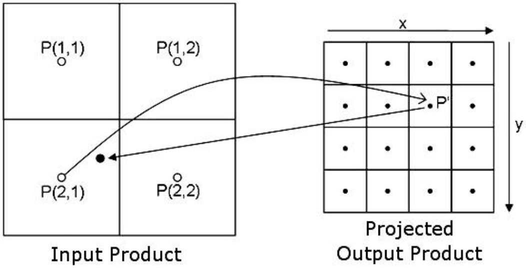
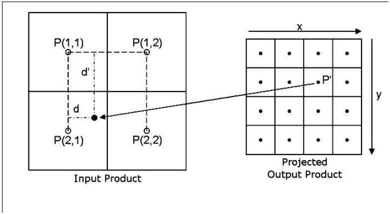


Image Resampling - Bi-linear interpolation

Calculation of the new pixel value is performed by the weight of the four surrounding pixels.

Pros	**Cons**
Extremas are balanced	Less contrast compared to Nearest Neighbour
Image losses sharpness compared to Nearest Neighbour	New values are calculated which are not present in th

Following figure demonstrates the calculation of the new pixel value.



The bilinear interpolation is performed as:

$$\mathbf{I}_w(u,v) = \mathbf{I}(1,1)(1-d)(1-d') + \mathbf{I}(1,2)d(1-d') + \mathbf{I}(2,1)d'(1-d) + \mathbf{I}(2,2)dd'$$

Image Resampling - Cubic convolution

Calculation of the new pixel value is performed by weighting the 16 surrounding pixels.

Pros
Extremas are balanced
Image is sharper compared to Bi-linear Interpolation

Cons
Less contrast compared to Nearest Neighbour
New values are calculated which are not present in the i
Slow, compared to Nearest Neighbour resampling

Following figure demonstrates the calculation of the new pixel value.

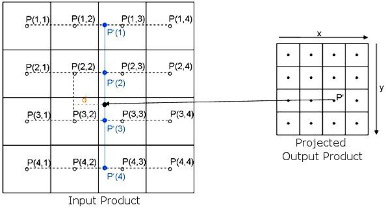


Image Resampling - Cubic convolution

The bi-linear interpolation is performed as:

$$
\begin{aligned}
\mathbf{I}_w(k) = \; & \mathbf{I}(k,1)(4 - 8(1+d) + (1+d)^2 - (1+d)^3) + \\
& \mathbf{I}(k,2)(1 - 2d^2 + d^3) + \\
& \mathbf{I}(k,3)(1 - 2(1-d)^2 + (1-d)^3) + \\
& \mathbf{I}(k,4)(4 - 8(2-d) + 5(2-d)^2 - (2-d)^3
\end{aligned}
$$

In the first step the average value for each line is calculated, afterwards the new pixel value is calculated with the four new average values $\mathbf{I}_w(1) - \mathbf{I}_w(4)$ similar to the preceding calculation.

Matlab - Homography warping example

Outline

Andrew Comport

Objectif - Estimation of unknown Image projection parameters

- The image of a point is modeled by:

$$\overline{\mathbf{p}} = \frac{1}{Z}\mathbf{K}\Pi\mathbf{T}(\mathbf{x})\overline{\mathbf{P}}, \quad \Pi\mathbf{T}(\mathbf{x}) = [\mathbf{R}, \mathbf{t}] \in \mathbb{R}^{3\times4} \tag{24}$$

- If all parameters are known then the projected point gives the correct image-point.
- If not we minimise the error $\mathbf{e} = \overline{\mathbf{p}}^* - \overline{\mathbf{p}}$ between the image-point $\overline{\mathbf{p}}^*$ and the projected model-point $\overline{\mathbf{p}}$.
- Estimate unknown paramters : Intrinsic camera parameters $\mathbf{K}$, Camera pose $\mathbf{T}(\mathbf{x})$, 3D scene structure Z, etc.

Non-linear Estimation - Linearisation

- Taylor series expansion of $\mathbf{e}(\mathbf{x})$:

  $$\mathbf{e}(\mathbf{x}) = \mathbf{e}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0},\mathbf{x})\mathbf{x} + \mathbf{O}(||\mathbf{X}||^3)$$

  where

  $\mathbf{J}(\mathbf{x}) = \nabla_{\mathbf{x}}\mathbf{e}(\mathbf{x})$ is the Jacobian matrix of dimension nx$n$

  and

  $\mathbf{M}(\mathbf{x}_1,\mathbf{x}_2) = \begin{bmatrix} \frac{\partial^2 \mathbf{e}_1(\mathbf{x}_1)}{\partial \mathbf{x}_1^2}\mathbf{x}_2 & \dots & \frac{\partial^2 \mathbf{e}_n(\mathbf{x}_1)}{\partial \mathbf{x}_1^2}\mathbf{x}_2 \end{bmatrix}^\top$ composed of nxn Hessian matrices of size n x n $\times$ 6.

- Solve iteratively by gradient descent.

Image Jacobian of a Point - Interaction Matrix

- The motion of a 3D point is modeled by:

$$\dot{\mathbf{P}} = \boldsymbol{\omega} \times \mathbf{P} + \boldsymbol{\upsilon} \tag{25}$$

- Differentiating the perspective projection $\mathbf{p} = \frac{f}{Z}\mathbf{P}$ gives:

3D point

rate of change of pixel →

$$\dot{\mathbf{p}} = \frac{1}{\lambda}\dot{\mathbf{P}} - \frac{\dot{\lambda}}{\lambda}\mathbf{p}, \quad \lambda = \frac{Z}{f} \tag{26}$$

- Substituting (25) into (26), the image Jacobian is then:

v (linear velocity　　w (rotational velocity

$$\mathbf{J} = \begin{array}{c} u \\ v \end{array} \begin{bmatrix} \frac{f}{Z} & 0 & -\frac{x}{Z} & -\frac{xy}{f} & \frac{f^2+x^2}{f} & y \\ 0 & \frac{f}{Z} & -\frac{y}{Z} & -\frac{(f^2+y^2)}{f} & \frac{xy}{f} & x \end{bmatrix} \tag{27}$$

only the linear velocity depends on the 3d point while the rotational one does not

pixe p(dot) = J * [v w]'

Non-linear Estimation - Newton

- Newton minimisation compute:

$$\mathbf{x} = -\mathbf{Q}^{-1}\mathbf{J}(\mathbf{0})^\top \mathbf{e}$$

where

$$\mathbf{Q} = \mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0}) + \sum_{i=0}^{n} \frac{\partial^2 \mathbf{e}_i(\mathbf{x})}{\partial \mathbf{x}^2}\Big|_{\mathbf{x}=0} \mathbf{e}_i$$

but it requires computing the *mn* Hessian matrices which is computationally expensive.

Non-linear Estimation - Guass-Newton

- Gauss Newton: approximate the Hessian and solve:

$$\mathbf{x} = -\left(\mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0})\right)^{-1} \mathbf{J}(\mathbf{0})^\top \mathbf{e}(\mathbf{x})$$

which corresponds to taking the pseudo-inverse $\mathbf{J}^+ = (\mathbf{J}^\top \mathbf{J})^{-1} \mathbf{J}^\top$.

Non-linear Estimation - Levenberg-Marquardt

- The Levenberg-Marquardt technique:

$$\mathbf{x} = -\left(\mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0}) + \lambda \mathbf{I}\right)^{-1} \mathbf{J}(\mathbf{0})^\top \mathbf{e}(\mathbf{x})$$

- The damping factor λ can be adjusted according to the curvature of the function so that larger steps are taken when the gradient is smaller so as to avoid slow convergence in the direction of small gradient by using $\lambda diag(\mathbf{J}(\mathbf{0})^\top \mathbf{J}(\mathbf{0}))$ for the second term.

Non-linear Estimation - Inverse compositional

- Inverse compositional:

$$\mathbf{J}(\mathbf{0}) \approx \mathbf{J}(\mathbf{x})$$

which allows to compute the Jacobian and its pseudo-inverse only once.

Non-linear Estimation - Efficient Second Order

- Efficient Second Order Minimisation (ESM):

$$\mathbf{e}(\tilde{\mathbf{x}}) \approx \mathbf{e}(\mathbf{0}) + \frac{1}{2}(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\tilde{\mathbf{x}}))\tilde{\mathbf{x}}$$

Robust Estimation

- **Objective:**
- Handle noise, miss-tracking and outliers in visual features extraction
  - Matching error between current and desired positions leads to positioning error or control law divergence
- Converge upon the correct position even in the presence of image outliers.

Robust Estimation Techniques

- **Local solution:**
- Robust feature extraction
  - Impossible to model a general disturbance processes [Tomasini98][Brautigam98]
- **Global solution:**
- Many pose estimations for small sampled data sets
  - Robust statistics: L-Meds
  - Computer Vision: RANSAC
- One estimation procedure from entire data set
  - Robust statistics: M-estimation.

Random Sample and Consensus: RANSAC

1. Randomly select a sub-sample S_i of size s from the entire data and compute the model parameters (i.e. $s = 4$ points in the case of the pose $\mathbf{x}$)
2. Determine the number of data η_i which are within the threshold bound τ around the solution of set i. These point are considered to be in consensus with this model.
3. *If* the number of data η_i is greater than some threshold, then re-estimate the model using all the data and stop.
   Else select a new sub-sample and goto 1.
4. After N trials the largest consensus set S_i is selected.

Least median squares: LMedS

1. Draw N subsamples $\mathbf{s}_J$, $J = 1 \dots N$ of d independent visual features. The maximum number of subsamples is $N_{\max} = \binom{nm}{d}$, therefore if nm is large $N_{\max}$ may be very large!
2. For each subsample $\mathbf{s}J$, the unknown parameters $\mathbf{x}_J$ are computed according to:
$$\mathbf{x}_J = -\mathbf{J}^{+}_{\mathbf{s}_J}(\mathbf{e})$$
3. For each $\mathbf{x}_J$, the median of square residuals is determined, denoted M_J, with respect to the whole set of features:
$$M_J = Med_{i=1 \dots n}\big(\mathbf{J}_{s_i}\mathbf{x}_J + (e_i)\big)^2$$
4. The minimal value M^* is retained amongst all N M_J's along with each corresponding $\mathbf{x}_J$.

Robust M-estimation

Algorithm to reject the outliers,

- Handle noise, miss-tracking and outliers in visual features extraction - global solution.
- The new residue is given by:
  - $\mathbf{e} = \sum_i \rho(\boldsymbol{\mathcal{I}}_i - \boldsymbol{\mathcal{I}}_i^*)$,
  - where $\rho(.)$ is a robust function (M-Estimation) [Huber81]
- Scale estimation
  - the Median Absolute Deviation:
    $\mu_{\mathcal{R}} = Median_i(\mathbf{e})$
    $\sigma_{\mathcal{R}} = Median(|\mathbf{e} - \mu_{\mathcal{R}}|)$

Influence fucntions

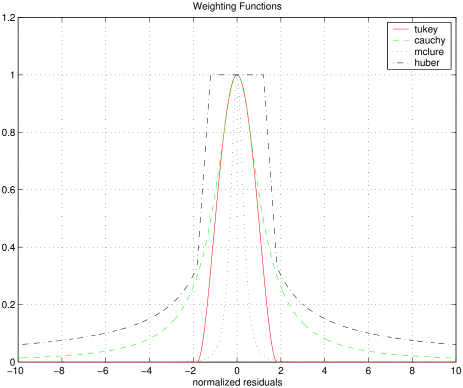
- Tukey's influence function:

$$\phi(u) = \begin{cases} u(C^2 - u^2)^2, & if\,|u| \leq X \\ 0 & else \end{cases}$$

- Weight functions:

Huber: converges faster and it tends to infinity but it takes some computational power since it doesn't fully reject the outliers

Tukey is very strict on the outliers but it can reject some good data



Robust Estimation

- Robust iteratively re-weighted least squares:
  - M-estimation and
  - Iteratively Re-weighted Least Squares (IRLS),
  - Uses all data and works better with the entire image.
- The optimsation equation which minimizes $(\mathbf{s} - \mathbf{s}^*)$ is given by:

$$\mathbf{x} = -\lambda(\mathbf{DJ})^+(\mathbf{s} - \mathbf{s}^*)$$

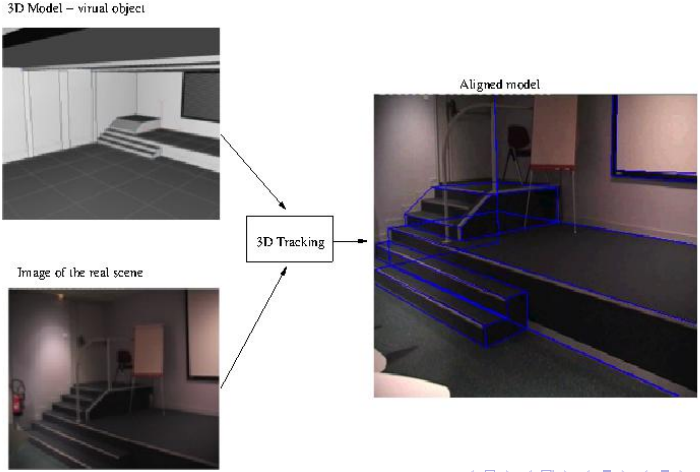
  - where

$$\mathbf{D} = \begin{bmatrix} w_1 & & 0 \\ & \dots & \\ 0 & & w_n \end{bmatrix}$$

Outline

What is 3D localisation and tracking?

