Multi-camera multi-target Tracking

François Fleuret

October 1, 2012

Joint work with H. Benshitrit, J. Berclaz,
R. Lengagne, E. Türetken, and P. Fua

Introduction

What is tracking ?

Given video streams from several cameras, multi-person tracking consists of estimating where individuals are located on the ground plane in each frame.

Original sequence [video]

Introduction
Applications

Detecting, localizing, and tracking multiple pedestrians is the core component of multiple applications:

- Preprocessing for behavioral studies.
- Hazardous area control.
- People counting.
- Activity surveillance.

Pedestrian Detection
Monocular

✎ M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Conference on Computer Vision and Pattern Recognition*, 2009

✎ Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *European Conference on Computer Vision*, 2010

✎ O. Barinova, V. Lempitsky, and P. Kohli. On the detection of multiple object instances using Hough transforms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010

Multi-camera

✎ A. Ess, B. Leibe, and L. Van Gool. Depth and appearance for mobile scene analysis. In *International Conference on Computer Vision*, 2007

✎ S. M. Khan and M. Shah. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):505–519, 2009

✎ S. Sternig, T. Mauthner, A. Irschara, P. M. Roth, and H. Bischof. Multi-camera multi-object tracking by robust hough-based homography projections. In *ICCV Workshop on Visual Surveillance*, pages 1689–1696, 2011

✎ A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and Hu W. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Conference on Computer Vision and Pattern Recognition*, 2006

✎ M. Isard and J. MaxCormick. Bramble: A bayesian multiple-blob tracker. In *International Conference on Computer Vision*, 2001

✎ K. Smith, D. Gatica-Perez, and J-M. Odobez. Using particles to track varying numbers of interacting people. In *Conference on Computer Vision and Pattern Recognition*, 2005

✎ M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. J. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *International Conference on Computer Vision*, 2009

Adaptive Graph

✎ H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. In *Conference on Computer Vision and Pattern Recognition*, 2007

✎ L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Conference on Computer Vision and Pattern Recognition*, 2008

✎ B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *Conference on Computer Vision and Pattern Recognition*, 2012

Pre-defined Graph

✎ F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):267–282, 2008

✎ A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In *European Conference on Computer Vision*, 2010

✎ A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst. Sparsity driven people localization with a heterogeneous network of cameras. *Journal of Mathematical Imaging and Vision*, 41(1-2):39–58, 2011

PROBABILISTIC OCCUPANCY MAP (POM)

OBJECTIVE
SUMMARY

Given the images from multiple cameras taken synchronously, we want to compute a probability of occupancy for every location on the ground.

We use a real-time background subtraction to pre-process the image. It estimates a background image and which parts of the current frame differ from it.

Segmented sequence [video]

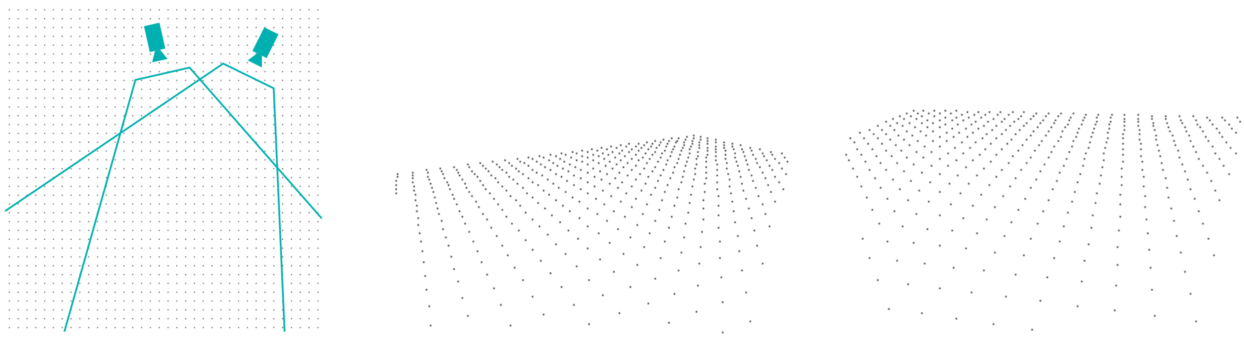
OBJECTIVE
DESIRED PROPERTIES

Our estimation of the occupancy should

- Work on isolated temporal frames.
- Combine several views.
- Provide probabilistic estimates.
- Handle noisy background subtraction.
- Take into account silhouette sizes for distance.
- Handle occlusions consistently.

MODELING
GROUND DISCRETIZATION

The ground plane is regularly divided into N locations. Here $N = 1332$, corresponding to a 20cm accuracy.



Top-view From camera 0 From camera 1

Modeling
Camera calibration

The calibration of the C cameras is provided by a collection of rectangles

$$\mathcal{A}_1^c, \dots, \mathcal{A}_N^c, \quad c = 1, \dots, C$$

of human size.

Collection of "human shapes" [video]

Modeling
Notation

We work on one frame and introduce the following random variables

$\mathbf{X} = (X^1, \dots, X^N)$ on $\{0, 1\}^N$ the grid occupancy (unknown).

$\mathbf{B} = (B^1, \dots, B^C)$ on $\left(\{0, 1\}^{w \times h}\right)^C$ the images (known).

The quantity of interest is $P(\mathbf{X} \mid \mathbf{B})$.

MODELING
INDEPENDENCE ASSUMPTIONS

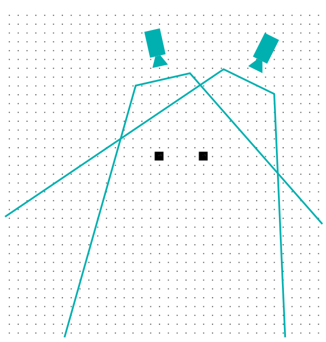
We make the following independence assumptions

$$P(X^1, \ldots, X^N) = \prod_n P(X^n)$$

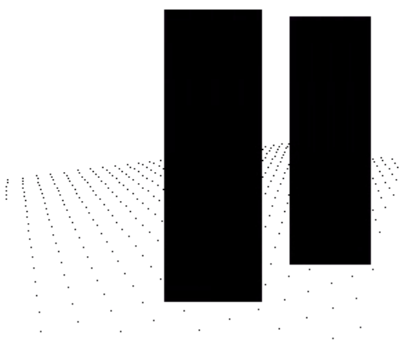
$$P(B^1, \ldots, B^C \mid X^1, \ldots, X^N) = \prod_c P(B^c \mid X^1, \ldots, X^N)$$

MODELING
CRUDE SYNTHESIS

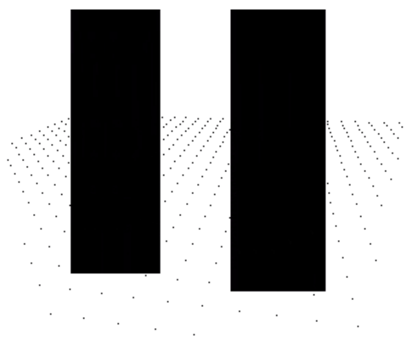
Let A^c denote a synthetic picture obtained by putting rectangles $\mathcal{A}_n^c$ where $X^n = 1$.



