Bayesian approaches and ML techniques in multisource fusion systems

Emanuel Aldea

Data Analysis Group, SATIE Laboratory, Université Paris-Saclay

Outline

Outline

Multisource fusion

Combine information from multiple sources and sensors in order to achieve analysis and decision-supporting inferences that cannot be achieved with a single sensor or source.

Main objectives

- Data fusion : combine multiple sources of information (e.g. sensor data) in order to provide an optimal estimation of the state of the system of interest
- Information fusion : take the optimal decision, given multiple sources of information (e.g. classifiers)

Multisource fusion

Combine information from multiple sources and sensors in order to achieve analysis and decision-supporting inferences that cannot be achieved with a single sensor or source.

Main objectives

- Data fusion : combine multiple sources of information (e.g. sensor data) in order to provide an optimal estimation of the state of the system of interest
- Information fusion : take the optimal decision, given multiple sources of information (e.g. classifiers)

Multisource fusion

Combine information from multiple sources and sensors in order to achieve analysis and decision-supporting inferences that cannot be achieved with a single sensor or source.

Main objectives

- Data fusion : combine multiple sources of information (e.g. sensor data) in order to provide an optimal estimation of the state of the system of interest
- Information fusion : take the optimal decision, given multiple sources of information (e.g. classifiers)

Multisource fusion

Combine information from multiple sources and sensors in order to achieve analysis and decision-supporting inferences that cannot be achieved with a single sensor or source.

Main objectives

- Data fusion : combine multiple sources of information (e.g. sensor data) in order to provide an optimal estimation of the state of the system of interest
- Information fusion : take the optimal decision, given multiple sources of information (e.g. classifiers)

Typical challenges:

- Varying levels of uncertainty of sources
- Missing or sparse data
- Non homogeneity of spatio-temporal scales
- Computational cost

Estimation

The first focus will be on estimation and we will focus on:

- Bayesian recursive filtering and how to apply it in a tractable manner
- Kalman filtering (and friends)
- Data fusion for estimation with a complete example

Estimation

The first focus will be on estimation and we will focus on:

- Bayesian recursive filtering and how to apply it in a tractable manner
- Kalman filtering (and friends)
- Data fusion for estimation with a complete example

Estimation

The first focus will be on estimation and we will focus on:

- Bayesian recursive filtering and how to apply it in a tractable manner
- Kalman filtering (and friends)
- Data fusion for estimation with a complete example

Outline

The principles

A relatively simple and general framework:

- Data fusion objective : obtain an estimation $\hat{x}$ of an unknown vector state x given a noisy measurement vector z provided by a range of sensors:

$$z = h(x, v)$$

  where v represents the noise.

- The famous Bayes rule : $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$
- $p(x)$ and $p(y)$: pdf (probability density function) of x and y
- $p(x, y)$: joint pdf of x and y
- $p(x|y)$: conditional pdf of x given y

The principles

A relatively simple and general framework:

- Data fusion objective : obtain an estimation $\hat{\mathrm{x}}$ of an unknown vector state x given a noisy measurement vector z provided by a range of sensors:

$$\mathrm{z} = h(\mathrm{x}, \mathrm{v})$$

  where v represents the noise.
- The famous Bayes rule : $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$
- $p(x)$ and $p(y)$: pdf (probability density function) of x and y
- $p(x, y)$: joint pdf of x and y
- $p(x|y)$: conditional pdf of x given y

The principles

A relatively simple and general framework:

- Data fusion objective : obtain an estimation $\hat{x}$ of an unknown vector state x given a noisy measurement vector z provided by a range of sensors:

$$z = h(x, v)$$

  where v represents the noise.
- The famous Bayes rule : $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$
- $p(x)$ and $p(y)$: pdf (probability density function) of x and y
- $p(x, y)$: joint pdf of x and y
- $p(x|y)$: conditional pdf of x given y

The principles

A relatively simple and general framework:

- Data fusion objective : obtain an estimation $\hat{\mathrm{x}}$ of an unknown vector state x given a noisy measurement vector z provided by a range of sensors:

$$\mathrm{z} = h(\mathrm{x}, \mathrm{v})$$

  where v represents the noise.
- The famous Bayes rule : $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$
- $p(x)$ and $p(y)$: pdf (probability density function) of x and y
- $p(x, y)$: joint pdf of x and y
- $p(x|y)$: conditional pdf of x given y

The principles

A relatively simple and general framework:

- Data fusion objective : obtain an estimation $\hat{\mathrm{x}}$ of an unknown vector state x given a noisy measurement vector z provided by a range of sensors:

$$\mathrm{z} = h(\mathrm{x}, \mathrm{v})$$

  where v represents the noise.
- The famous Bayes rule : $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$
- $p(x)$ and $p(y)$: pdf (probability density function) of x and y
- $p(x, y)$: joint pdf of x and y
- $p(x|y)$: conditional pdf of x given y

Our estimation

Considering the Bayes rule, it is straightforward to obtain:

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)}$$

- Likelihood $p(z|x)$: the link between the observation and the unknown state x
- Prior $p(x)$: the prior knowledge we have on x
- $p(z)$ is the knowledge we have on z irrespective of x; since it does not involve the state, it is rarely used in practice

Our estimation

Considering the Bayes rule, it is straightforward to obtain:

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)}$$

- Likelihood $p(z|x)$: the link between the observation and the unknown state x
- Prior $p(x)$: the prior knowledge we have on x
- $p(z)$ is the knowledge we have on z irrespective of x; since it does not involve the state, it is rarely used in practice

Our estimation

Considering the Bayes rule, it is straightforward to obtain:

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)}$$

- Likelihood $p(z|x)$: the link between the observation and the unknown state x
- Prior $p(x)$: the prior knowledge we have on x
- $p(z)$ is the knowledge we have on z irrespective of x; since it does not involve the state, it is rarely used in practice

Our estimation

Considering the Bayes rule, it is straightforward to obtain:

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)}$$

- Likelihood $p(z|x)$: the link between the observation and the unknown state x
- Prior $p(x)$: the prior knowledge we have on x
- $p(z)$ is the knowledge we have on z irrespective of x; since it does not involve the state, it is rarely used in practice

The likelihood

- Note that z is a random variable while x is known : measurement characterization knowing x
- Typically provided by $z = h(x, v)$

Estimators

Even if we know $p(x|z)$, we usually want a good estimation $\hat{x}$ of the variable x

Properties

- We characterize an estimator by its bias $B_{\hat{x}} = \mathbb{E}[\hat{x} - x] = \mathbb{E}[\hat{x}] - x$, which is ideally 0 (i.e. unbiased estimator)
- And by its variance $\mathbb{E}[\tilde{x}] = \mathbb{E}[(\hat{x} - x)^2] = \mathbb{E}[\hat{x}^2] - \mathbb{E}[x]^2$
- The variance should be as small as possible
- The minimum variance can be estimated theoretically: it is the CRLB

Estimators

Even if we know $p(x|z)$, we usually want a good estimation $\hat{x}$ of the variable x

Properties

- We characterize an estimator by its bias $B_{\hat{x}} = \mathbb{E}[\hat{x} - x] = \mathbb{E}[\hat{x}] - x$, which is ideally 0 (i.e. unbiased estimator)
- And by its variance $\mathbb{E}[\hat{x}] = \mathbb{E}[(\hat{x} - x)^2] = \mathbb{E}[\hat{x}^2] - \mathbb{E}[x]^2$
- The variance should be as small as possible
- The minimum variance can be estimated theoretically: it is the CRLB

Estimators

Even if we know $p(x|z)$, we usually want a good estimation $\hat{x}$ of the variable x

Properties

- We characterize an estimator by its bias $B_{\hat{x}} = \mathbb{E}[\hat{x} - x] = \mathbb{E}[\hat{x}] - x$, which is ideally 0 (i.e. unbiased estimator)
- And by its variance $\mathbb{E}[\hat{x}] = \mathbb{E}[(\hat{x} - x)^2] = \mathbb{E}[\hat{x}^2] - \mathbb{E}[x]^2$
- The variance should be as small as possible
- The minimum variance can be estimated theoretically: it is the CRLB

Estimators

Even if we know $p(x|z)$, we usually want a good estimation $\hat{x}$ of the variable x

Properties

- We characterize an estimator by its bias $B_{\hat{x}} = \mathbb{E}[\hat{x} - x] = \mathbb{E}[\hat{x}] - x$, which is ideally 0 (i.e. unbiased estimator)
- And by its variance $\mathbb{E}[\hat{x}] = \mathbb{E}[(\hat{x} - x)^2] = \mathbb{E}[\hat{x}^2] - \mathbb{E}[x]^2$
- The variance should be as small as possible
- The minimum variance can be estimated theoretically: it is the CRLB

Estimators

Even if we know $p(x|z)$, we usually want a good estimation $\hat{x}$ of the variable x

Properties

- We characterize an estimator by its bias $B_{\hat{x}} = \mathbb{E}[\hat{x} - x] = \mathbb{E}[\hat{x}] - x$, which is ideally 0 (i.e. unbiased estimator)
- And by its variance $\mathbb{E}[\hat{x}] = \mathbb{E}[(\hat{x} - x)^2] = \mathbb{E}[\hat{x}^2] - \mathbb{E}[x]^2$
- The variance should be as small as possible
- The minimum variance can be estimated theoretically: it is the CRLB

Common estimators

- The MAP $\hat{x}_{MAP}$ is the value x such that $p(x|z)$ is maximum: $\hat{x}_{MAP} = \arg\max_x p(x|z)$
- The MMSE mimizes the squared error sum:

$$\hat{x}_{MMSE} = \arg\min_x \mathbb{E}[(\hat{x}] - x)^T(\hat{x}] - x)]$$

- We can demonstrate that $\hat{x}_{MMSE} = \mathbb{E}[x|z] = \int x p(x|z) dx$

Estimators

Even if we know $p(x|z)$, we usually want a good estimation $\hat{x}$ of the variable x

Properties

- We characterize an estimator by its bias $B_{\hat{x}} = \mathbb{E}[\hat{x} - x] = \mathbb{E}[\hat{x}] - x$, which is ideally 0 (i.e. unbiased estimator)
- And by its variance $\mathbb{E}[\hat{x}] = \mathbb{E}[(\hat{x} - x)^2] = \mathbb{E}[\hat{x}^2] - \mathbb{E}[x]^2$
- The variance should be as small as possible
- The minimum variance can be estimated theoretically: it is the CRLB

Common estimators

- The MAP $\hat{x}_{MAP}$ is the value x such that $p(x|z)$ is maximum: $\hat{x}_{MAP} = \arg\max_x p(x|z)$
- The MMSE mimizes the squared error sum:

$$\hat{x}_{MMSE} = \arg\min_x \mathbb{E}[(\hat{x}] - x)^T(\hat{x}] - x)]$$

- We can demonstrate that $\hat{x}_{MMSE} = \mathbb{E}[x|z] = \int x p(x|z) dx$

Estimators

Even if we know $p(x|z)$, we usually want a good estimation $\hat{x}$ of the variable x

Properties

- We characterize an estimator by its bias $B_{\hat{x}} = \mathbb{E}[\hat{x} - x] = \mathbb{E}[\hat{x}] - x$, which is ideally 0 (i.e. unbiased estimator)
- And by its variance $\mathbb{E}[\hat{x}] = \mathbb{E}[(\hat{x} - x)^2] = \mathbb{E}[\hat{x}^2] - \mathbb{E}[x]^2$
- The variance should be as small as possible
- The minimum variance can be estimated theoretically: it is the CRLB

Common estimators

- The MAP $\hat{x}_{MAP}$ is the value x such that $p(x|z)$ is maximum: $\hat{x}_{MAP} = \arg\max_x p(x|z)$
- The MMSE mimizes the squared error sum:

$$\hat{x}_{MMSE} = \arg\min_x \mathbb{E}[(\hat{x}] - x)^T(\hat{x}] - x)]$$

- We can demonstrate that $\hat{x}_{MMSE} = \mathbb{E}[x|z] = \int x p(x|z) dx$

Estimators - in practice

- The behaviour is approximately the same for symmetrical distributions $p(x|z)$
- Multimodal posterior distributions : significant errors / unstable estimations
- The MAP is usually easier to compute
- The denominator in the Bayes rule $p(z)$ does not matter since it does not depend on x

Exercise

Let us consider a scalar measurement z such that $z = \theta + w$ with :

- w : noise such that $w \sim \mathcal{N}(0, \sigma_w^2)$
- the prior distribution for $\theta \sim \mathcal{N}(\theta_0, \sigma_\theta^2)$

Determine the Bayesian estimator $\hat{\theta}$ of θ.

