Performance enhancement of the vision system for the augmented reality Thymio robot

Presentation Bachelor-Thesis

Michael A. Flückiger

Dr. Stéphane Magnenat, Dr. Fabio Zünd

Prof. Dr. Robert W. Sumner

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Game Technology Center

Initial situation

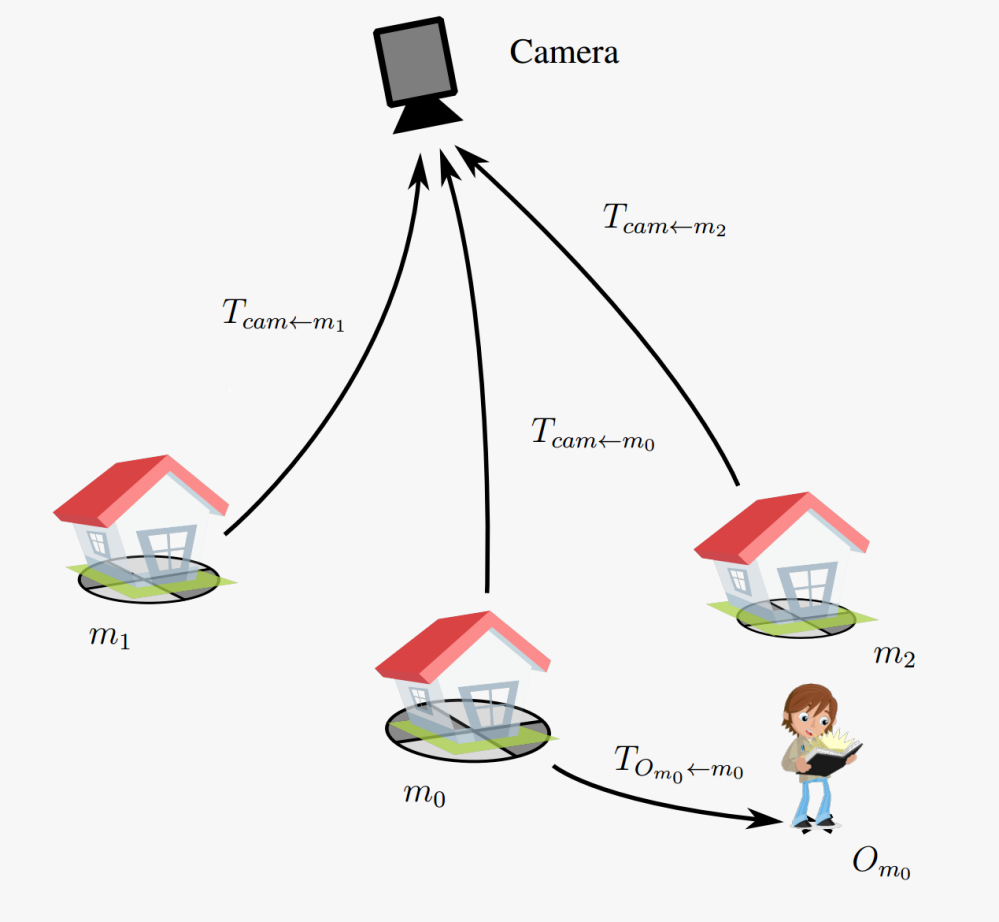
Thymio Programming Adventure

- Involves robotics and augmented reality to help children discover programming
- World observed through camera, augmented with virtual objects relative to the position of printable markers