$X^1, \ldots, X^N$



A^0



A^1

Modeling
Modeling $P(\mathbf{B} \mid \mathbf{X})$

We introduce a pseudo-distance Ψ between images

$$\Psi(B^c, A^c) = \frac{1}{\sigma} \frac{\|B^c - A^c\|^2}{\|A^c\|^2}$$

and use the following model

$$\begin{aligned} P(\mathbf{B} \mid \mathbf{X}) &= \prod_c P(B^c \mid \mathbf{X}) \\ &= \frac{1}{Z} \prod_c e^{-\Psi(B^c, A^c)} \end{aligned}$$

Estimation of the q^n
Approximation of $P(\mathbf{X} \mid \mathbf{B})$

Let Q denote a product law approximating the true posterior density $P(\,\cdot \mid \mathbf{B})$

$$Q(\mathbf{X} = \mathbf{x}) = \prod_n Q(X^n = x_n)$$

We want to compute the marginals q^n of Q

$$q^n = Q(X^n = 1)$$

.../...

ESTIMATION OF THE q^n
APPROXIMATION OF $P(\mathbf{X} \mid \mathbf{B})$ (CONT.)

We are looking for q^ns so that Q is close to $P(\,\cdot \mid \mathbf{B})$ with respect to the Kullback-Leibler divergence. Solving

$$\frac{\partial}{\partial q^n} KL(Q,\, P(\,\cdot \mid \mathbf{B})) \;=\; 0$$

leads to, with E_Q denoting $E_{\mathbf{X} \sim Q}$

$$q^n = \frac{1}{1 + \exp\left(\lambda - \underbrace{E_Q(\log P(\mathbf{B}|\mathbf{X})|X^n = 1) + E_Q(\log P(\mathbf{B}|\mathbf{X})|X^n = 0)}_{\text{Function of } q^1,\ldots,q^{n-1},q^{n+1},\ldots,q^N}\right)}$$

ESTIMATION OF THE q^n
PROOF

$$\begin{aligned}
&\frac{\partial}{\partial q^n} KL(Q,\, P(\,\cdot \mid \mathbf{B})) \\
&\quad = \frac{\partial}{\partial q^n} E_Q\left[\log \frac{Q(\mathbf{X})}{P(\mathbf{X} \mid \mathbf{B})}\right] \\
&\quad = \frac{\partial}{\partial q^n} E_Q\left[\log \frac{Q(\mathbf{X})}{P(\mathbf{X})} \frac{P(\mathbf{B})}{P(\mathbf{B} \mid \mathbf{X})}\right] \\
&\quad = \frac{\partial}{\partial q^n} E_Q\left[\sum_l \log \frac{Q(X^l)}{P(X^l)} + \log P(\mathbf{B}) - \log P(\mathbf{B} \mid \mathbf{X})\right] \\
&\quad = \frac{\partial}{\partial q^n} E_Q\left[\log \frac{Q(X^n)}{P(X^n)} - \log P(\mathbf{B} \mid \mathbf{X})\right]
\end{aligned}$$

.../...

Estimation of the q^n

Proof (cont.)

$$= \frac{\partial}{\partial q^n} E_Q\left[\log \frac{Q(X^n)}{P(X^n)} - \log P(\mathbf{B} \mid \mathbf{X})\right]$$

$$= \frac{\partial}{\partial q^n}\, q^n \left(\log \frac{q^n}{\epsilon} - E_Q\left[\log P(\mathbf{B} \mid \mathbf{X}) \mid X^n = 1\right]\right)$$

$$+ \frac{\partial}{\partial q^n}\,(1-q^n)\left(\log \frac{1-q^n}{1-\epsilon} - E_Q\left[\log P(\mathbf{B} \mid \mathbf{X}) \mid X^n = 0\right]\right)$$

$$= \log \frac{q^n}{\epsilon} + 1 - E_Q\left[\log P(\mathbf{B} \mid \mathbf{X}) \mid X^n = 1\right]$$

$$- \log \frac{1-q^n}{1-\epsilon} - 1 + E_Q\left[\log P(\mathbf{B} \mid \mathbf{X}) \mid X^n = 0\right]$$

$$= -\log\left(\frac{1}{q^n} - 1\right) + \log \frac{1-\epsilon}{\epsilon}$$

$$- E_Q\left[\log P(\mathbf{B} \mid \mathbf{X}) \mid X^n = 1\right] + E_Q\left[\log P(\mathbf{B} \mid \mathbf{X}) \mid X^n = 0\right]$$

$\square$

Estimation of the q^n

Local linearization of Ψ

Under our model

$$P(\mathbf{B} \mid \mathbf{X}) = \frac{1}{Z} \prod_c e^{-\Psi(B^c, A^c)},$$

we have

$$E_Q\left(\log P(\mathbf{B} \mid \mathbf{X}) \mid X^n = \xi\right) \;=\; -\sum_c E_Q(\Psi(B^c, A^c) \mid X^n = \xi) + cst$$

Computing $E_Q(\Psi(B^c, A^c) \mid X^n = \xi)$ is intractable. However, since A^c is highly picked around B^c for $\mathbf{X} \sim Q$, we use $\forall n, \forall \xi$

$$E_Q(\Psi(B^c, A^c) \mid X^n = \xi) \;\simeq\; \Psi(B^c,\, E_Q(A^c \mid X^n = \xi))$$

ESTIMATION OF THE q^n
AVERAGE IMAGE

The synthetic image A^c is a function of $\mathbf{X}$, thus $E_Q(A^c)$ is well defined, and a function of the q^n:

$$\begin{aligned} E_Q(A^c(x,y)) &= Q(A^c(x,y)=1) \\ &= 1 - \prod_{n:\mathcal{A}_n^c(x,y)=1} (1-q^n) \end{aligned}$$

Averaging images [video]

ESTIMATION OF THE q^n
SUMMARY

Finally, our algorithm solves, $\forall n$

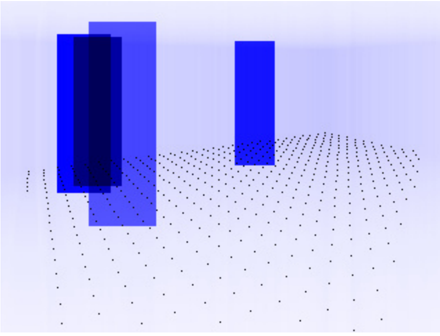
$$q^n = \frac{1}{1 + \exp\left(\lambda + \sum_c \Psi(B^c,\, E_Q(A^c|X^n\!=\!1)) - \Psi(B^c, E_Q(A^c|X^n\!=\!0))\right)}$$

Where Ψ is the pseudo-distance and λ is the prior log-ratio.

