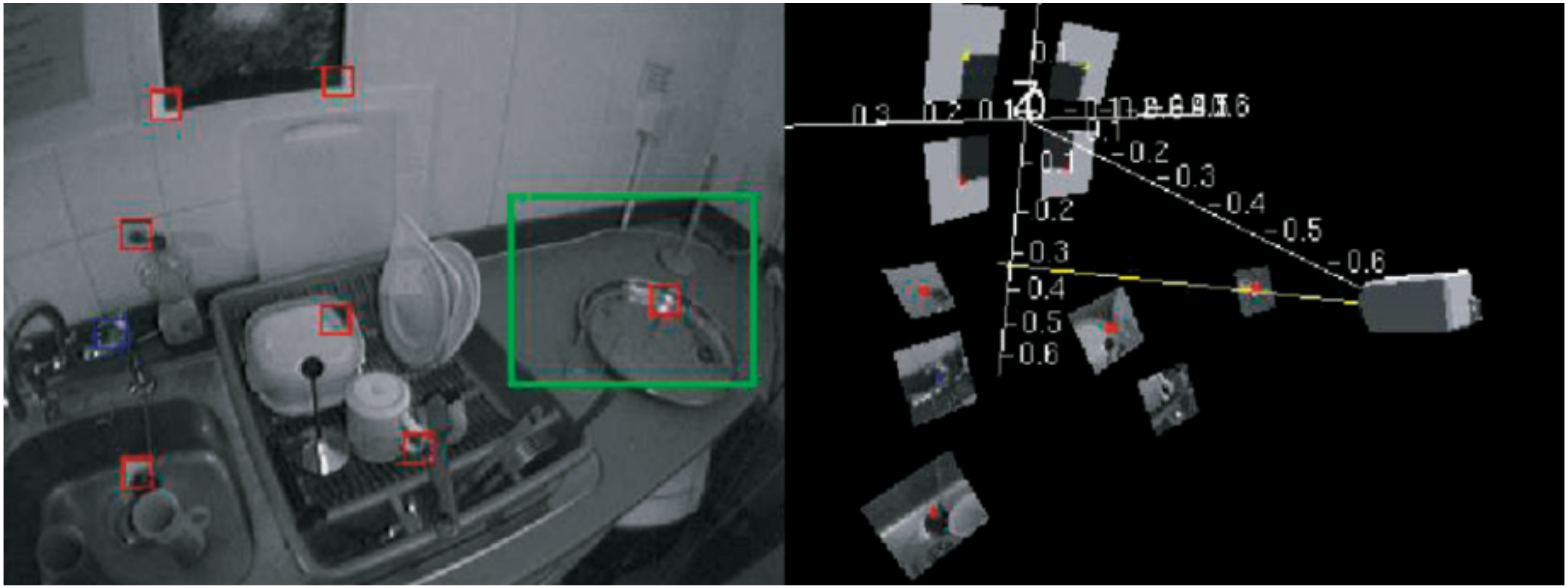
# MonoSLAM: Real-Time Single Camera SLAM

Andrew J. Davison, Ian D. Reid, *Member, IEEE*, Nicholas D. Molton, and Olivier Stasse, *Member, IEEE*

**Abstract**—We present a real-time algorithm which can recover the 3D trajectory of a monocular camera, moving rapidly through a previously unknown scene. Our system, which we dub *MonoSLAM*, is the first successful application of the SLAM methodology from mobile robotics to the "pure vision" domain of a single uncontrolled camera, achieving real time but drift-free performance inaccessible to Structure from Motion approaches. The core of the approach is the online creation of a sparse but persistent map of natural landmarks within a probabilistic framework. Our key novel contributions include an *active* approach to mapping and measurement, the use of a general motion model for smooth camera movement, and solutions for monocular feature initialization and feature orientation estimation. Together, these add up to an extremely efficient and robust algorithm which runs at 30 Hz with standard PC and camera hardware. This work extends the range of robotic systems in which SLAM can be usefully applied, but also opens up new areas. We present applications of *MonoSLAM* to real-time 3D localization and mapping for a high-performance full-size humanoid robot and live augmented reality with a hand-held camera.