Augmented world scene through the application

Thymio educational robot

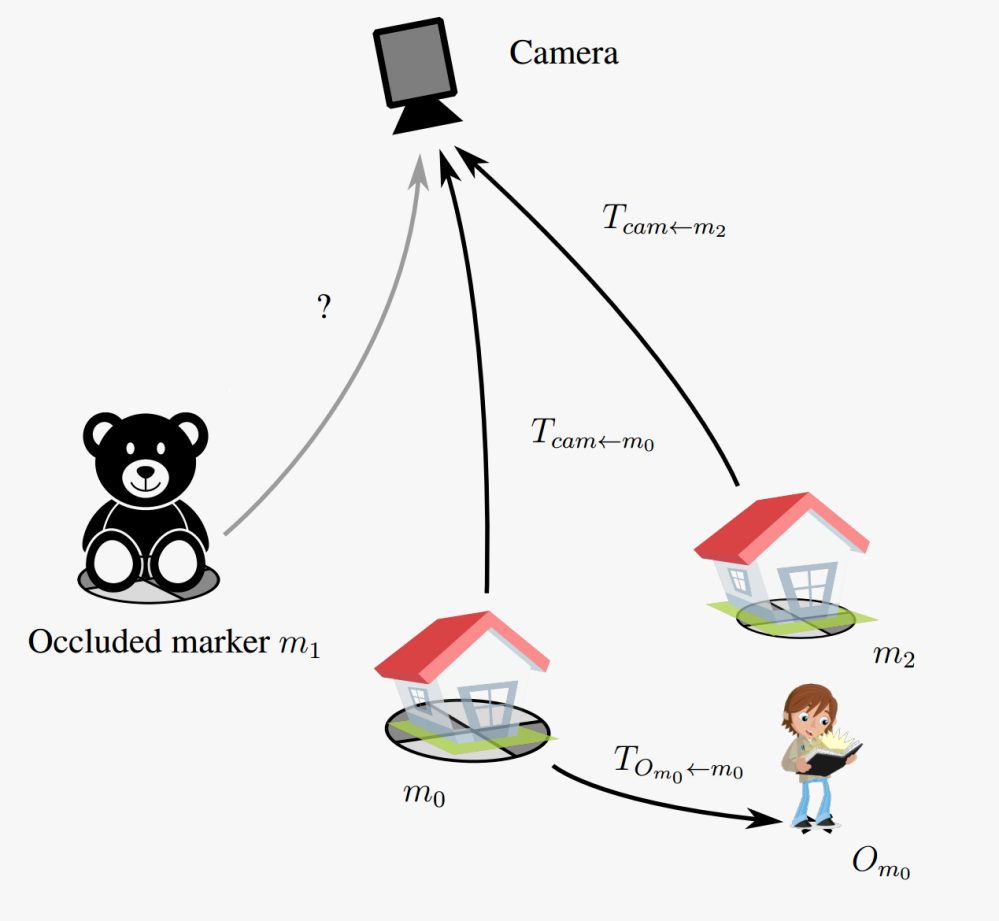
Statement of the problem



All markers are visible to the camera

As long as a marker is visible, its corresponding virtual objects can be drawn

Statement of the problem



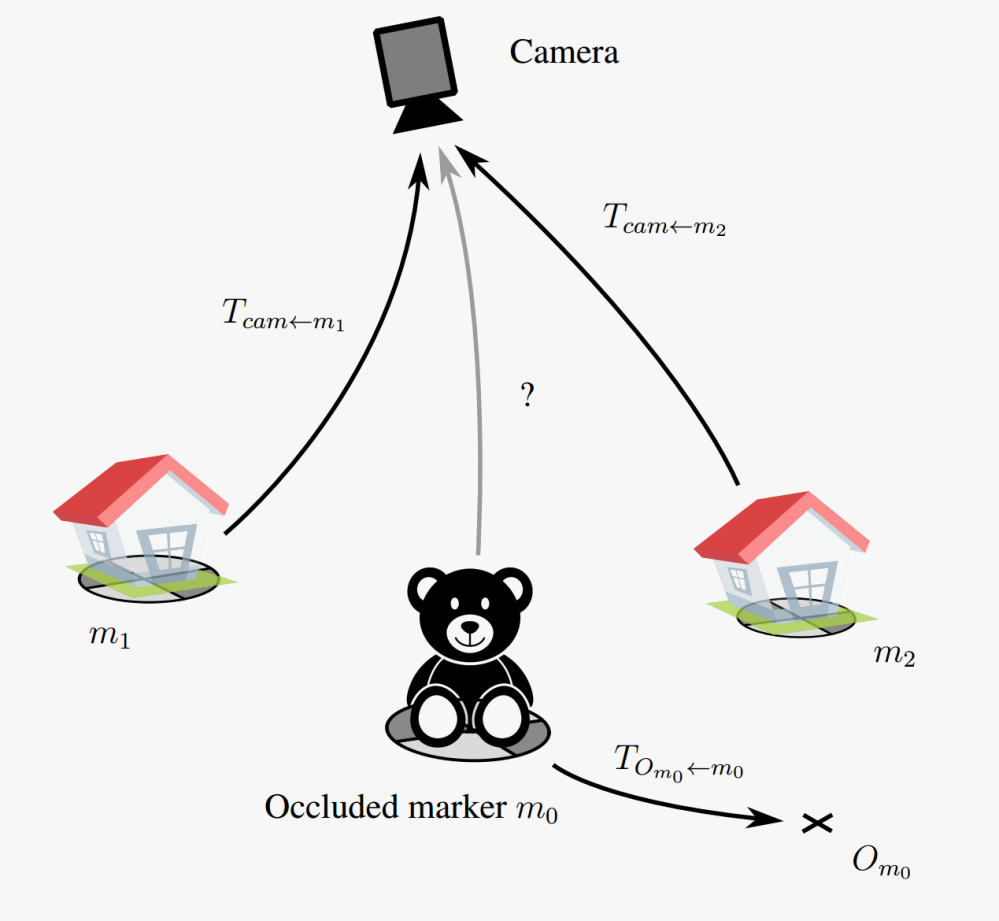
Marker m_1 is hidden

As long as a marker is visible, its corresponding virtual objects can be drawn

Position of hidden marker is no longer known

Can no longer draw objects described relative to this hidden marker

Statement of the problem



Marker m_0 is hidden

As long as a marker is visible, its corresponding virtual objects can be drawn

Position of hidden marker is no longer known

Can no longer draw objects described relative to this hidden marker

Especially disadvantageous when virtual object not direct at position of marker

Reordering of the scene graph as a possible solution

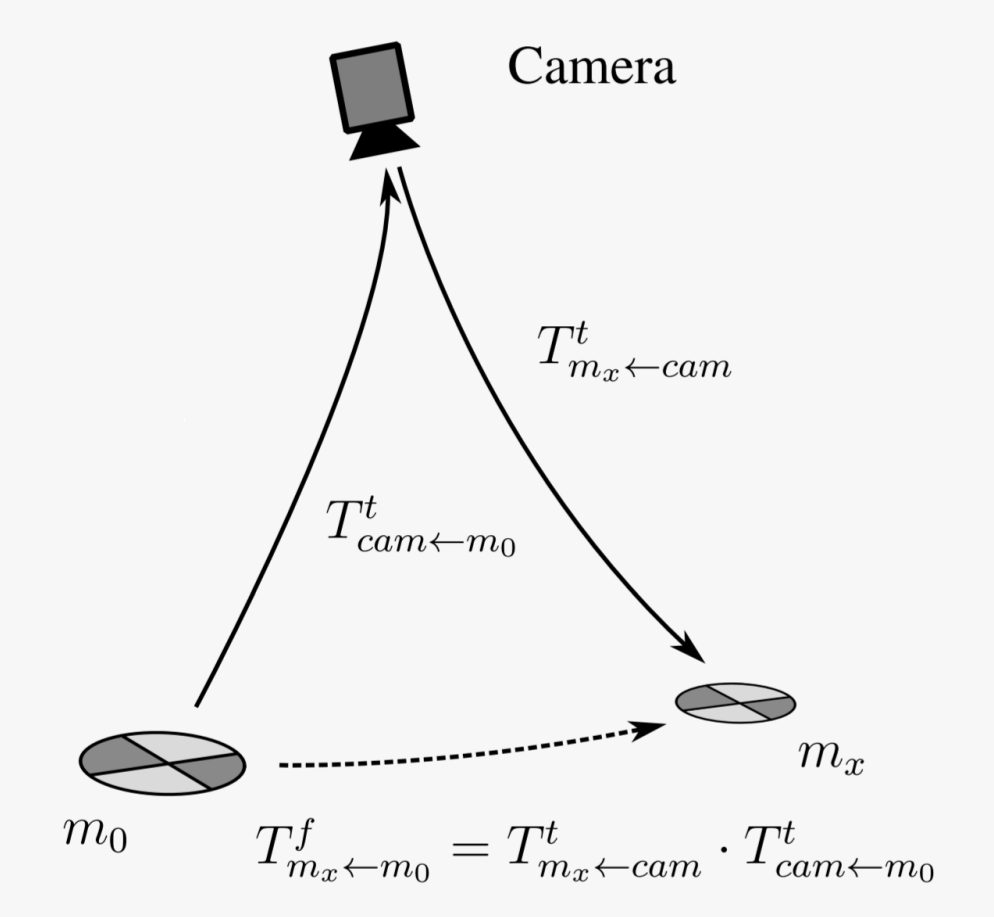
Solution chosen

Maintain the position of a world center marker which is known as long as at least one of multiple markers is visible

Allows to express all virtual objects relative to this world center marker

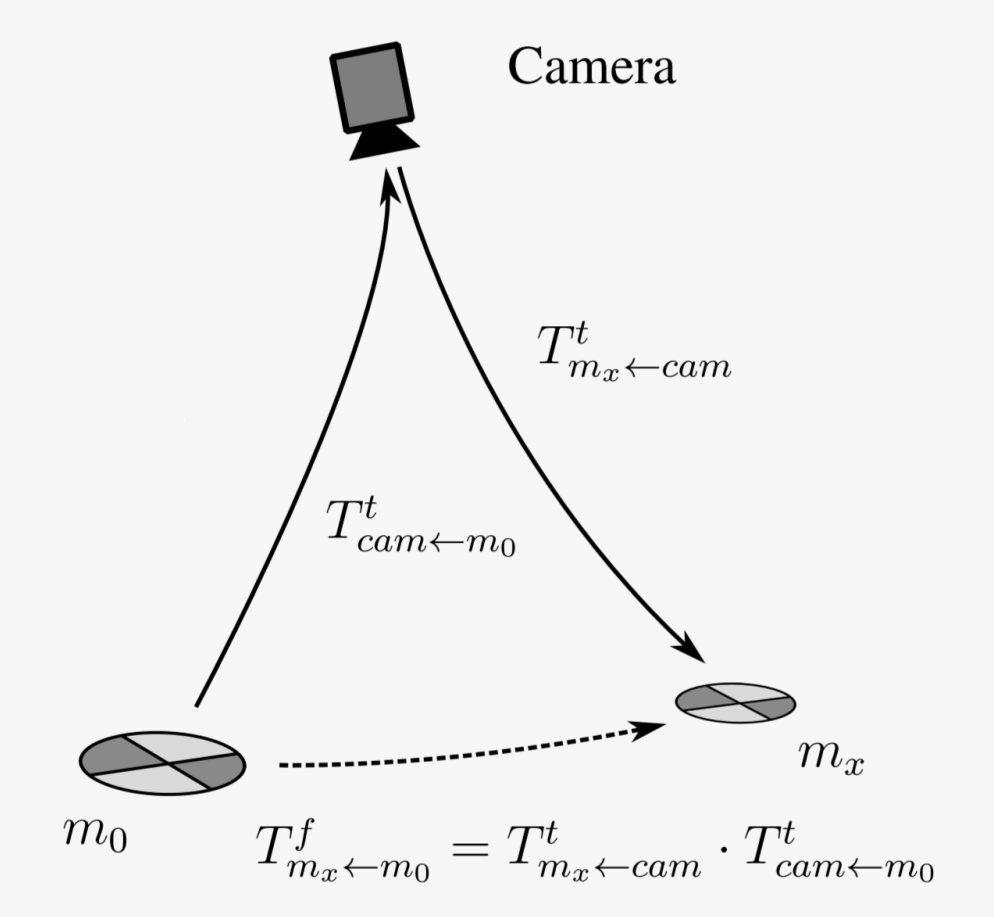
How to maintain the position of such a world center marker?

Storing transformations in time

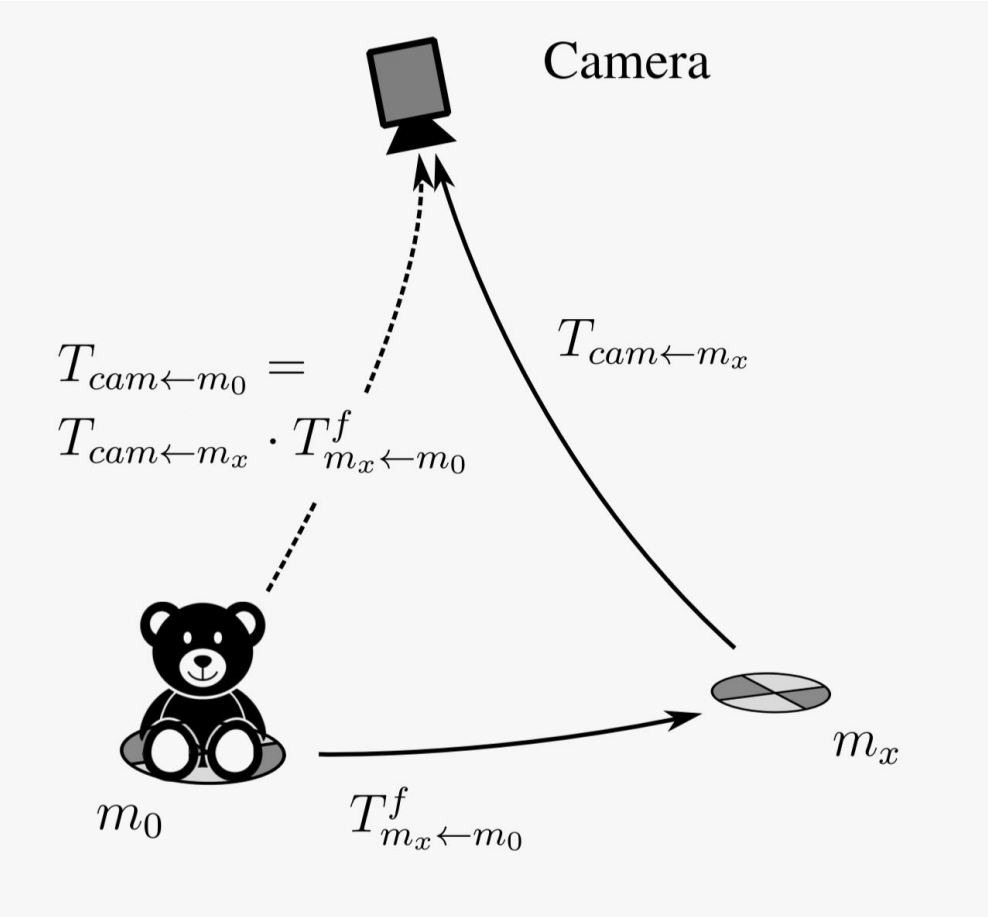


World center marker m_0 and marker m_x are both visible at *t*, store fixed transformation

Storing transformations in time



World center marker m_0 and marker m_x are both visible at *t*, store fixed transformation



World center marker m_0 hidden, use stored fixed transformation to calculate its position

TransMem – Transformation memory

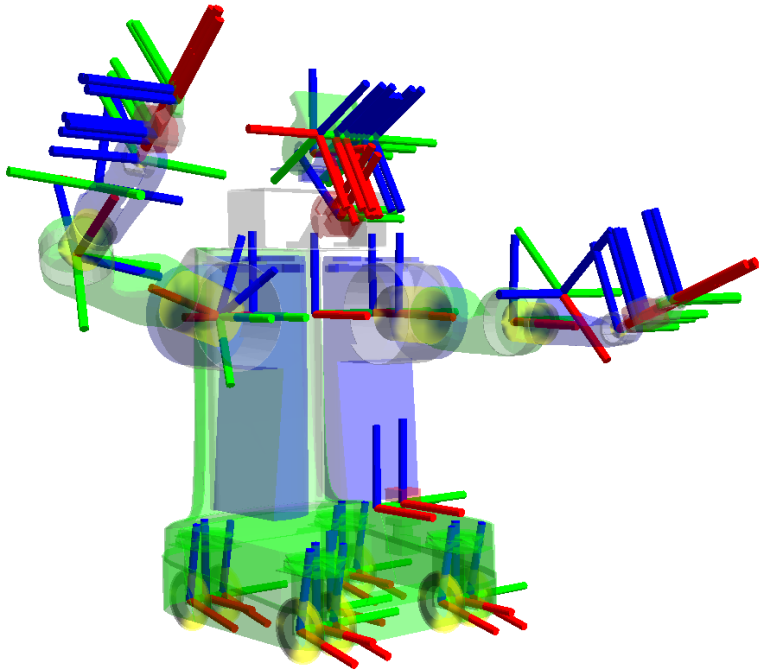
Data structure for storing and retrieving transformation in time

Based on an underlying graph data structure

Design inspired by *ROS's tf library*

Two main functionalities:

- Storage of a transformation
- Retrieval of a transformation

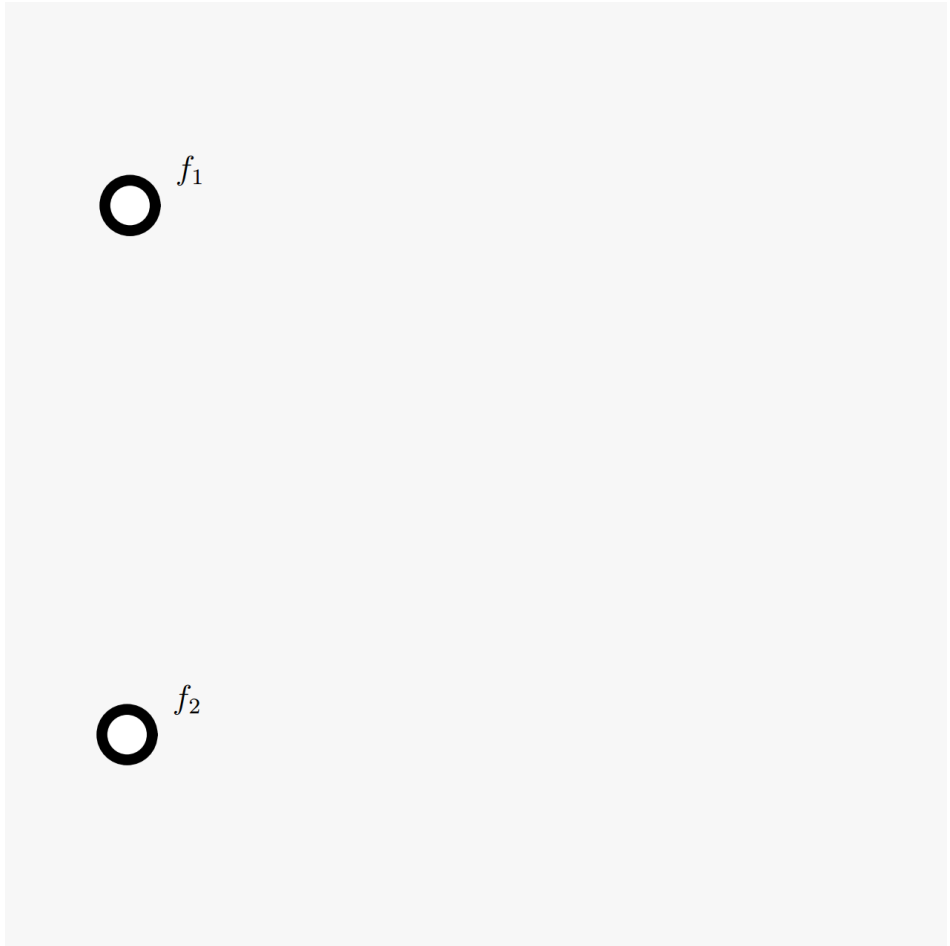


http://wiki.ros.org/tf

TransMem – Storing a transformation

Want to store transformation $T_{f_2 \leftarrow f_1}^{t_0}$

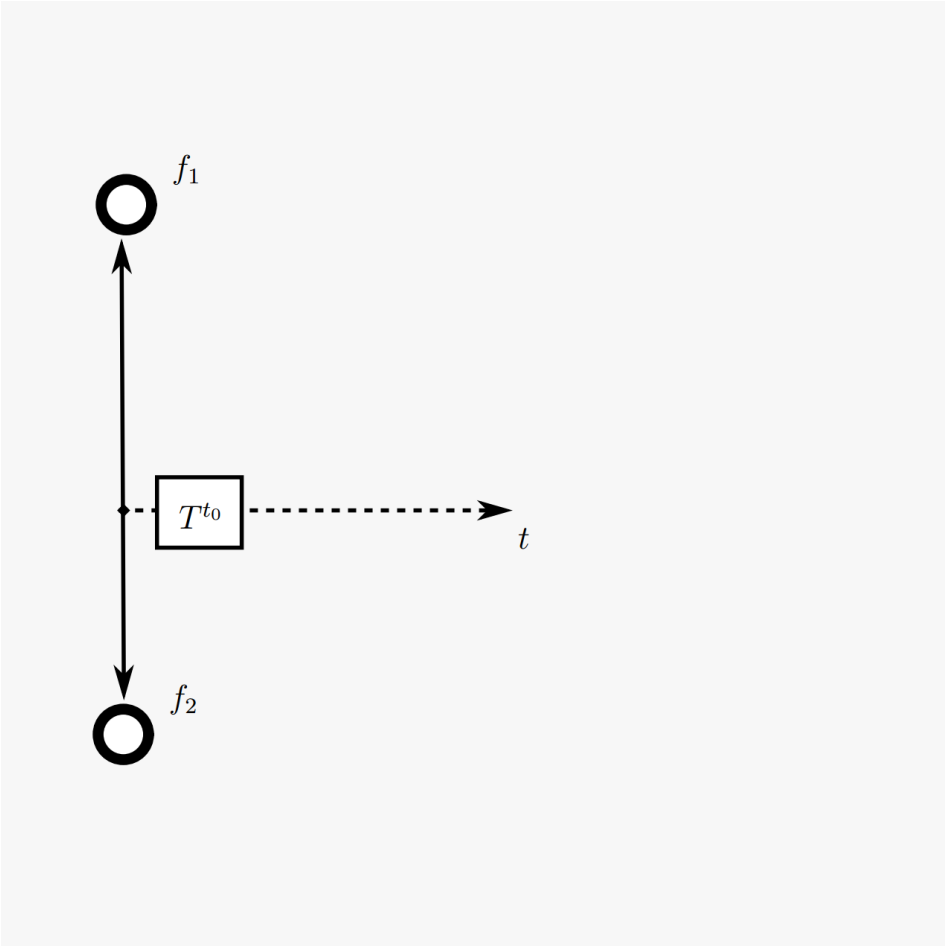
TransMem – Storing a transformation



Want to store transformation $T_{f_2 \leftarrow f_1}^{t_0}$

Creates frames f_1 and f_2

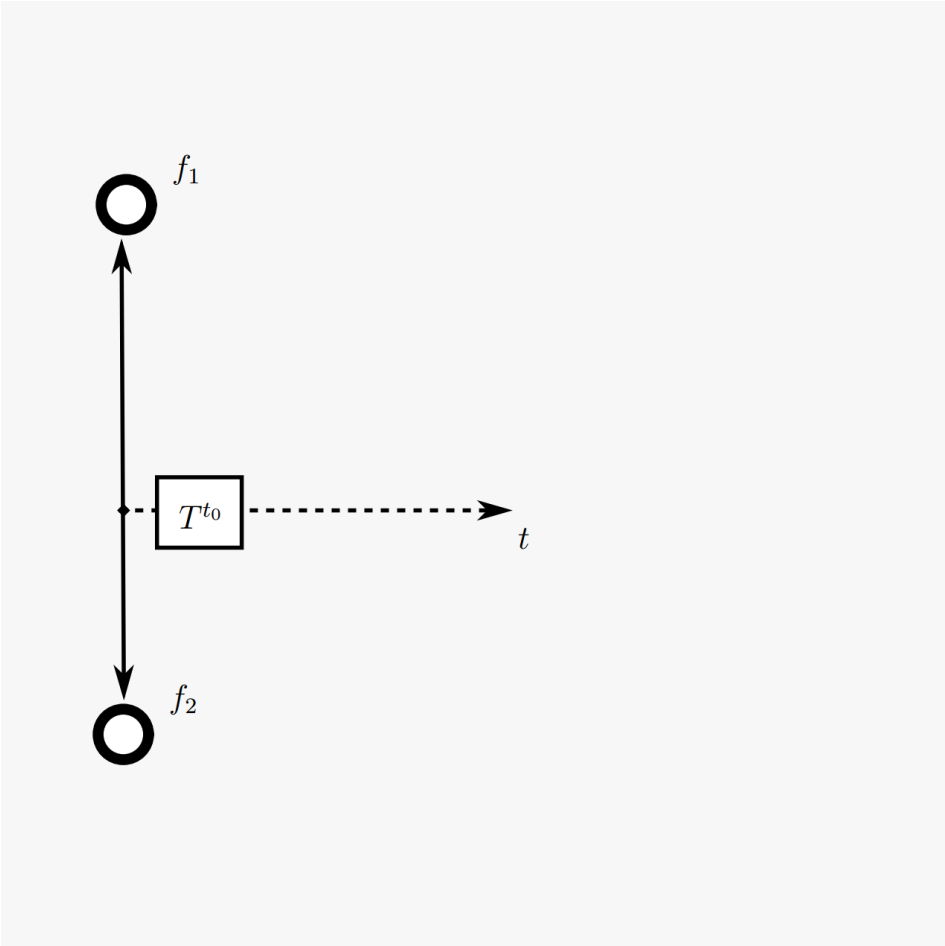
TransMem – Storing a transformation



Want to store transformation $T_{f_2 \leftarrow f_1}^{t_0}$

Creates frames f_1 and f_2 and stores transformation on link between the frames

TransMem – Storing a transformation

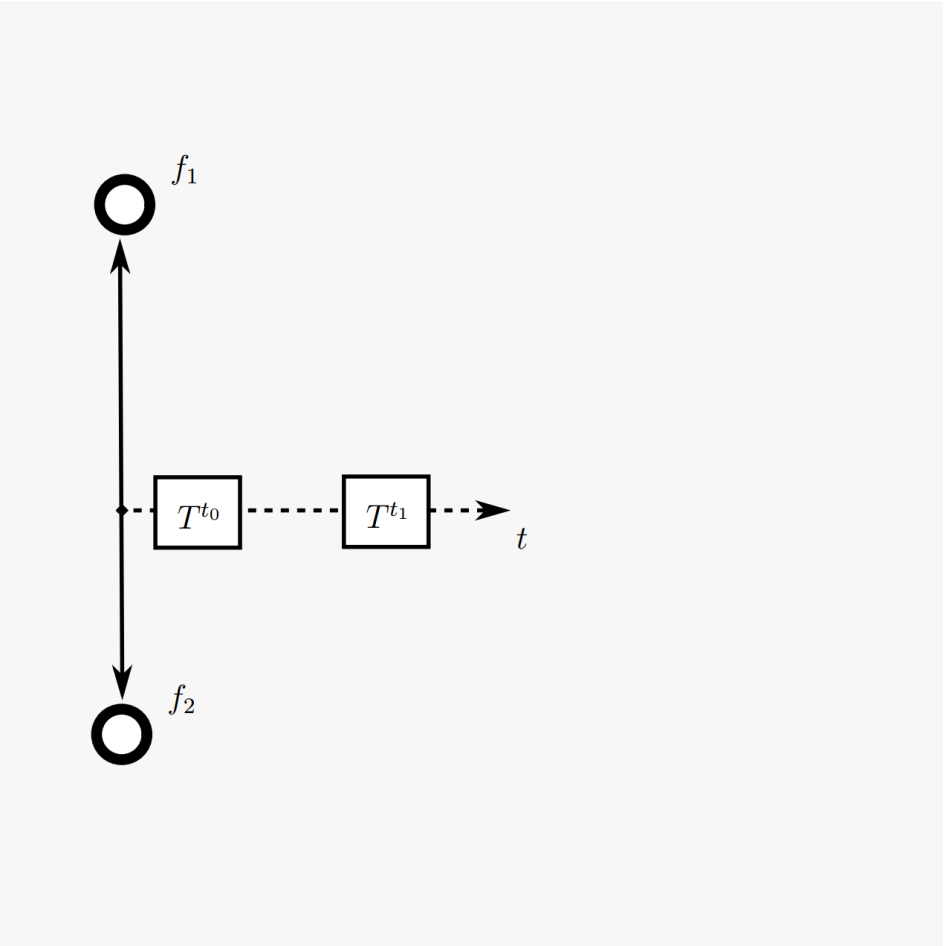


Want to store transformation $T_{f_2 \leftarrow f_1}^{t_0}$

Creates frames f_1 and f_2 and stores transformation on link between the frames

Storing another transformation $T_{f_2 \leftarrow f_1}^{t_1}$

TransMem – Storing a transformation



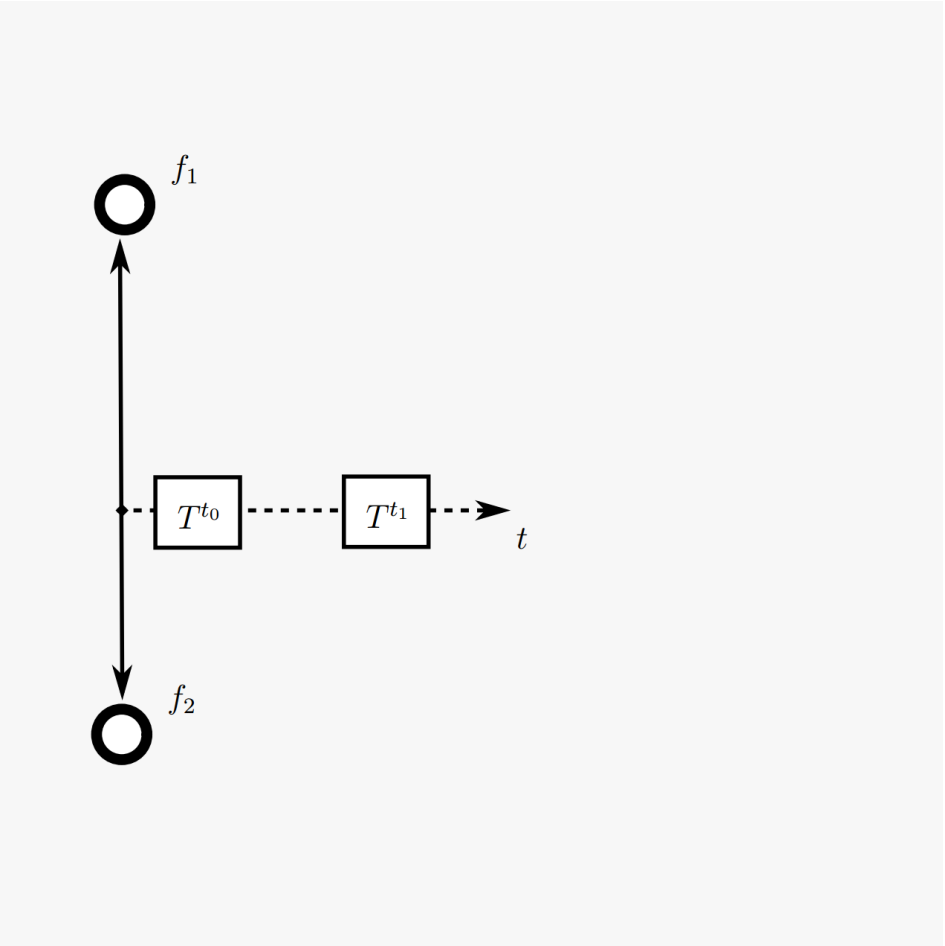