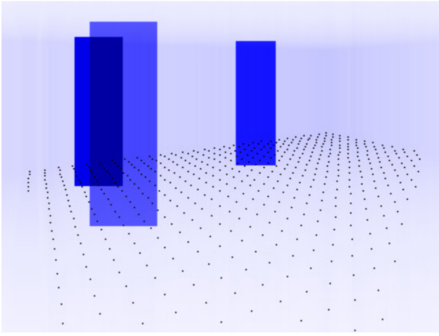
ESTIMATION OF THE q^n

INTUITIVELY

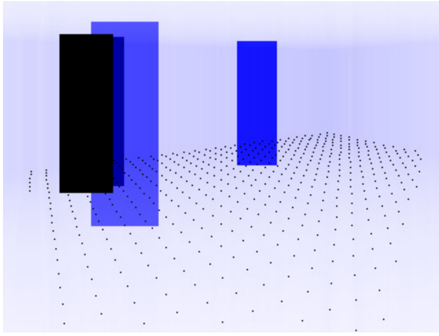
The algorithm updates q^n according to the fit between the $E_Q(A^c \mid X^n = \xi)$ and the real images B^c, with the current estimates of the q^ls for Q



$E_Q(A^c)$



$E_Q(A^c \mid X^n = 0)$



$E_Q(A^c \mid X^n = 1)$

ESTIMATION OF THE q^n

ALGORTIHM

Algorithm 1 Probabilistic Occupancy Map

for $n = 1, \dots, N$ **do**
  $q_n^0 \leftarrow \epsilon$
end for
repeat
  for $n = 1, \dots, N$ **do**
    for $c = 1, \dots, C$ **do**
      $\overline{A}_0^c \leftarrow E_{Q^k}(A^c | X^n = 0)$
      $\overline{A}_1^c \leftarrow E_{Q^k}(A^c | X^n = 1)$
    end for
    $q_n^{k+1} \leftarrow \frac{1}{1 + \exp\left(\lambda + \sum_c \Psi(B^c, \overline{A}_1^c) - \Psi(B^c, \overline{A}_0^c)\right)}$
  end for
until Convergence

RESULTS
CONVERGENCE

The algorithm chooses the q^ns so that the average synthetic images match the result of the background subtraction.

Two cameras (106) [video]

Two cameras (112) [video]

Two cameras (156) [video]

Two cameras (175) [video]

Four cameras (1563) [video]

Four cameras (3258) [video]

Four cameras, kids [video]

RESULTS
FULL SEQUENCES

This procedure can then be applied on every single frame of a sequence:

Two person sequence [video]

EPFL Exterior [video]

FIBA match [video]

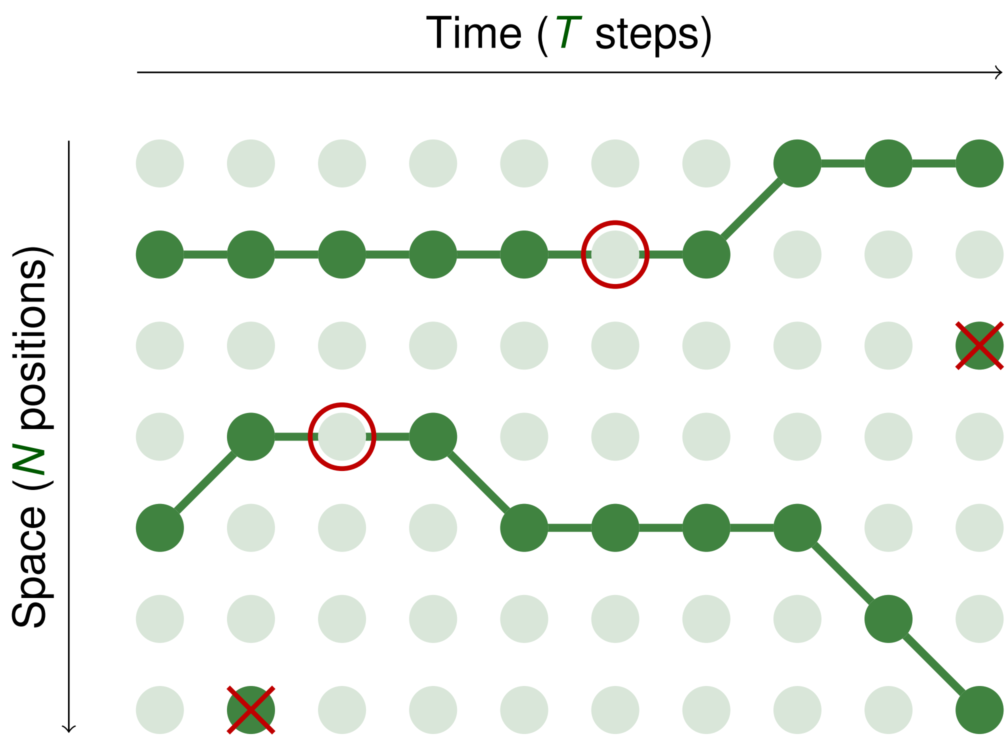
Since $E_Q(A^c)$ and $E_Q(A^c \,|\, X^n = \xi)$ differ only in $\mathcal{A}_n^c$, the computation of $\Psi(B^c, E_Q(A^c \,|\, X^n = \xi))$ can be done at constant time with integral images.

With $N = 1332$, two cameras and one hundred iterations to converge, it takes 150ms per time frame on a 2.4Ghz Pentium.

Real speed [video]

MULTI-TRACKED PATHS (MTP)

GRAPH-BASED TRACKING
TIME CONSISTENCY ALLOWS TO FILTER MISTAKES



GRAPH-BASED TRACKING
ABSTRACTING A BIT OUR GOAL

If $\mathcal{M} \subset \{0,1\}^{N \times T}$ is the set of Binary maps corresponding to physically possible presences of people, we are looking for

$$
\begin{aligned}
& \underset{(x_1^1,\dots,x_T^N)\in\mathcal{M}}{\arg\max} \quad \log P(\forall n, t,\ X_t^n = x_t^n \mid B_1^1, \dots, B_T^C) \\
= \ & \underset{(x_1^1,\dots,x_T^N)\in\mathcal{M}}{\arg\max} \quad \sum_{t,n} \log P(X_t^n = x_t^n \mid B_t^1, \dots, B_t^C) \\
= \ & \underset{(x_1^1,\dots,x_T^N)\in\mathcal{M}}{\arg\max} \quad \sum_{t,n} \log \frac{P(X_t^n = 1 \mid B_t^1, \dots, B_t^C)}{P(X_t^n = 0 \mid B_t^1, \dots, B_t^C)} x_t^n \\
= \ & \underset{(x_1^1,\dots,x_T^N)\in\mathcal{M}}{\arg\max} \quad \underbrace{\sum_{t,n} \left(\log \frac{q_t^n}{1-q_t^n} \right) x_t^n}_{S(x_1^1,\dots,x_T^N)}
\end{aligned}
$$

The score S to maximize is linear.