**Index Terms**—Autonomous vehicles, 3D/stereo scene analysis, tracking.

# Simultaneous Localization and Mapping




Given a **single camera** feed,
estimate the 3D **position of the camera** and
the 3D **positions of all landmark** points in the world

# Real-Time Camera Tracking in Unknown Scenes

# General Filtering Equations

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t)\int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

**Prediction:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

**Update:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t)P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

# General Filtering Equations

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t)\int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

*What is the state representation?*

**Prediction:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

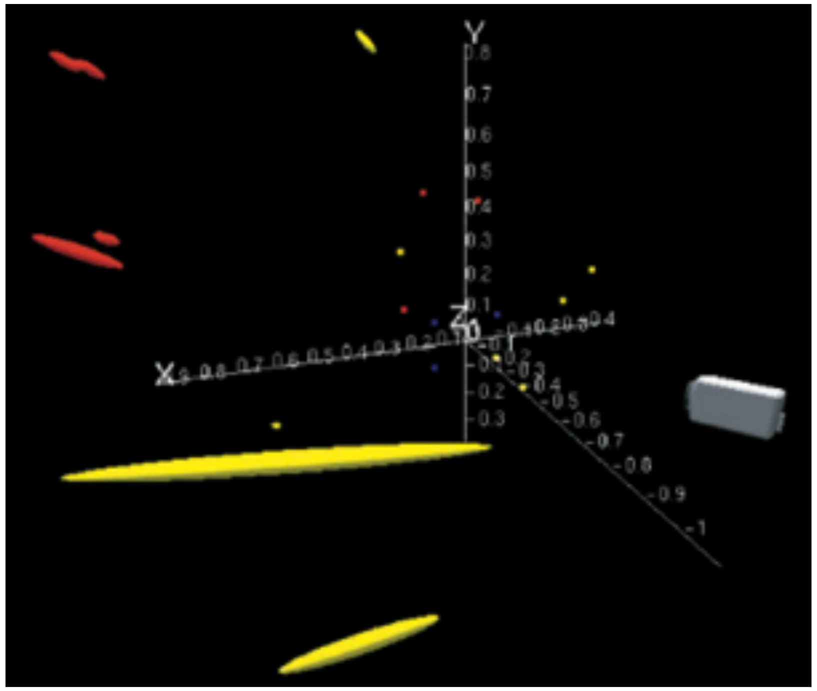
**Update:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t)P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

## *What is the camera (robot) state?*

*What are the dimensions?*

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{r} \\ \mathbf{q} \\ \mathbf{v} \\ \omega \end{bmatrix} \begin{array}{l} \text{position} \\ \text{rotation (quaternion)} \\ \text{velocity} \\ \text{angular velocity} \end{array}$$




**13 total**

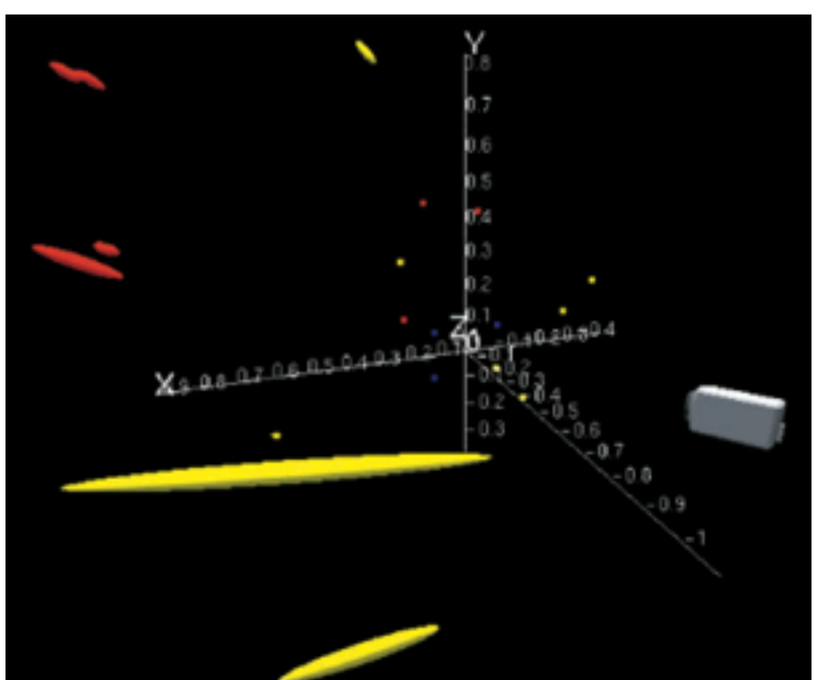
# *What is the camera (robot) state?*

*What are the dimensions?*

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{r} \\ \mathbf{q} \\ \mathbf{v} \\ \omega \end{bmatrix}$$

| | | |
|---|---|---|
| $\mathbf{r}$ | position | 3 |
| $\mathbf{q}$ | rotation (quaternion) | 4 |
| $\mathbf{v}$ | velocity | 3 |
| $\omega$ | angular velocity | 3 |