Want to store transformation $T_{f_2 \leftarrow f_1}^{t_0}$

Creates frames f_1 and f_2 and stores transformation on link between the frames

Storing another transformation $T_{f_2 \leftarrow f_1}^{t_1}$

Transformations on a link arranged according to their validity time

TransMem – Storing a transformation



Want to store transformation $T^{t_0}_{f_2 \leftarrow f_1}$

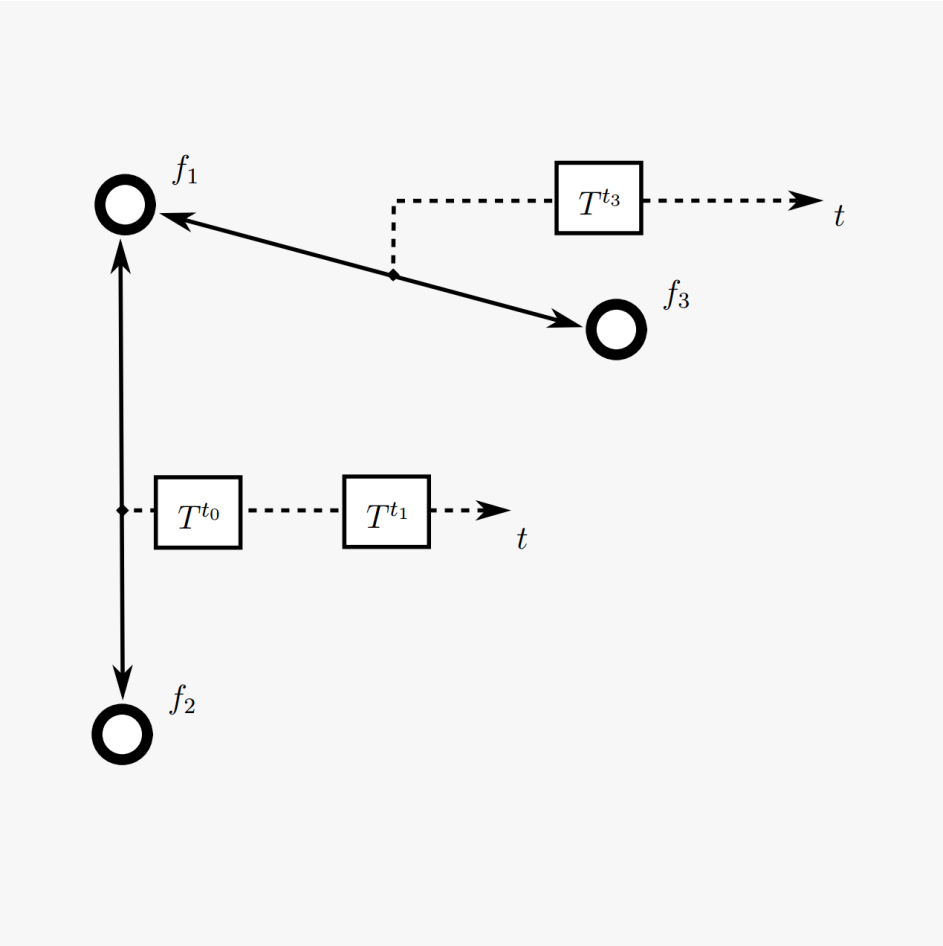
Creates frames f_1 and f_2 and stores transformation on link between the frames

Storing another transformation $T^{t_1}_{f_2 \leftarrow f_1}$

Transformations on a link arranged according to their validity time

Storing a third transformation $T^{t_2}_{f_3 \leftarrow f_1}$

TransMem – Storing a transformation



Want to store transformation $T^{t_0}_{f_2 \leftarrow f_1}$

Creates frames f_1 and f_2 and stores transformation on link between the frames

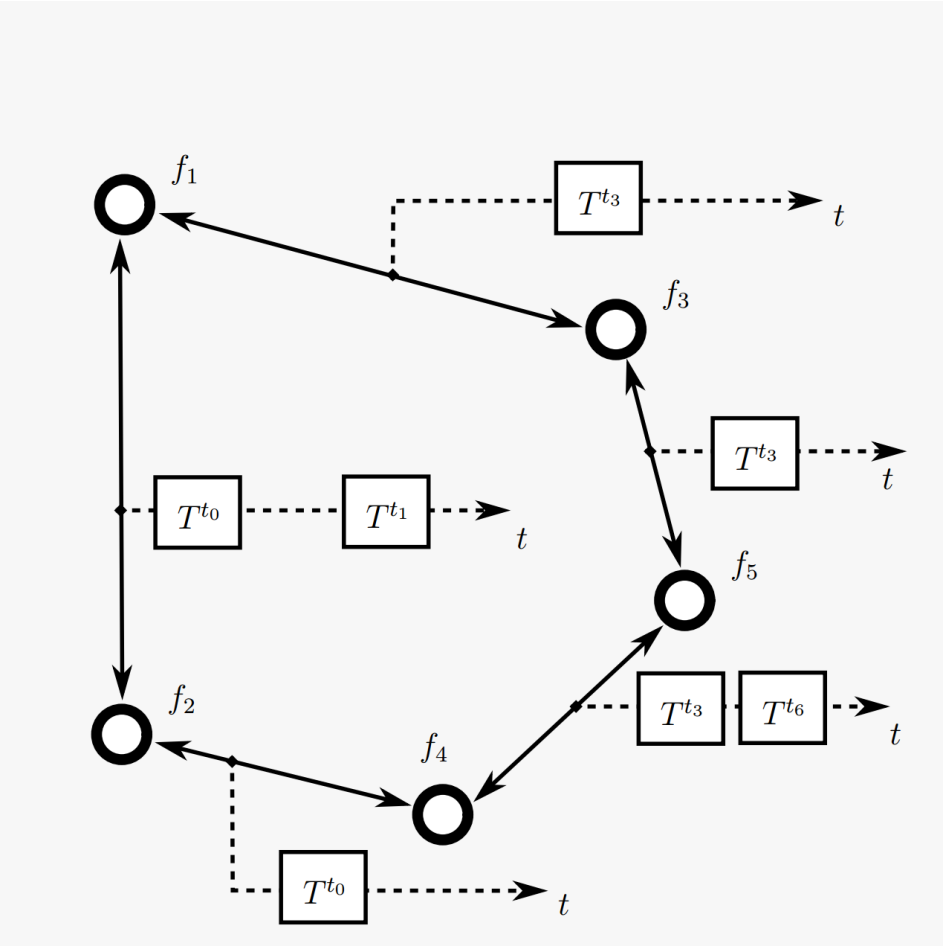
Storing another transformation $T^{t_1}_{f_2 \leftarrow f_1}$