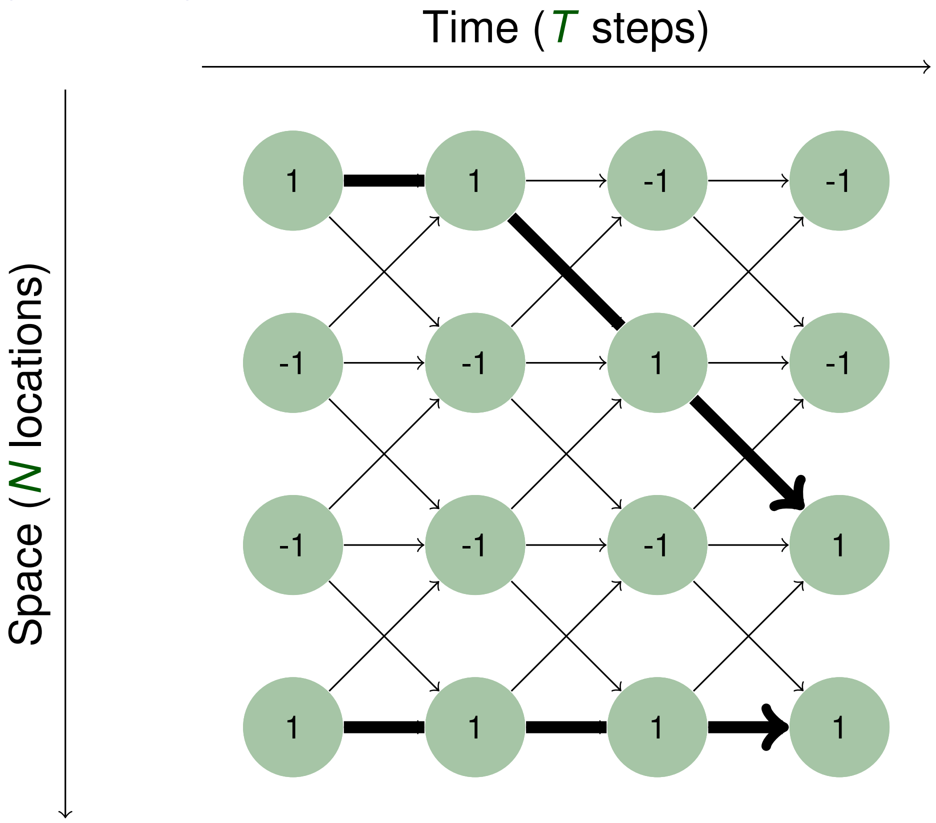
Graph-based Tracking
Union of disjoint trajectories

The set $\mathcal{M}$ contains all maps which are the unions of disjoint continuous trajectories.

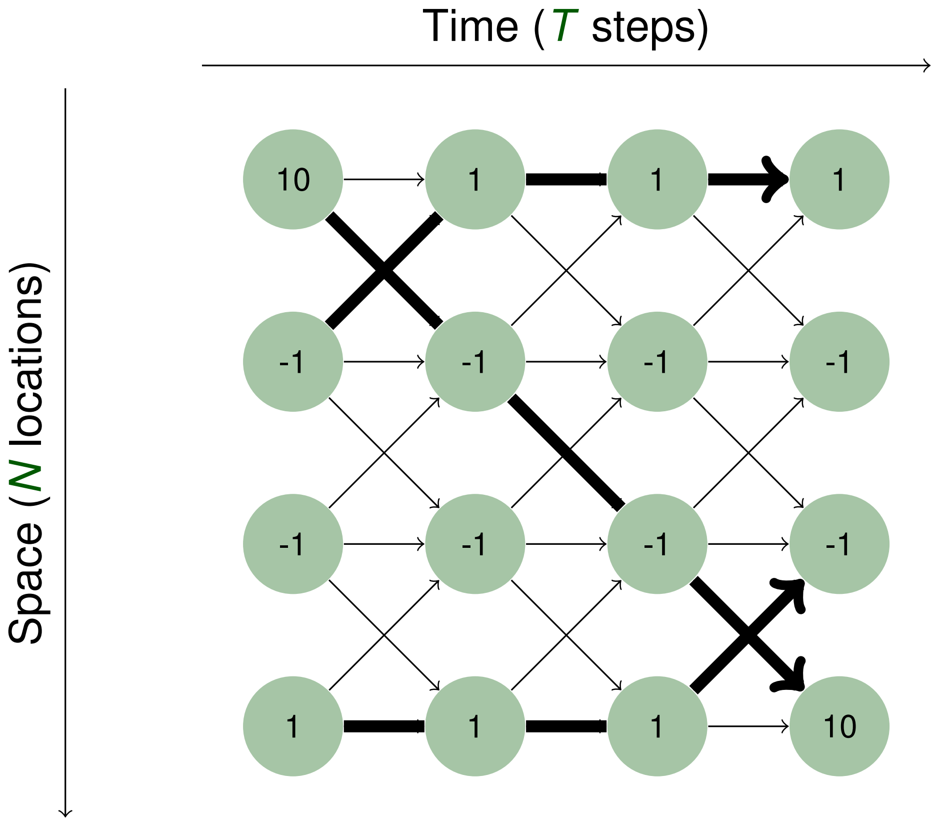
Since the value S is linear, if x is an union of disjoint trajectories, $S(x)$ is the sum of the values of the trajectories composing x.

A straight-forward strategy to compute $\arg\max_x S(x)$ is to add trajectories one after another.

Graph-based Tracking
Greedy optimization



Graph-based Tracking
Typical problem



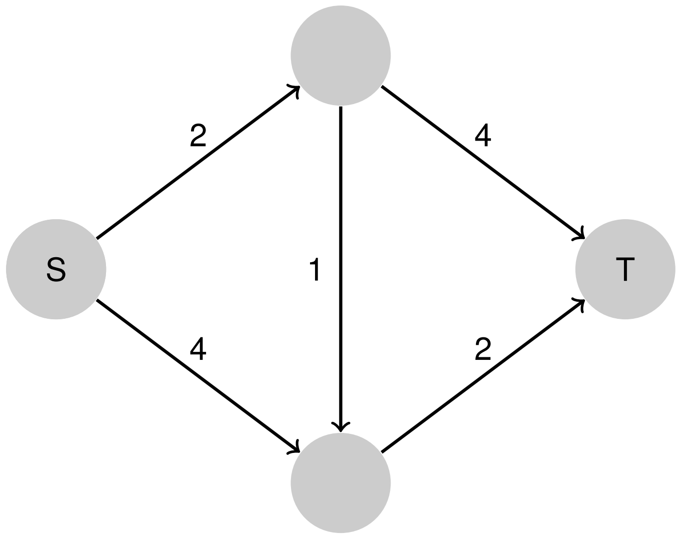
Graph-based Tracking
KSP

Given a graph, with a source and a sink, there is an algorithm to compute the family of K disjoint paths from source to sink with minimal total length.