Solution

We need to compute $p(z|\theta)p(\theta)$.
Since z $= \theta +$ w and w $\sim \mathcal{N}(0, \sigma_w^2)$, we have z $\sim \mathcal{N}(\theta, \sigma_w^2)$.
Therefore, $p(z|\theta) = \mathcal{N}(\theta, \sigma_w^2)$, and we also know that $p(\theta) = \mathcal{N}(\theta_0, \sigma_\theta^2)$. The product of two Gaussian functions $\mathcal{N}(\mu_1, C_1)$ and $\mathcal{N}(\mu_2, C_2)$ is also a Gaussian function given by $k\mathcal{N}(\mu, C)$, with:

$$C = [C_1{}^{-1} + C_2{}^{-1}]^{-1} \qquad \mu = [C_1{}^{-1} + C_2{}^{-1}]^{-1}[C_1{}^{-1}\mu_1 + C_2{}^{-1}\mu_2]$$

So, $p(z|\theta)p(\theta)$ will be an un-normalized Gaussian function

$$p(z|\theta)p(\theta) \propto \mathcal{N}(\mu, \sigma^2) \propto \mathcal{N}(z, \sigma_w^2)\mathcal{N}(\theta_0, \sigma_\theta^2)$$

With :

$$\sigma^2 = \frac{\sigma_w^2\sigma_\theta^2}{\sigma_w^2 + \sigma_\theta^2} \qquad \mu = \frac{\sigma_w^2\theta_0 + \sigma_\theta^2 z}{\sigma_w^2 + \sigma_\theta^2}$$

Since the result is a Gaussian function, we have for MAP/MMSE the same result $\hat{\theta} = \mu$

Dynamic Estimation

This problem is more challenging : take into account the temporal evolution of the system.

Notations

- x_k : real state vector at time k
- z_k : measurement performed at time k
- Z_k : set of measurements performed up to time k

The objective is to estimate x_k by exploiting all the available observations:

$$p(x_k|Z_k) = \frac{p(Z_k|x_k) \times p(x_k)}{p(Z_k)}$$

Definition

Under a Markovian assumption, we can formulate :

$$p(x_{0:t}|z_{1:t}) \propto p(z_t|x_t)p(x_t|x_{t-1})p(x_{0:t-1}|z_{1:t-1})$$

Proof

$$p(x_{0:t}|z_{1:t}) \quad = \quad p(x_{0:t}|z_t, z_{1:t-1})$$

Proof

$$\begin{aligned} p(x_{0:t}|z_{1:t}) &= p(x_{0:t}|z_t, z_{1:t-1}) \\ &= \frac{p(z_t, z_{1:t-1}|x_{0:t})p(x_{0:t})}{p(z_t, z_{1:t-1})} \end{aligned}$$

Bayes rule

$$p(A|B,C) = p(B,C|A)\frac{p(A)}{p(B,C)}$$

Proof

$$
\begin{aligned}
p(x_{0:t}|z_{1:t}) &= p(x_{0:t}|z_t, z_{1:t-1}) \\
&= \frac{p(z_t, z_{1:t-1}|x_{0:t})p(x_{0:t})}{p(z_t, z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(z_{1:t-1}|x_{0:t})p(x_{0:t})}{p(z_t, z_{1:t-1})}
\end{aligned}
$$

Bayes rule

$$p(A, B|C) = p(A|B, C)p(B|C)$$

Proof

$$
\begin{aligned}
p(x_{0:t}|z_{1:t}) &= p(x_{0:t}|z_t, z_{1:t-1}) \\
&= \frac{p(z_t, z_{1:t-1}|x_{0:t})p(x_{0:t})}{p(z_t, z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(z_{1:t-1}|x_{0:t})p(x_{0:t})}{p(z_t, z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_{0:t}|z_{1:t-1})p(z_{1:t-1})p(x_{0:t})}{p(x_{0:t})p(z_t, z_{1:t-1})}
\end{aligned}
$$

Bayes rule

$$p(A|B) = p(B|A)\frac{p(A)}{p(B)}$$

Proof

$$
\begin{aligned}
p(x_{0:t}|z_{1:t}) &= p(x_{0:t}|z_t, z_{1:t-1}) \\
&= \frac{p(z_t, z_{1:t-1}|x_{0:t})p(x_{0:t})}{p(z_t, z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(z_{1:t-1}|x_{0:t})p(x_{0:t})}{p(z_t, z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_{0:t}|z_{1:t-1})p(z_{1:t-1})p(x_{0:t})}{p(x_{0:t})p(z_t, z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_{0:t}|z_{1:t-1})p(z_{1:t-1})}{p(z_t|z_{1:t-1})p(z_{1:t-1})}
\end{aligned}
$$

Bayes rule

$$p(A, B) = p(A|B)p(B)$$

Proof

$$p(x_{0:t}|z_{1:t}) = \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_{0:t}|z_{1:t-1})}{p(z_t|z_{1:t-1})}$$

Proof

$$
\begin{aligned}
p(x_{0:t}|z_{1:t}) &= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_{0:t}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_t, x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})}
\end{aligned}
$$

Proof

$$
\begin{aligned}
p(x_{0:t}|z_{1:t}) &= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_{0:t}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_t, x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_t|x_{0:t-1}, z_{1:t-1})p(x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})}
\end{aligned}
$$

Bayes rule

$$p(A, B|C) = p(A|B, C)p(B|C)$$

Proof

$$
\begin{aligned}
p(x_{0:t}|z_{1:t}) &= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_{0:t}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_t, x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\
&= \frac{p(z_t|z_{1:t-1}, x_{0:t})p(x_t|x_{0:t-1}, z_{1:t-1})p(x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\
&\propto p(z_t|z_{1:t-1}, x_{0:t})p(x_t|x_{0:t-1}, z_{1:t-1})p(x_{0:t-1}|z_{1:t-1})
\end{aligned}
$$

Proof

Markovian hypotheses

- the current state depends only on the previous state:

$$p(x_t|x_{0:t-1}, z_{1:t-1}) = p(x_t|x_{t-1})$$

- the observations are conditionally independent with respect to the states, and depend only on the current state:

$$p(z_t|z_{1:t-1}, x_{0:t}) = p(z_t|x_t)$$

Proof

Markovian hypotheses

- the current state depends only on the previous state:

$$p(x_t|x_{0:t-1}, z_{1:t-1}) = p(x_t|x_{t-1})$$

- the observations are conditionally independent with respect to the states, and depend only on the current state:

$$p(z_t|z_{1:t-1}, x_{0:t}) = p(z_t|x_t)$$

Proof

Markovian hypotheses

- the current state depends only on the previous state:

$$p(x_t|x_{0:t-1}, z_{1:t-1}) = p(x_t|x_{t-1})$$

- the observations are conditionally independent with respect to the states, and depend only on the current state:

$$p(z_t|z_{1:t-1}, x_{0:t}) = p(z_t|x_t)$$

$$\begin{aligned} p(x_{0:t}|z_{1:t}) &\propto p(z_t|z_{1:t-1}, x_{0:t})p(x_t|x_{0:t-1}, z_{1:t-1})p(x_{0:t-1}|z_{1:t-1}) \\ &\propto p(z_t|x_t)p(x_t|x_{t-1})p(x_{0:t-1}|z_{1:t-1}) \end{aligned}$$

Using recursion

$$p(x_{0:t}|z_{1:t}) \propto \underbrace{p(z_t|x_t)}_{\text{likelihood}} \underbrace{p(x_t|x_{t-1})}_{\text{transition}} p(x_{0:t-1}|z_{1:t-1})$$

Solutions

- Kalman filter : linear functions, Gaussian densities, analytical solution
- Extended Kalman filter : linearized functions
- Particle filter : tractable approximation for general functions

Outline

Kalman Filter

State equations

$$x_t = F_t x_{t-1} (+B_t u_t) + w_t$$

- x_t : state vector at time t
- u_t : control vector
- F_t : state transition matrix between $t-1$ and t
- B_t : control matrix
- w_t : state noise vector, $w_t \sim \mathcal{N}(0, W_t), \mathbb{E}[w_t w_t^T] = W_t$

Kalman Filter

State equations

$$x_t = F_t x_{t-1} (+B_t u_t) + w_t$$

- x_t : state vector at time t
- u_t : control vector
- F_t : state transition matrix between $t-1$ and t
- B_t : control matrix
- w_t : state noise vector, $w_t \sim \mathcal{N}(0, W_t), \mathbb{E}[w_t w_t^T] = W_t$

Kalman Filter

State equations

$$x_t = F_t x_{t-1} (+B_t u_t) + w_t$$

- x_t : state vector at time t
- u_t : control vector
- F_t : state transition matrix between $t-1$ and t
- B_t : control matrix
- w_t : state noise vector, $w_t \sim \mathcal{N}(0, W_t), \mathbb{E}[w_t w_t^T] = W_t$

Kalman Filter

State equations

$$x_t = F_t x_{t-1} (+B_t u_t) + w_t$$

- x_t : state vector at time t
- u_t : control vector
- F_t : state transition matrix between $t-1$ and t
- B_t : control matrix
- w_t : state noise vector, $w_t \sim \mathcal{N}(0, W_t), \mathbb{E}[w_t w_t^T] = W_t$

Kalman Filter

State equations

$$x_t = F_t x_{t-1} (+B_t u_t) + w_t$$

- x_t : state vector at time t
- u_t : control vector
- F_t : state transition matrix between $t-1$ and t
- B_t : control matrix
- w_t : state noise vector, $w_t \sim \mathcal{N}(0, W_t), \mathbb{E}[w_t w_t^T] = W_t$

Kalman Filter

State equations

$$x_t = F_t x_{t-1} (+B_t u_t) + w_t$$

- x_t : state vector at time t
- u_t : control vector
- F_t : state transition matrix between $t-1$ and t
- B_t : control matrix
- w_t : state noise vector, $w_t \sim \mathcal{N}(0, W_t), \mathbb{E}[w_t w_t^T] = W_t$

In the following, we will ignore the control term:

$$x_t = F_t x_{t-1} + w_t$$

Kalman Filter

Measurement equation

$$z_t = H_t x_t + v_t$$

- z_t : observation vector at time t
- H_t : measurement matrix
- v_t : measurement noise vector, $v_t \sim \mathcal{N}(0, V_t), \mathbb{E}[v_t v_t^T] = V_t$

Kalman Filter

Measurement equation

$$z_t = H_t x_t + v_t$$

- z_t : observation vector at time t
- H_t : measurement matrix
- v_t : measurement noise vector, $v_t \sim \mathcal{N}(0, V_t), \mathbb{E}[v_t v_t^T] = V_t$

Kalman Filter

Measurement equation

$$z_t = H_t x_t + v_t$$

- z_t : observation vector at time t
- H_t : measurement matrix
- v_t : measurement noise vector, $v_t \sim \mathcal{N}(0, V_t), \mathbb{E}[v_t v_t^T] = V_t$

Kalman Filter

Two-step recursion

- Given $x_{t-1|t-1}$ the optimal non-biased estimator of x_{t-1}
- Given the observation z_t

1. Compute the non-biased estimator $x_{t|t-1}$ of x_t without using the observation z_t
2. Compute the non-biased estimator $x_{t|t}$ of x_t using the observation z_t

Kalman Filter

Two-step recursion

- Given $x_{t-1|t-1}$ the optimal non-biased estimator of x_{t-1}
- Given the observation z_t