Transformations on a link arranged according to their validity time

Storing a third transformation $T^{t_3}_{f_3 \leftarrow f_1}$

Creates a new link since destination is different

TransMem – Storing a transformation

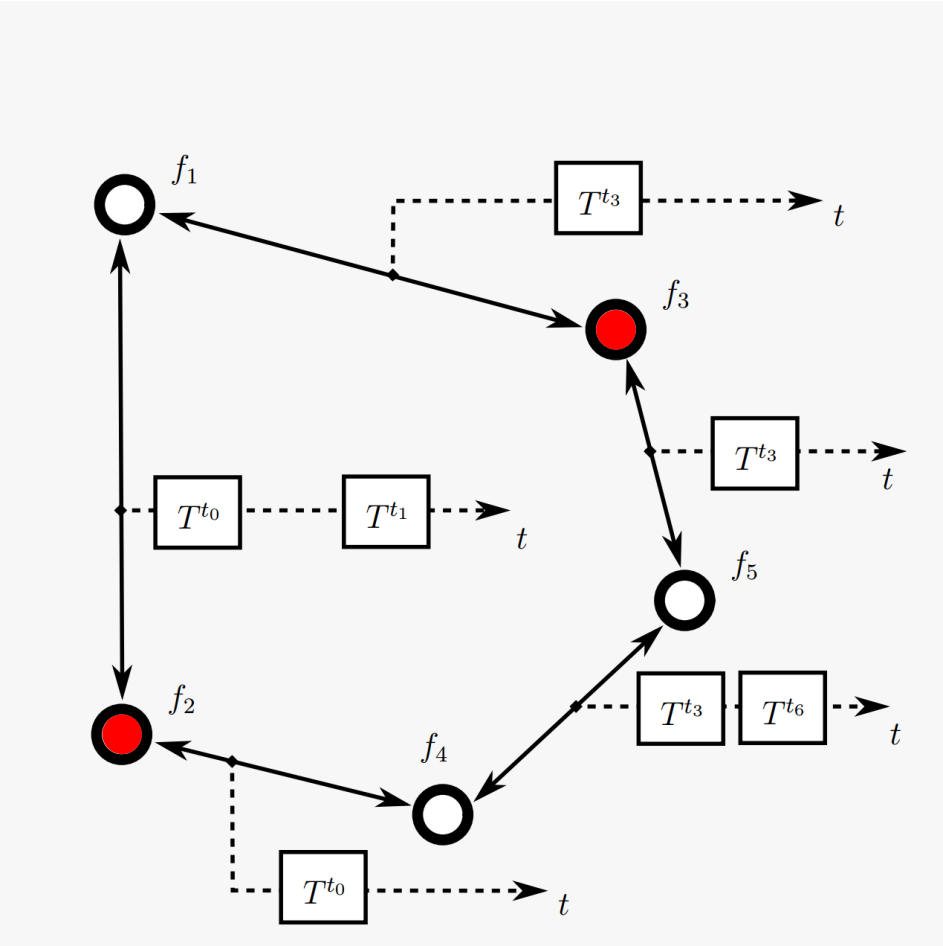


TransMem after a series of insertions

Transformations which are too old are removed

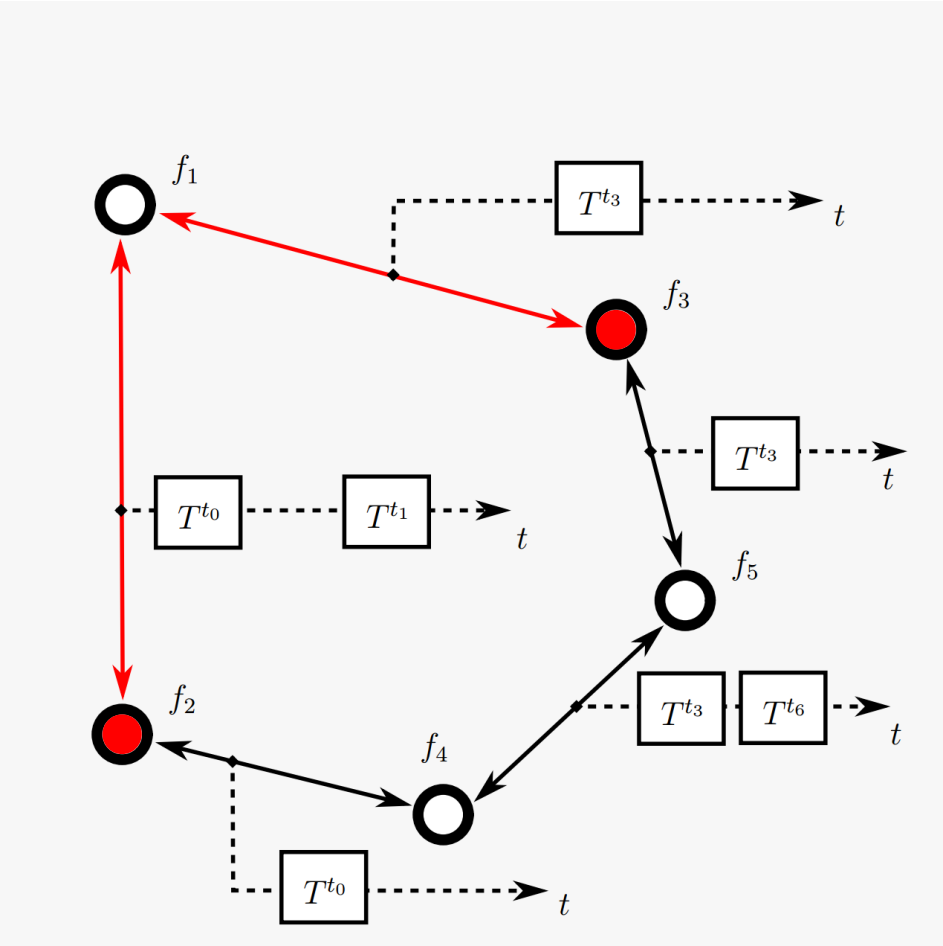
Duration of storage specified by the user

TransMem – Querying for a transformation



Want to know transformation from f_2 to f_3 valid at time t_q: $T^{t_q}_{f_3 \leftarrow f_2}$

TransMem – Querying for a transformation

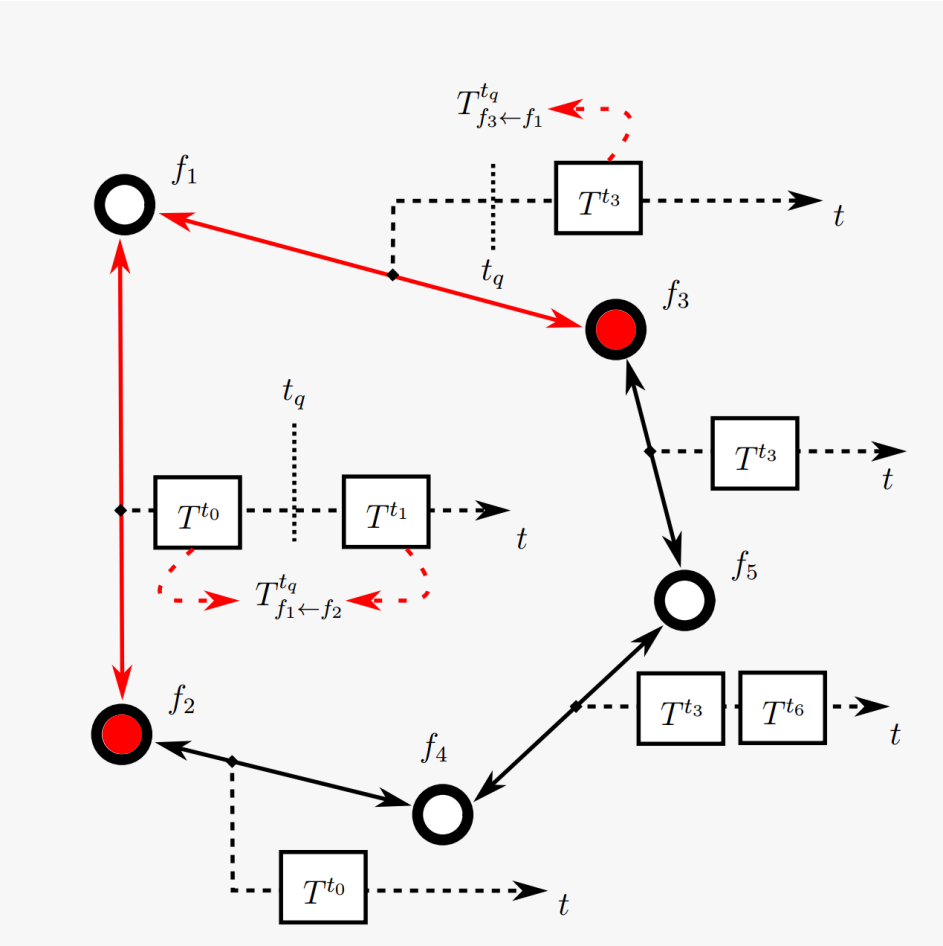


Want to know transformation from f_2 to f_3 valid at time t_q: $T^{t_q}_{f_3 \leftarrow f_2}$

Determination of shortest path

Consists of links $L_{f_1 \leftrightarrow f_2}$ and $L_{f_1 \leftrightarrow f_3}$

TransMem – Querying for a transformation



Want to know transformation from f_2 to f_3 valid at time t_q: $T^{t_q}_{f_3 \leftarrow f_2}$

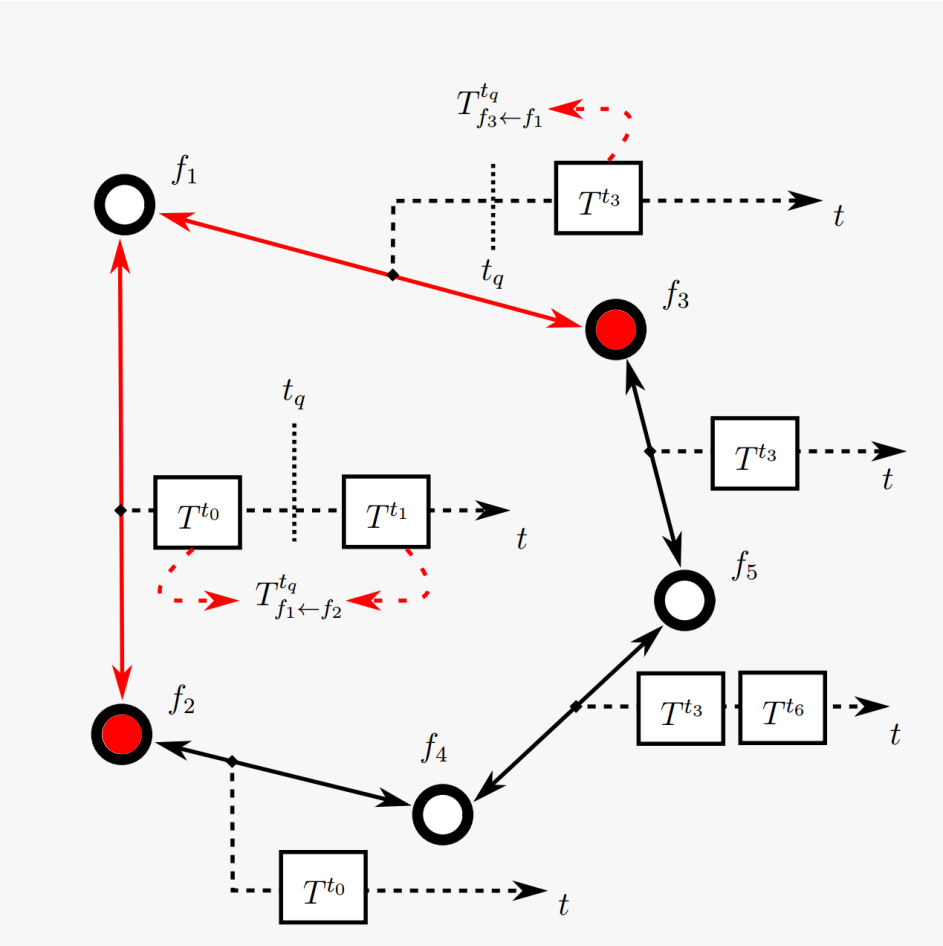
Determination of shortest path

Consists of links $L_{f_1 \leftrightarrow f_2}$ and $L_{f_1 \leftrightarrow f_3}$