- Determine the view-point between a camera and its environment.



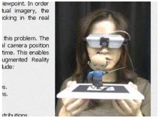
3D Position Sensors: Active, Passive, Intrusive

- Target mounted devices: GPS, gyroscopes, accelerometers, magnetic.
- Active : LEDs, range finders, ultrasonic, striped light projectors,
- Passive devices: multiple cameras, infrared sensors.
- Intrusive: markers [ARToolkit].

GPS

Razor 9dof inertial sensor

Velodyne LIDAR LIDAR



ARToolkit markers

Monocular vision sensor

- Pros
  - Versatile,
  - Non-intrusive,
  - Cheap,
  - Provides spatio-temporal RGB measurements of the environment.
- Cons
  - Complex modeling,
  - 2D Spatio-temporal information of a 3D world.

Webcams - Unibrain FW

Professional sensors - Point Grey Flea 3

Active RGB-D sensors: Kinect, Swissranger

- Pros
  - Versatile,
  - Cheap,
  - Provides spatio-tempora RGB-D measurements of the environment.
- Cons
  - Doesnt work in sun-light (IR-light),
  - Intrusive - projection onto the scene (although limited to non-visible spectrum) .

ASUS Xtion Pro Live

Microsoft Kinect

SwissRanger

Passive RGB-D vision sensors: Stereo, Plenoptic camera

- Pros
  - Versatile,
  - Non-intrusive,
  - Less cost than profesional scanners ,
  - Provides spatio-temporal RGB-D measurements of the environment.
- Cons
  - Difficult dense matching in textureless environments,
  - No embedded matching processors (for the moment).

Stereo Camera

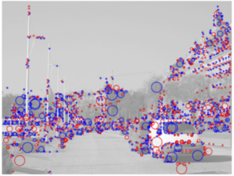
Raytrix light feild camera

Lytro plenoptic cam

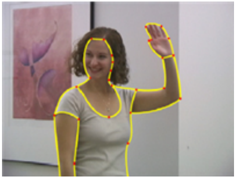
2D Visual Tracking

- 2D feature matching
  - Local Interest Points - Maximum Image Gradient [Stephens and Harris 88]
  - Scale Invariant Feature (SIFT) [Tommasini98, Lindeberg98, Lowe04],
  - Active Contour tracking [Bouthemy89, Blake93, Hager96],

Harris features



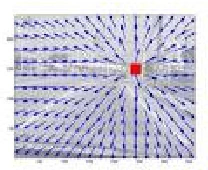
SIFT features shown at their detected scale



Active contours

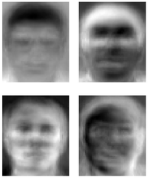
2D Visual Tracking

- Direct 2D Tracking
  - Optic Flow [Horn and Schun 81]
  - Local smoothing and global image registration [Lucas-Kanade]
  - KLT feature tracking [Tomasi and Kanade 91]
  - Principal component Analysis [Murase95]
  - Region of Interest Tracking - Affine [Hager98], Planar Homography [Malis04]



Optical flow

Region of interest tracking - Affine,



Principal componenet image analysis

2D Visual Tracking and Matching

- **2D Tracking**
- **Pros**
  - Good for learning in unknown environments,
  - Can be used where little prior information is available,
  - Local tracking methods are efficient,
  - Low level elements are flexible to manipulate within Bayesian frameworks.
- **Cons**
  - No 3D information available,
  - 3D occlusions difficult to handle,
  - Imprecision in position estimation,
  - Little resistance to external noise.

3D Model-based Tracking

- **3D model known apriori**
  - Reconstructed surfaces, Depth maps, Contours, Template images, Texture [Jurie02, Vachetti04, Pressigout04], Planar structure [Simon02], Structure from motion, Active vision.
- **Structure**[Lowe91, Gennery92, Koller94, Wunsch97, Berger98, Haag98, Drummond02]
  - Pertinent information about the position of the object,
  - Rigid 3D motion constraint,
  - Improved robustness, Handles self occlusion, Improved computational efficiency.

3D Lines

Cylinders, Circles

Contours

Pose Estimation

- **Depends on correspondences** between the model and image features.
- **Estimation techniques**
  - Purely geometric, perspective-n-point [Ganapathy84, Dhome89, Horaud89 Haralick91,Navab93]
  - Linear least squares and Non-linear iterative methods, Gauss-Newton, Levenberg-Marquardt [Lowe87, Lu00, Drummond02]
  - Combined numerical and iterative [Dementhon95]
- **Various Features**
  - Points [Fischler81, Haralick89, Dementhon95]
  - Lines [Lowe97, Dhome89, Haralick89, Gennery92]
  - Intensity-based [Koller97]
  - Combinations [Liu90, Phong95, Marchand02]
  - Circles, Cylinders, Contours

Pose and Visual Servoing

- **Visual servoing:**
  - Move a camera in order to observe an object at a given position in the image.
  - Non-linear image-based minimization of image features.
  - The robot reaches a desired pose defined by a target image.

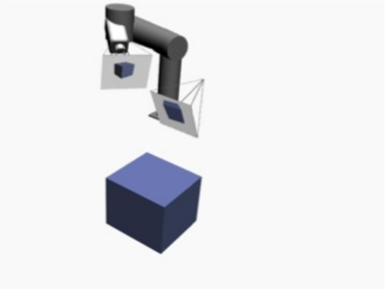
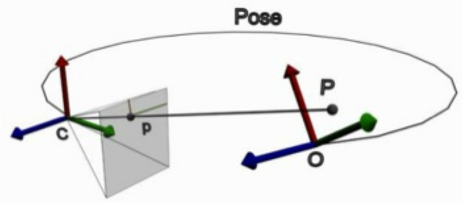


Image-based Visual Servoing

Pose Estimation: Non-linear Minimisation

- **Goal:**
  - Estimate the pose $\mathbf{T} \in SE(3)$ of an object with respect to the camera frame.
  - Example for point features:



  - Minimizing the error between the observation and the projection of the model in the image:

$$\widehat{\mathbf{T}} = \operatorname{argmin}_{\mathbf{T}} \sum_i \left(\mathbf{p}^* - \pi(\mathbf{TP})\right)^2$$

where $\mathbf{P}$ are the coordinates of the same points in the object frame.

Virtual Visual Servoing

- The objective is to minimize the error:

$$\mathbf{e} = (\mathbf{s}(\mathbf{x}(t)) - \mathbf{s}^*) = (\pi(\mathbf{x}(t), \xi, \mathbf{P}) - \mathbf{s}^*)$$

  where

  - $\mathbf{s}^*$ is the desired value of the visual features in the image,
  - $\mathbf{s}(\mathbf{x}(t))$ is the current value of the features for the pose $\mathbf{T}(\mathbf{x})$.

- The error is related to the camera velocity twist $\mathbf{x}$ by derivation using a first order Taylor series expansion:

$$\dot{\mathbf{s}} = \mathbf{J}_s\mathbf{x},$$

  where the Jacobian matrix $\mathbf{J}$ depends on the type of visual features $\mathbf{s}$.

Control Law

- Iterative minimization of the error by and exponential decrease:

$$\dot{\mathbf{e}} = -\lambda \mathbf{e}$$

- The control law which regulates the error $e \to 0$ is given by:

$$\mathbf{x} = -\lambda \mathbf{J}_{\mathbf{s}}^{+}(\mathbf{s}(\mathbf{x}) - \mathbf{s}^*)$$

- After convergence the new pose is updated as:

$$\mathbf{T} = e^{[\mathbf{x}]_\wedge} \mathbf{T}$$

- where $\mathbf{T}$ is the homogeneous representation of the pose vector $\mathbf{x}$ and $e^{[\mathbf{x}]_\wedge}$ is the exponential map of a twist to is pose representation.

Convergence and Stability

- Stability of the non-linear minimisation can be shown if the following criterion is true:

$$\widehat{\mathbf{J}}_{\mathbf{s}}^{+}\widehat{\mathbf{J}}_{\mathbf{s}}^{+} > 0, \forall t,$$

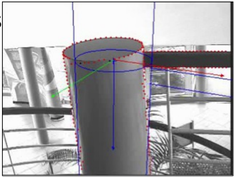
- The choice of $\mathbf{J_s}$ is important and different choices are possible:
  1. At solution : Calculated with the final estimate of the pose and image features:
  $$\widehat{\mathbf{J}}_{\mathbf{s}}^{+} = \widehat{\mathbf{J}}_{\mathbf{s}}^{+}(\mathbf{s}^*, \mathbf{x}^*)$$
  2. Forward compositional : Calculated with the current estimate of the pose and image features:
  $$\widehat{\mathbf{J}}_{\mathbf{s}}^{+} = \widehat{\mathbf{J}}_{\mathbf{s}}^{+}(\mathbf{s}^*(\mathbf{x}(t)))$$
  3. Inverse compositional : Calculated with the initial estimate of the pose and final or initial image features:
  $$\widehat{\mathbf{J}}_{\mathbf{s}}^{+} = \widehat{\mathbf{J}}_{\mathbf{s}}^{+}(\mathbf{s}(\mathbf{0}), \mathbf{x}(\mathbf{0}))$$
  $$\widehat{\mathbf{J}}_{\mathbf{s}}^{+} = \widehat{\mathbf{J}}_{\mathbf{s}}^{+}(\mathbf{s}^*, \mathbf{x}(\mathbf{0}))$$

1D/3D error function

- **Points**, **lines**, **circle**, **sphere**, **cylinder**, **moments** or different combinations of these features [Chaumette90]
- **Distance to a contour**
  - Stack each distance jacobian matrix so as to for a matrix representing the distance to the entire object:

$$\begin{bmatrix} \dot{\mathbf{s}}_1 \\ \vdots \\ \dot{\mathbf{s}}_n \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_n \end{bmatrix} \mathbf{x}$$

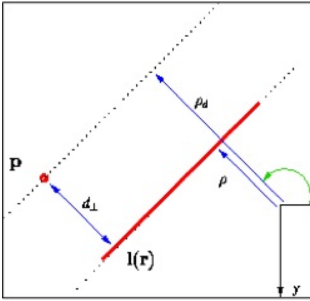
  - Many visual features may be considered in a simultaneous manner.



[ISMAR03]

Distance-based Jacobian

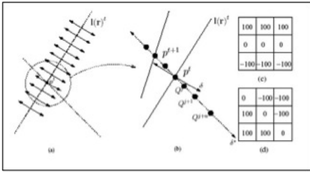
- **Distance point-to-line**
  $d_l = d_\perp(\mathbf{p}, \mathbf{l}(\mathbf{x})) = \rho(\mathbf{l}(\mathbf{x})) - \rho_d$,
  $x\cos\theta + y\sin\theta = \rho, \forall(x, y) \in \mathbf{l}(\mathbf{x})$,
  where $\mathbf{p}$: image point, $\mathbf{l}(\mathbf{x})$: 3D model at pose $\mathbf{x}$,
  $$\mathbf{J}_{\mathbf{d}_\perp} = \begin{pmatrix} \lambda_d \cos\theta \\ \lambda_d \sin\theta \\ \lambda_d \rho \\ (1+\rho^2)\sin\theta - \alpha\rho\cos\theta \\ -(1+\rho^2)\cos\theta - \alpha\rho\sin\theta \\ -\alpha \end{pmatrix}^\top$$

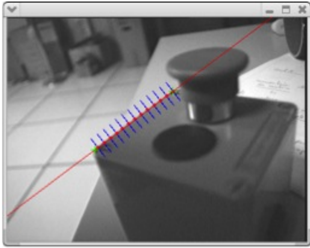


- **Distance point-to-cylinder**
- **Distance point-to-ellipse** - derive in a similar way

Local 1D Edge Tracking

- **Edges**
  - Regions of strong position information [Haralick81, Canny83],
  - Efficient 1D search along the normal,
  - Convolution efficiency,
  - Invariant to illumination changes,
- **Maximum Likelihood Framework** [Bouthemy89]
  - Model edges as spatio-temporal patches,
  - Oriented edge masks,
  - Sub-pixel precision.
- **Scale Invariant Edges**
  - Propagating likelihood pose estimation.
  - No detection threshold needed.
  - Handle edges with varying strength.