1. Compute the non-biased estimator $x_{t|t-1}$ of x_t without using the observation z_t
2. Compute the non-biased estimator $x_{t|t}$ of x_t using the observation z_t

Kalman Filter

Two-step recursion

- Given $x_{t-1|t-1}$ the optimal non-biased estimator of x_{t-1}
- Given the observation z_t

1. Compute the non-biased estimator $x_{t|t-1}$ of x_t without using the observation z_t
2. Compute the non-biased estimator $x_{t|t}$ of x_t using the observation z_t

Principle : feedback control

Objective : recursively minimize the error between estimation and state

1. Prediction (theoretical value) step : with the parameters from $t-1$
2. Correction step : using the current observation to correct the current prediction

Kalman Filter

Two-step recursion

- Given $x_{t-1|t-1}$ the optimal non-biased estimator of x_{t-1}
- Given the observation z_t

1. Compute the non-biased estimator $x_{t|t-1}$ of x_t without using the observation z_t
2. Compute the non-biased estimator $x_{t|t}$ of x_t using the observation z_t

Principle : feedback control

Objective : recursively minimize the error between estimation and state

1. Prediction (theoretical value) step : with the parameters from $t-1$
2. Correction step : using the current observation to correct the current prediction

Kalman Filter

Densities

- Posterior density at $t-1$:

$$p(\mathbf{x}_{0:t-1}|\mathbf{z}_{1:t-1}) = \mathcal{N}(\mathbf{x}_{t-1|t-1}, \mathbf{P}_{t-1|t-1})$$

- Prior density at t :

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t-1}) = \mathcal{N}(\mathbf{x}_{t|t-1}, \mathbf{P}_{t|t-1})$$

- Posterior density at t :

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) = \mathcal{N}(\mathbf{x}_{t|t}, \mathbf{P}_{t|t})$$

Kalman Filter

Densities

- Posterior density at $t-1$:

$$p(\mathrm{x}_{0:t-1}|\mathrm{z}_{1:t-1}) = \mathcal{N}(\mathrm{x}_{t-1|t-1}, \mathrm{P}_{t-1|t-1})$$

- Prior density at t :

$$p(\mathrm{x}_{0:t}|\mathrm{z}_{1:t-1}) = \mathcal{N}(\mathrm{x}_{t|t-1}, \mathrm{P}_{t|t-1})$$

- Posterior density at t :

$$p(\mathrm{x}_{0:t}|\mathrm{z}_{1:t}) = \mathcal{N}(\mathrm{x}_{t|t}, \mathrm{P}_{t|t})$$

Kalman Filter

Densities

- Posterior density at $t-1$:

$$p(\mathbf{x}_{0:t-1}|\mathbf{z}_{1:t-1}) = \mathcal{N}(\mathbf{x}_{t-1|t-1}, \mathbf{P}_{t-1|t-1})$$

- Prior density at t :

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t-1}) = \mathcal{N}(\mathbf{x}_{t|t-1}, \mathbf{P}_{t|t-1})$$

- Posterior density at t :

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) = \mathcal{N}(\mathbf{x}_{t|t}, \mathbf{P}_{t|t})$$

General algorithm

Initialization

- $x_0 = x_{0|0} = \mathbb{E}[x_0] = \mu_0, P_0 = \mathbb{E}[(x_0 - \mu_0)(x_0 - \mu_0)^T]$

Prediction of $x_{t|t-1}$

Using only the state equation :

$$x_{t|t-1} = \mathbb{E}[x_t | z_{1:t-1}]$$

General algorithm

Initialization

- $x_0 = x_{0|0} = \mathbb{E}[x_0] = \mu_0, P_0 = \mathbb{E}[(x_0 - \mu_0)(x_0 - \mu_0)^T]$

Prediction of $x_{t|t-1}$

Using only the state equation :

$$\begin{aligned} x_{t|t-1} &= \mathbb{E}[x_t | z_{1:t-1}] \\ &= \mathbb{E}[F_t x_{t-1} + w_t] \end{aligned}$$

General algorithm

Initialization

- $x_0 = x_{0|0} = \mathbb{E}[x_0] = \mu_0, P_0 = \mathbb{E}[(x_0 - \mu_0)(x_0 - \mu_0)^T]$

Prediction of $x_{t|t-1}$

Using only the state equation :

$$\begin{aligned} x_{t|t-1} &= \mathbb{E}[x_t | z_{1:t-1}] \\ &= \mathbb{E}[F_t x_{t-1} + w_t] \\ &= \mathbb{E}[F_t x_{t-1}] + \mathbb{E}[w_t] \end{aligned}$$

General algorithm

Initialization

- $x_0 = x_{0|0} = \mathbb{E}[x_0] = \mu_0, P_0 = \mathbb{E}[(x_0 - \mu_0)(x_0 - \mu_0)^T]$

Prediction of $x_{t|t-1}$

Using only the state equation :

$$\begin{aligned} x_{t|t-1} &= \mathbb{E}[x_t | z_{1:t-1}] \\ &= \mathbb{E}[F_t x_{t-1} + w_t] \\ &= \mathbb{E}[F_t x_{t-1}] + \mathbb{E}[w_t] \\ &= F_t x_{t-1|t-1} \end{aligned}$$

General algorithm

Prediction error

$$e_{t|t-1} = x_t - x_{t|t-1}$$

General algorithm

Prediction error

$$\begin{aligned} e_{t|t-1} &= x_t - x_{t|t-1} \\ &= F_t x_{t-1} + w_t - F_t x_{t-1|t-1} \end{aligned}$$

General algorithm

Prediction error

$$\begin{aligned} e_{t|t-1} &= x_t - x_{t|t-1} \\ &= F_t x_{t-1} + w_t - F_t x_{t-1|t-1} \\ &= F_t(x_{t-1} - x_{t-1|t-1}) + w_t \end{aligned}$$

General algorithm

Prediction error

$$
\begin{aligned}
e_{t|t-1} &= x_t - x_{t|t-1} \\
&= F_t x_{t-1} + w_t - F_t x_{t-1|t-1} \\
&= F_t(x_{t-1} - x_{t-1|t-1}) + w_t \\
&= F_t e_{t-1|t-1} + w_t
\end{aligned}
$$

General algorithm

Prediction of the state covariance error $P_{t|t-1}$

By definition :

$$P_{t|t-1} = \mathbb{E}[e_{t|t-1} e_{t|t-1}^T]$$

General algorithm

Prediction of the state covariance error $P_{t|t-1}$

By definition :

$$\begin{aligned} P_{t|t-1} &= \mathbb{E}[e_{t|t-1} e_{t|t-1}^T] \\ &= \mathbb{E}[(F_t e_{t-1|t-1} + w_t)(F_t e_{t-1|t-1} + w_t)^T] \end{aligned}$$

General algorithm

Prediction of the state covariance error $P_{t|t-1}$

By definition :

$$\begin{aligned} P_{t|t-1} &= \mathbb{E}[e_{t|t-1} e_{t|t-1}^T] \\ &= \mathbb{E}[(F_t e_{t-1|t-1} + w_t)(F_t e_{t-1|t-1} + w_t)^T] \\ &= F_t \mathbb{E}[e_{t-1|t-1} e_{t-1|t-1}^T] F_t^T + \mathbb{E}[w_t w_t^T] \end{aligned}$$

General algorithm

Prediction of the state covariance error $P_{t|t-1}$

By definition :

$$\begin{aligned} P_{t|t-1} &= \mathbb{E}[e_{t|t-1}e_{t|t-1}^T] \\ &= \mathbb{E}[(F_t e_{t-1|t-1} + w_t)(F_t e_{t-1|t-1} + w_t)^T] \\ &= F_t \mathbb{E}[e_{t-1|t-1}e_{t-1|t-1}^T]F_t^T + \mathbb{E}[w_t w_t^T] \\ &= F_t P_{t-1|t-1} F_t^T + W_t \end{aligned}$$

General algorithm

Correction : exploiting the observation z_t

Update of the prediction based on the difference between the predicted observation $z_{t|t-1} = H_t x_{t|t-1}$ and the observation (innovation)

$$x_{t|t} = x_{t|t-1} + Q_t(z_t - H_t x_{t|t-1})$$

General algorithm

Correction : exploiting the observation z_t

Update of the prediction based on the difference between the predicted observation $z_{t|t-1} = H_t x_{t|t-1}$ and the observation (innovation)

$$\begin{aligned} x_{t|t} &= x_{t|t-1} + Q_t(z_t - H_t x_{t|t-1}) \\ &= (I - Q_t H_t) x_{t|t-1} + Q_t z_t \end{aligned}$$

General algorithm

Correction : exploiting the observation z_t

Update of the prediction based on the difference between the predicted observation $z_{t|t-1} = H_t x_{t|t-1}$ and the observation (innovation)

$$\begin{aligned} x_{t|t} &= x_{t|t-1} + Q_t(z_t - H_t x_{t|t-1}) \\ &= (I - Q_t H_t) x_{t|t-1} + Q_t z_t \\ &= Q'_t x_{t|t-1} + Q_t z_t \end{aligned}$$

General algorithm

Correction : exploiting the observation z_t

Update of the prediction based on the difference between the predicted observation $z_{t|t-1} = H_t x_{t|t-1}$ and the observation (innovation)

$$\begin{aligned} x_{t|t} &= x_{t|t-1} + Q_t(z_t - H_t x_{t|t-1}) \\ &= (I - Q_t H_t)x_{t|t-1} + Q_t z_t \\ &= Q'_t x_{t|t-1} + Q_t z_t \end{aligned}$$

Question : how to set the gain matrix Q_t so that it minimizes the MSE between the state and its estimation?

General algorithm

Prediction error at t

$$e_{t|t} = x_t - x_{t|t}$$

General algorithm

Prediction error at t

$$\begin{aligned} e_{t|t} &= x_t - x_{t|t} \\ &= x_t - Q_t' x_{t|t-1} - Q_t z_t \end{aligned}$$

General algorithm

Prediction error at t

$$\begin{aligned} e_{t|t} &= x_t - x_{t|t} \\ &= x_t - Q_t' x_{t|t-1} - Q_t z_t \\ &= x_t - Q_t'(x_t - e_{t|t-1}) - Q_t(H_t x_t + v_t) \end{aligned}$$

General algorithm

Prediction error at t

$$
\begin{aligned}
e_{t|t} &= x_t - x_{t|t} \\
&= x_t - Q_t' x_{t|t-1} - Q_t z_t \\
&= x_t - Q_t'(x_t - e_{t|t-1}) - Q_t(H_t x_t + v_t) \\
&= (I - Q_t' - Q_t H_t)x_t + Q_t' e_{t|t-1} - Q_t v_t
\end{aligned}
$$

General algorithm

Prediction error at t

$$
\begin{aligned}
e_{t|t} &= x_t - x_{t|t} \\
&= x_t - Q_t' x_{t|t-1} - Q_t z_t \\
&= x_t - Q_t'(x_t - e_{t|t-1}) - Q_t(H_t x_t + v_t) \\
&= (I - Q_t' - Q_t H_t)x_t + Q_t' e_{t|t-1} - Q_t v_t \\
&= (I - Q_t' - Q_t H_t)x_t + Q_t'(x_t - x_{t|t-1}) - Q_t v_t
\end{aligned}
$$

General algorithm

Prediction error at t

$$
\begin{aligned}
e_{t|t} &= x_t - x_{t|t} \\
&= x_t - Q_t' x_{t|t-1} - Q_t z_t \\
&= x_t - Q_t'(x_t - e_{t|t-1}) - Q_t(H_t x_t + v_t) \\
&= (I - Q_t' - Q_t H_t)x_t + Q_t' e_{t|t-1} - Q_t v_t \\
&= (I - Q_t' - Q_t H_t)x_t + Q_t'(x_t - x_{t|t-1}) - Q_t v_t \\
&= x_t - Q_t H_t x_t - Q_t' x_{t|t-1} - Q_t v_t
\end{aligned}
$$

