UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

IAS-Lab

Intelligent Autonomous Systems Laboratory

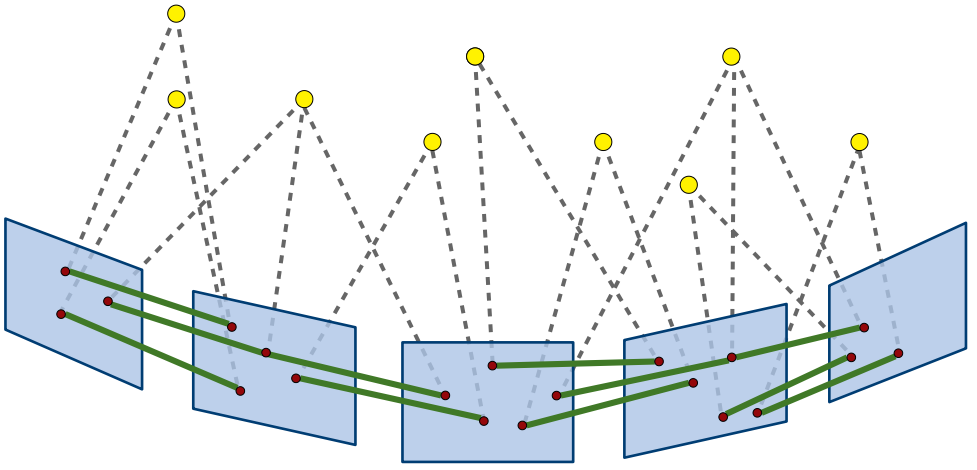
3D Data Processing

Lab 2: Structure from Motion (notes)

Alberto Pretto

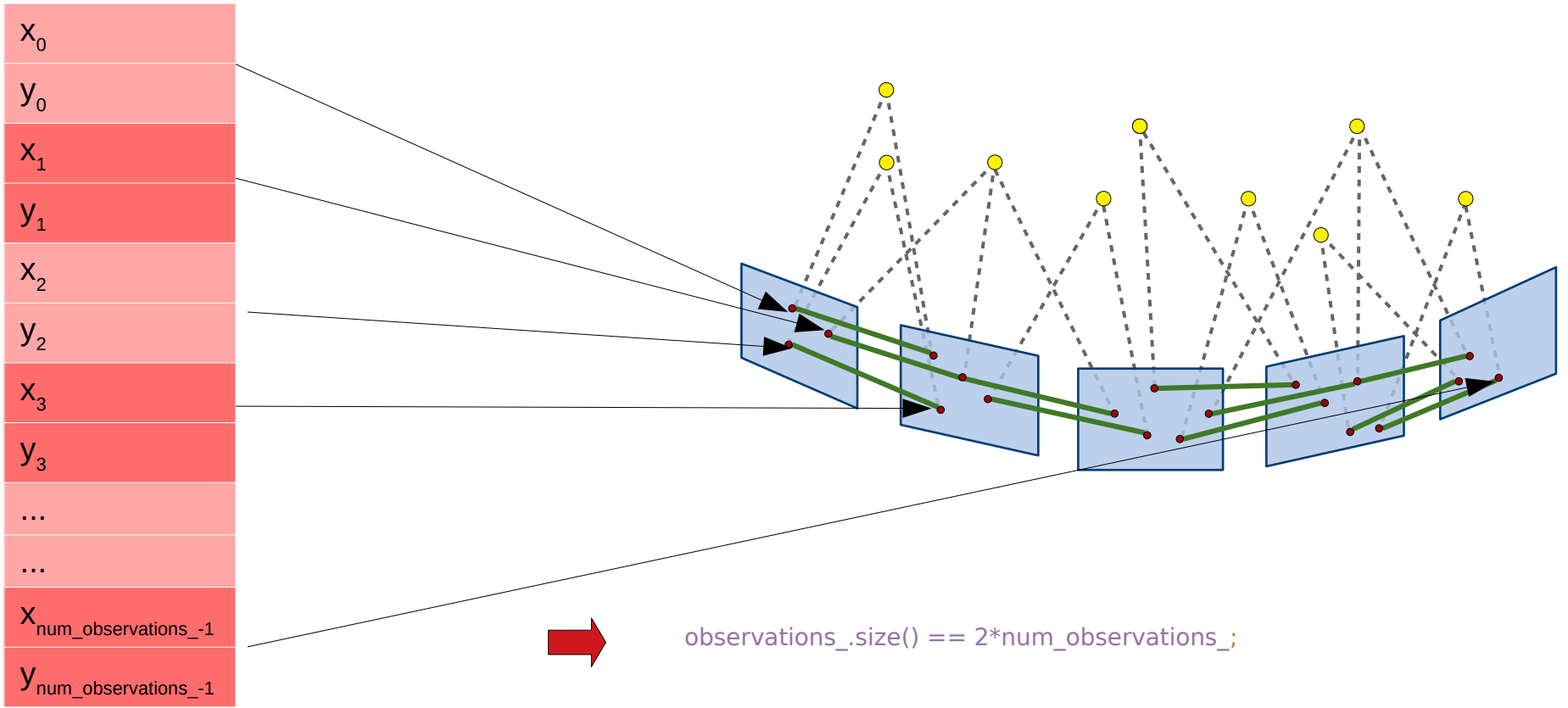
Scene Graph

```
int num_cam_poses_ = 5;

int num_points_ = 9;

int num_observations_ = 20;

// 6 * num_cam_poses_ + 3 * num_points_
int num_parameters_ = 90;
```