Robust Tracking

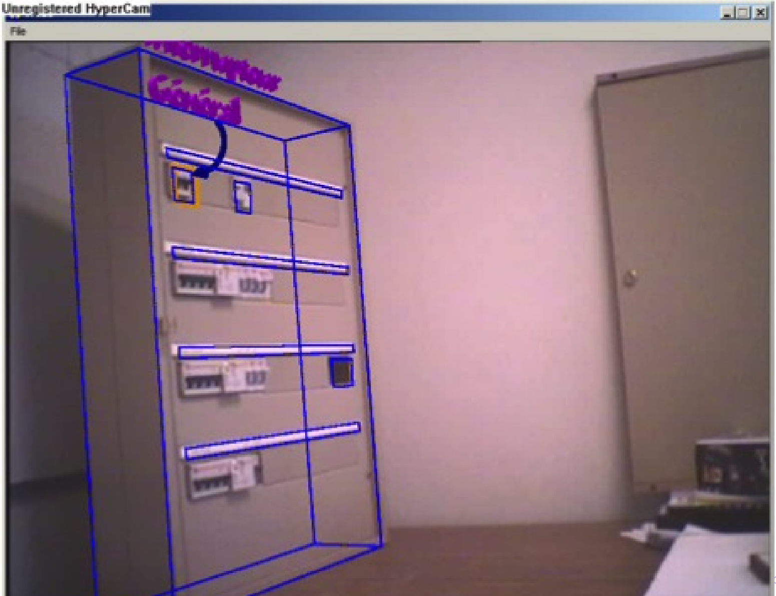
- Robust iteratively re-weighted least squares:
  - M-estimation and
  - Iteratively Re-weighted Least Squares (IRLS),
  - Uses all data and works better with the entire image.
- The optimsation equation which minimizes $(\mathbf{s} - \mathbf{s}^*)$ is given by:

$$\mathbf{x} = -\lambda(\mathbf{DJ})^+(\mathbf{s} - \mathbf{s}^*)$$

  - where

$$\mathbf{D} = \begin{bmatrix} w_1 & & 0 \\ & \dots & \\ 0 & & w_n \end{bmatrix}$$

Robust Tracking

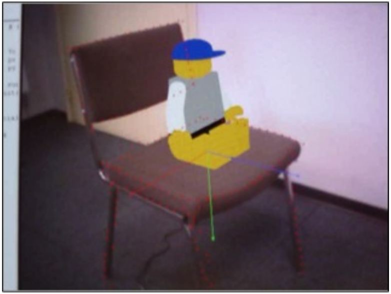


Robust Tracking with Augmented Reality

Augmented Reality

A real-time, robust method of pose calculation for augmented reality:

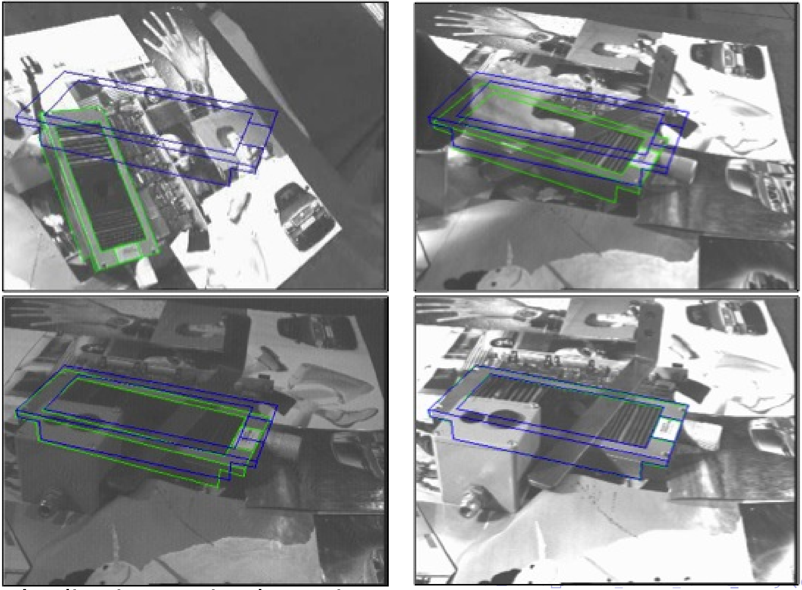
Application to an Augmented Reality game.



Application to an Augmented Reality game.

Interaction between virtual and real objects.

3D model-based tracking in visual servoing



Application to visual servoing

Planar patch tracking

- Planar target tracking problem in a video stream (small inter-frame motion).
- Use planar patch warping to define an error to optimise (inverse compositional):

$$\mathbf{e} = \mathcal{I}\left(w\left(\mathbf{p}^*; \widehat{\mathbf{H}}\mathbf{H}(\mathbf{x})\right) - \mathcal{I}^*(\mathbf{p}^*)\right)$$

The error is minimised using a least squares optimisation criterion:

$$O(\mathbf{x}) = \arg\min_{\mathbf{x}} \sum_{\mathbf{p}_i^* \in \mathcal{R}^*} (\mathbf{e}_i)^2$$

Planar patch tracking

- Solved iteratively for $\mathbf{x}$ by performing a Taylor series expansion of the cost function.
- Reference Jacobian computed once.
- Estimate obtined with a pseudo-inverse of the Jacobian Matrix $\mathbf{J}^{+} = (\mathbf{J}^{\top}\mathbf{J})^{-1}\mathbf{J}$:

$$\widehat{\mathbf{x}} = \mathbf{J}^{+}\mathbf{e}$$

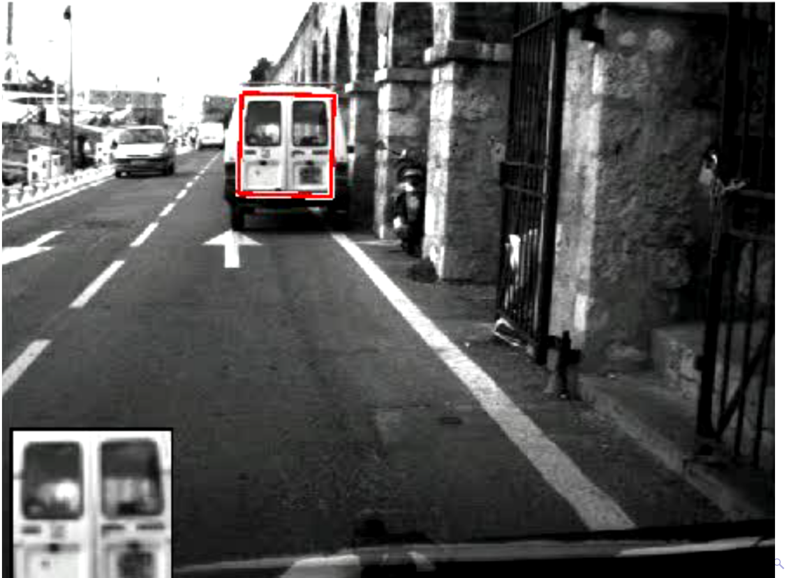
where $\mathbf{J} = \mathbf{J}_{\mathcal{I}}\mathbf{J}_{w}\mathbf{J}_{\mathbf{x}}$ is the reference Jacobian matrix decomposed into the spatial image intensity gradient $\mathbf{J}_{\mathcal{I}}$, the geometric warping gradient $\mathbf{J}_{w}$ and the gradient of the homography with respect to its Lie algebra parametrisation $\mathbf{J}_{\mathbf{x}}$.

- The planar homography update:

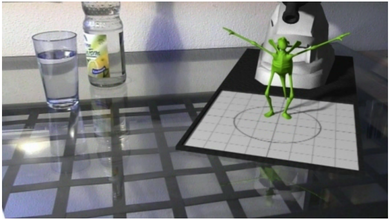
$$\widehat{\mathbf{H}} \leftarrow \widehat{\mathbf{H}} e^{(\mathbf{A}(\mathbf{x}))}.$$

- Convergence is stopped at a predetermined threshold $|\mathbf{x}|$ or $|\mathbf{e}|$, whichever comes first.

Example: Planar patch tracking



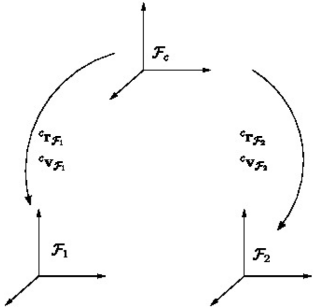
AR Examples: Planar patch tracking



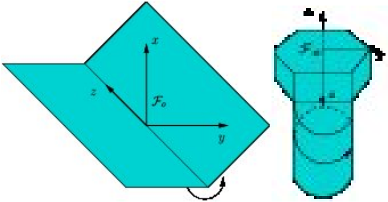
Multi-object tracking

Andrew Comport

Example: Pose calculation for two components requires 2 X 6 degrees of freedom.



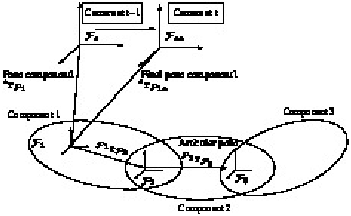
Constrain a 3D model composed of several rigid parts. In this example, the constraint matrices for two different types of class 1 links are given.



(a) A class one rotational hinge link, where component 1 and component 2 are both rectangular components , (b) A class one helical link where component 1 is a screw and component 2 is a square plate that is not shown in the diagram.

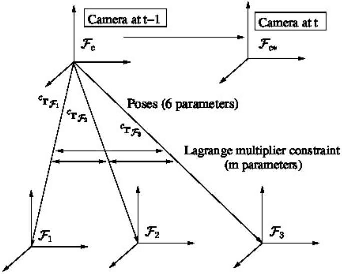
Kinematic chain - velocity contraints

Kinematic chain method: The pose of an articulated object is determined via a kinematic chain of rigid bodies extending to sub components. Each circle represents a rigid component. The root component is rigidly linked to the camera via a pose and each subsequent component is linked by their articulated dof.



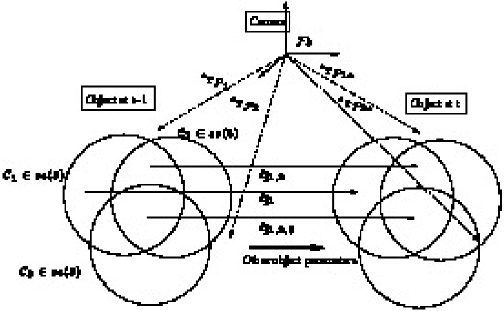
Lagrange Multiplier Contraints

Lagrange Multiplier method: The poses between the camera and each part of the object are calculated directly in a first step. Constraints between the components are then enforced in a second minimization step via Lagrange multipliers. In this case, a kinematic chain or tree is also used to represent the articulated structure.

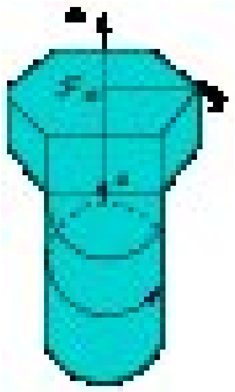


Kinematic set method - velocity contraints

Kinematic set method: Each circle represents a component as set of velocities in $se(3)$ which are rigidly linked together. The intersection of these sets define the minimal subspaces that correspond to the minimal parameters corresponding to the intersection between individual velocities.



Example



Outline

Mapping Introduction

Mapping example: large scale urban environments.

- Poor precision of low cost GPS units.
- GPS satellite occlusion.
- Drift of classical wheel encoder odometry.
- **Computer vision**: Robustness and accuracy, but large amount of information.

Collecting visual and geometric information during a learning phase and use it for online localization.

- Model based representation.
- Visual memory.

Model Based Representation

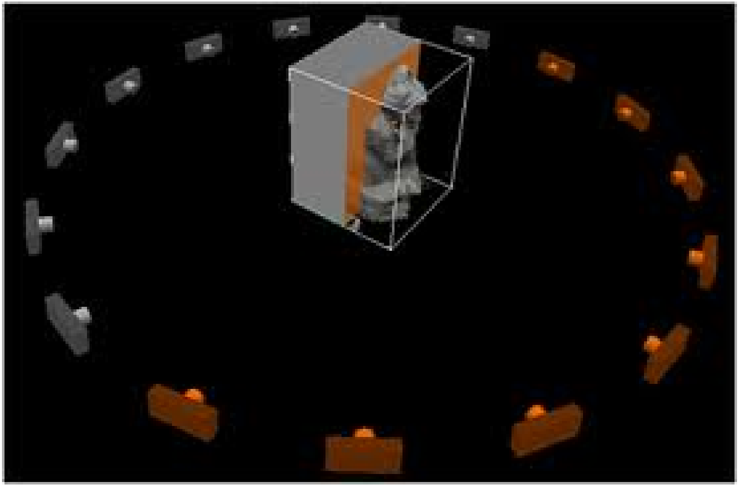
Using 3D wire-frame CAD model to register a current camera image.

Inaccuracy:

- Reconstructions errors.
- Photometric inconsistency.
- Yield bad localization quality or impossible registration.
- **Not suitable for image based registration**.

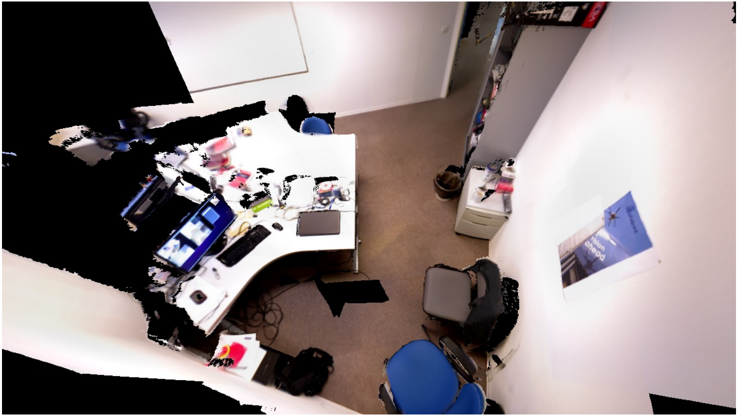
Figure: A 3D CAD model

Mapping using a volumetric voxel-based octree



A voxel-based 3D object model obtained from Space carving. Surfaces can be represented in this space by the Truncated Signed Distance Funcion.