General algorithm

Prediction error at t

$$
\begin{aligned}
e_{t|t} &= x_t - x_{t|t} \\
&= x_t - Q_t' x_{t|t-1} - Q_t z_t \\
&= x_t - Q_t'(x_t - e_{t|t-1}) - Q_t(H_t x_t + v_t) \\
&= (I - Q_t' - Q_t H_t)x_t + Q_t' e_{t|t-1} - Q_t v_t \\
&= (I - Q_t' - Q_t H_t)x_t + Q_t'(x_t - x_{t|t-1}) - Q_t v_t \\
&= x_t - Q_t H_t x_t - Q_t' x_{t|t-1} - Q_t v_t \\
&= x_t - Q_t H_t x_t - (I - Q_t H_t)x_{t|t-1} - Q_t v_t
\end{aligned}
$$

General algorithm

Prediction error at t

$$
\begin{aligned}
e_{t|t} &= x_t - x_{t|t} \\
&= x_t - Q_t' x_{t|t-1} - Q_t z_t \\
&= x_t - Q_t'(x_t - e_{t|t-1}) - Q_t(H_t x_t + v_t) \\
&= (I - Q_t' - Q_t H_t)x_t + Q_t' e_{t|t-1} - Q_t v_t \\
&= (I - Q_t' - Q_t H_t)x_t + Q_t'(x_t - x_{t|t-1}) - Q_t v_t \\
&= x_t - Q_t H_t x_t - Q_t' x_{t|t-1} - Q_t v_t \\
&= x_t - Q_t H_t x_t - (I - Q_t H_t)x_{t|t-1} - Q_t v_t \\
&= (I - Q_t H_t)e_{t|t-1} - Q_t v_t
\end{aligned}
$$

General algorithm

Prediction of the state covariance error $P_{t|t}$

By definition :

$$P_{t|t} = \mathbb{E}[e_{t|t} e_{t|t}^T]$$

General algorithm

Prediction of the state covariance error $P_{t|t}$

By definition :

$$\begin{aligned} P_{t|t} &= \mathbb{E}[e_{t|t} e_{t|t}^T] \\ &= \mathbb{E}[((I - Q_t H_t) e_{t|t-1} - Q_t v_t)((I - Q_t H_t) e_{t|t-1} - Q_t v_t)^T] \end{aligned}$$

General algorithm

Prediction of the state covariance error $P_{t|t}$

By definition :

$$\begin{aligned} P_{t|t} &= \mathbb{E}[e_{t|t}e_{t|t}^T] \\ &= \mathbb{E}[((I - Q_t H_t)e_{t|t-1} - Q_t v_t)((I - Q_t H_t)e_{t|t-1} - Q_t v_t)^T] \\ &= (I - Q_t H_t)\mathbb{E}[e_{t|t-1}e_{t|t-1}^T](I - Q_t H_t)^T - Q_t \mathbb{E}[v_t v_t^T]Q_t^T \end{aligned}$$

General algorithm

Prediction of the state covariance error $P_{t|t}$

By definition :

$$\begin{aligned} P_{t|t} &= \mathbb{E}[e_{t|t}e_{t|t}^T] \\ &= \mathbb{E}[((I - Q_t H_t)e_{t|t-1} - Q_t v_t)((I - Q_t H_t)e_{t|t-1} - Q_t v_t)^T] \\ &= (I - Q_t H_t)\mathbb{E}[e_{t|t-1}e_{t|t-1}^T](I - Q_t H_t)^T - Q_t \mathbb{E}[v_t v_t^T]Q_t^T \\ &= (I - Q_t H_t)P_{t|t-1}(I - Q_t H_t)^T - Q_t V_t Q_t^T \end{aligned}$$

Discussion of the gain

The Kalmain gain must account for the uncertainties of the current estimation and of the observation :

- if the observation uncertainty V_t is small compared to $P_{t|t-1}$: strong gain (the observation is reliable)
- if the observation uncertainty V_t is large compared to $P_{t|t-1}$: small gain (we should not modify too much the current prediction)

Discussion of the gain

The Kalmain gain must account for the uncertainties of the current estimation and of the observation :

- if the observation uncertainty V_t is small compared to $P_{t|t-1}$: strong gain (the observation is reliable)
- if the observation uncertainty V_t is large compared to $P_{t|t-1}$: small gain (we should not modify too much the current prediction)

Discussion of the gain

The Kalmain gain must account for the uncertainties of the current estimation and of the observation :

- if the observation uncertainty V_t is small compared to $P_{t|t-1}$: strong gain (the observation is reliable)
- if the observation uncertainty V_t is large compared to $P_{t|t-1}$: small gain (we should not modify too much the current prediction)

Finding the gain

- It minimizes the MSE of the estimation : trace $(P_{t|t})$
- We can show that the optimal gain is:

$$Q_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + V^T)^{-1}$$

Discussion of the gain

The Kalmain gain must account for the uncertainties of the current estimation and of the observation :

- if the observation uncertainty V_t is small compared to $P_{t|t-1}$: strong gain (the observation is reliable)
- if the observation uncertainty V_t is large compared to $P_{t|t-1}$: small gain (we should not modify too much the current prediction)

Finding the gain

- It minimizes the MSE of the estimation : trace $(P_{t|t})$
- We can show that the optimal gain is:

$$Q_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + V^T)^{-1}$$

Conclusion

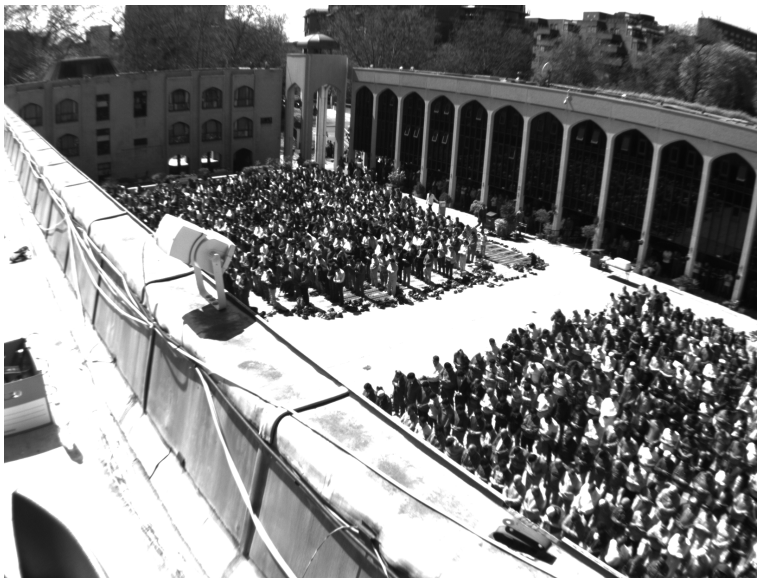
- Linear system and Gaussian noise assumptions : analytical, optimal solution
- Positive definite matrixes : matrix calculus
- Ideally suited for sensor fusion
- However, in reality it has its limitations (e.g. computer vision, tracking etc.): strongly nonlinear or unknown dynamics, non-parametric and multimodal densities
- More general approaches are needed (e.g., sequential Monte Carlo methods), but the cost is significant

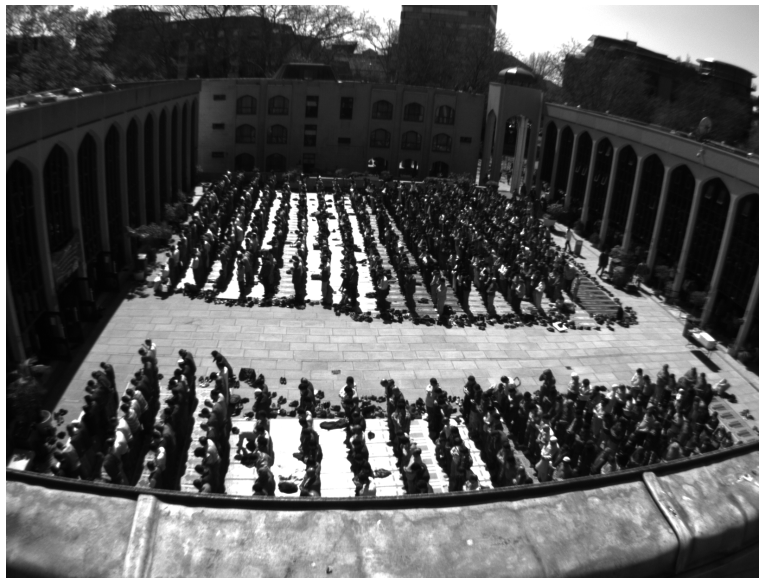
Outline

The context of this work

Head detection in crowded urban environments:

- Video surveillance
- Crowd modeling
- Action recognition

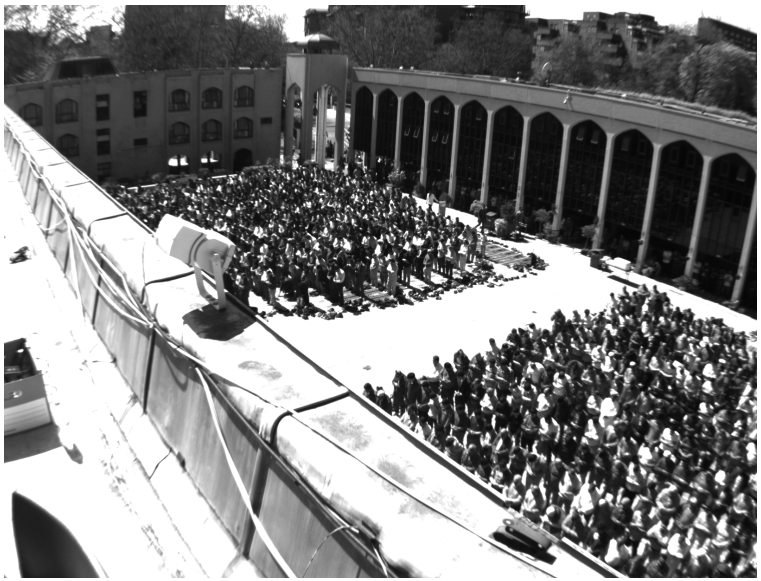


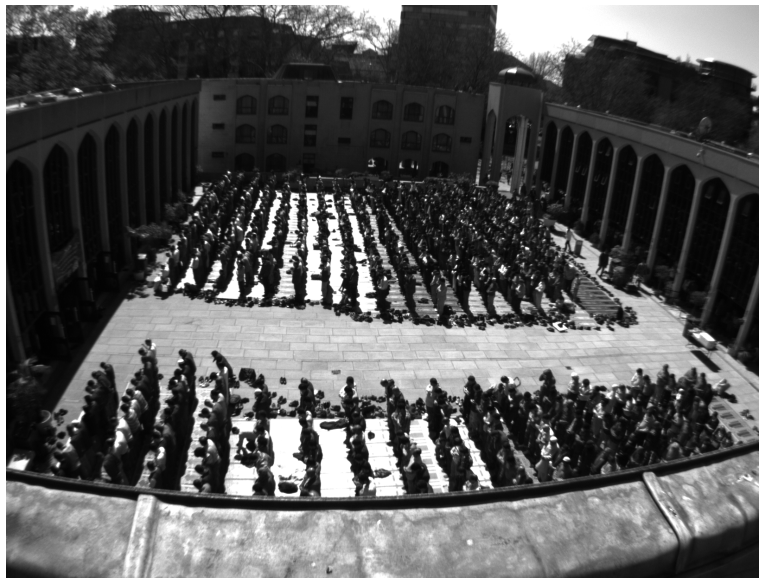


The context of this work

Head detection in crowded urban environments:

- Video surveillance
- Crowd modeling
- Action recognition

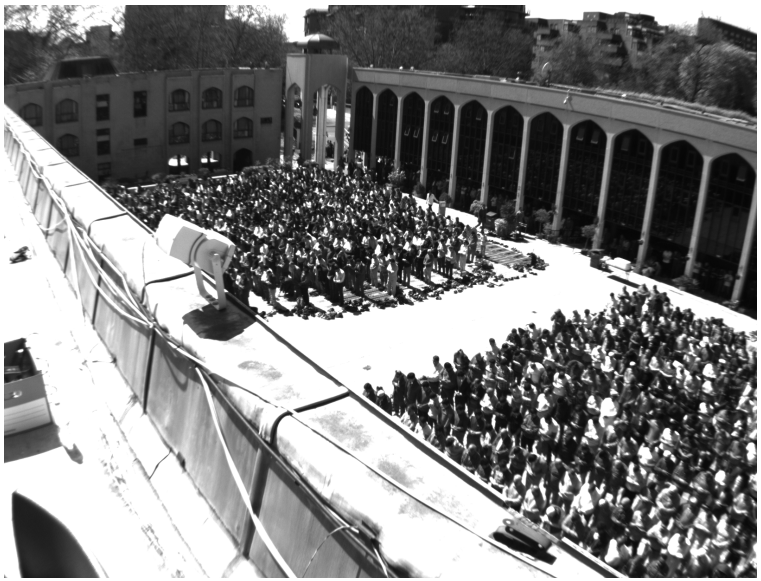


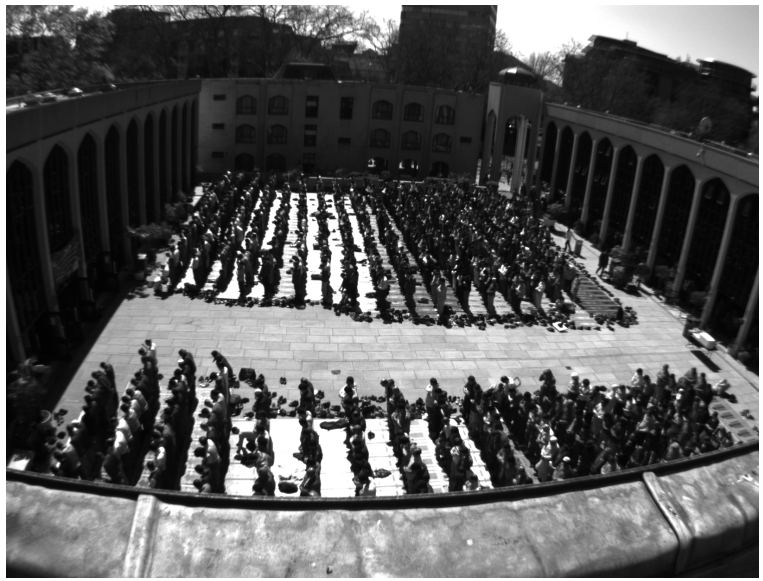


The context of this work

Head detection in crowded urban environments:

- Video surveillance
- Crowd modeling
- Action recognition





The context of this work

Dense crowds scenarios introduce novel challenges:

- Strong occlusion
- Background homogeneity
- Invisible ground plane and body parts
- Low detection performance in individual sensors

The context of this work

Dense crowds scenarios introduce novel challenges:

- Strong occlusion
- Background homogeneity
- Invisible ground plane and body parts
- Low detection performance in individual sensors

The context of this work

Dense crowds scenarios introduce novel challenges:

- Strong occlusion
- Background homogeneity
- Invisible ground plane and body parts
- Low detection performance in individual sensors

The context of this work

Dense crowds scenarios introduce novel challenges:

- Strong occlusion
- Background homogeneity
- Invisible ground plane and body parts
- Low detection performance in individual sensors

The context of this work

How to overcome dense crowd challenges in an unsupervised manner?

Camera network with multiple overlapping views.

- Exploit redundant information from multiple cameras.
- Sensor data fusion **prior to detection**.

The context of this work

How to overcome dense crowd challenges in an unsupervised manner?

Camera network with multiple overlapping views.

- Exploit redundant information from multiple cameras.
- Sensor data fusion **prior to detection**.

The context of this work

How to overcome dense crowd challenges in an unsupervised manner?

Camera network with multiple overlapping views.

- Exploit redundant information from multiple cameras.
- Sensor data fusion **prior to detection**.

Fundamental problems:

- Robust camera and scene geometry estimation.
- Data association (among neighboring views)
- Information fusion among all views.

The context of this work

How to overcome dense crowd challenges in an unsupervised manner?

Camera network with multiple overlapping views.

- Exploit redundant information from multiple cameras.
- Sensor data fusion **prior to detection**.

Fundamental problems:

- Robust camera and scene geometry estimation.
- Data association (among neighboring views)
- Information fusion among all views.

The context of this work

How to overcome dense crowd challenges in an unsupervised manner?

Camera network with multiple overlapping views.

- Exploit redundant information from multiple cameras.
- Sensor data fusion **prior to detection**.

Fundamental problems:

- Robust camera and scene geometry estimation.
- Data association (among neighboring views)
- Information fusion among all views.

The context of this work

How to overcome dense crowd challenges in an unsupervised manner?

Camera network with multiple overlapping views.

- Exploit redundant information from multiple cameras.
- Sensor data fusion **prior to detection**.

Fundamental problems:

- Robust camera and scene geometry estimation.
- Data association (among neighboring views)
- Information fusion among all views.

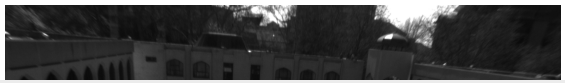
The setting

Images recorded in Regents Park Mosque, London.

- Camera registration is performed at low densities.



- Head detection is performed on the crowd leaving the mosque.

Pedestrian map computation

Pairwise MRF energy function

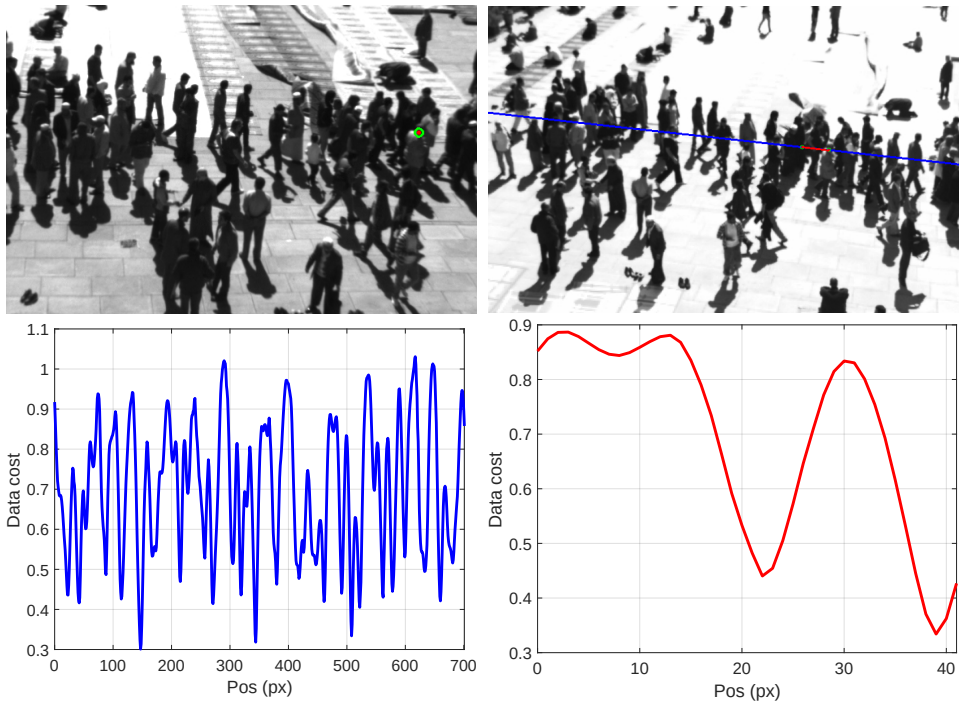
$$E(I) = \sum_{p \in \mathcal{I}} D_p(I_p) + \lambda \sum_{(p,q) \in \mathcal{N}} V_{p,q}(I_p, I_q)$$

- p pixel belonging to the image $\mathcal{I}$.
- $I \in \mathcal{L}$ labeling assigment.
- $\mathcal{N}$ set of edges of the image graph (4-connectivity).
- D_p **data** cost function.
- $V_{p,q}$ **discontinuity** cost function.
- λ regularization parameter.

Geometry of the problem

Inferring scene and camera geometry

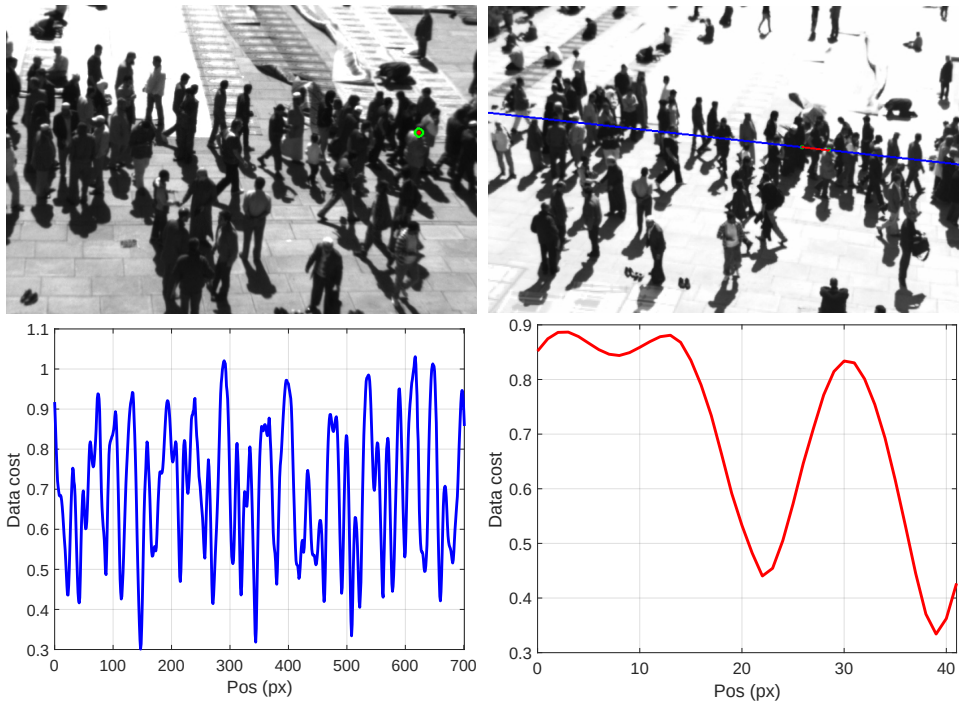
- **Objective**: Restrict the search volume of a head in the 3D space.
- Volume contained between the planes at height 1.4 and 2 meters.
- In the image space it corresponds to a **segment** on the epipolar line.
- Extremes of the segments are found by variable-height homographies projections.



Geometry of the problem

Inferring scene and camera geometry

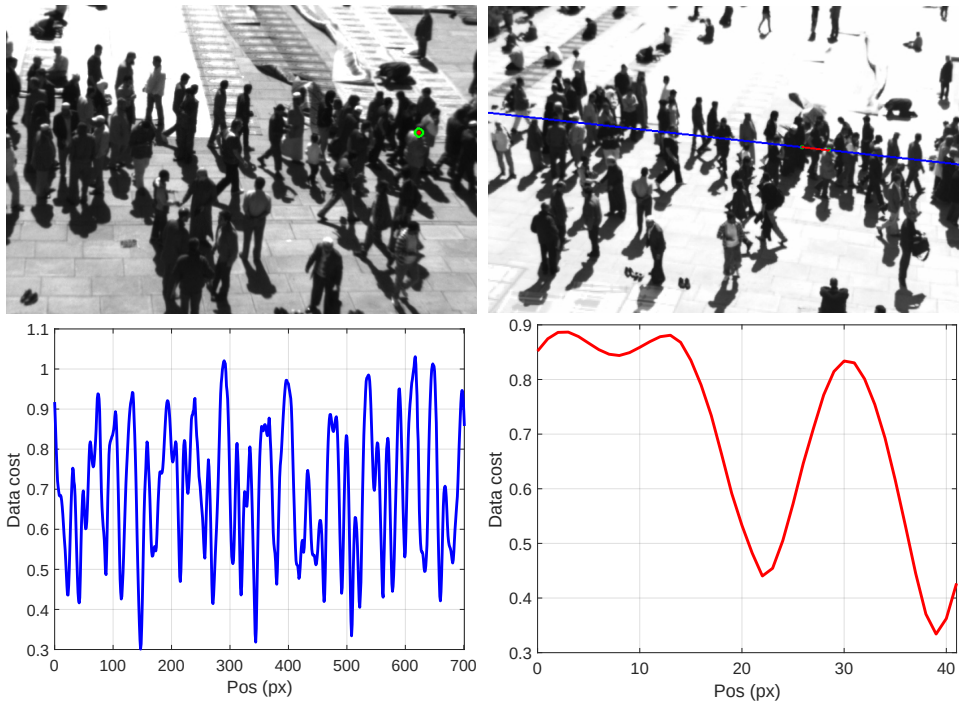
- **Objective**: Restrict the search volume of a head in the 3D space.
- Volume contained between the planes at height 1.4 and 2 meters.
- In the image space it corresponds to a **segment** on the epipolar line.
- Extremes of the segments are found by variable-height homographies projections.