Observations

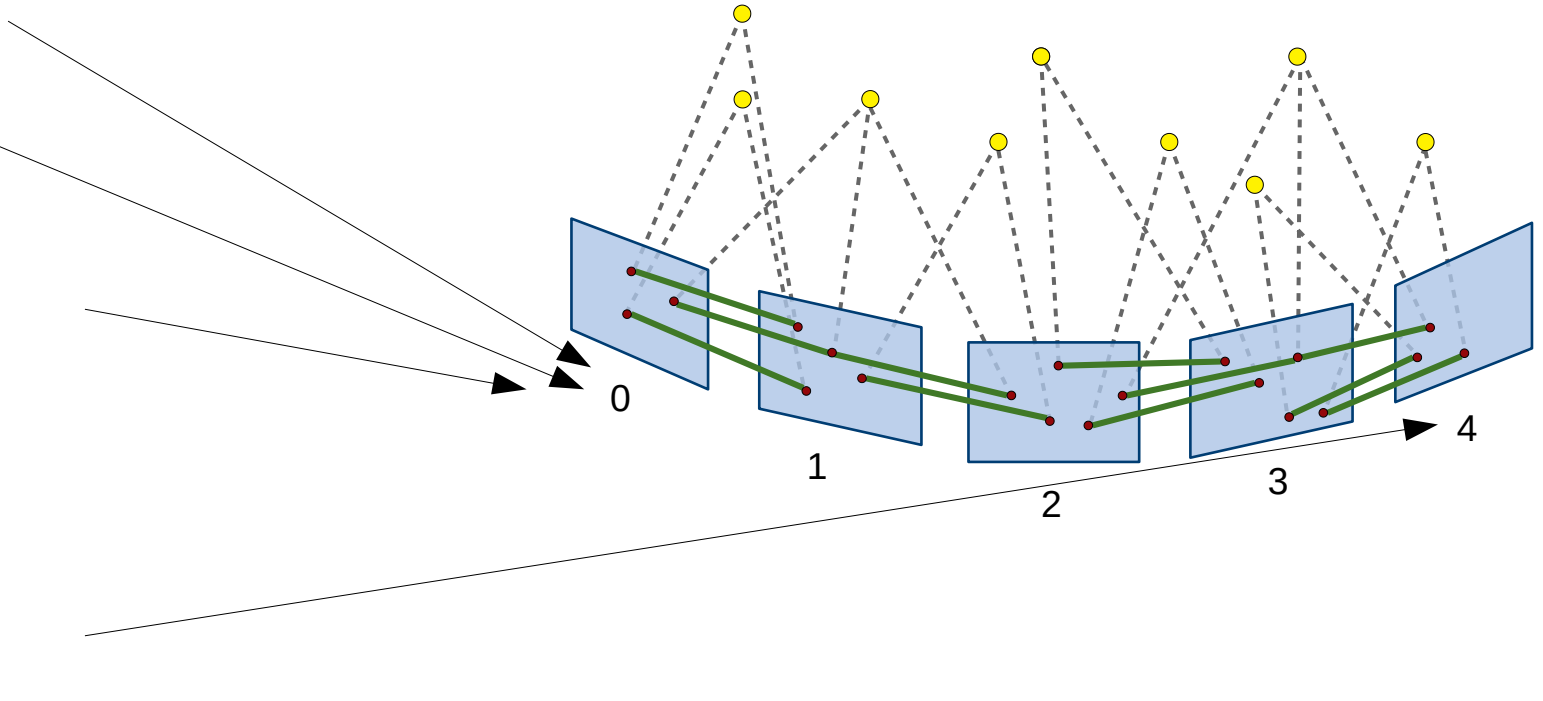
std::vector<double> observations_; // Vector of observations, i.e. 2D point projections in all images of the observed 3D points.