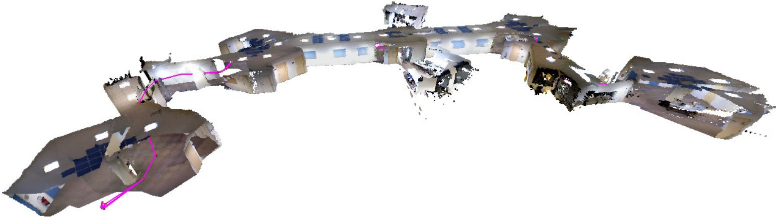
Tracking from a keyframe graph



Bird eye view of a reconstructed office rendered in real-time using a multi-key-frame fusion approach.

A dense key-frame map representation of a building

Bird eye view



Side view

Dense reconstruction of an entire floor obtained in real-time from a 100 meters trajectory containing 67 key-frames.

Ego-centered Spherical Representation

Advantages:

1. Generic model: Any type of online sensor can be registered.
2. Omnidirectional.
3. Locally very accurate.
4. Photometric consistency: enhances image-based techniques' performances.

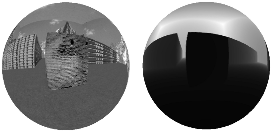


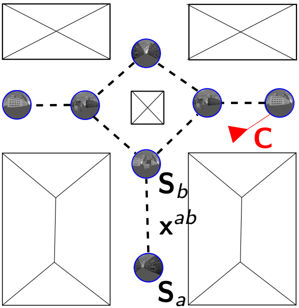
Figure: A Spherical image and its associated depth map

Spherical ego-centered representation

Global Representation: Graph

$\mathcal{G} = \{\mathbf{S}_1, \ldots, \mathbf{S}_n; \mathbf{x}_1, \ldots, \mathbf{x}_m\}$,
$\mathbf{x}_n \in \mathbb{R}^6$: 6 d.o.f. twist between each sphere.

- Learning phase - high computation and low rate.
- A set of augmented spherical images sampled along a trajectory.
- Edges: $\mathbf{x}^{ab}$
- Nodes: $\mathbf{S}_{1\ldots n}$



Local representation: Augmented sphere

Local representation: Augmented sphere

$$\mathbf{S} = \{\mathcal{I}_s, \mathcal{P}_{\boldsymbol{s}}, \mathbf{Z}_s, \mathbf{W}_s\}$$

Description

- $\mathcal{P}_{\boldsymbol{s}}$: Unit sphere sampling.
- $\mathcal{I}_s$: Photometric spherical image.
- $\mathbf{Z}_s$: Depth-map.
- $\mathbf{W}_s$: Saliency image (pixel ordering).

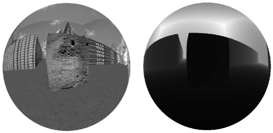


Figure: A Spherical image and its associated depth map.

Acquisition: Commercial spherical imaging sensors

Omnidirectional camera [?]

- Poor and non uniform spatial resolution.
- Limited vertical FOV.

Image stitching [?]

- Multiple perspective cameras.
- High resolution panoramas.
- Unique center of projection **approximation**.
- Parallax artefacts.

Depth information **cannot** be extracted in a single frame.

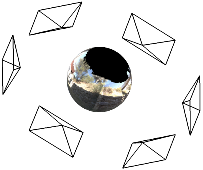
[?] S.K Nayar, Catadioptric omnidirectional camera, CVPR 97.
[?] R. Szeliski, Image alignment and stitching: a tutorial, 06

Multi-camera system

Multi-camera system with baselines - version 1

- 6 wide angle (125°) **stereo** cameras with wide baselines (65 cm).
- 360° of overlap.
- Stereo dense matching [?] for depth extraction.

Depth information **can** be extracted in a single frame - constrains general 6dof motion estimation.

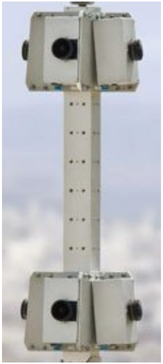


[?] H. Hirschmuller, Stereo processing by semi-global matching and mutual information, PAMI 08.

Multi-camera system - version 2

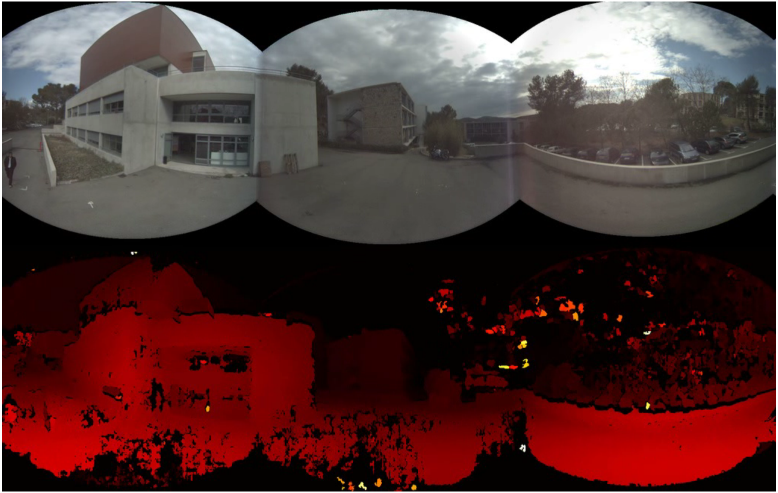
Multi-camera system with baselines

- 6 wide angle (125°) **stereo** cameras with a single baseline (50 cm).
- 360° of overlap.
- Spherical panorama rotation calibration [?].
- Initial stereo dense matching for bootstrapping depth extraction.
- Spherical RGB-D localisation for extrinsic parameters.



[?] S. Lovegrove and A. Davison, Real-time spherical mosaicing using whole image alignment, Computer Vision ECCV, 2010, pp. 7386. .

An augmented spherical image sequence



Automatic dense mapping: Sphere positioning

Accurate dense visual odometry [?], **direct** minimisation of intensity errors:

Multi-camera robust dense minimisation

$$\mathbf{e} = \sum_{i=1\ldots6} \rho\left(\mathcal{I}_i\left(w\left(\mathbf{T}(\mathbf{x}_i^c)\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); \mathcal{P}_s, \mathbf{Z}_s\right)\right) - \mathcal{I}_s(\mathcal{P}_s, \mathbf{Z}_s)\right),$$

- $\mathcal{I}_s$: reference sphere intensities.
- $\mathcal{I}_i$: perspective images,
- $w(.)$ warping function
- $\widehat{T} \in \mathbb{SE}(3)$: initial motion estimation.
- $\mathbf{x} \in \mathbb{R}^6$: unknown 3D motion.
- $\rho(.)$: robust outlier rejection from M-estimation.

[?] A.I. Comport, E. Malis, and P. Rives, Real-time quadrifocal visual odometry, IJRR 2010.

New sphere selection

Accurate robust dense multi-camera visual odometry [?] :

Direct minimisation of intensity errors

$$\mathbf{e} = \sum_{i=1\ldots6} \rho\left(\mathcal{I}_i\left(w\left(\mathbf{T}(\mathbf{x}_i^c)\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); \mathcal{P}_s, \mathbf{Z}_s\right)\right) - \mathcal{I}_s(\mathcal{P}_s, \mathbf{Z}_s)\right),$$

Improve depth maps over time

- Maintain as long as possible the reference sphere.
- Integrate the dense matching incrementally in time [?],

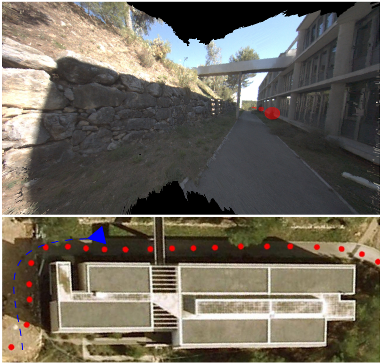
Robust update criteria: Median absolute deviation

$$\lambda < Median(\mathbf{e} - Median(\mathbf{e})).$$

[?] Tykkala, T.M. and Comport, A.I, A Dense Structure Model for Image Based Stereo SLAM, ICRA 11

Immersive navigation

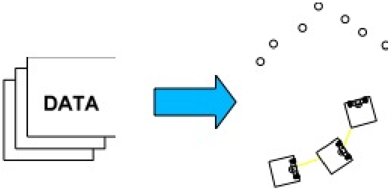
- Virtual navigation using the spherical graph.
- Photo-realistic rendering.

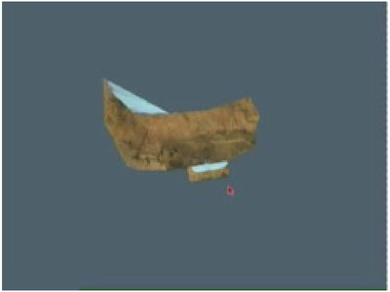


Outline

Post production approaches (not real-time)

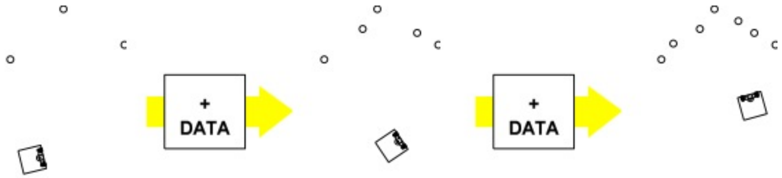
- **Computer vision community:**
  - Structure from motion,
  - 3D model reconstruction,
  - Bundle adjustment over long sequences.





An old example from Oxford's Visual Geometry Group [Torr, Zisserman]

Sequential real-time processing



- Simultaneous Localisation and Mapping (SLAM)
- Building a long-term map by propagating and correcting uncertainty,
- Mostly used in simplified 2D environments with specialised sensors such as laser range-finders, gps, sonars, etc...
- Robotic community:
  - Probabilistic approaches have been studied extensively
  - For example: work of Durrant-Whyte and Thrun.
  - Primarily aimed at autonomous robots, often slow moving with additional control input, e.g. odometry

Classical approaches to Visual SLAM

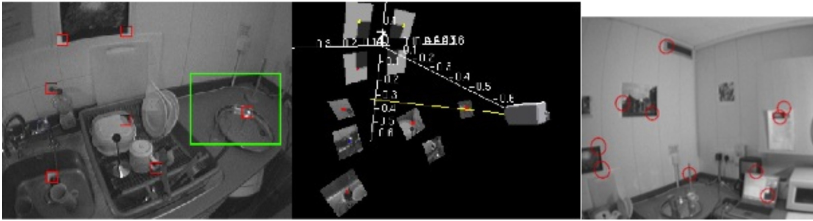
- Davison, ICCV 2003
  - Traditional SLAM approach (Extended Kalman Filter)
  - Maintains full camera and feature covariance
  - Limited to Gaussian uncertainty only
- Nister, ICCV 2003
  - Structure-from-motion approach (Preemptive RANSAC)
  - Frame-to-frame motion only
  - Drift: No repeatable localisation

Detection and tracking

- The predictor-corrector framework enables active tracking of features.
- Measurement uncertainty regions derived from the filter constrain the search for matching features
  - reduces image processing operations: real-time performance
- Contrast with detection methodologies - detect all potential features and find best set of matches with previous frame (e.g. using optimisation, RANSAC, etc).
- NB: as uncertainty in camera pose increases: search regions increase $\rightarrow$ detection $+$ matching takes over.

Case Study 1: MonoSLAM

- Davison, Mayol and Murray 2003.
- Initialisation with known target
- Extended Kalman Filter
  - *Constant velocity* motion model
  - Image patch features with Active Search
- Automatic Map Measurement
- Particle filter for initialisation of new features



MonoSLAM State Vector

- Camera state representation: 3D position $\mathbf{t}$, orientation $\mathbf{q}$ in quaternions, velocity $\mathbf{v}$ and angular velocity $\boldsymbol{\omega}$:

$$\mathbf{x}_{\upsilon} = \begin{pmatrix} \mathbf{t} \\ \mathbf{q} \\ \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix}$$

- Each feature state is a 3D position vector:

$$\mathbf{y}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$$

First Order Uncertainty Propagation

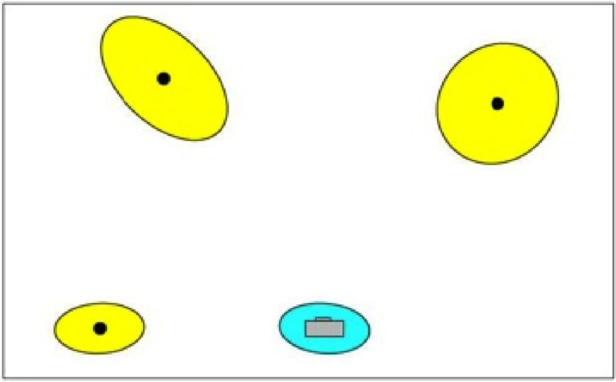
covariance matrix

$$\widehat{\mathbf{x}} = \begin{pmatrix} \widehat{\mathbf{x}}_v \\ \widehat{\mathbf{y}}_1 \\ \widehat{\mathbf{y}}_2 \\ \vdots \end{pmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xy_1} & \mathbf{P}_{xy_2} & \cdots \\ \mathbf{P}_{y_1x} & \mathbf{P}_{y_1y_1} & \mathbf{P}_{y_1y_2} & \cdots \\ \mathbf{P}_{y_2x} & \mathbf{P}_{y_2y_1} & \mathbf{P}_{y_2y_2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

- PDF over robot and map parameters is modelled as a single multi-variate Gaussian and we can use the Extended Kalman Filter.