Geometry of the problem

Inferring scene and camera geometry

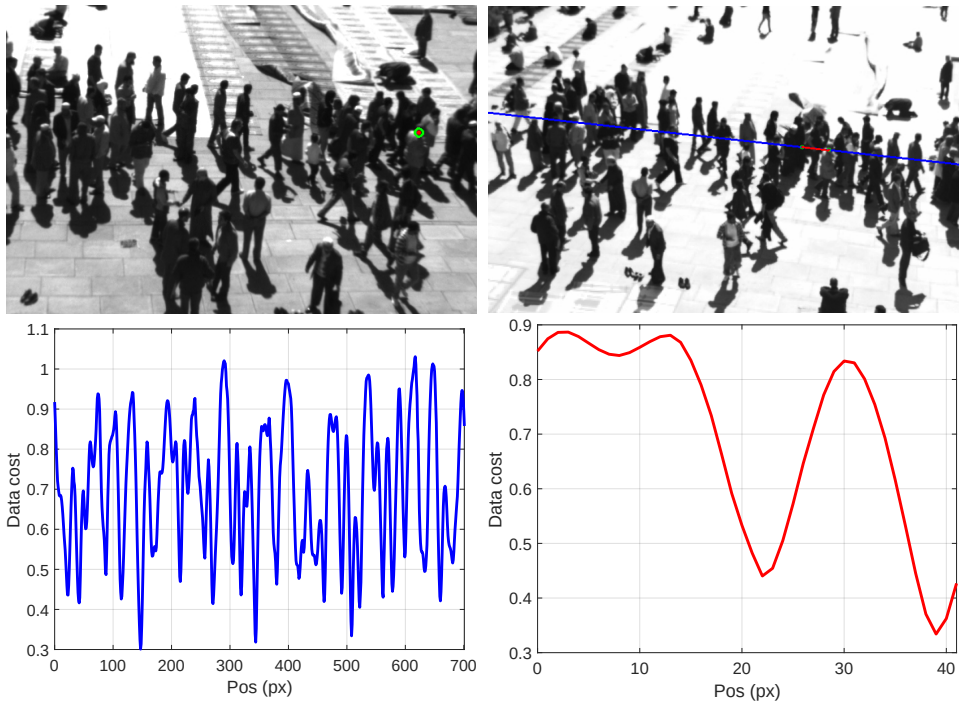
- **Objective**: Restrict the search volume of a head in the 3D space.
- Volume contained between the planes at height 1.4 and 2 meters.
- In the image space it corresponds to a **segment** on the epipolar line.
- Extremes of the segments are found by variable-height homographies projections.



Geometry of the problem

Inferring scene and camera geometry

- **Objective**: Restrict the search volume of a head in the 3D space.
- Volume contained between the planes at height 1.4 and 2 meters.
- In the image space it corresponds to a **segment** on the epipolar line.
- Extremes of the segments are found by variable-height homographies projections.



Label definition: height-based optimization

- State-of-the-art stereo matching: *depth* as label.
- This method performs a **height-based optimization**.

Advantages of height over depth

- Use of height comes naturally from the search space definition.
- Height is a ground plane related variable.
- Height allows more sophisticated discontinuity constraints than the classic constant-depth assumption.

- Discretized label set, ranging from 1.4 meters to 2 meters, with step $\Delta_h = 2.5$ centimeters.
- The label set is augmented with an *unknown label*, meaning no head is present.

Label definition: height-based optimization

- State-of-the-art stereo matching: *depth* as label.
- This method performs a **height-based optimization**.

Advantages of height over depth

- Use of height comes naturally from the search space definition.
- Height is a ground plane related variable.
- Height allows more sophisticated discontinuity constraints than the classic constant-depth assumption.

- Discretized label set, ranging from 1.4 meters to 2 meters, with step $\Delta_h = 2.5$ centimeters.
- The label set is augmented with an *unknown label*, meaning no head is present.

Label definition: height-based optimization

- State-of-the-art stereo matching: *depth* as label.
- This method performs a **height-based optimization**.

Advantages of height over depth

- Use of height comes naturally from the search space definition.
- Height is a ground plane related variable.
- Height allows more sophisticated discontinuity constraints than the classic constant-depth assumption.

- Discretized label set, ranging from 1.4 meters to 2 meters, with step $\Delta_h = 2.5$ centimeters.
- The label set is augmented with an *unknown label*, meaning no head is present.

Label definition: height-based optimization

- State-of-the-art stereo matching: *depth* as label.
- This method performs a **height-based optimization**.

Advantages of height over depth

- Use of height comes naturally from the search space definition.
- Height is a ground plane related variable.
- Height allows more sophisticated discontinuity constraints than the classic constant-depth assumption.

- Discretized label set, ranging from 1.4 meters to 2 meters, with step $\Delta_h = 2.5$ centimeters.
- The label set is augmented with an *unknown label*, meaning no head is present.

Label definition: height-based optimization

- State-of-the-art stereo matching: *depth* as label.
- This method performs a **height-based optimization**.

Advantages of height over depth

- Use of height comes naturally from the search space definition.
- Height is a ground plane related variable.
- Height allows more sophisticated discontinuity constraints than the classic constant-depth assumption.

- Discretized label set, ranging from 1.4 meters to 2 meters, with step $\Delta_h = 2.5$ centimeters.
- The label set is augmented with an *unknown label*, meaning no head is present.

Label definition: height-based optimization

- State-of-the-art stereo matching: *depth* as label.
- This method performs a **height-based optimization**.

Advantages of height over depth

- Use of height comes naturally from the search space definition.
- Height is a ground plane related variable.
- Height allows more sophisticated discontinuity constraints than the classic constant-depth assumption.

- Discretized label set, ranging from 1.4 meters to 2 meters, with step $\Delta_h = 2.5$ centimeters.
- The label set is augmented with an *unknown label*, meaning no head is present.

Label definition: height-based optimization

- State-of-the-art stereo matching: *depth* as label.
- This method performs a **height-based optimization**.

Advantages of height over depth

- Use of height comes naturally from the search space definition.
- Height is a ground plane related variable.
- Height allows more sophisticated discontinuity constraints than the classic constant-depth assumption.

- Discretized label set, ranging from 1.4 meters to 2 meters, with step $\Delta_h = 2.5$ centimeters.
- The label set is augmented with an *unknown label*, meaning no head is present.

Relative pose estimation

Estimation of the **fundamental matrix** $F_{i,j}$ between any pair of cameras using [Pellicanò et al., 2016]:

- Iterative estimation of $F_{i,j}$ from a synchronized video stream.
- Moving pedestrians add new information at each frame.
- **Output**: $F_{i,j}$, $S_{i,j}$ (inliers matches set).

Bundle adjustment is used to enforce a global scale.

The only metric information needed is the **laser distance** between cameras

`https://github.com/MOHICANS-project/fundvid`

Relative pose estimation

Estimation of the **fundamental matrix** $F_{i,j}$ between any pair of cameras using [Pellicanò et al., 2016]:

- Iterative estimation of $F_{i,j}$ from a synchronized video stream.
- Moving pedestrians add new information at each frame.
- **Output**: $F_{i,j}$, $S_{i,j}$ (inliers matches set).

Bundle adjustment is used to enforce a global scale.

The only metric information needed is the **laser distance** between cameras

`https://github.com/MOHICANS-project/fundvid`

Relative pose estimation

Estimation of the **fundamental matrix** $F_{i,j}$ between any pair of cameras using [Pellicanò et al., 2016]:

- Iterative estimation of $F_{i,j}$ from a synchronized video stream.
- Moving pedestrians add new information at each frame.
- **Output**: $F_{i,j}$, $S_{i,j}$ (inliers matches set).

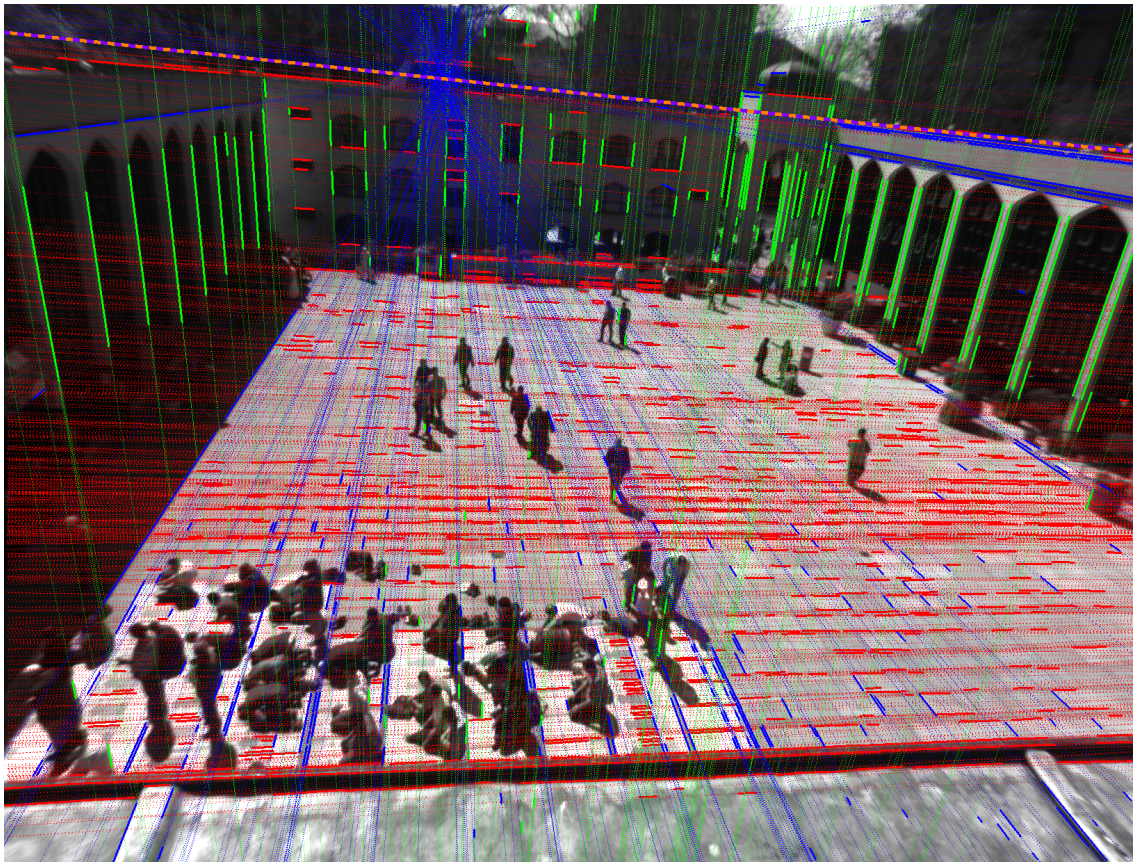
Bundle adjustment is used to enforce a global scale.