Choosing appropriate transformation on each link part of the path

$T^{t_q}_{f_1 \leftarrow f_2}$ and $T^{t_q}_{f_3 \leftarrow f_1}$

TransMem – Querying for a transformation



Want to know transformation from f_2 to f_3 valid at time t_q: $T^{t_q}_{f_3 \leftarrow f_2}$

Determination of shortest path

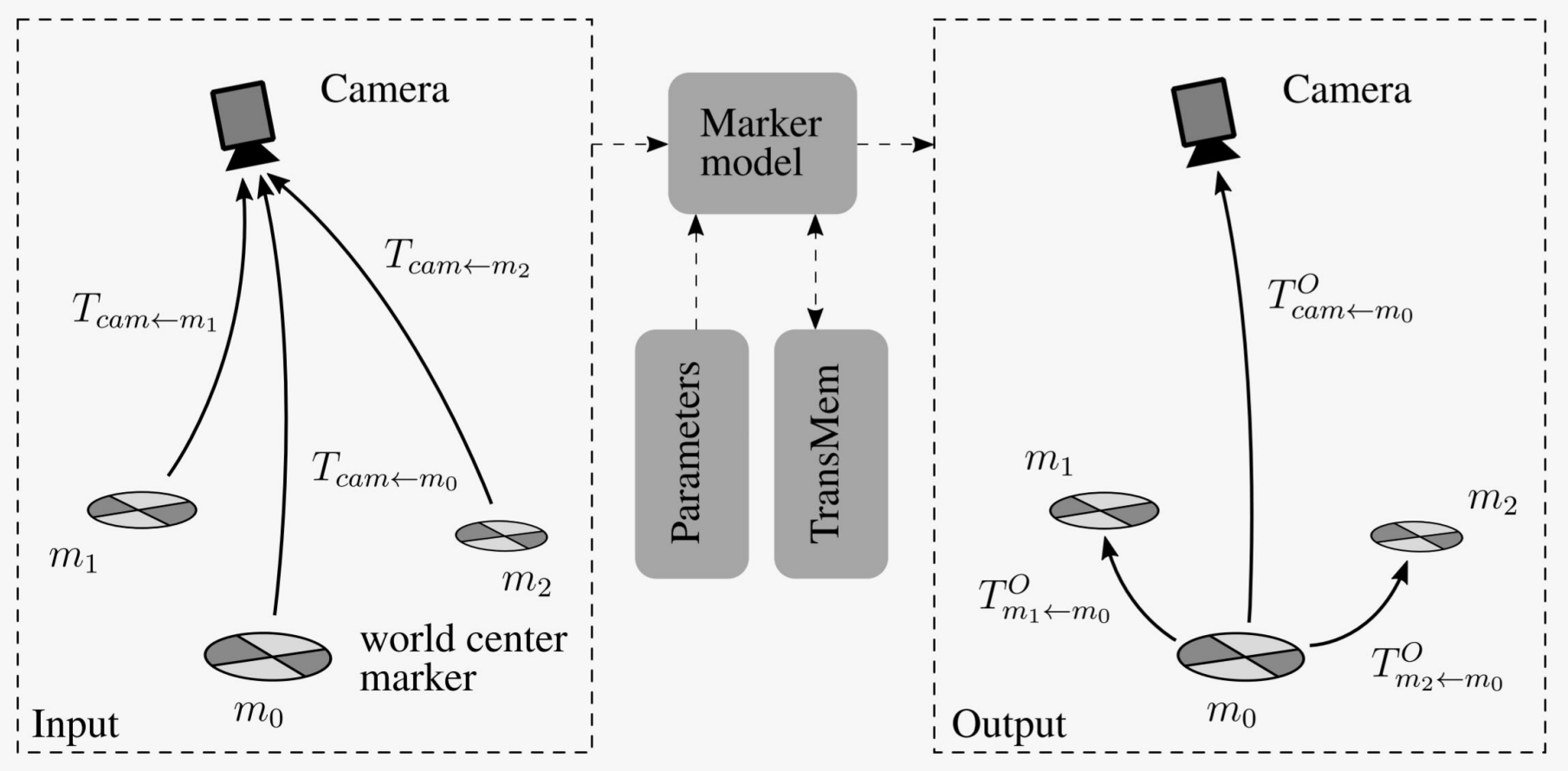
Consists of links $L_{f_1 \leftrightarrow f_2}$ and $L_{f_1 \leftrightarrow f_3}$

Choosing appropriate transformation on each link part of the path

$T^{t_q}_{f_1 \leftarrow f_2}$ and $T^{t_q}_{f_3 \leftarrow f_1}$

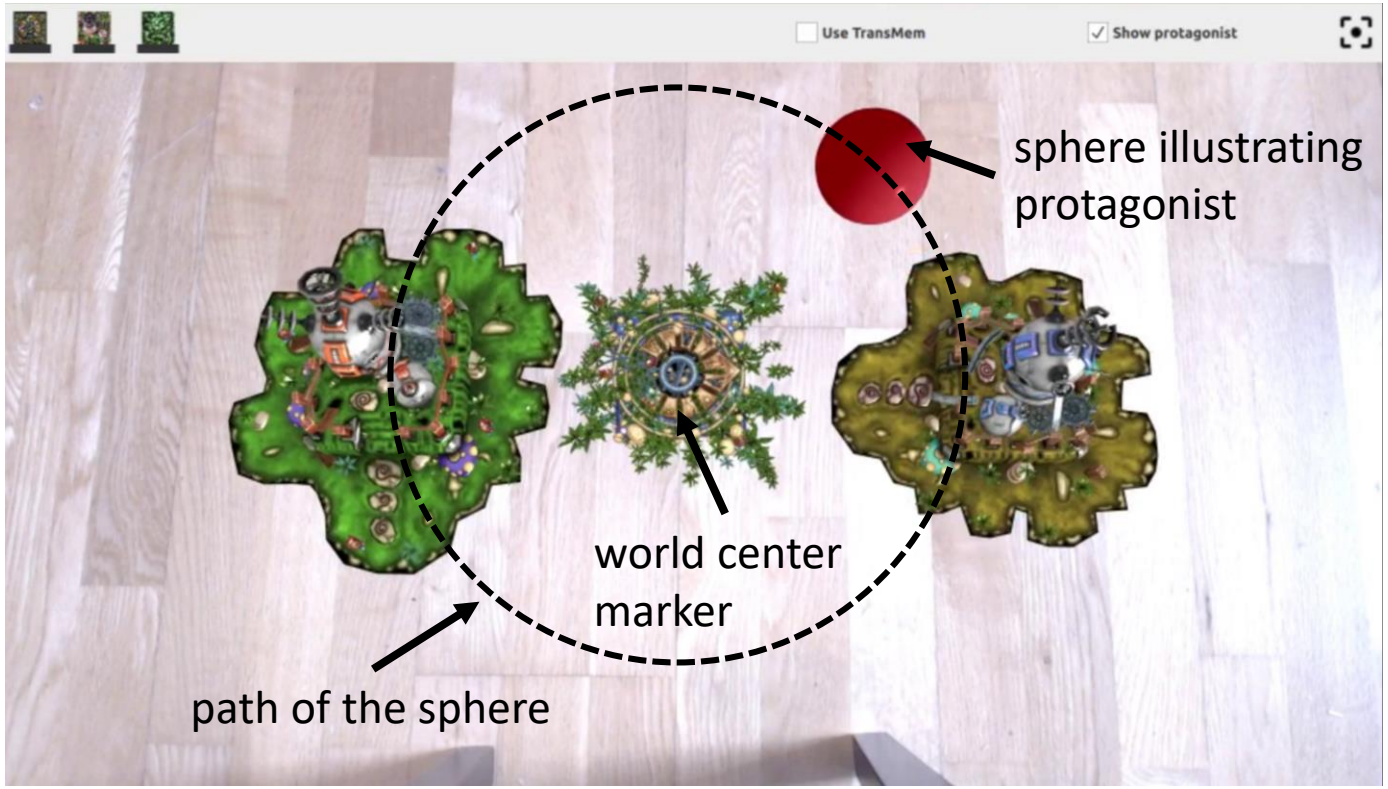
Calculate $T^{t_q}_{f_3 \leftarrow f_2} = T^{t_q}_{f_3 \leftarrow f_1} \cdot T^{t_q}_{f_1 \leftarrow f_2}$

Integration of *TransMem* into a marker model



Marker model takes a number of markers as input and creates a new relationship between the markers as output