Extended Kalman Filter: Prediction Step

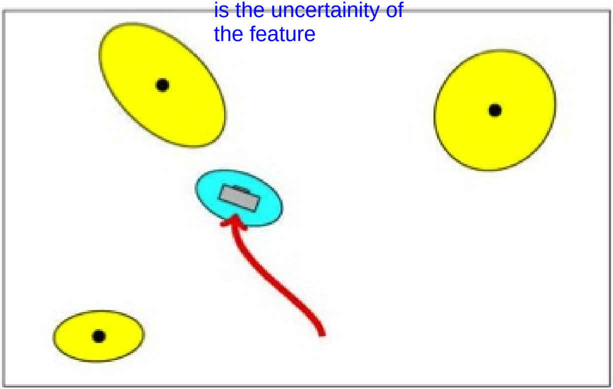
- Time Update
  1. .
  2. .



Extended Kalman Filter: Prediction Step

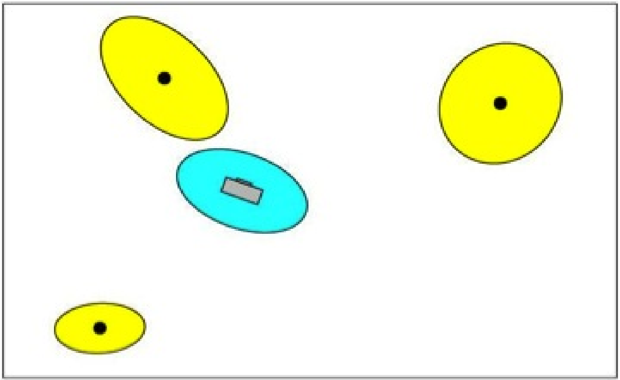
- Time Update
  1. Estimate new location
  2. .

the yellow ellipsoid
is the uncertainity of
the feature



Extended Kalman Filter: Prediction Step

- Time Update
  1. Estimate new location
  2. Add process noise



Extended Kalman Filter: Prediction Step

- Time Update
  1. Project the state ahead:

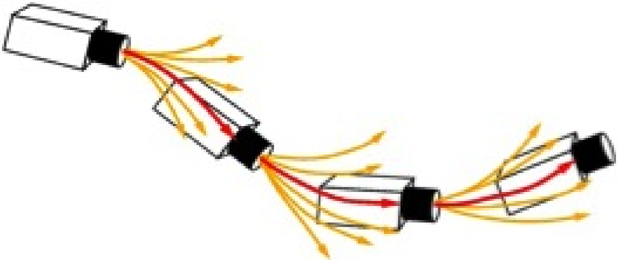
$$\widehat{\mathbf{x}} = f(\widehat{\mathbf{x}}, \mathbf{u})$$

  2. Project the error covariance ahead:

$$\mathbf{P}_{new} = \frac{\partial f}{\partial \mathbf{x}} \mathbf{P} \frac{\partial f}{\partial \mathbf{x}}^\top + \mathbf{Q}$$

where $\mathbf{Q}$ is the additive process noise.

Extended Kalman Filter: Motion Models



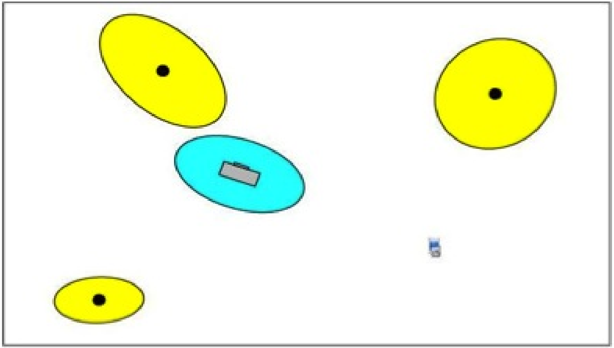
- Constant velocity
  - Assume bounded, Gaussian-distributed linear and angular acceleration:

adding

$$f_v = \begin{pmatrix} \mathbf{t} \\ \mathbf{q} \\ \mathbf{v} \\ \omega \end{pmatrix} = \begin{pmatrix} \mathbf{t} + (\mathbf{v} + \mathbf{V})\delta t \\ \mathbf{q} \times \mathbf{q}((\omega + \mathbf{\Omega})\delta t) \\ \mathbf{v} + \mathbf{V} \\ \omega + \mathbf{\Omega} \end{pmatrix}$$

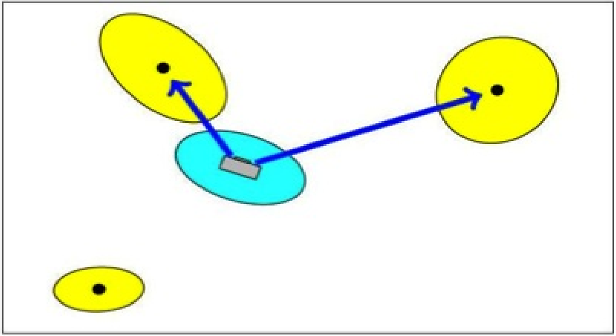
Extended Kalman Filter: Update Step

- Measurement Update
  1. .
  2. .



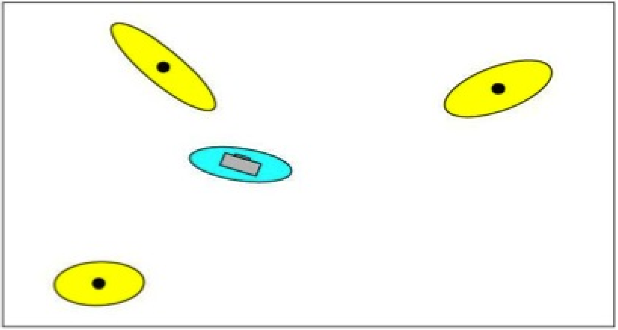
Extended Kalman Filter: Update Step

- Measurement Update
  1. Measure feature(s),
  2. .



Extended Kalman Filter: Update Step

- Measurement Update
  1. Measure feature(s),
  2. Update positions and uncertainties.



Extended Kalman Filter: Update Step

- Measurement Update

covariance matrix tells us, how certain we are for a certain pose, forexample

1. Make measurement z to give the innovation ν

$$\boldsymbol{\nu} = \mathbf{z} - \mathbf{h}\widehat{\mathbf{x}}$$

gives u the predicted measurement

2. Calculate innovation covariance S and Kalman gain W

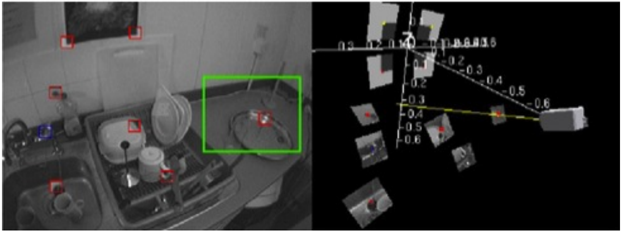
$$\mathbf{S} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \mathbf{P} \frac{\partial \mathbf{h}}{\mathbf{x}}^{\top} + \mathbf{R}$$
$$\mathbf{W} = \mathbf{P} \frac{\partial \mathbf{h}}{\partial \mathbf{x}}^{\top} \mathbf{S}^{-1}$$

3. Update estimate and error covariance:

$$\widehat{\mathbf{x}}_{\mathrm{new}} = \widehat{\mathbf{x}} + \mathbf{W}\boldsymbol{\nu}$$
$$\mathbf{P}_{\mathrm{new}} = \mathbf{P} - \mathbf{W}\mathbf{S}\mathbf{W}^{\top}$$

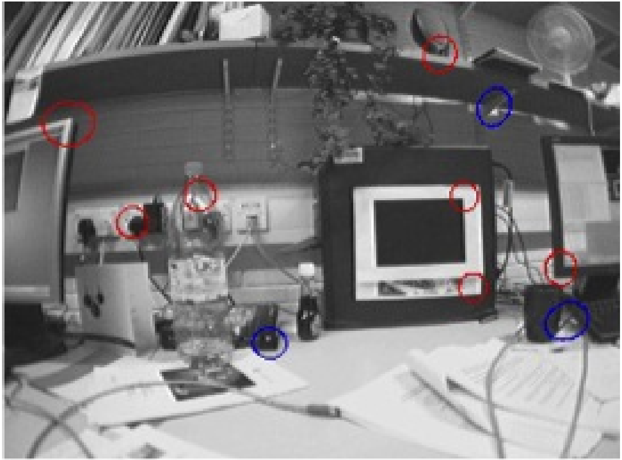
Measurement Step: Image Features and the Map

- Feature measurements are the locations of salient image patches.
- Patches are detected once to serve as long-term visual landmarks.
- Sparse set of landmarks gradually accumulated and stored indefinitely.



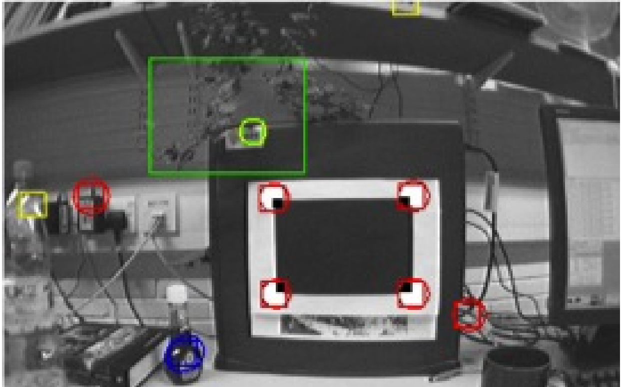
Measurement Step: Active Search

- Active search within elliptical search regions defined by the feature innovation covariance.
- Template matching via exhaustive correlation search.



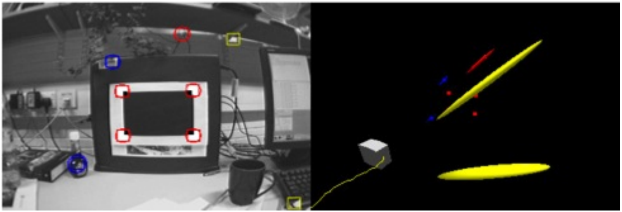
Automatic Map Management

- Initialise system from a few known features.
- Add a new feature if number of visible features drops below a threshold (e.g. 12).
- Choose salient image patch from a search box in an under populated part of the image.



Monocular Feature Initialisation with Depth Particles

- A new feature has unknown depth
  1. Populate the line with 100 particles, spaced uniformly between 0.5m and 5m from the camera.
  2. Match each particle in successive frames to find probability of that depth.
  3. When depth covariance is small, convert to Gaussian.



Inverse depth parametrisation

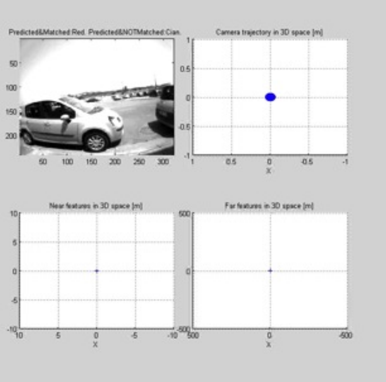
Andrew Comport

A scene 3D point $\mathbf{P}_i$ is defined by the state vector:

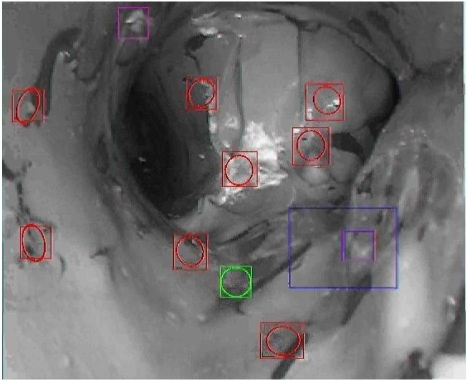
$$\mathbf{y}_i = \begin{pmatrix} x_i & y_i & z_i & \theta_i & \phi_i & \rho_i \end{pmatrix}^\top$$

which models a 3D point located at:

$$\begin{pmatrix} x_i & y_i & z_i \end{pmatrix}^\top + \frac{1}{\rho_i} m(\theta_i, \phi_i)$$



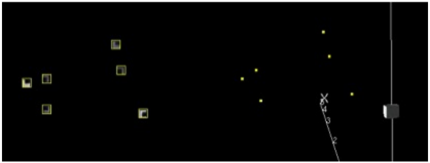
Example: MonoSLAM in Medical Inspection



[Mountney, Stoyanov, Davison, Yang, MICCAI 2006]

Example: HRP2 Humanoid from AIST, Japan

- Small circular loop within a large room
- No re-observation of 'old' features until closing of large loop.