**13 total**



*What is the world (robot+environment) state?*

*What are the dimensions?*

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}$$

state of the camera

location of feature 1

location of feature 2

location of feature N

**13+3N total**

## *What is the world (robot+environment) state?*

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}$$

| | *What are the dimensions?* |
|---|---|
| state of the camera | 13 |
| location of feature 1 | 3 |
| location of feature 2 | 3 |
| location of feature N | 3 |

**13+3N total**

*What is the covariance (uncertainty) of the world state?*

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{x}_c\mathbf{x}_c} & \boldsymbol{\Sigma}_{\mathbf{x}_c\mathbf{y}_1} & \cdots & \boldsymbol{\Sigma}_{\mathbf{x}_c\mathbf{y}_N} \\ \boldsymbol{\Sigma}_{\mathbf{y}_1\mathbf{x}_c} & \boldsymbol{\Sigma}_{\mathbf{y}_1\mathbf{y}_1} & \cdots & \boldsymbol{\Sigma}_{\mathbf{y}_1\mathbf{y}_N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{\mathbf{y}_N\mathbf{x}_c} & \boldsymbol{\Sigma}_{\mathbf{y}_N\mathbf{y}_1} & \cdots & \boldsymbol{\Sigma}_{\mathbf{y}_N\mathbf{y}_N} \end{bmatrix}$$

*What are the dimensions?*

**(13+3N) x (13+3N)**

# General Filtering Equations

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

*What are the observations?*

**Prediction:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

**Update:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

# General Filtering Equations

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

*What are the observations?*

**Prediction:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

**Update:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

Observations are…

detected visual features of landmark points.

# General Filtering Equations

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

**Prediction:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

**Update:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

# General Filtering Equations

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

**Prediction:**

$$\boxed{P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}}$$

*What does the prediction step look like?*

**Update:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

*What is the motion model?* $P(\boldsymbol{x}_t | \boldsymbol{x}_{t-1})$

*What is the form of the belief?* $P(\boldsymbol{x}_t | \boldsymbol{z}_{1:t-1})$

*What is the motion model?* $P(\boldsymbol{x}_t | \boldsymbol{x}_{t-1})$

**Landmarks**:
constant position
(identity matrix)

**Camera**:
constant velocity
(not identity matrix!)

*What is the form of the belief?* $P(\boldsymbol{x}_t | \boldsymbol{z}_{1:t-1})$

*What is the motion model?* $P(\boldsymbol{x}_t | \boldsymbol{x}_{t-1})$

**Landmarks**:
constant position
(identity matrix)

**Camera**:
constant velocity
(not identity matrix!)

*What is the form of the belief?* $P(\boldsymbol{x}_t | \boldsymbol{z}_{1:t-1})$

**Gaussian!**
(everything is parametrized by a mean and Gaussian)

# Constant Velocity Motion Model

$$\mathbf{r}_t = \mathbf{r}_{t-1} + \mathbf{v}_{t-1}\Delta t$$ position

$$\mathbf{q}_t = \mathbf{q}_{t-1} \times [\mathbf{q}(\omega)\Delta t]$$ rotation (quaternion)

$$\mathbf{v}_t = \mathbf{v}_{t-1}$$ velocity

$$\omega_t = \omega_{t-1}$$ angular velocity

# Gaussian noise uncertainty (only on velocity)

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \mathbf{V}$$

$$\omega_t = \omega_{t-1} + \boldsymbol{\Omega}$$

$$\mathbf{V} \sim \mathcal{N}(\mathbf{0}, \left[ \begin{array}{ccc} \sigma_v & 0 & 0 \\ 0 & \sigma_v & 0 \\ 0 & 0 & \sigma_v \end{array} \right])$$

$$\boldsymbol{\Omega} \sim \mathcal{N}(\mathbf{0}, \left[ \begin{array}{ccc} \sigma_w & 0 & 0 \\ 0 & \sigma_w & 0 \\ 0 & 0 & \sigma_w \end{array} \right])$$

## Prediction (mean of camera state):

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

$$\mathbf{f}_t = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{q}_t \\ \mathbf{v}_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{t-1} + \mathbf{v}_{t-1}\Delta t \\ \mathbf{q}_{t-1} + \mathbf{q}(\omega)_{t-1}\Delta t \\ \mathbf{v}_{t-1} \\ \omega_{t-1} \end{bmatrix}$$

# Prediction (covariance of camera state):

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

$$\bar{\mathbf{\Sigma}}_{\mathbf{xx}} = \boxed{\frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}} \mathbf{\Sigma}_{\mathbf{xx}} \frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}^\top + \mathbf{Q}_t$$

new covariance

change around new state

old covariance

change around new state

system noise (process noise)