The only metric information needed is the **laser distance** between cameras

`https://github.com/MOHICANS-project/fundvid`

Vanishing points

Automatic vanishing points extraction from image segments [Lezama et al., 2014].



The position of the ground plane is also estimated automatically based on the Manhattan-world assumption.

Data cost function

- We need a local region descriptor suited for a wide-baseline scenario.
- **DAISY descriptor**:
  - Good computational efficiency.
  - Robust to perspective and illumination changes.
- Data cost expressed as a **dissimilarity function** between the DAISY descriptors.
- Given N cameras, $N-1$ dissimilarity functions will be produced.
- **Data cost fusion**:
  - If $N = 3$, simple cost averaging will be effective.
  - If $N > 3$, outlier robust cost merging is needed [Vogiatzis et al., 2007].
- *Unknown label* is assigned with a constant cost $K_{d,u}$.

Data cost function

- We need a local region descriptor suited for a wide-baseline scenario.
- **DAISY descriptor**:
  - Good computational efficiency.
  - Robust to perspective and illumination changes.
- Data cost expressed as a **dissimilarity function** between the DAISY descriptors.
- Given N cameras, $N-1$ dissimilarity functions will be produced.
- **Data cost fusion**:
  - If $N = 3$, simple cost averaging will be effective.
  - If $N > 3$, outlier robust cost merging is needed [Vogiatzis et al., 2007].
- *Unknown label* is assigned with a constant cost $K_{d,u}$.

Data cost function

- We need a local region descriptor suited for a wide-baseline scenario.
- **DAISY descriptor**:
  - Good computational efficiency.
  - Robust to perspective and illumination changes.
- Data cost expressed as a **dissimilarity function** between the DAISY descriptors.
- Given N cameras, $N-1$ dissimilarity functions will be produced.
- **Data cost fusion**:
  - If $N = 3$, simple cost averaging will be effective.
  - If $N > 3$, outlier robust cost merging is needed [Vogiatzis et al., 2007].
- *Unknown label* is assigned with a constant cost $K_{d,u}$.

Data cost function

- We need a local region descriptor suited for a wide-baseline scenario.
- **DAISY descriptor**:
  - Good computational efficiency.
  - Robust to perspective and illumination changes.
- Data cost expressed as a **dissimilarity function** between the DAISY descriptors.
- Given N cameras, $N-1$ dissimilarity functions will be produced.
- **Data cost fusion**:
  - If $N = 3$, simple cost averaging will be effective.
  - If $N > 3$, outlier robust cost merging is needed [Vogiatzis et al., 2007].
- *Unknown label* is assigned with a constant cost $K_{d,u}$.

Data cost function

- We need a local region descriptor suited for a wide-baseline scenario.
- **DAISY descriptor**:
  - Good computational efficiency.
  - Robust to perspective and illumination changes.
- Data cost expressed as a **dissimilarity function** between the DAISY descriptors.
- Given N cameras, $N-1$ dissimilarity functions will be produced.
- **Data cost fusion**:
  - If $N = 3$, simple cost averaging will be effective.
  - If $N > 3$, outlier robust cost merging is needed [Vogiatzis et al., 2007].
- *Unknown label* is assigned with a constant cost $K_{d,u}$.

Data cost function

- We need a local region descriptor suited for a wide-baseline scenario.
- **DAISY descriptor**:
  - Good computational efficiency.
  - Robust to perspective and illumination changes.
- Data cost expressed as a **dissimilarity function** between the DAISY descriptors.
- Given N cameras, $N-1$ dissimilarity functions will be produced.
- **Data cost fusion**:
  - If $N = 3$, simple cost averaging will be effective.
  - If $N > 3$, outlier robust cost merging is needed [Vogiatzis et al., 2007].
- *Unknown label* is assigned with a constant cost $K_{d,u}$.

Discontinuity cost function

- We model the head surface as a planar patch which is **perpendicular to the ground plane**.
- At any pixel location p, the direction of maximum height variation is on the r_p line connecting p to the vertical vanishing point.
- Assuming local planarity, the height varies with a constant gradient along r_p.
- The variation gradient $|\nabla_p|$ can be estimated **from the geometry** and is dependent on the pixel position.
  - Height variation per pixel gets larger in locations further from the reference camera.

Discontinuity cost function

- We model the head surface as a planar patch which is **perpendicular to the ground plane**.
- At any pixel location p, the direction of maximum height variation is on the r_p line connecting p to the vertical vanishing point.
- Assuming local planarity, the height varies with a constant gradient along r_p.
- The variation gradient $|\nabla_p|$ can be estimated **from the geometry** and is dependent on the pixel position.
  - Height variation per pixel gets larger in locations further from the reference camera.

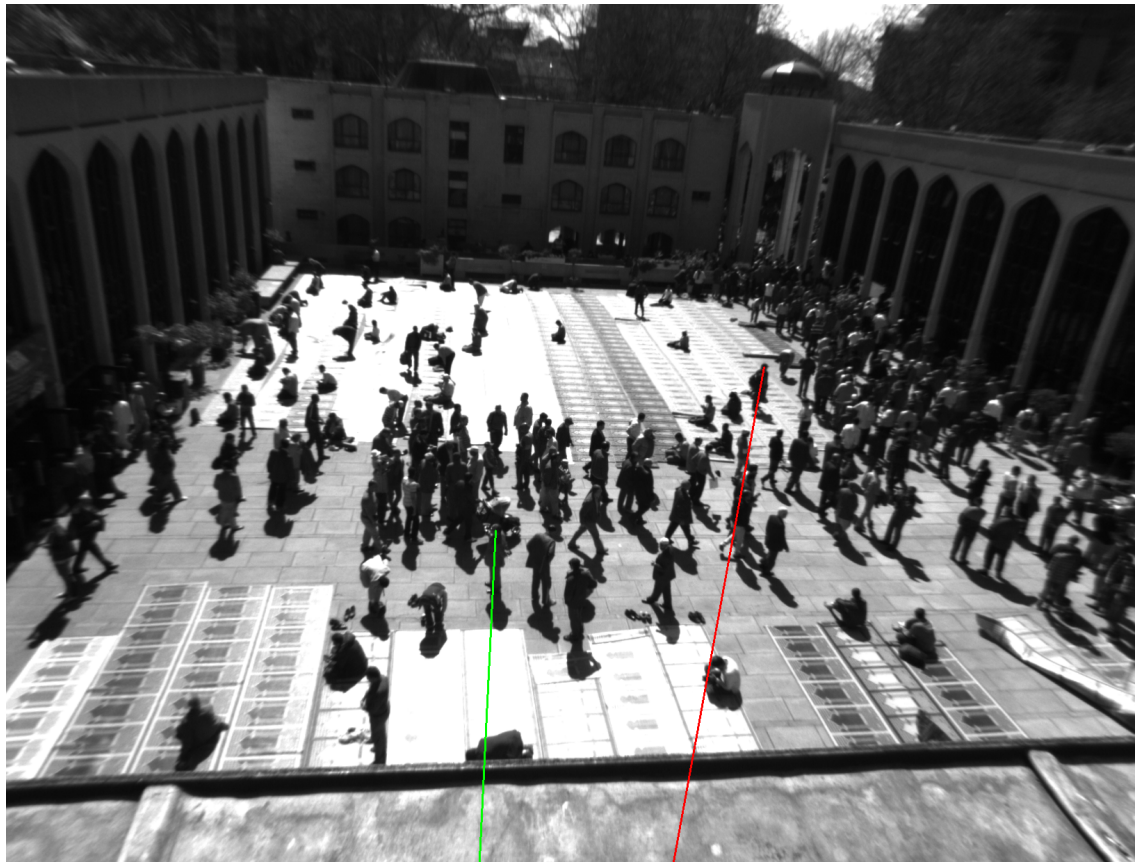
Discontinuity cost function

- We model the head surface as a planar patch which is **perpendicular to the ground plane**.
- At any pixel location p, the direction of maximum height variation is on the r_p line connecting p to the vertical vanishing point.
- Assuming local planarity, the height varies with a constant gradient along r_p.
- The variation gradient $|\nabla_p|$ can be estimated **from the geometry** and is dependent on the pixel position.
  - Height variation per pixel gets larger in locations further from the reference camera.

Discontinuity cost function

- We model the head surface as a planar patch which is **perpendicular to the ground plane**.
- At any pixel location p, the direction of maximum height variation is on the r_p line connecting p to the vertical vanishing point.
- Assuming local planarity, the height varies with a constant gradient along r_p.
- The variation gradient $|\nabla_p|$ can be estimated **from the geometry** and is dependent on the pixel position.
  - Height variation per pixel gets larger in locations further from the reference camera.

Discontinuity cost function



- Red: Head radius $\approx 4px$; $|\nabla_p| = 2.5cm$.
- Green: Head radius $\approx 6px$; $|\nabla_p| = 1.6cm$.

Discontinuity cost function

Given the point p and the neighbor point q:

- Compute the orthogonal projection $q^{\perp}$ of q on the line r_p
- Expected height variation ξ_h: $|\nabla_p| d(p, q^{\perp})$.
- Distance function:

$$D_{p,q}(I_p, I_q) = \left| I_p - I_q - s_p \xi_h \right|$$

where s_p is the expected sign of the variation.

Discontinuity cost function

Given the point p and the neighbor point q:

- Compute the orthogonal projection $q^{\perp}$ of q on the line r_p
- Expected height variation ξ_h: $|\nabla_p| d(p, q^{\perp})$.
- Distance function:

$$D_{p,q}(I_p, I_q) = \left| I_p - I_q - s_p \xi_h \right|$$

where s_p is the expected sign of the variation.

Discontinuity cost function

Given the point p and the neighbor point q:

- Compute the orthogonal projection $q^{\perp}$ of q on the line r_p
- Expected height variation ξ_h: $|\nabla_p| d(p, q^{\perp})$.
- Distance function:

$$D_{p,q}(I_p, I_q) = \left| I_p - I_q - s_p \xi_h \right|$$

where s_p is the expected sign of the variation.

Discontinuity cost function

Given the point p and the neighbor point q:

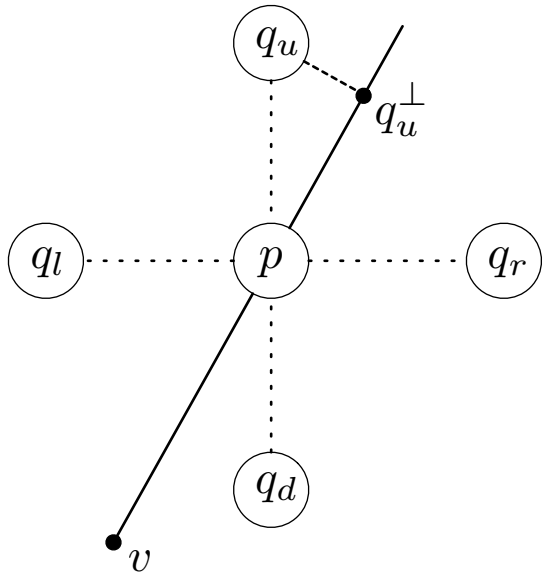
- Compute the orthogonal projection $q^{\perp}$ of q on the line r_p
- Expected height variation ξ_h: $|\nabla_p| d(p, q^{\perp})$.
- Distance function:

$$D_{p,q}(l_p, l_q) = \left| l_p - l_q - s_p \xi_h \right|$$

where s_p is the expected sign of the variation.

When r_p is almost completely vertical:

- $\xi_h(q_l) = \xi_h(q_r) \approx 0$
- $\xi_h(q_u) = \xi_h(q_d) \approx |\nabla_p|$



Discontinuity cost function

Truncated distance function

$$\hat{V}_{p,q}(l_p, l_q) = \min\left[\frac{\max\left(D_{p,q}(l_p, l_q), D_{q,p}(l_q, l_p)\right)}{\Delta_h}, K\right]$$

Final discontinuity cost function

$$V_{p,q}(l_p, l_q) = \begin{cases} \hat{V}_{p,q}(l_p, l_q) & l_p \neq unknown \wedge l_q \neq unknown \\ K & l_p = unknown \wedge l_q \neq unknown \\ K & l_p \neq unknown \wedge l_q = unknown \\ 0 & l_p = unknown \wedge l_q = unknown \end{cases}$$

Optimization algorithm

- $V_{p,q}(l_p, l_q)$ is not **submodular**: not suitable for Graph-Cuts.
- Results provided with Loopy Belief Propagation.