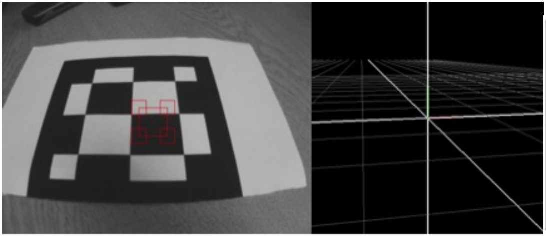
[Stasse, Davison, Zellouati, Yokoi, IROS 2006]

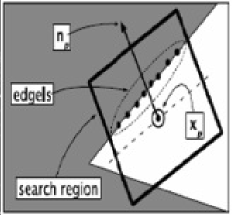
Other feature-based Visual SLAM approaches

- Pupilli & Calway, BMVC 2005
  - Traditional SLAM approach (Particle Filter)
  - Greater robustness: handles multi-modal cases
  - New features not rigorously initialised
- Eade & Drummond, 2006
  - FastSLAM approach (Particle Filter/Kalman Filter)
  - Particle per camera hypothesis, Kalman filter for features
  - Allows larger maps: update O(M logN) instead of O(N2)

FastSLAM-Based Monocular SLAM

- Edglets:
  - Locally straight section of gradient Image.
  - Parameterized as 3D point + direction.
  - Avoid regions of conflict (e.g. close parallel edges).
  - Deal with multiple matches through robust estimation





[Eade and Drummond, CVPR 2006]

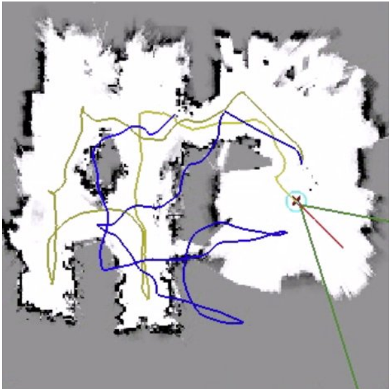
SLAM with Lines (Smith et al. BMVC 2006)

- In undistorted image:
  - Detect FAST corners [Rosten and Drummond, 2005].
  - Quickly verify there is an edge between two corners by bisecting checks.
  - Remove overlapping lines.
  - To measure a line, also use normal to projected line as in [Harris 1992].

Also approaches by Lemaire et al., Glee et al.

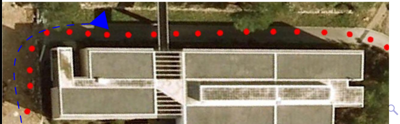
SLAM: Stereo Vision SLAM

- Simultaneous Localization and Mapping (SLAM) using the Rao-Blackwellised Particle Filter (RBPF) [Elinas, Sim, Little]
- Scale Invariant Feature Transform (SIFT) landmark matching.



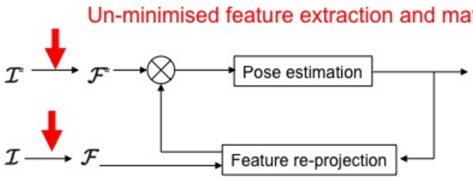
Case Study 2: Dense Stereo Visual SLAM

- Comport, Malis and Rives 2007.
- Dense stereo matching to acquire model.
- Direct Visual Odometry
  - *Real-time* computational efficiency
  - Multi-resolution approach.
- Automatic Map Management
- Robust M-estimation



3D SFM

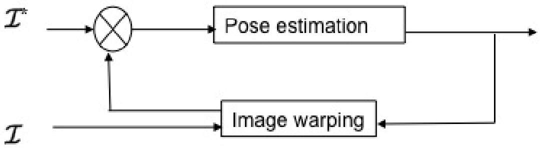
- Feature based [Nister04, Royer04] :
  - Real large scale scenes,
  - Difficult to model all features: usually point feature based,
  - Propagates feature extraction uncertainty to pose estimation,
  - Often have a temporal feature matching step,
  - Require corrective techniques such as EKF and Bundle adjustment.



Direct 3D Tracking

- Intensity-based [Hagar98, Baker01, Benhimane04]:
  - Restrictive affine or planar homography assumptions,
  - Accurate non-linear iterative closed loop estimation,
  - Uses all information (points, lines, planes,...) in a general manner,
  - Robust and precise.

All errors minimised in closed loop estimation



Direct muli-planar SLAM

- Intensity-based direct method [Silveira, Malis, Rives]

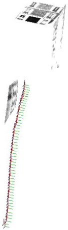
$$\arg\min_{\widehat{\mathbf{x}}\in\mathbb{R}^6} \frac{1}{2}\sum_{\mathbf{p}_i\in\mathbb{R}^+}\left[\mathcal{I}(w(\mathbf{p}_i^*;\mathbf{G}(\widehat{\mathbf{T}}\mathbf{T}(\tilde{\mathbf{x}})))) - \mathcal{I}^*(\mathbf{p}_i^*)\right]^2$$

where

$$\mathbf{G}(\mathbf{T};\mathbf{n}^*) = \mathbf{K}\frac{\mathbf{R}+\mathbf{t}\mathbf{n}^\top}{\sqrt[3]{1+\mathbf{t}^\top\mathbf{R}\mathbf{n}^*}}\mathbf{K}^{-1}$$

- **Pros**
  - Accurate, efficient and robust,
  - No local matching, only SLAM,
  - Full 3D deformation of planar image patches (i.e. no erroneous point detection)
- **Cons**
  - Planar Homography assumption must detect planar regions,
  - Unknown Scale factor

Direct SLAM



Stereo Visual Slam

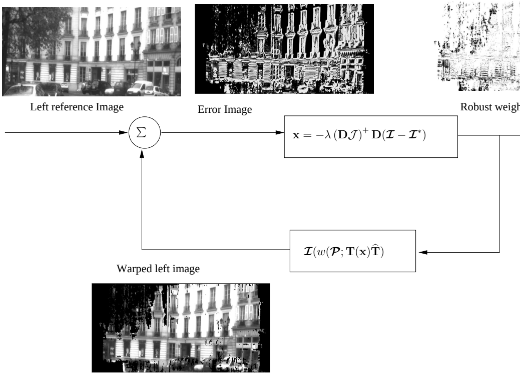
- Non-trivial problem:
  - Potentially infinite source of information
  - Occlusions, Illumination conditions,
  - Large and fast movements,
  - Drift.
- Stereo:
  - Provides constraints for estimating 3D movement,
  - More robust than monocular.

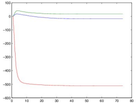
Trajectory Estimation

- Image pair: $\mathcal{I} = (\mathbf{I}, \mathbf{I}')^\top \in \mathbb{R}^{2n}$
- Pixel correspondences: $\mathcal{P}^* = \{\mathbf{p}, \mathbf{p}'\}$
- The warping function: w
- Novel view synthesis: $\mathcal{I}^*(\mathcal{P}^*) = \mathcal{I}(w(\mathcal{P}^*; \overline{\mathbf{T}})).\forall \mathcal{P}^* \in \mathcal{R}^*.$
- Current pose estimate: $\widehat{\mathbf{T}}$
- Unknown incremental pose: $\mathbf{T}(\mathbf{x})$ where $\exists \tilde{\mathbf{x}} : \mathbf{T}(\tilde{\mathbf{x}})\widehat{\mathbf{T}} = \overline{\mathbf{T}}$
  - Robust optimisation criterion:

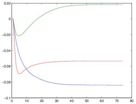
$$\mathcal{O}(\mathbf{x}) = \sum_{\mathcal{P}^* \in \mathcal{R}^*} \rho \left(\mathcal{I}(w(\mathcal{P}^*; \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}) - \mathcal{I}^*(\mathcal{P}^*) \right)$$

Non-linear iterative estimation





Translation (mm)



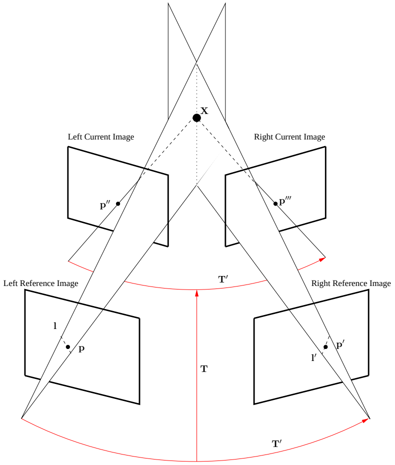
Rotation (rad)

Dense Correspondence

- Known Stereo Geometry:
  - Intrinsic and Extrinsic parameters,
  - Precise calibration,
  - 1D search along Epipolar lines.
- Many techniques offering similar results:

  [Birchfield98, Scharstein01, Ogale05, ...],

Warping : Quadrifocal Geometry



Warping : Novel View Synthesis

- Quadrifocal geometry allows definition of two trifocal tensors:

$$\mathcal{T}_i^{jk} = \mathbf{k}'^j_m \mathbf{r}'^m_n \mathbf{k}^{-1}{}^n_i . \mathbf{k}''^k_o \mathbf{t}''^o(t) - \mathbf{k}'^j_p \mathbf{t}'^p . \mathbf{k}''^k_q \mathbf{r}''^q_r(t) \mathbf{k}^{-1}{}^r_i,$$

- Each trifocal tensor defines an image transfer of points in the reference pair to points in another pair of images:

$$\begin{bmatrix} \mathbf{p}''^k \\ \mathbf{p}'''^n \end{bmatrix} = \begin{bmatrix} \mathbf{p}^i \mathbf{l}'_j \mathcal{T}_i^{jk} \\ \mathbf{p}'^l \mathbf{l}_m \mathcal{T}_l^{mn} \end{bmatrix},$$

- This warping operator is a $\mathbb{SE}(3)$ group *action*.

Robust Efficient Second-order Minimisation

- Efficient second order approximation:

$$\mathcal{I}(\tilde{\mathbf{x}}) \approx \mathcal{I}(\mathbf{0}) + \frac{\mathbf{J}(\mathbf{0})+\mathbf{J}(\tilde{\mathbf{x}})}{2}\tilde{\mathbf{x}} + \mathcal{R}(||\tilde{\mathbf{x}}||^3),$$

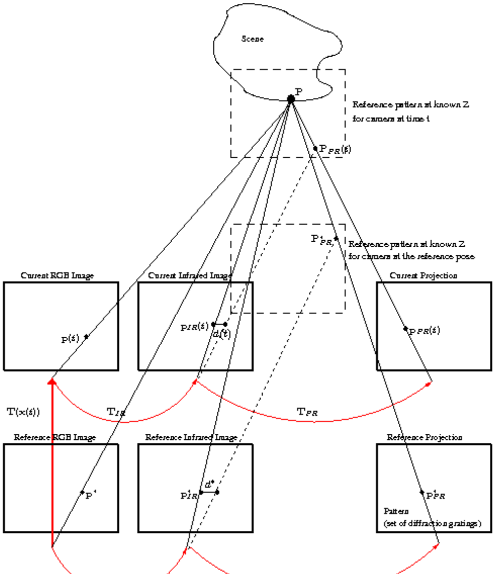
- Second Order approximation using the current and reference images:
(requires the warping function to be a group action):

$$\mathcal{J}(\tilde{\mathbf{x}}) = \frac{(\mathbf{J}_{\mathcal{I}}+\mathbf{J}_{\mathcal{I}^*})}{2}\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{T}}\mathbf{J}_{\mathcal{V}},$$

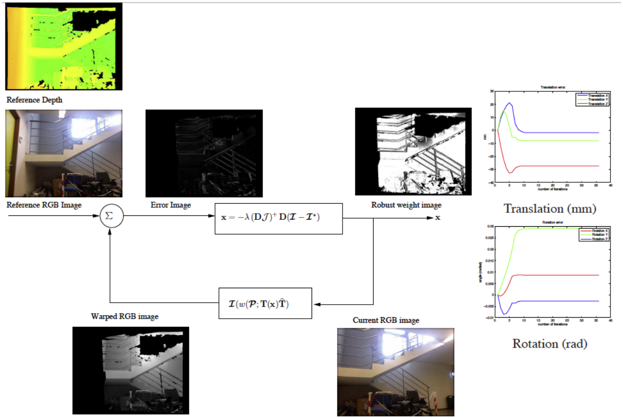
- Robust ESM minimisation is solved iteratively :

$$\tilde{\mathbf{x}} = -\lambda\left(\mathbf{D}\mathcal{J}\right)^{+}\mathbf{D}(\mathcal{I} - \mathcal{I}^*),$$

RGB-D Kinect SLAM - Projective Light Geometry



RGB-D Kinect SLAM - Projective Light



The unknown pose $\mathbf{T}(\mathbf{x}(t))$ is estimated via a non-linear warping function which warps all points in the reference RGB-D image $\mathbf{p}^*$ to the current $\mathbf{p}(t)$.

Direct Iterative Closest Point

The pose and structure can be estimated by minimising a non-linear least squares cost function

$$C(\mathbf{x}) = \mathbf{e}_{\mathcal{I}}^T \mathbf{W}_{\mathcal{I}} \mathbf{e}_{\mathcal{I}} + \lambda^2 \mathbf{e}_{\mathbf{Z}}^T \mathbf{W}_{\mathbf{Z}} \mathbf{e}_{\mathbf{Z}},$$

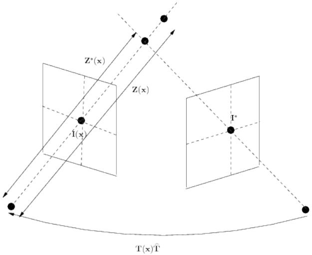
where

$$\begin{aligned} \mathbf{e}_{\mathcal{I}} &= \mathcal{I}\left(w(\mathcal{P}; \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}})\right) - \mathcal{I}^*\left(w(\mathcal{P}; \mathbf{I})\right) \\ \mathbf{e}_{\mathbf{Z}} &= \mathbf{Z}\left(w(\mathcal{P}; \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}})\right) - [\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}\mathcal{P}]_z, \end{aligned}$$

and $\mathbf{W}_{\mathcal{I}}$ and $\mathbf{W}_{\mathbf{Z}}$ are the diagonal weight matrices obtained from a M-estimator.