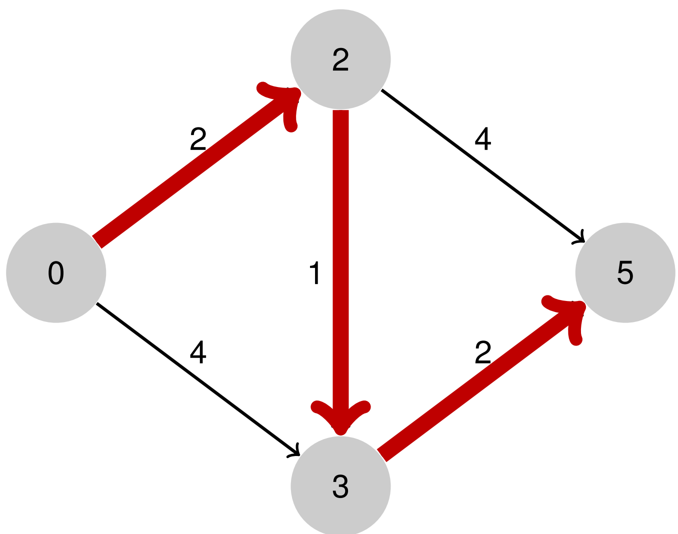
It consists of iteratively finding the shortest path, after inverting all the edges occupied by the paths found in the previous iterations.

The computed family is globally optimal.

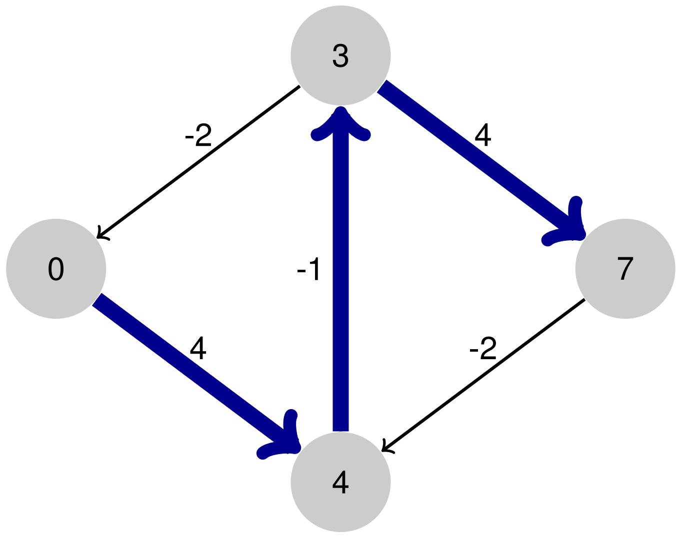
Graph-based Tracking
KSP



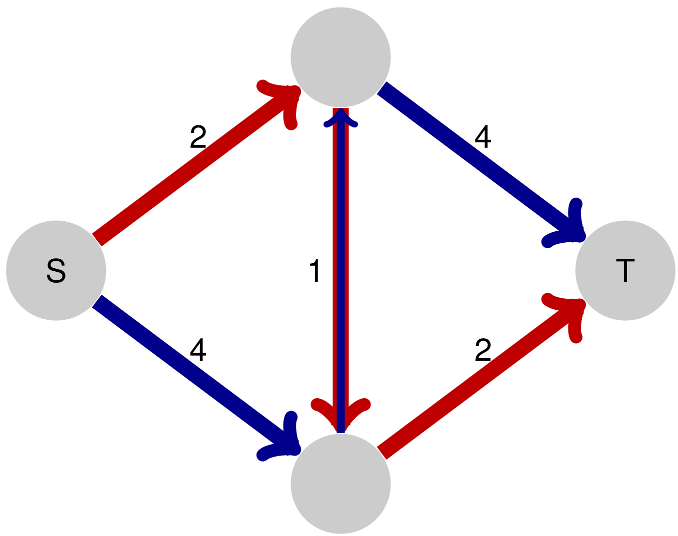
Graph-based Tracking
KSP

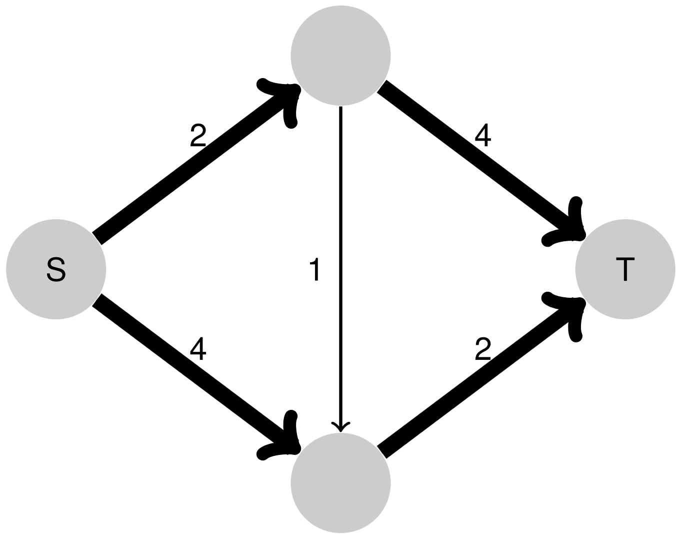


Graph-based Tracking
KSP



Graph-based Tracking
KSP





KSP
Pseudo-code

Algorithm 2 k-shortest Paths

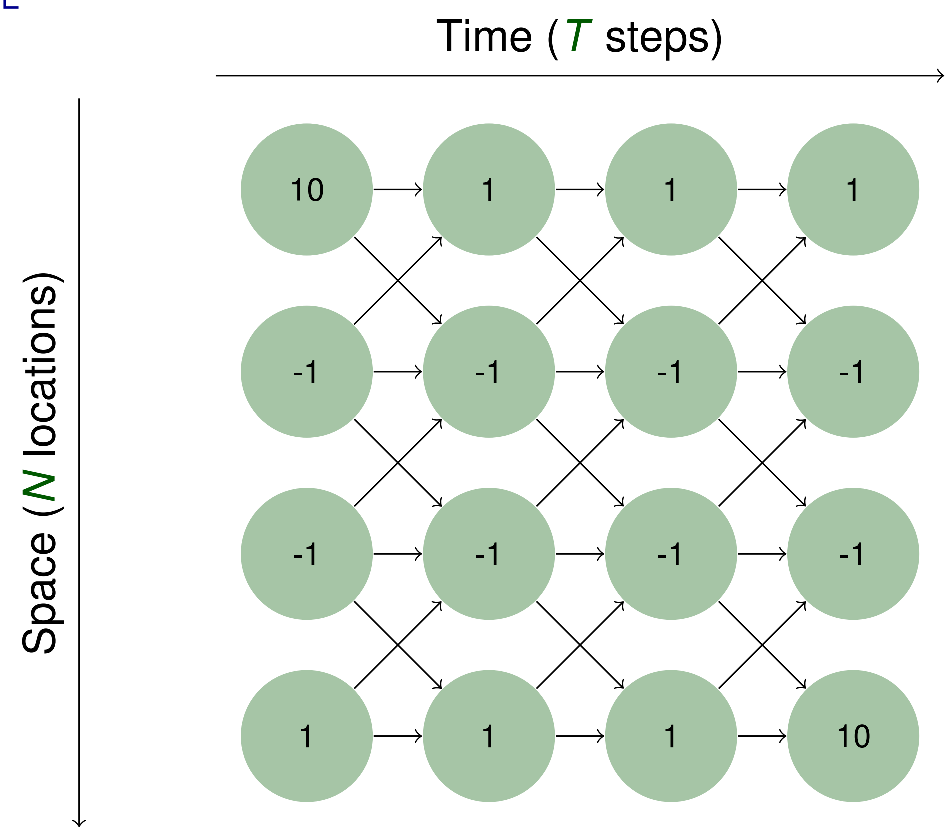
1: **for** $k = 1, \ldots, K$ **do**
2: Compute distances to source
3: Find best path
4: Invert occupied edges
5: Use distances to make all lengths positive
6: **end for**