Demo videos with and without *TransMem*



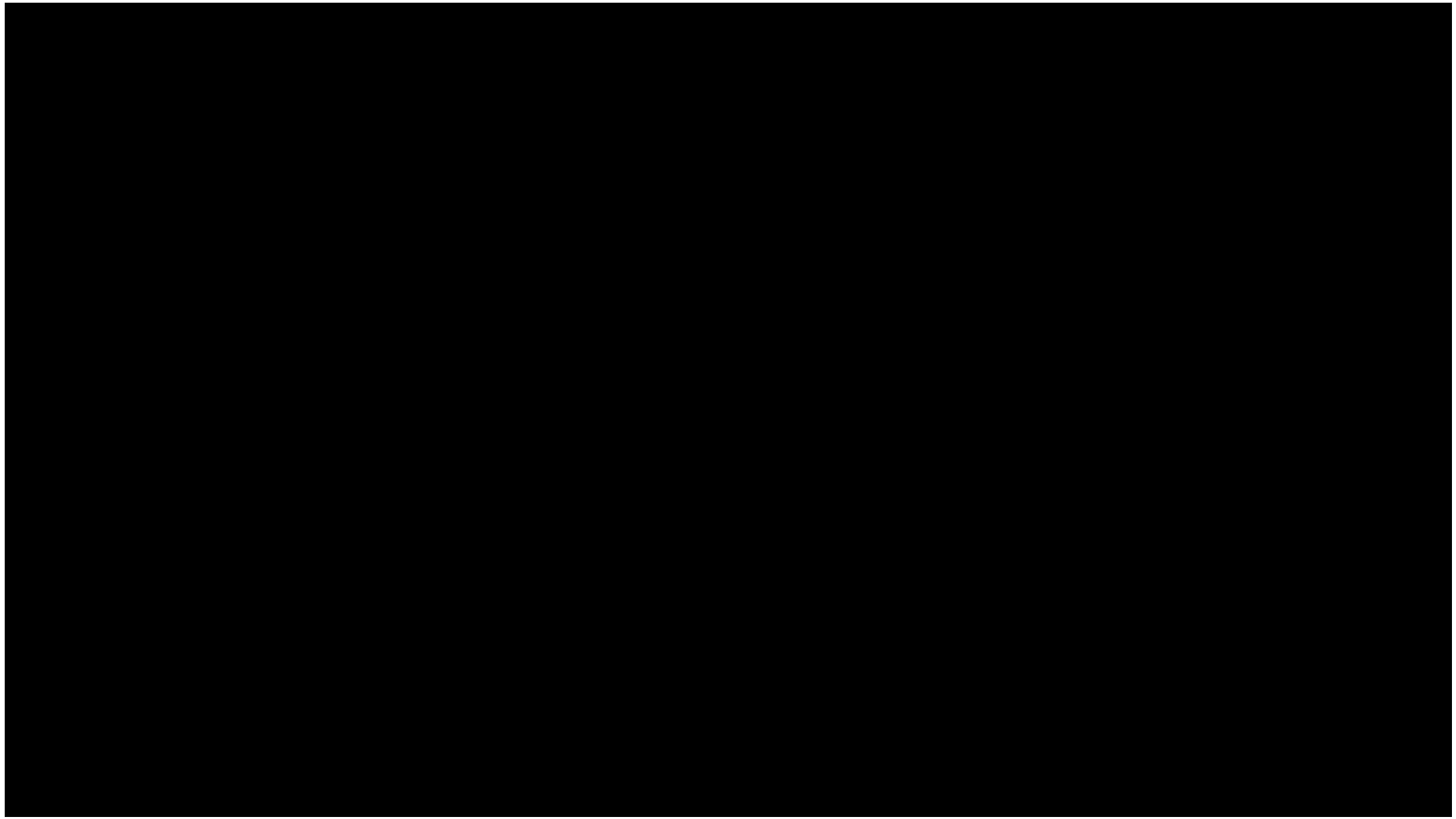
Video without use of *TransMem*

Video with use of *TransMem*

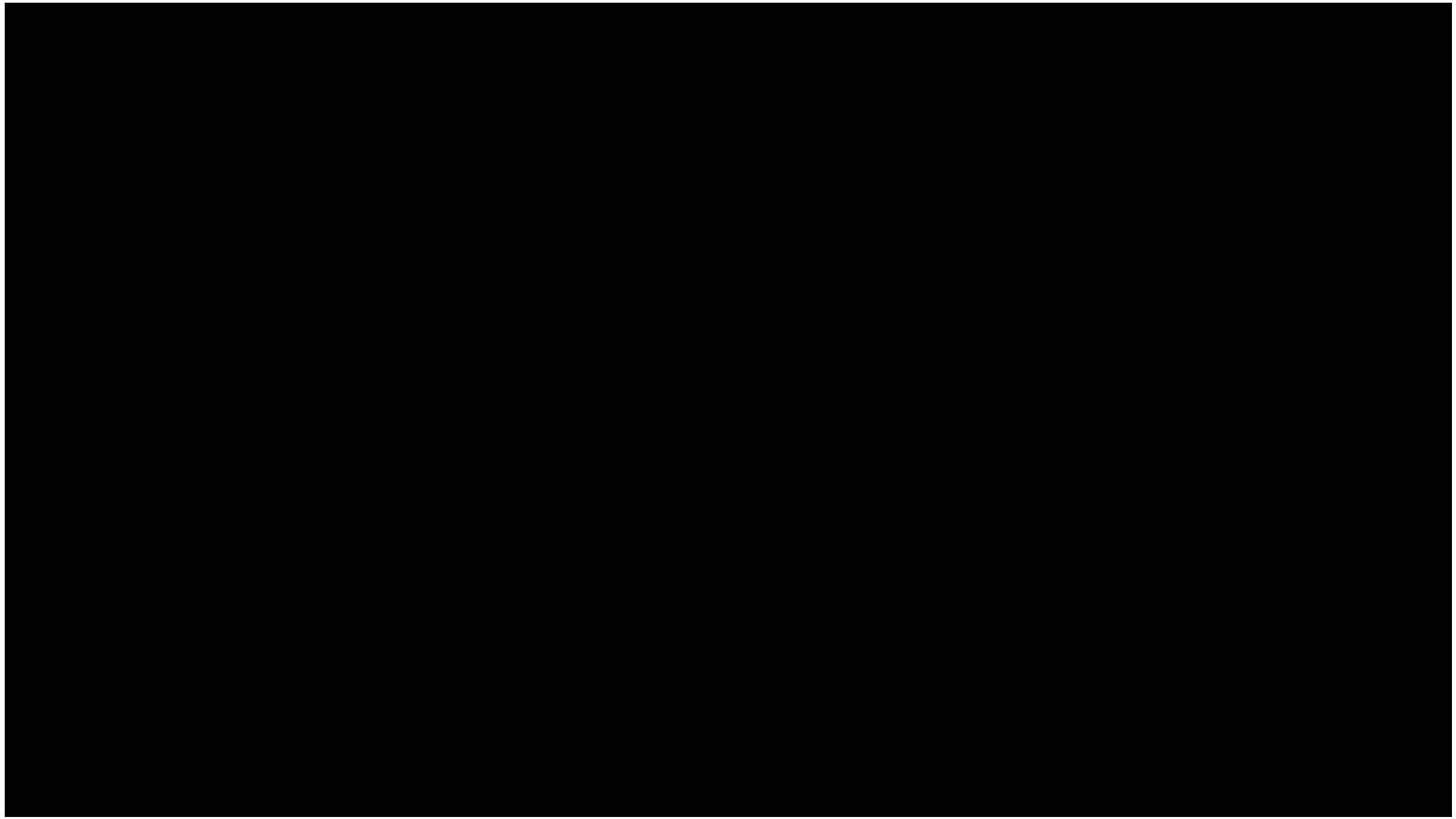
Marker in the middle is the world center marker