Direct Iterative Closest Point

Photometric and depth error are minimized simultaneously between subsequent image frames.



Estimation is performed by computing

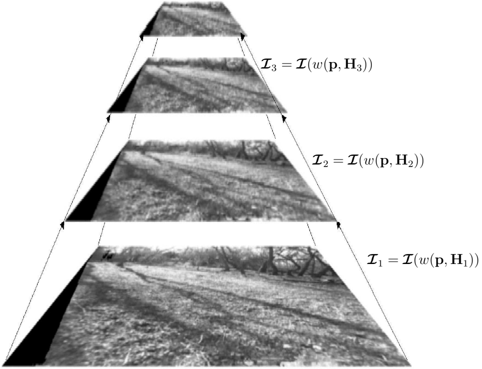
$\Delta \mathbf{x} = -(\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}\begin{bmatrix}\mathbf{e}_{\mathcal{I}} \\ \lambda \mathbf{e}_{\mathbf{Z}}\end{bmatrix}$ where λ accounts for the relatives uncertainty between the depth and image measurements.

Large Scale SLAM

- Update reference image-pair:
  - the weighted average error
  - the Median Absolute Deviation
  - become too large.
  - Or
  - Update every image.
  - Compromise between image interpolation, drift propagation and computational efficiency.
- Computational Efficiency:
  - Use reference images (Calculation of Jacobian and dense correspondences every N images),
  - Predict the motion of the vehicle using an Extended Kalman Filter,
  - Do an offline learning learning step,
  - Change resolution of the image and/or choose the strongest gradients.
  - Algorithm is highly parallelizable.

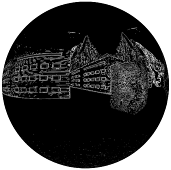
Real-time optimisation

- Multi-resolution pyramid: Estimate larger scale movement at higher levels with less computational cost, only a few iterations a higher precision lower levels.
- Choose strongest gradients



Real-time computing: Selection of the best pixels minimized in the cost function.

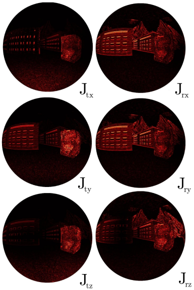
- Highly redundant number of pixels to warp at each iteration.
- Reduction of computing time without degrading accuracy and robustness.
- Classical selection: best image intensity gradients (i.e Harris): **geometry not considered**.



Real-time optimisation - Pixel Selection

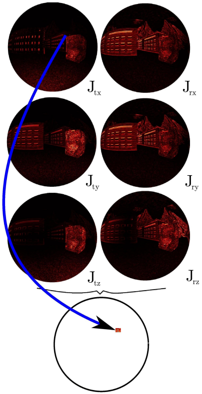
Selection directly done on an analytical spherical warping Jacobian matrix : $\mathbf{J} \in \mathbb{R}^{n \times 6}$.

- Gradient of the state vector from the cost function.
- Combines directly Luminance and geometric gradients.
- 6 DOF of $\mathbf{J} \Rightarrow$ 6 saliency maps.



Real-time optimisation - Pixel Selection

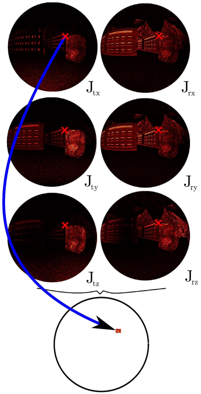
1. Select the pixel with the maximum gradient in the first saliency map: $argmax(\mathbf{J}_0)$.



Saliency Selection

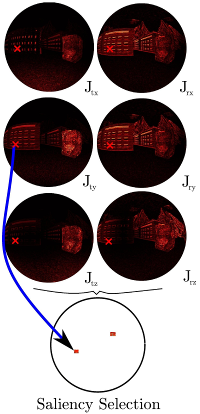
Real-time optimisation - Pixel Selection

1. Select the pixel with the maximum gradient in the first saliency map: $argmax(\mathbf{J}_0)$.
2. Mark the pixel as selected in the 6 saliency maps.



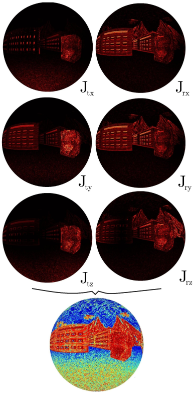
Real-time optimisation - Pixel Selection

1. Select the pixel with the maximum gradient in the first saliency map: $argmax(\mathbf{J}_0)$.
2. Mark the pixel as selected in the 6 saliency maps.
3. Switch to the next saliency map and repeat 1 and 2.



Real-time optimisation - Pixel Selection

- The process is iteratively done for all pixels across the 6 saliency maps.
- **Resulting selection**: an equal number of maximum gradient for each DOF sorted in a list.
- During online tracking only best pixels are picked-up in the pre-computed saliency map.

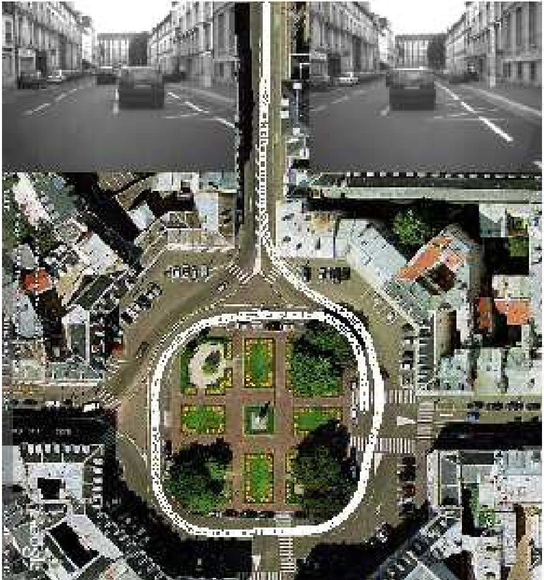


Saliency Selection

Trajectory Estimation - Urban Canyon

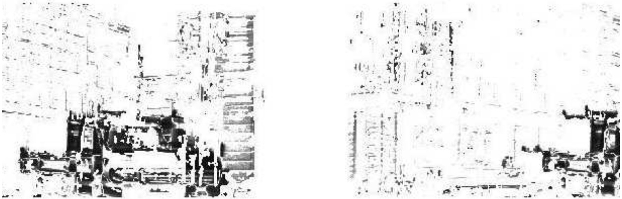
- 700 759x280 images, Baseline = 1m,
- 440m long sequence measured on Google Earth (2.9cm/pixel) and by our Odometry.
- Other moving traffic and pedestrians.
- Car stops at the traffic lights and overtakes another car.

Trajectory Estimaton - Round-about



Robust Estimation

Vehicule

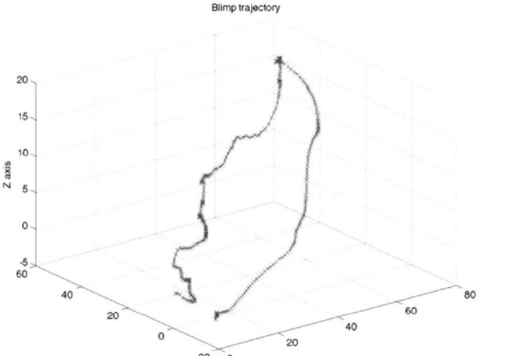


Pedestrian

Notice the occluded corner data.

Trajectory Estimaton: LAAS Blimp

- Blimp sequence demonstrating full 6DOF Visual Odometry.
- Results obtained with (image resolution)/6 and only the 50,000 strongest gradients.
- Full loop of a parking lot.
- Very unpredictable large movements.



Outline

Real-time dense SLAM

Direct 3D Model-based (MB) tracking

The unknown 3D motion $\mathbf{x}$ between an augmented reference image $\boldsymbol{S} = \{\mathcal{I}^*, \mathcal{P}\}$ and the current camera $\mathcal{I}_t$ can be iteratively estimated by minimising a robust error between the warped image and the reference image:

$$\mathbf{e}_{MB} = \rho\left(\mathcal{I}_t\left(w(\mathcal{P}; \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}))\right) - \beta_{MB} - \mathcal{I}^*(\mathcal{P})\right).$$

Dynamic environments - illumination change

- *Global* illumination - $\beta_{MB} = \mathit{Median}(\mathbf{e}_{MB})$ [**?**].
- *Local* illumination - robust diagonal weighting matrix $\mathbf{D}$ [**?**].

Inconvenient: Large environment changes

Visual odometry(VO) tracking

A non-classic visual odometry approach - improves convergence speed and robustness to dynamic changes (current and previous image intensities are minimised with model geometry).

$$\mathbf{e}_{VO} = \rho\left(\mathcal{I}_t(w(\mathcal{P}; \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}))) - \beta_{VO} - \mathcal{I}^w_{t-1}(\mathcal{P})\right),$$

Advantages

- 3D geometry is shared with the original 3D model.
- Very small local illumination changes can be expected between successive frames (ie. $\geq$ 20Hz) $\Rightarrow$ fast convergence.
- Still robust to global illumination changes (due to the global bias model).

however **visual odometry** **drifts** with time.

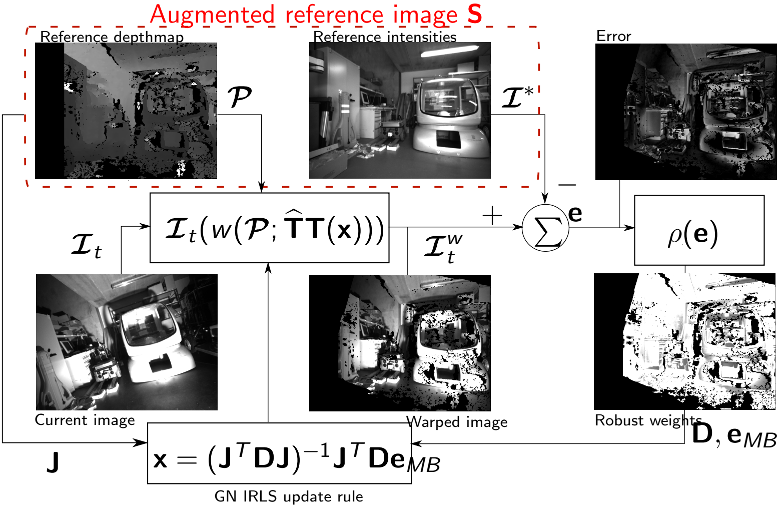
Hybrid model-based and visual odometry (H) tracking

Proposed method: Global minimisation of the error functions, combines the advantages of both techniques.

$$\mathbf{e}_H = \left[\mathbf{e}_{MB} \quad \mathbf{e}_{VO}\right]^T .$$

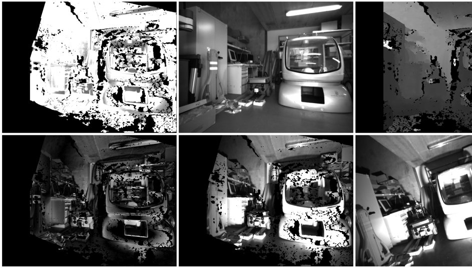
- **Fast** convergence (due to VO).
- **No drift** since raw sensor measurement is maintained in the minimisation process (due to MB).

Minimisation process



Tracking in dynamic environments

- Local reflections and illumination variation.
- Global illumination change.
- Intensity saturation.
- Varying focal length.



A general RGB-D localisation model

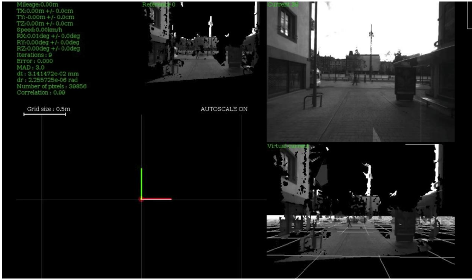
- Kinect based localisation and mapping at 30Hz [?].
- Direct Iterative Closest Point [?] (tomorrow).

[?] C. Audras, A.I Comport, M. Meilland, and P. Rives. Real-time dense RGB-D localisation and mapping. ACRA 11

[?] Tykkälä, T.M. and Comport, A.I. Direct Iterative Closest Point for Real-time Visual Odometry CVVT-ICCV 11

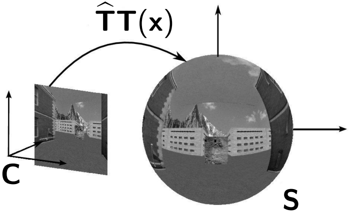
Environment mapping in Urban city environments

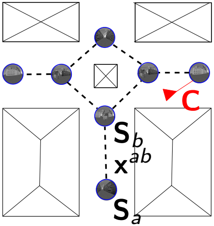
- Real-world complex environment with pedestrians, cars, trams, and illumination change.
- Dense and fast model acquisition.



Online localisation

Real time localization and navigation of a vehicle using a **single** camera.