*Where does this motion model approximation come from?*

change in
position

$$\boxed{\frac{\partial \mathbf{f}_t}{\partial \mathbf{x}_{t-1}}} = \begin{bmatrix} \boxed{\frac{\partial \mathbf{r}_t}{\partial \mathbf{r}_{t-1}}} & \frac{\partial \mathbf{q}_t}{\partial \mathbf{r}_{t-1}} & \frac{\partial \mathbf{v}_t}{\partial \mathbf{r}_{t-1}} & \frac{\partial \omega_t}{\partial \mathbf{r}_{t-1}} \\ \frac{\partial \mathbf{r}_t}{\partial \mathbf{q}_{t-1}} & \frac{\partial \mathbf{q}_t}{\partial \mathbf{q}_{t-1}} & \frac{\partial \mathbf{v}_t}{\partial \mathbf{q}_{t-1}} & \frac{\partial \omega_t}{\partial \mathbf{q}_{t-1}} \\ \frac{\partial \mathbf{r}_t}{\partial \mathbf{v}_{t-1}} & \frac{\partial \mathbf{q}_t}{\partial \mathbf{v}_{t-1}} & \frac{\partial \mathbf{v}_t}{\partial \mathbf{v}_{t-1}} & \frac{\partial \omega_t}{\partial \mathbf{v}_{t-1}} \\ \frac{\partial \mathbf{r}_t}{\partial \omega_{t-1}} & \frac{\partial \mathbf{q}_t}{\partial \omega_{t-1}} & \frac{\partial \mathbf{v}_t}{\partial \omega_{t-1}} & \frac{\partial \omega_t}{\partial \omega_{t-1}} \end{bmatrix}$$

change in
camera state

*What are the dimensions?*

Skipping over many details…

$$\frac{\partial \mathbf{f}_t}{\partial \mathbf{x}_{t-1}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{I}\Delta t & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{q}_t}{\partial \mathbf{q}_{t-1}} & \mathbf{0} & \frac{\partial \omega_t}{\partial \mathbf{q}_{t-1}} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

# Prediction (covariance of camera state):

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

$$\bar{\boldsymbol{\Sigma}}_{\mathbf{xx}} = \boxed{\frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}} \boldsymbol{\Sigma}_{\mathbf{xx}} \frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}^{\top} + \mathbf{Q}_t$$

| new covariance | change around new state | old covariance | change around new state | system noise (process noise) |
|---|---|---|---|---|

Bit of a pain to compute this term…

We just covered the **prediction** step for the camera state

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1})d\boldsymbol{x}_{t-1}$$

$$\mathbf{f}_t = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{q}_t \\ \mathbf{v}_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{t-1} + \mathbf{v}_{t-1} \\ \mathbf{q}_{t-1} + \mathbf{q}(\omega)_{t-1} \\ \mathbf{v}_{t-1} \\ \omega_{t-1} \end{bmatrix}$$

$$\bar{\boldsymbol{\Sigma}}_{\mathbf{xx}} = \frac{\partial \mathbf{f}_t}{\partial \mathbf{x}} \boldsymbol{\Sigma}_{\mathbf{xx}} \frac{\partial \mathbf{f}_t}{\partial \mathbf{x}}^\top + \mathbf{Q}_t$$

Now we need to do the **update** step!

# General Filtering Equations

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \propto P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

**Prediction:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) P(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{1:t-1}) d\boldsymbol{x}_{t-1}$$

**Update:**

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t) P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

Belief state | State observation | Predicted State

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t)P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