Red sphere rotating around world center marker illustrates protagonist

Demo video without use of *TransMem*



Demo video with use of *TransMem*



Marker evaluation

Impression during development: tracker not able to track all marker equally well

Attempt to quantify this impression

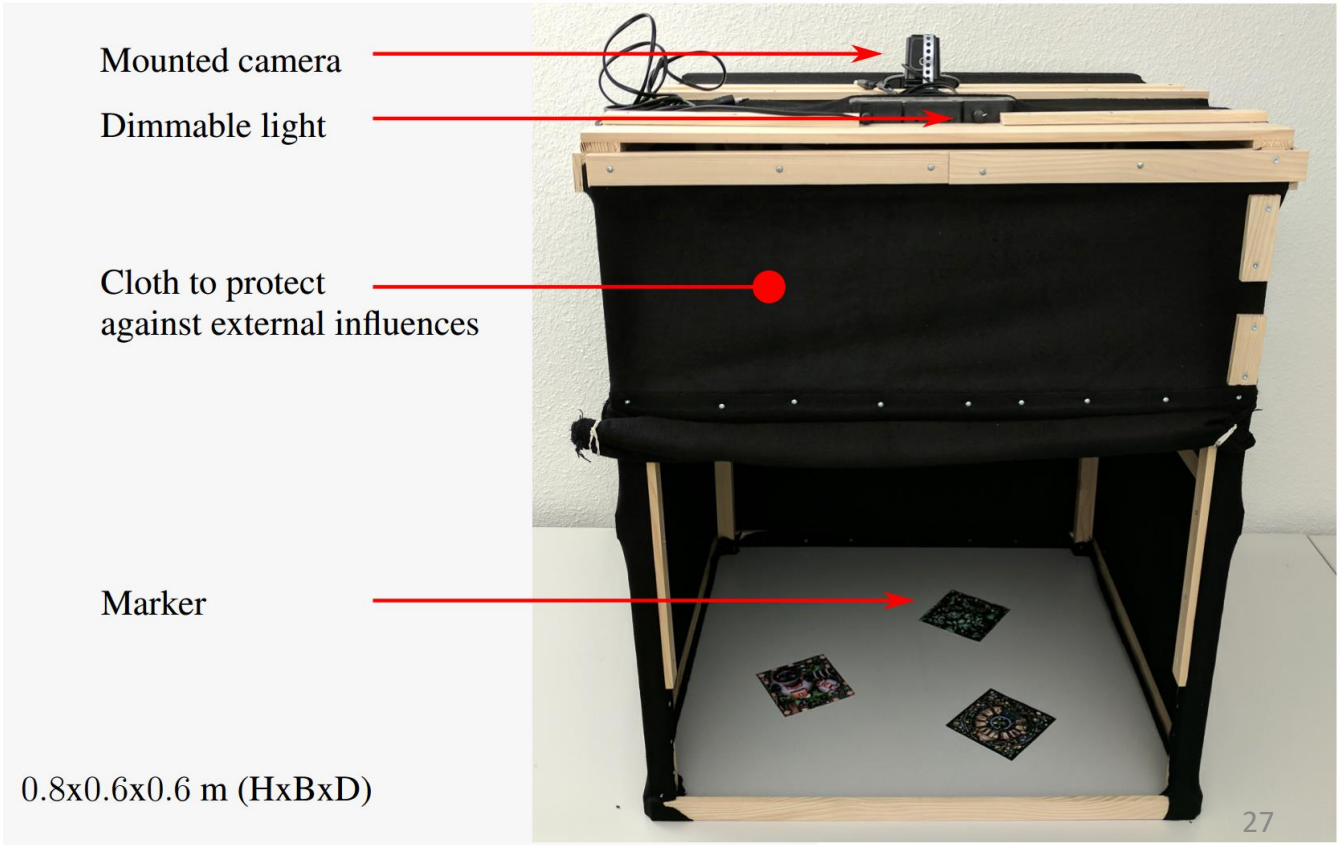
Marker evaluation

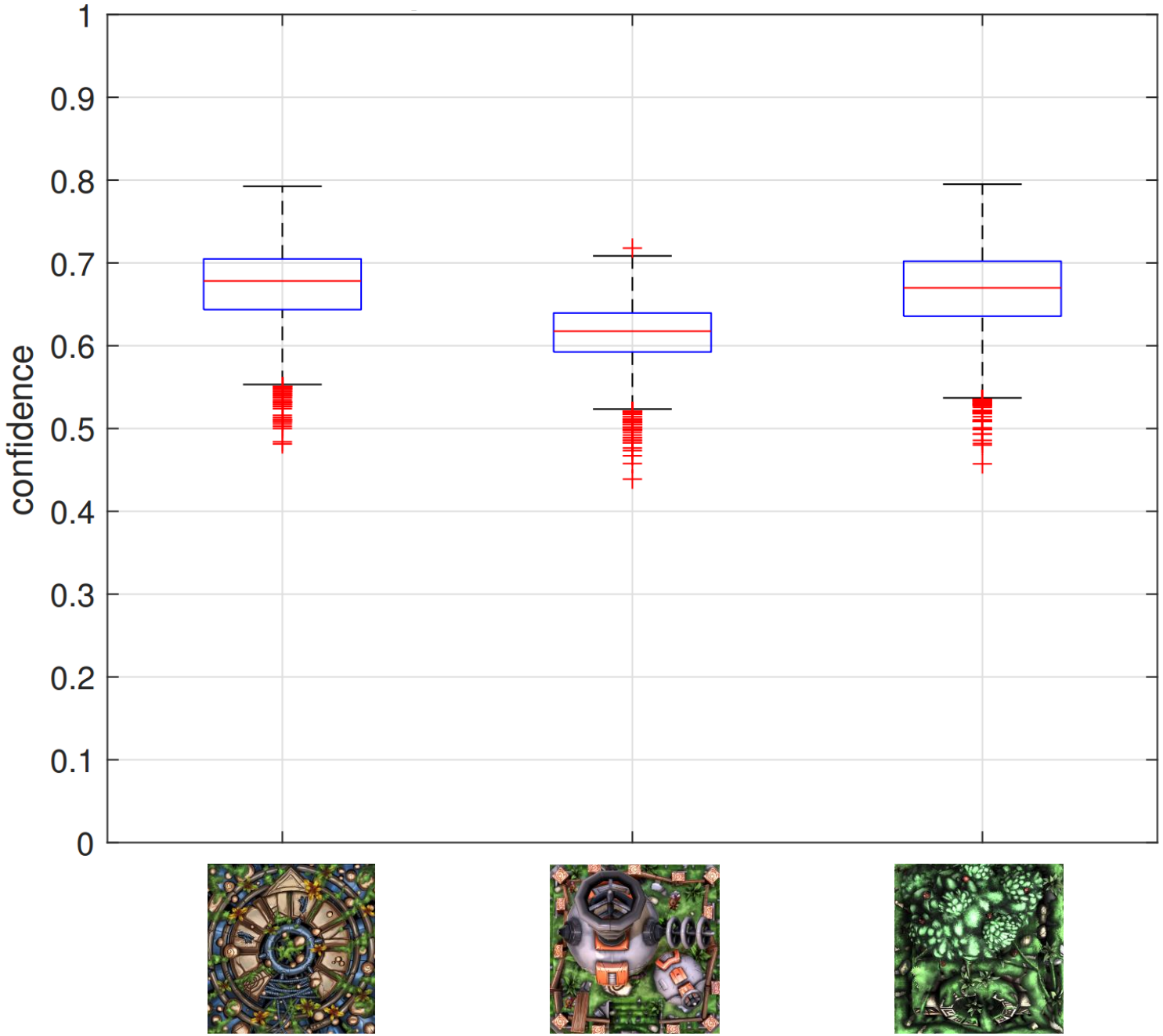
Impression during development: tracker not able to track all marker equally well

Attempt to quantify this impression

Construction of an experiment box

Allows experiments under similar conditions with all markers




confidence
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

Conclusion

Can determine position of a world center marker as long as at least one marker is visible

Can use its position as reference for the positioning of all virtual objects

Investigation into marker quality
recommended since crucial for pleasing vision system

Improve performance of tracking as a further step

Using the stored transformation to localize marker faster

Live demo

Thank you for your attention!

Image sources

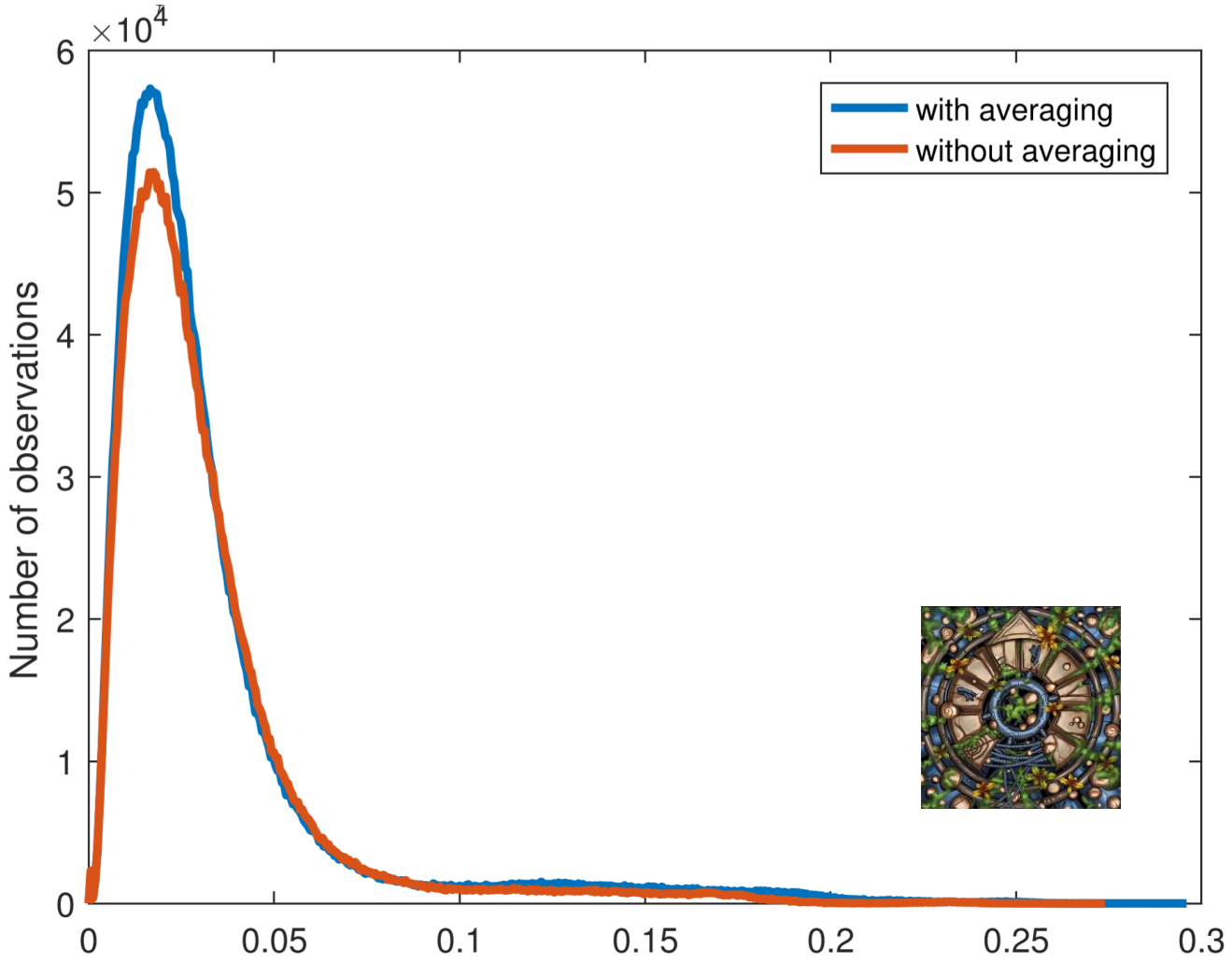
Slide 2

Thymio robot https://www.techykids.com/wp-content/uploads/2013/05/Thymio-Updated.png
Augmented world http://www.mobsya.org/ext-media/Thymio_adventure%20-%20small.pdf

Slide 9

ROS's tf library http://wiki.ros.org/tf

No positive effect of averaging



Pairwise comparison of transformations

Using introduced metric for comparison

TransMem – Querying for the best transformation

Just source and destination specified by the user

Time t_q when transformation is valid determined by *TransMem*

TransMem – Querying for the best transformation

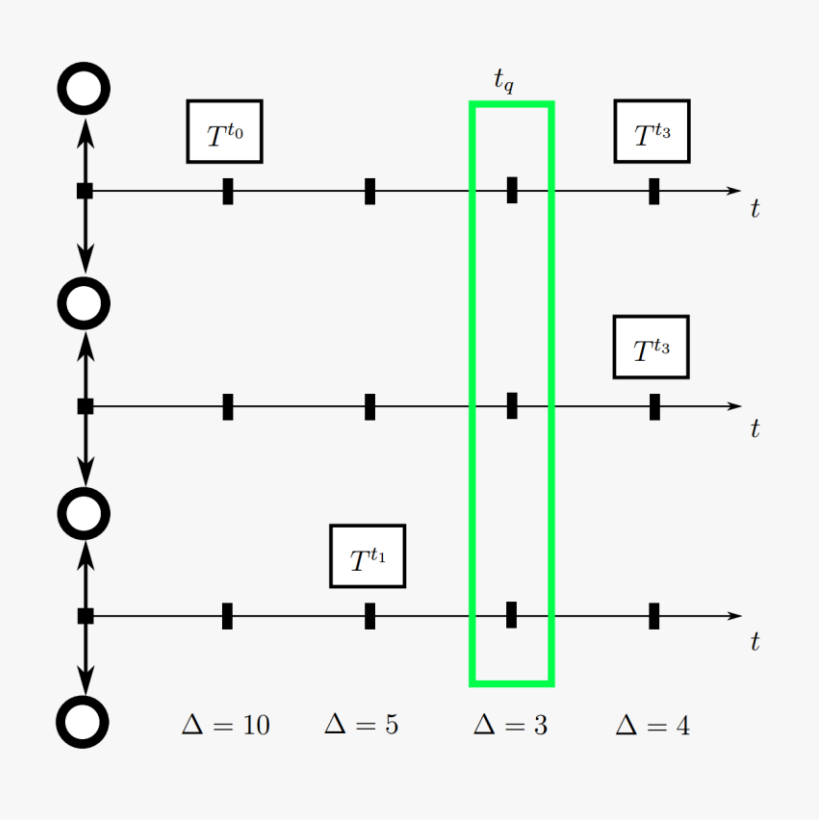
Just source and destination specified by the user

Time t_q when transformation is valid determined by *TransMem*

Also determines shortest path in first step

TransMem chooses time t_q:

> Time t_q minimizes the sum of the quadratic distances Δ to the next closest transformation on each link of the shortest path



TransMem – Querying for the best transformation

Just source and destination specified by the user

Time t_q when transformation is valid determined by *TransMem*

Also determines shortest path in first step

TransMem chooses time t_q:

> Time t_q minimizes the sum of the quadratic distances Δ to the next closest transformation on each link of the shortest path

Remaining steps similar to previous presented "Querying for a transformation"