MTP
Pseudo-code

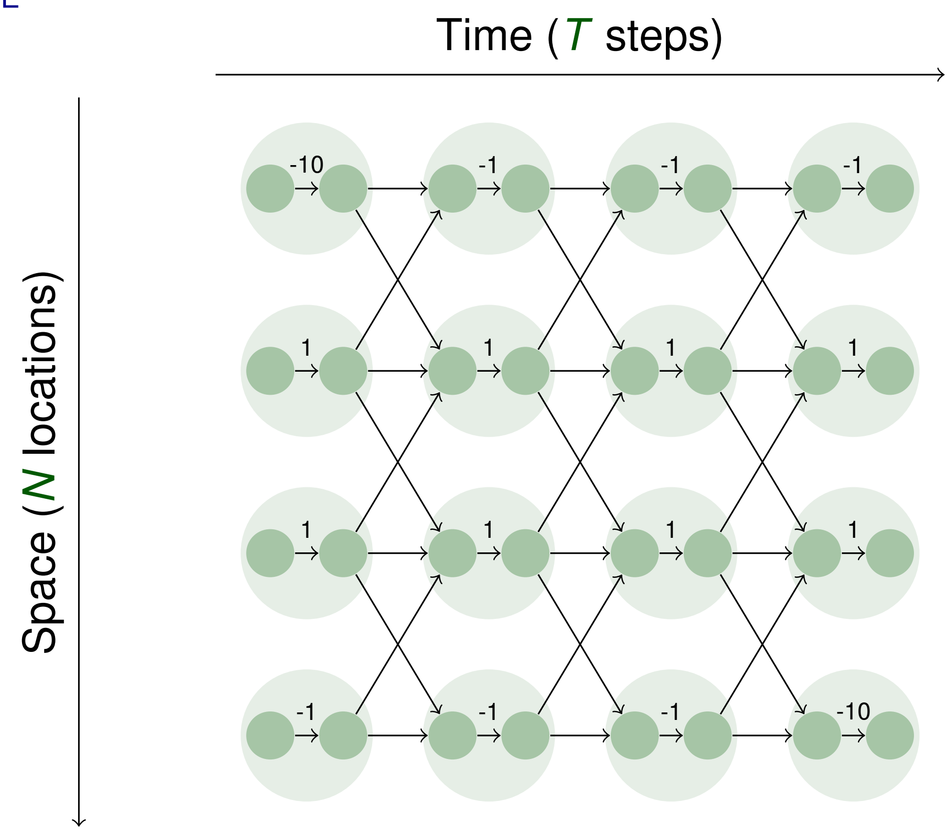
Algorithm 3 Multi-tracked Paths

```
repeat
    Compute distances to source
    Find best path
    Invert occupied edges
    Use distances to make all lengths positive
until Best path is of positive (original) length
```