Discontinuity cost function

Truncated distance function

$$\hat{V}_{p,q}(l_p, l_q) = \min\left[\frac{\max\left(D_{p,q}(l_p, l_q), D_{q,p}(l_q, l_p)\right)}{\Delta_h}, K\right]$$

Final discontinuity cost function

$$V_{p,q}(l_p, l_q) = \begin{cases} \hat{V}_{p,q}(l_p, l_q) & l_p \neq unknown \wedge l_q \neq unknown \\ K & l_p = unknown \wedge l_q \neq unknown \\ K & l_p \neq unknown \wedge l_q = unknown \\ 0 & l_p = unknown \wedge l_q = unknown \end{cases}$$

Optimization algorithm

- $V_{p,q}(l_p, l_q)$ is not **submodular**: not suitable for Graph-Cuts.
- Results provided with Loopy Belief Propagation.

Height map example

Color ranges from red (1.4 meters) to green (2 meters).

Temporal filtering

Objective: Output only motion consistent fragments of the height map.

Tracklets building stage

1. Project the height map on the reference plane (geometry+PCA)
2. Find height local maxima in the projected data.
3. Cluster points around closest maxima and compute a centroid per cluster.
4. Tracklets creation and extension. Associate centroid to an existing tracklet if:
   - It is located closer than $\theta_h = 20cm$ from the predicted location of the tracklet.
   - Its height closer than $\theta_h = 10cm$ to the tracklet average height.
5. Any tracklet which is not extended is terminated.

Temporal filtering

Objective: Output only motion consistent fragments of the height map.

Tracklets building stage

1. Project the height map on the reference plane (geometry+PCA)
2. Find height local maxima in the projected data.
3. Cluster points around closest maxima and compute a centroid per cluster.
4. Tracklets creation and extension. Associate centroid to an existing tracklet if:
   - It is located closer than $\theta_h = 20cm$ from the predicted location of the tracklet.
   - Its height closer than $\theta_h = 10cm$ to the tracklet average height.
5. Any tracklet which is not extended is terminated.

Temporal filtering

Objective: Output only motion consistent fragments of the height map.

Tracklets building stage

1. Project the height map on the reference plane (geometry+PCA)
2. Find height local maxima in the projected data.
3. Cluster points around closest maxima and compute a centroid per cluster.
4. Tracklets creation and extension. Associate centroid to an existing tracklet if:
   - It is located closer than $\theta_h = 20cm$ from the predicted location of the tracklet.
   - Its height closer than $\theta_h = 10cm$ to the tracklet average height.
5. Any tracklet which is not extended is terminated.

Temporal filtering

Objective: Output only motion consistent fragments of the height map.

Tracklets building stage

1. Project the height map on the reference plane (geometry+PCA)
2. Find height local maxima in the projected data.
3. Cluster points around closest maxima and compute a centroid per cluster.
4. Tracklets creation and extension. Associate centroid to an existing tracklet if:
   - It is located closer than $\theta_h = 20cm$ from the predicted location of the tracklet.
   - Its height closer than $\theta_h = 10cm$ to the tracklet average height.
5. Any tracklet which is not extended is terminated.

Temporal filtering

Objective: Output only motion consistent fragments of the height map.

Tracklets building stage

1. Project the height map on the reference plane (geometry+PCA)
2. Find height local maxima in the projected data.
3. Cluster points around closest maxima and compute a centroid per cluster.
4. Tracklets creation and extension. Associate centroid to an existing tracklet if:
   - It is located closer than $\theta_h = 20cm$ from the predicted location of the tracklet.
   - Its height closer than $\theta_h = 10cm$ to the tracklet average height.
5. Any tracklet which is not extended is terminated.

Temporal filtering

Objective: Output only motion consistent fragments of the height map.

Tracklets building stage

1. Project the height map on the reference plane (geometry+PCA)
2. Find height local maxima in the projected data.
3. Cluster points around closest maxima and compute a centroid per cluster.
4. Tracklets creation and extension. Associate centroid to an existing tracklet if:
   - It is located closer than $\theta_h = 20cm$ from the predicted location of the tracklet.
   - Its height closer than $\theta_h = 10cm$ to the tracklet average height.
5. Any tracklet which is not extended is terminated.

Temporal filtering

Objective: Output only motion consistent fragments of the height map.

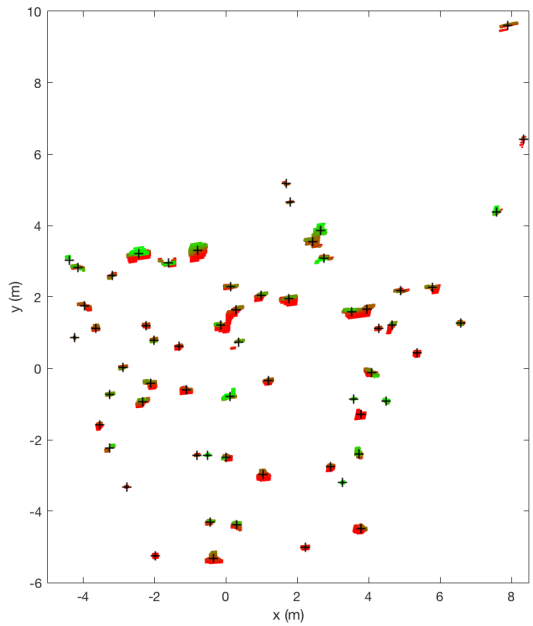
Tracklets building stage

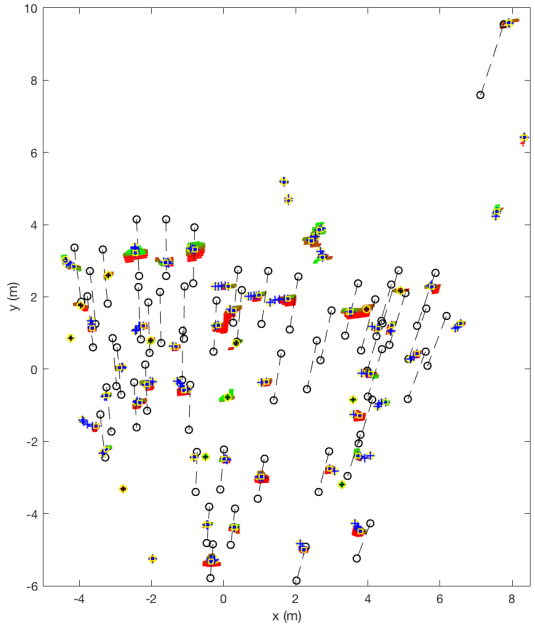
1. Project the height map on the reference plane (geometry+PCA)
2. Find height local maxima in the projected data.
3. Cluster points around closest maxima and compute a centroid per cluster.
4. Tracklets creation and extension. Associate centroid to an existing tracklet if:
   - It is located closer than $\theta_h = 20cm$ from the predicted location of the tracklet.
   - Its height closer than $\theta_h = 10cm$ to the tracklet average height.
5. Any tracklet which is not extended is terminated.

Temporal filtering

Filtering stage

- Select a tracklet thresholding paramater θ_l ($\theta_l = 0$, no filtering).
- Discard all the fragments associated to tracklets of length less or equal to θ_l.

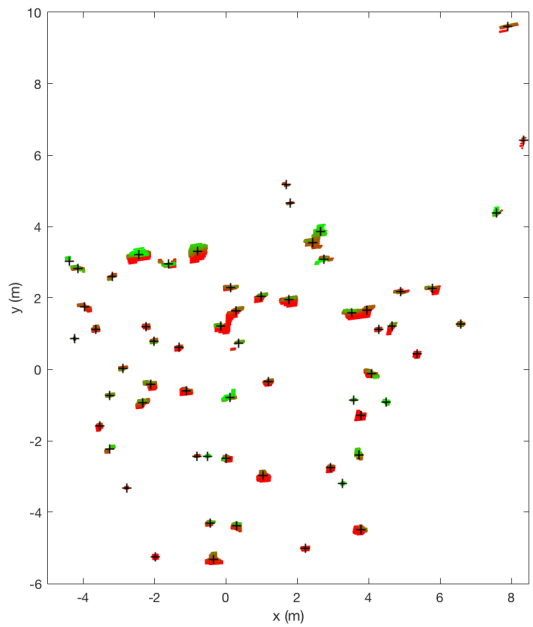


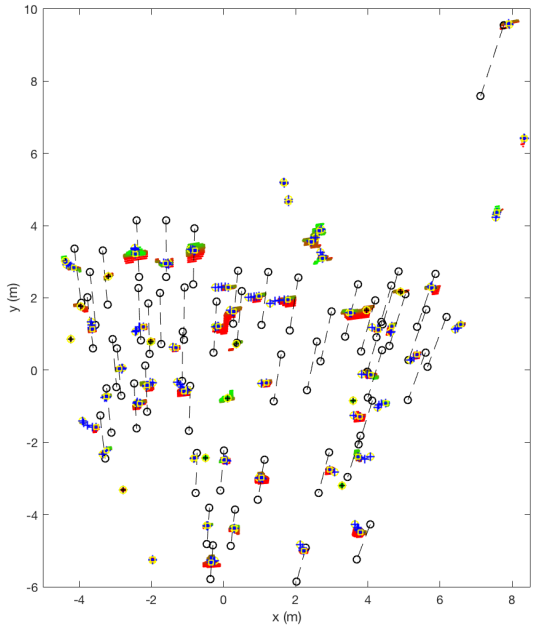


Temporal filtering

Filtering stage

- Select a tracklet thresholding paramater θ_l ($\theta_l = 0$, no filtering).
- Discard all the fragments associated to tracklets of length less or equal to θ_l.





Dataset

- Three synchronized video streams per sequence.
- *Dense*: 18567 pedestrians annotations, clustered towards exit areas.
- *Sparse*: 2969 pedestrians annotations.

- Quantitative evaluation performed with respect to manually annotated tracks.

Dense sequence results

θ_l value	Recall	Precision
0	85.60%	65.67%
1	80.20%	72.35%
2	75.10%	77.18%
3	70.26%	79.81%

Table: Recall and precision on the *Dense* sequence depending on the tracklet threshold θ_l.

- $\lambda = 0.07$
- High recall, even with strong occlusion and no appearance related term.
- Temporal filtering gives a strong benefit on the precision metric.

Sparse sequence results

λ value	Recall	Precision
0.08	89.89%	42.65%
0.15	74.37%	66.78%
0.20	62.23%	82.82%

Table: Recall and precision on the *Sparse* sequence depending on the regularization parameter λ. Here $\theta_{\mathrm{l}} = 1$.

- λ has a strong impact on the precision-recall results.
- At low densities ground related **persistent phantoms** can appear, being detrimental for precision.
- Absence of an appearance term makes impossible to differentiate phantoms from pedestrians.

Concluding remarks

Conclusions

- A method for locating pedestrian heads in cluttered outdoor scenes.
- **Low level** data fusion.
- Modeling the problem is very important : the right space for the hidden variable, the right optimizer
- Cost may be a significant issue, but hardware can help : 1000x acceleration on GPU for this problem

References I

Lezama, J., Grompone von Gioi, R., Randall, G., and Morel, J.-M. (2014).
Finding vanishing points via point alignments in image primal and dual domains.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 509–515.

Pellicanò, N., Aldea, E., and Le Hégarat-Mascle, S. (2016).
Robust wide baseline pose estimation from video.
In *Proceedings of the International Conference on Pattern Recognition (ICPR)*.

Vogiatzis, G., Esteban, C. H., Torr, P. H., and Cipolla, R. (2007).
Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(12):2241–2246.