Camera positions

```
std::vector<int> cam_pose_index_;
```

Index of the camera that made observation 0
Index of the camera that made observation 1
Index of the camera that made observation 2
...
Index of the camera that made observation num_observations_-1



3D Points

`std::vector<int> point_index_;`

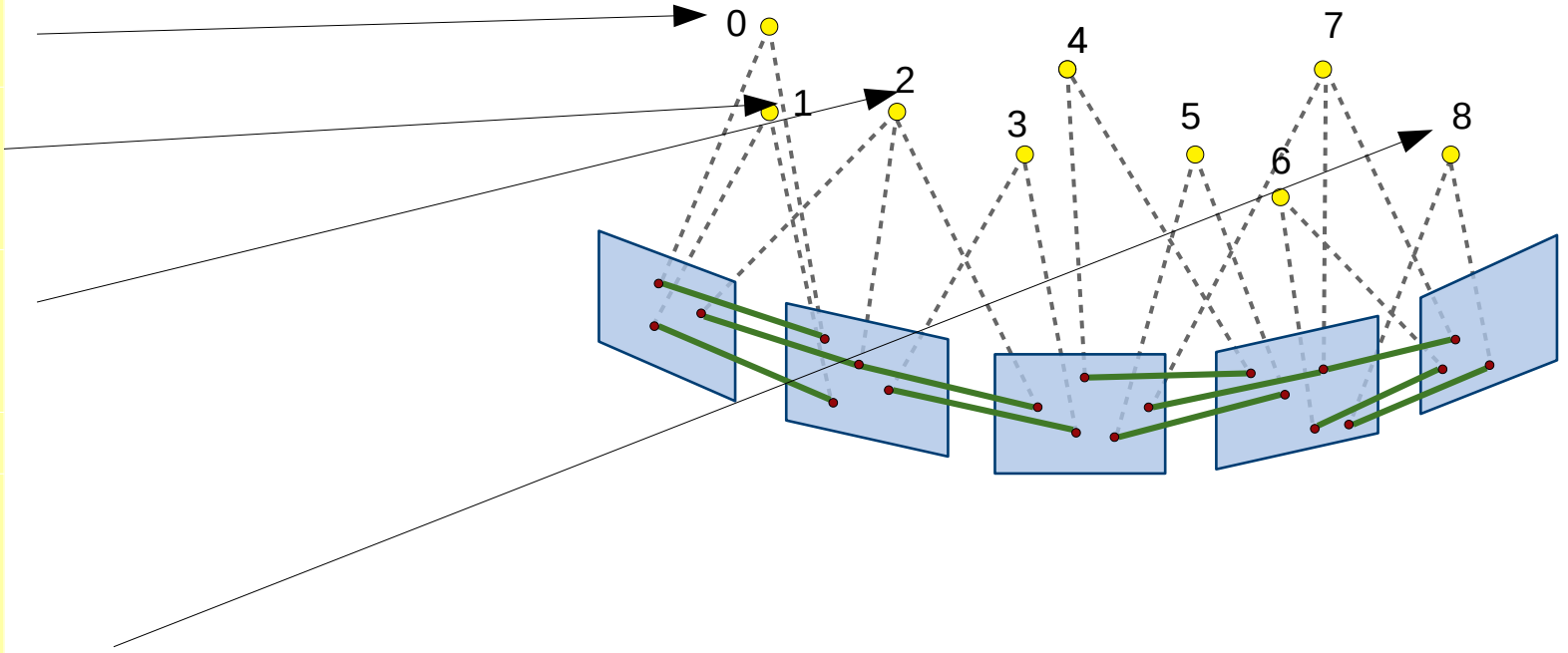
Index of the 3D point that generates sobservation 0

Index of the 3D point that generates sobservation 1

Index of the 3D point that generates sobservation 2

...

Index of the 3D point that generates sobservation num_observations_-1



Parameters

```
// Vector of all the parameters to be estimated: it is composed by num_cam_poses_ 6D blocks
// (3D axis-angle rotation and 3D translation, one for each camera view) followed by num_points_
// 3D blocks (one for each 3D point). Assume:
// n = num_cam_poses_
// m = num_points_
std::vector<double> parameters_;
```

$r_x cam_0$	$r_y cam_0$	$r_z cam_0$	$t_x cam_0$	$t_y cam_0$	$t_z cam_0$	$r_x cam_1$	$r_y cam_1$	$r_z cam_1$	$t_x cam_1$	$t_y cam_1$	$t_z cam_1$	...
...	$r_x cam_{n-1}$	$r_y cam_{n-1}$	$r_z cam_{n-1}$	$t_x cam_{n-1}$	$t_y cam_{n-1}$	$t_z cam_{n-1}$	...	x_p_0	y_p_0	z_p_0	...	...
...	...	...	...	...	...	...	...	...	...	x_p_{m-1}	y_p_{m-1}	z_p_{m-1}

Observation map

```
std::vector< std::map<int,int> >
  cam_observation_;

// For each camera index, a point – observation map

auto map = cam_observation_[ ▭ ]

// For each point index, the corresponding observation index for the corresponding camera (if any)

int obs_idx = map[ ● ]

// To check if a camera observed a point:

If( map.find(●) != map.end() ) {  .... }
```