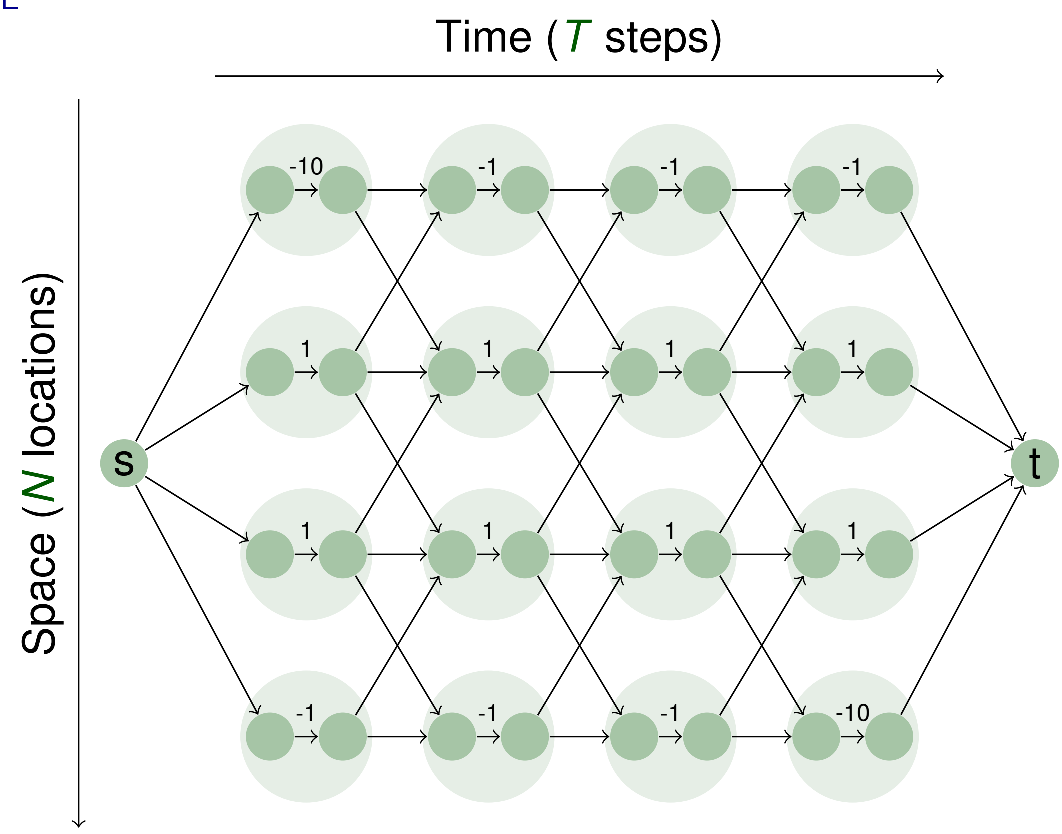

MTP
Example



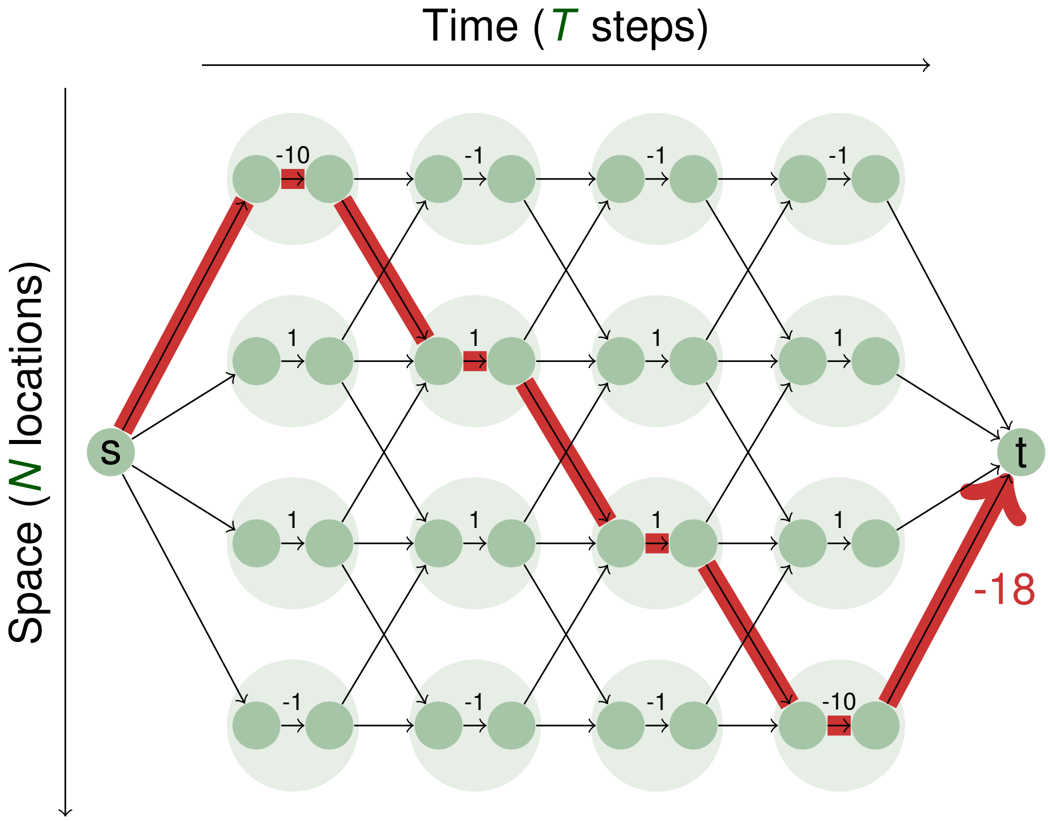
MTP
Example



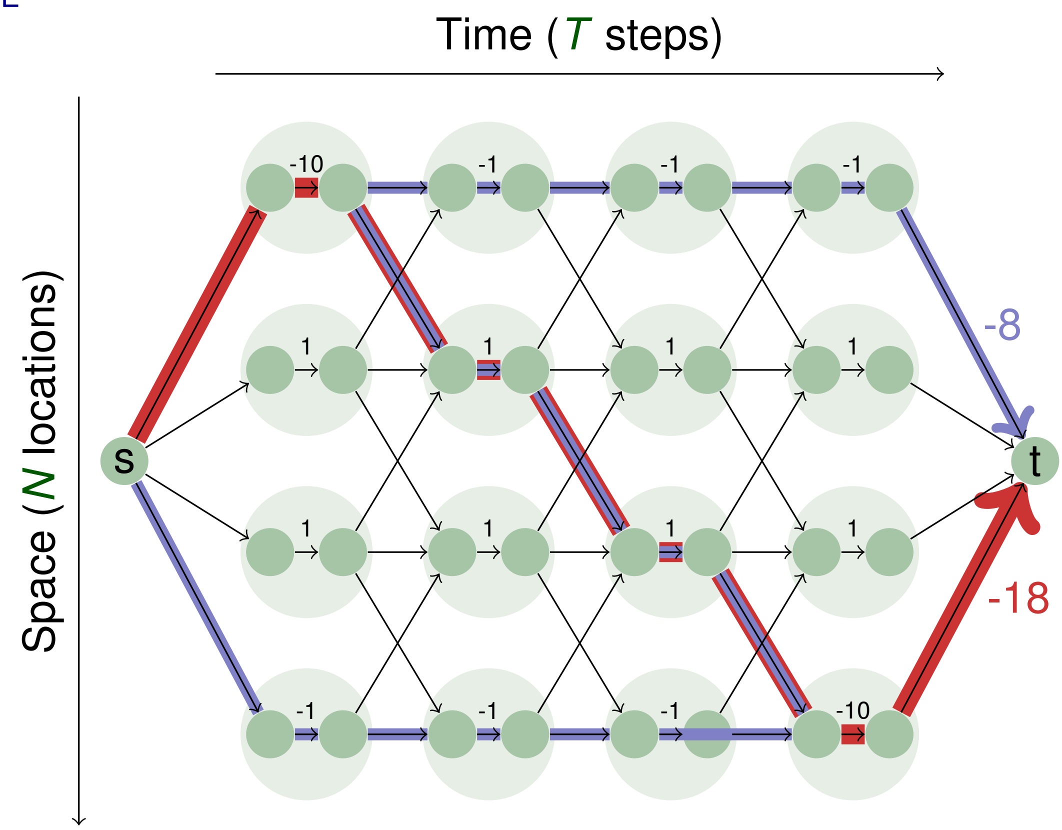
MTP
Example



MTP
Example

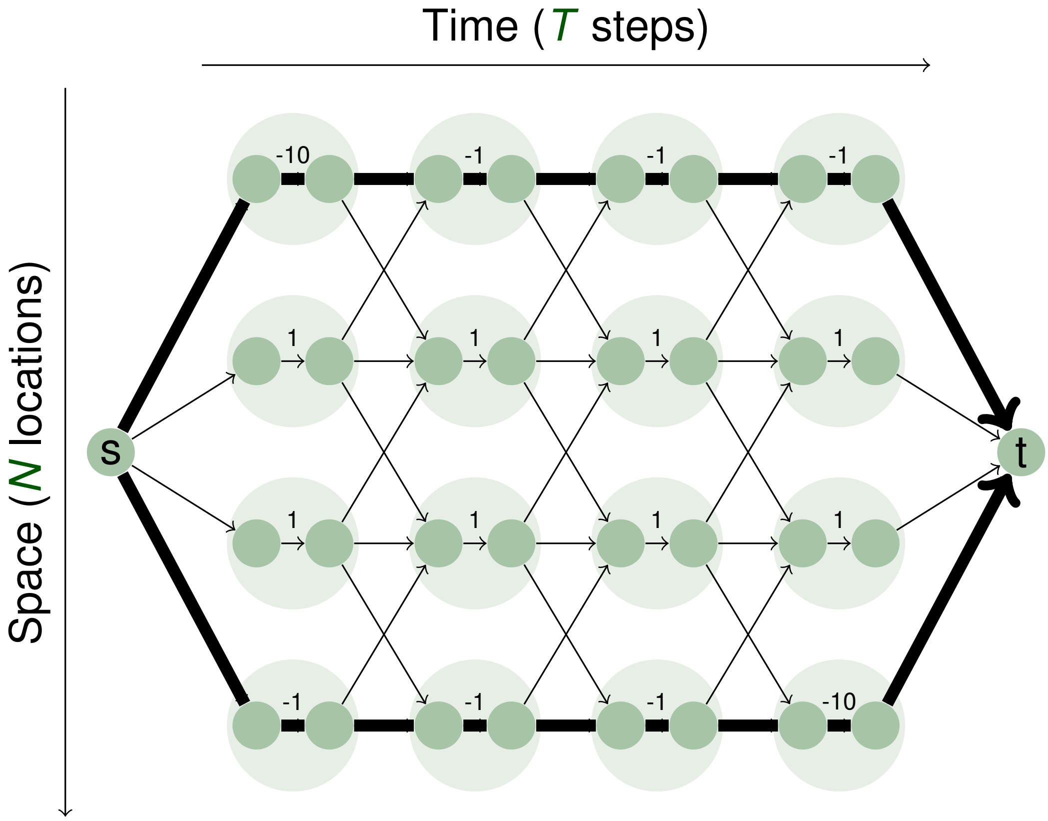


MTP
Example



MTP

Example



MTP GPL implementation

Example

```
// Initialize the tracker
MTPTracker *tracker = new MTPTracker();

tracker->allocate(nb_time_steps, nb_locations);

// Define the allowed target motions
for(int l = 0; l < nb_locations; l++) {
  for(int m = 0; m < nb_locations; m++) {
    tracker->allowed_motion[l][m] = abs(l - m) <= motion_amplitude;
  }
}

// Define the entrances (first frame and location 0) and exists
// (last frame and location nb_locations-1)
for(int t = 0; t < nb_time_steps; t++) {
  for(int l = 0; l < nb_locations; l++) {
    tracker->entrances[t][l] = (t == 0 || l == 0);
    tracker->exits[t][l] = (t == nb_time_steps - 1 || l == nb_locations-1);
  }
}

// Build the graph to us the modified KSP
tracker->build_graph();
```

MTP GPL IMPLEMENTATION
EXAMPLE

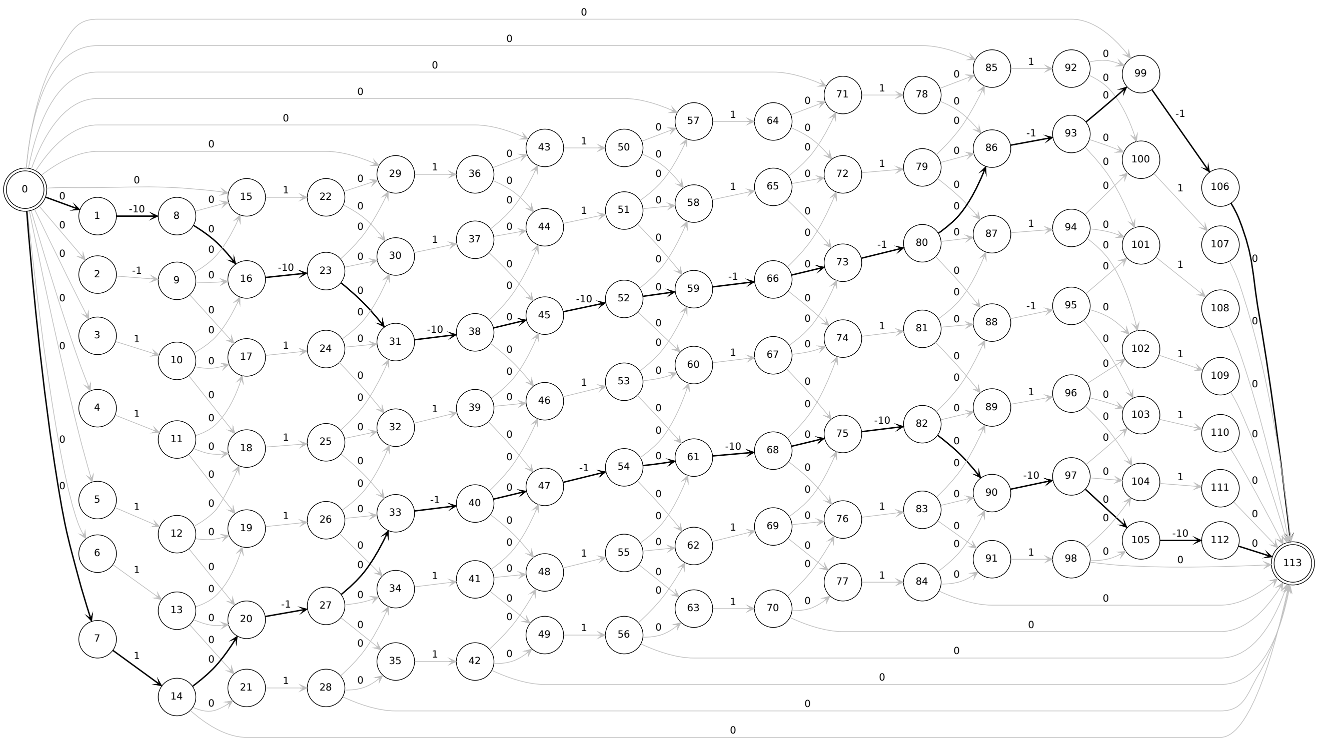
```
// Fill the detection map
for(int t = 0; t < nb_time_steps; t++) {
  for(int l = 0; l < nb_locations; l++) {
    tracker->detection_scores[t][l] = detect(t, l);
  }
}

// Performs the tracking per se
tracker->track();

// Prints the detected trajectories
for(int t = 0; t < tracker->nb_trajectories(); t++) {
  cout << t;
  for(int u = 0; u < tracker->trajectory_duration(t); u++) {
    cout << " " << tracker->trajectory_location(t, u);
  }
  cout << endl;
}

delete tracker;
```