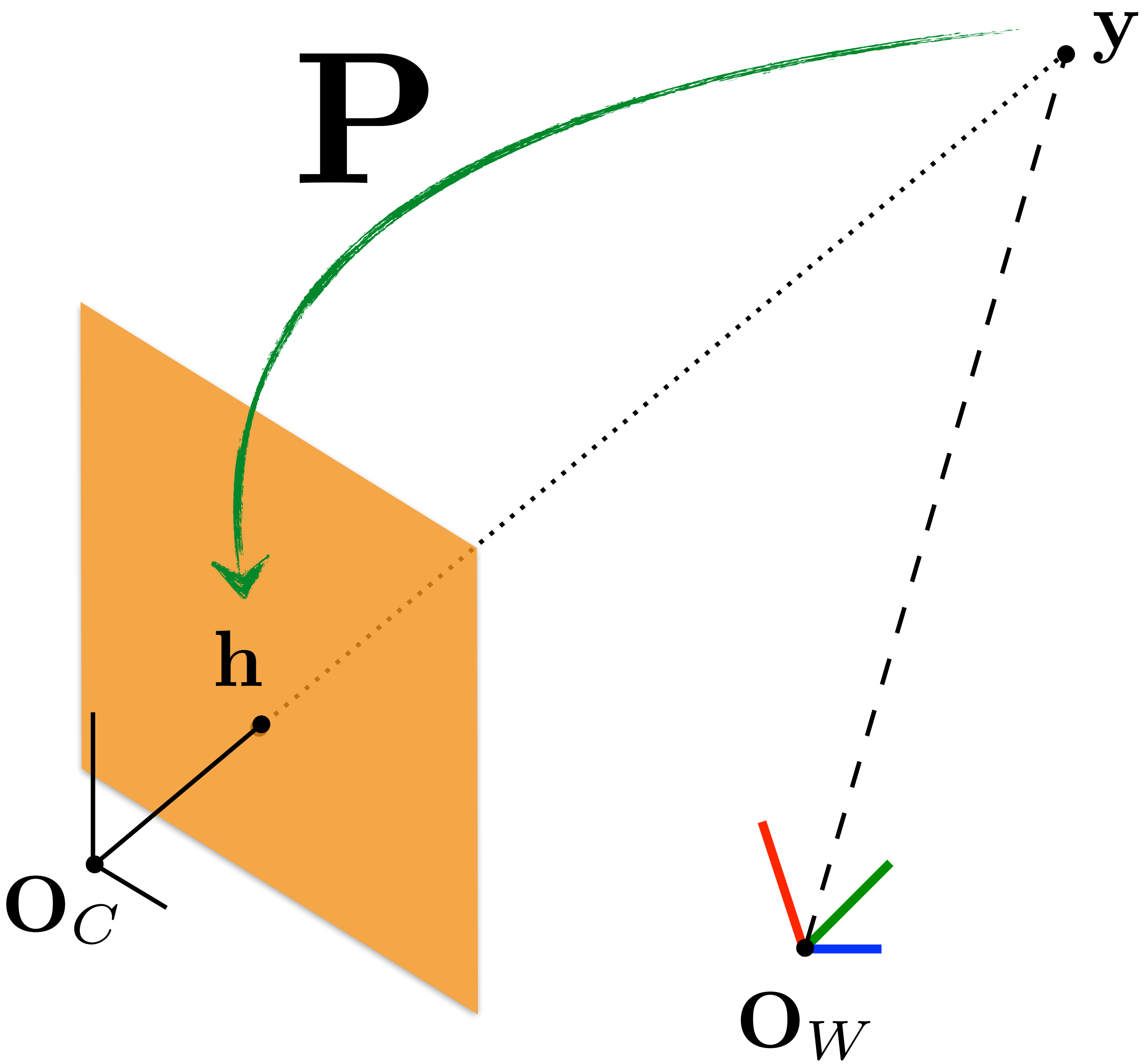
*What are the observations?*

2D projections of 3D landmarks

Recall, the state includes the 3D location of landmarks

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}$$

*What is the projection from 3D point to 2D image point?*



# Observation Model

$$P(\boldsymbol{z}_t | \boldsymbol{x}_t)$$

*If you know the 3D location of a landmark, what is the 2D projection?*

**Non-linear observation model**

$$\mathbf{h} \sim \mathbf{P}\ \mathbf{y}$$

2D Image Point

Camera matrix

3D World Point

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R}|\mathbf{T}\right]$$

*What do we know about **P**?*

*How do we make the observation model linear?*

$$H = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$$

(2n x 13)

n: number of visible points

I will spare you the pain of deriving the partial derivative…

$$P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = P(\boldsymbol{z}_t|\boldsymbol{x}_t)P(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$$

Update step (mean):

Kalman gain

$$\mathbf{x}_t = \mathbf{x}_t + \mathbf{K}_t(\mathbf{z}_t - \mathbf{h}(\mathbf{y};\mathbf{x}_t))$$

Updated state

Predicted state

Matched 2D features

2D projection of 3D point

Update step (covariance):

Identity

Kalman gain

$$\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\boldsymbol{\Sigma}_t$$

Covariance (updated)

Jacobian

Covariance (predicted)

# Kintinuous: Spatially Extended Kinect Fusion

Thomas Whelan, John McDonald
National University of Ireland Maynooth, Ireland

Michael Kaess, Maurice Fallon, Hordur Johannsson, John J. Leonard
Computer Science and Artificial Intelligence Laboratory, MIT, USA

NUI MAYNOOTH

CSAIL