Monocular direct 3D image registration

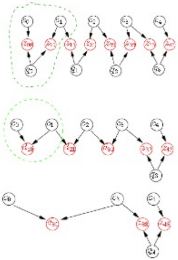
$$\mathbf{e} = \rho\left(\mathcal{I}_t\left(w\left(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); \mathcal{P}_s, \mathbf{Z}_s, \mathbf{W}_s\right)\right) - \mathcal{I}_s(\mathcal{P}_s, \mathbf{Z}_s, \mathbf{W}_s)\right).$$

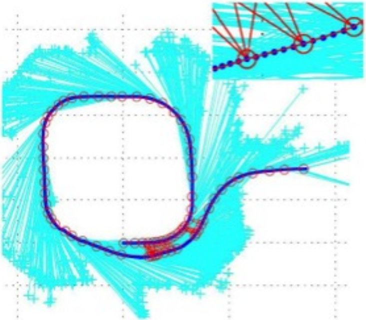
Real-time localisation results

- 300 augmented spheres extracted from the graph.
- Real-time **monocular** camera (robot) localisation at 45 Hz.

Loop closing: Frame SLAM

- Stereo Visual Odometry (feature based)
- Reduction of large loop closing procedures [Konolige, Aggrawal]
- 10km sequences, real-time





Super-resolution

- Combining a set of LR images into a SR image
- SR techniques often consider 2D registration
- Contribution: real-time full 6 dof. motion and 3D registration

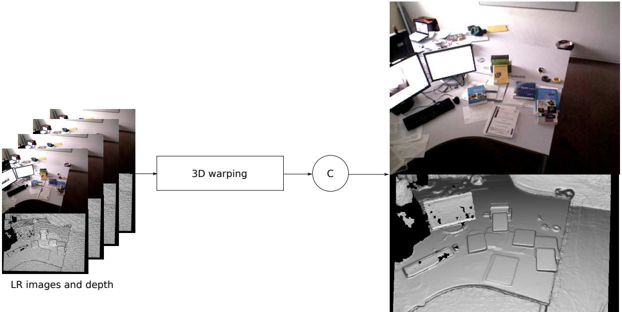


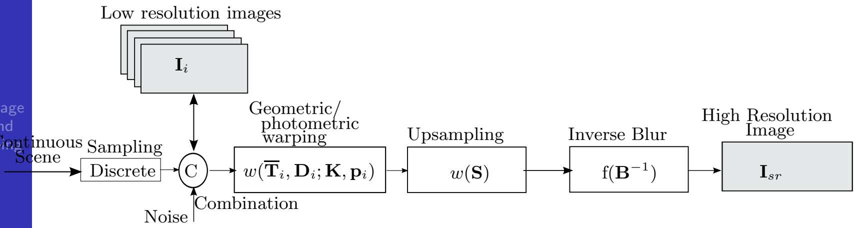
Image generation pipeline

Andrew Comport

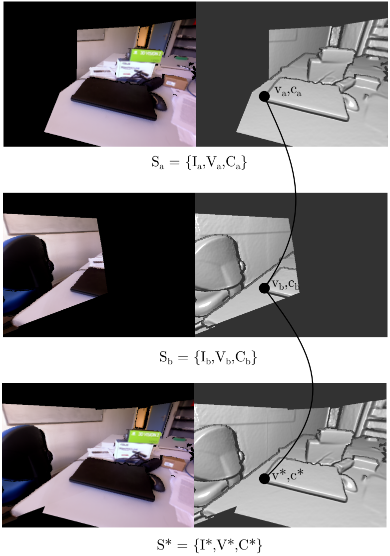
Part 1. Introduction
Part 2. Applications
Part 3. Rigid Body Motion
Part 4. Image formation
Part 5. 3D Visual Geometry
Part 5.2 Image synthesis and Image warping
Part 6. Non-linear Robust Estimation
Part 7. 6D Localisation

The image generation pipeline (inverse to degradation)

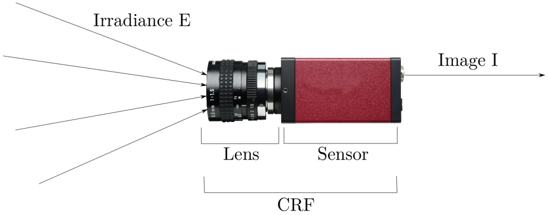
- Multiple LR images sampled from a continuous light field.
- Images weighted wrt. distance to ideal image (SR image).
- LR images warped to common reference frame (6D pose).
- LR images up-sampled and inverse blurring is applied.



Super-resolution SLAM results



High Dynamic Range Dense Mapping



Real-scene

- World irradiance is a **continuous** function
- High dynamic range (HDR): $[0, \infty]$

Image sensors

- Range of intensities is **quantified**
- Low dynamic range (LDR): $[0, 255]$ levels

Camera response function (CRF)

Camera exposure (Δ_t) and gain (g) allow to observe an interval of the dynamic range

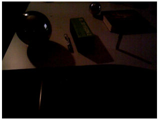
Camera response function

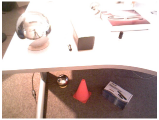
$$\mathbf{I} = (g\Delta_t \mathbf{E})^{\gamma} \tag{28}$$

$\mathbf{E}$ is the scene irradiance

$\mathbf{I}$ is the measured intensity

γ is a known value






Camera response function (CRF)

Camera exposure (Δ_t) and gain (g) allow to observe an interval of the dynamic range

Camera response function

$$\mathbf{I} = (g\Delta_t \mathbf{E})^{\gamma} \qquad (28)$$

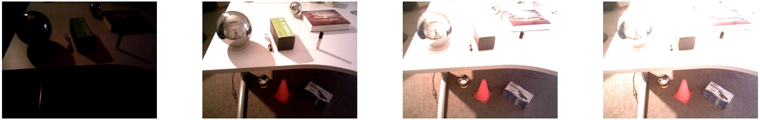
E is the scene irradiance

I is the measured intensity

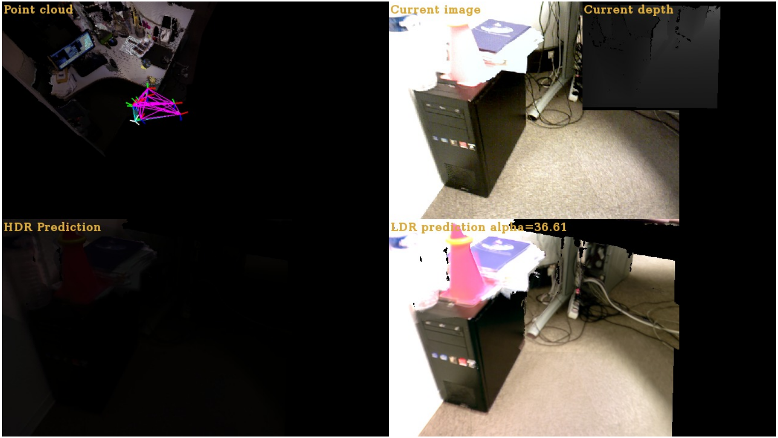
γ is a known value

Exposure time Δt and gain g

- Not available for low cost sensors
- Dynamically computed by the sensor



HDR mpping results



Towards photo-realistic augmented reality

Coherent virtual object rendering

- Shadows
- External reflections

Towards photo-realistic augmented reality

Coherent virtual object rendering

- Shadows
- External reflections

Classic reflection mapping

- HDR reflection map
- Light probe
- Cumbersome offline procedure

Figure: Light probe calibration [?].

Figure: Rendering example [?].

Towards photo-realistic augmented reality

Coherent virtual object rendering

- Shadows
- External reflections

Classic reflection mapping

- HDR reflection map
- Light probe
- Cumbersome offline procedure

Use the HDR 3D map

- Generate virtual light probes
- Any location in the map
- Detect 3D light positions
- Shadow casting

Figure: Light probe calibration [?].

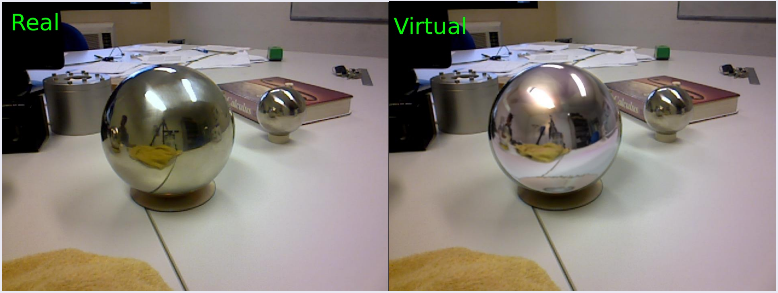
Figure: Rendering example [?].

Realistic AR

Ground truth comparison

Ground truth comparison

Ground truth comparison



Results

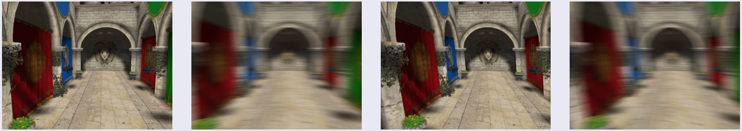
- Realistic reflections
- Parallax artefact

High-speed localsiation and mapping

Unified Rolling Shutter and Motion Blur Model - Two patents

The combined model depends only on the 6 d.o.f. camera velocity twist:

$$\mathbf{I}\left(w_{rs}(\mathbf{T}(\tau\widetilde{\mathbf{x}}_v)^{-1}\widetilde{\mathbf{T}},\mathbf{v}^*)\right) = \int_{t_i-t_e}^{t_i} \mathbf{I}^*\left(w(\widetilde{\mathbf{T}}^{-1}\mathbf{T}(-t\widetilde{\mathbf{x}}_v)\widetilde{\mathbf{T}},\mathbf{v}^*)\right)\mathrm{d}t$$



(a) Global shutter, without blur (b) Global shutter with blur (c) Rolling shutter without blur (d) Rolling shutter with blur

Semantic SLAM for aircraft manufacturing

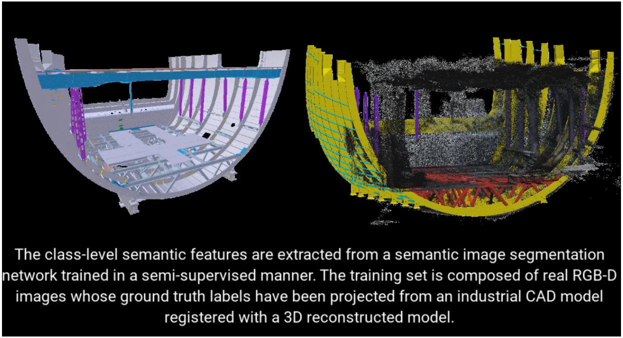


Figure: Semantic SLAM, Airbus Saint Nazaire 2019

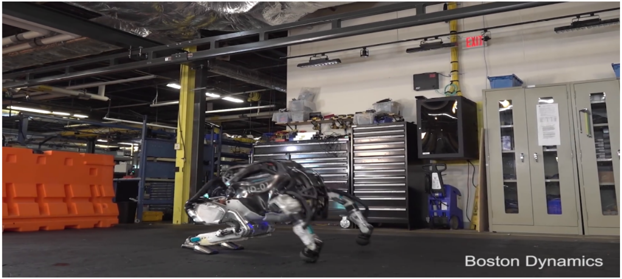
Open problems

- Long term localisation and mapping,
- Robust place recognition for loop closing,
- Non-rigid scenes,
- Robot control based on dense visual SLAM,
- Robustness to outliers (illumination changes, occlusions, etc...)
- Incorporate semantics and beyond-geometric scene understanding.

2019: Autonomous drones - Skydio

Facebook reality labs

Boston Dynamics Atlas



Personal assistance - Toyota Research Labs

References I

S. Nayar, "Catadioptric omnidirectional camera," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 1997, pp. 482–. [Online]. Available: http://portal.acm.org/citation.cfm?id=794189.794460

R. Szeliski, "Image alignment and stitching: a tutorial," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 1, pp. 1–104, 2006.

H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 328–341, 2008.

References II

- S. Lovegrove and A. Davison, “Real-time spherical mosaicing using whole image alignment,” in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6313. Springer Berlin / Heidelberg, 2010, pp. 73–86.
- A. Comport, E. Malis, and P. Rives, “Real-time quadrifocal visual odometry,” *In The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 245–266, 2010.
- T. Tykkala and A. Comport, “A dense structure model for image based stereo slam,” in *IEEE International Conference on Robotics and Automation, ICRA'11*, Shanghai, China, May 9-13 2011. [Online]. Available: http://www.i3s.unice.fr/~comport/publications/2011_ICRA_Tykkala.pdf

References III

T. Gonçalves and A. Comport, "Real-time direct tracking of color images in the presence of illumination variation," in *IEEE International Conference on Robotics and Automation*, May 2011.

P. Huber, *Robust Statistics*. New york, Wiley, 1981.

C. Audras, A. Comport, M. Meilland, and P. Rives, "Real-time dense rgb-d localisation and mapping," in *To Appear Australian Conference on Robotics and Automation*, Monash University, Australia, December 7-9 2011. [Online]. Available: http://www.i3s.unice.fr/~comport/publications/2011_ACRA_Comport.pdf

References IV

T. Tykkälä and A. Comport, "Direct iterative closest point for real-time visual odometry," in *The Second international Workshop on Computer Vision in Vehicle Technology: From Earth to Mars in conjunction with the International Conference on Computer Vision*, Barcelona, Spain, November 6-13 2011. [Online]. Available: http://www.i3s.unice.fr/~comport/publications/2011_CCVT_Tykkala.pdf