MTP GPL IMPLEMENTATION
RESULT GRAPH



$L = 7$ and $T = 8$

RESULTS

This algorithm has been tested on the tracking of basketball players in real-world situation:

Basketball POM+KSP [video]

We have recently put an appearance model back by adding an additional “identity” variable. KSP can not optimize it anymore, so we run LP on the locations selected without appearance by KSP.

Basketball POM+KSP+LP [video]

REFERENCES

✎ F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):267–282, 2008

✎ J. Berclaz, E. Turetken, F. Fleuret, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(9):1806–1819, 2011

✎ J. Berclaz, F. Fleuret, and P. Fua. Multi-camera tracking and atypical motion detection with behavioral maps. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 112–125, 2008

✎ J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Proceedings of the 12th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (Winter-PETS)*, pages 1–8, 2009

✎ H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple objects under global appearance constraints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 137–144, 2011

OPEN-SOURCE IMPLEMENTATIONS

You can get the implementation of POM, under GPL3, at

```
git clone http://www.idiap.ch/~fleuret/git/pom/
```

and the implementation of MTP, under GPL3, at

```
git clone http://www.idiap.ch/~fleuret/git/mtp/
```

The reference implementations, additional information, and videos can be found at

http://cvlab.epfl.ch/research/playground/

THANK YOU!

francois.fleuret@idiap.ch
http://www.idiap.ch/~fleuret/