

Coping with occlusions in visual tracking of multiple objects

Vincenzo Lippiello, Bruno Siciliano, and Luigi Villani

PRISMA Lab – Dipartimento di Informatica e Sistemistica

Università degli Studi di Napoli Federico II

Via Claudio 21, 80125 Napoli, Italy

`{vincenzo.lippiello,bruno.siciliano,luigi.villani}@unina.it`

Abstract

The problem of visual tracking of multiple objects is considered in this paper. Special emphasis is devoted to the case when two or more objects overlap with respect to the visual system causing occlusion. The algorithm is based on the Kalman filtering and Binary Space Partition tree representations of the objects geometry. The real-time implementation of the algorithm is experimentally tested for the case of visual tracking of two objects using two cameras.

1 Introduction

The performance of a robotic system during the execution of tasks such as grasping, assembling, mating mechanical parts, etc., can be significantly enhanced if visual measurements are available. In fact, visual information can be exploited for planning the end-effector trajectory and, in some cases, can be directly used for visual feedback control [1].

A typical problem in robotic vision is the real-time estimation of the position and orientation of moving objects of known geometry [2]. To this purpose, the visual system is in charge to measure some features of the objects, like the position of suitable points. Estimation accuracy grows when more cameras are used to guarantee different objects views and capture a larger number of feature points.

During motion, some feature points may be occluded to the visual system by the object itself (self occlusion) or may exit from the visual space, especially when a fixed-camera system is used. Moreover, during the execution of tasks like grasping or assembling, the feature points of an object may be occluded by another object (mutual occlusion). It should be no surprise that coping with occlusions is a critical issue for real-time visual tracking algorithms.

In this paper, the extended Kalman filter is adopted to estimate the pose of moving objects using a fixed stereo camera system. The systematic procedure

presented in [3] is used to compute the pose of the objects, combined to an efficient method of selection of the visible feature points [4]. This work is mainly focused on the selection procedure for multiple objects, based on Binary Space Partition (BSP) trees technique for real-time rendering of CAD models [5]. In detail, the prediction of the objects pose provided by Kalman filters is used to build a BSP tree of the whole system and to guide a visit of the tree which allows identifying all the feature points visible at the next sample time. Differently from other algorithms (see [6] and references therein), this method allows recognizing all the points of the objects which are hidden to the camera or occluded by some other objects or obstacles of known geometry. Two different versions of the algorithm are presented: a simplified version that can be used for the case of objects with non interposing parts and a general version to be used for objects with interposing parts, e.g., a gripper grasping a tool. Complex situations can be handled in real time thanks to the computational low cost of BSP tree visit algorithms, whose complexity grows linearly with the number of feature points.

Particular emphasis is devoted to the presentation of experimental tests realized on a system of two cameras, for the case of two objects. The results confirm the effectiveness of the proposed approach.

2 Kalman filter

The position and orientation of a moving object can be estimated via an extended Kalman filter, using a system of n video cameras fixed in the workspace. The pin-hole model is adopted to relate the position of a point in the 3-D space to the position of the corresponding projection on the image plane of the cameras (see [3] for the details).

The position and orientation of the object with respect to a base frame is specified by defining a frame $O_o-x_o y_o z_o$ attached to the object and considering the coordinate vector of the origin $\mathbf{o}_o = [x_o \ y_o \ z_o]^T$ and the rotation matrix $\mathbf{R}_o(\varphi_o)$,

where $\boldsymbol{\varphi}_o = [\phi_o \ \alpha_o \ \psi_o]^T$ is the vector of the Roll, Pitch and Yaw angles. The components of the vectors $\mathbf{o}_o$ and $\boldsymbol{\varphi}_o$ are the six quantities to be estimated.

Assuming that a CAD geometric model of the object is available, m feature points are considered. It can be shown (see [3]) that the vectors $\mathbf{o}_o$ and $\boldsymbol{\varphi}_o$ can be computed by solving a system of $2mn$ nonlinear equations depending on the measurements of the position of the m feature points in the image planes of the n cameras. To derive these equations, the camera calibration parameters are assumed to be known, e.g., they can be measured via a suitable calibration procedure.

An implicit solution can be computed using the extended Kalman filter in discrete time, based on a suitable model of the object motion. The recursive form of the equations for the problem at issue is reported in [3].

3 Feature points selection

The accuracy of the estimate provided by the Kalman filter depends on the number of the available feature points. Inclusion of extra points improves the estimation accuracy but increases the computational cost. It has been shown that a number of five or six feature points, if properly chosen, may represent a good trade-off [7]; to this purpose, automatic selection algorithms have been developed [6]. It should be pointed out, however, that the complexity of the selection algorithms grows at factorial rate. Hence, in case of objects with a large number of feature points, it is crucial to perform a pre-selection of the points. In [4], a computationally efficient pre-selection method is presented, which allows identifying all the points that are visible to the camera at a given sample time. This method, based on Binary Space Partition (BSP) trees, exhibits a computational complexity which grows linearly with the number of object feature points.

The pre-selection algorithm does not guarantee that all the visible points are “well” localizable, i.e., their positions can be effectively measured with a given accuracy. Moreover, the number of the well localizable feature points may be larger than the optimal number of points ensuring the best pose estimation accuracy.

To this purpose, a windowing test is adopted first to select the projections of the feature points that can be well localized. In particular, only the points that can be centered into suitable rectangular windows of the image plane are considered for the next step of selection, while the points that are out of the field of view of the camera as well as the points that are too close each other or to the boundaries of the image

plane, are discarded.

After the pre-selection and the windowing test, an optimal set of feature points is identified by using an optimal cost function based on a combination of suitable quality indexes. These must be able to provide accuracy, robustness and to minimize the oscillations in the pose estimation variables. To achieve this goal it is necessary to ensure an optimal spatial distribution of the projections of the feature points on the image plane and to avoid chattering events between different optimal subsets of feature points chosen during the object motion. Moreover, in order to exploit the potentialities of a multi-camera system, it is important to achieve an optimal distribution of the feature points among the different cameras.

The quality indexes used to evaluate the optimality of a set Γ of feature points are:

- Spatial distribution index:

$$Q_{si} = \frac{1}{q_i} \sum_{k=1}^{q_i} \min_{j \in \{1, \dots, q_i\} \atop j \neq k} \|\mathbf{p}_j - \mathbf{p}_k\|,$$

where q_i is a subset of selected points for the i -th camera, $i = 1, 2$.

- Angular distribution index:

$$Q_{ai} = 1 - \sum_{k=1}^{q_i} \left| \frac{\alpha_k}{2\pi} - \frac{1}{q_i} \right|,$$

where q_i is a subset of selected points for the i -th camera, $i = 1, 2$, α_k is the angle between the vector $\mathbf{p}_{k+1} - \mathbf{p}_{Ci}$ and the vector $\mathbf{p}_k - \mathbf{p}_{Ci}$, being $\mathbf{p}_{Ci}$ the central gravity point of the whole subset of feature points. The q_i points of the subset are considered in a counter-clockwise ordered sequence with respect to $\mathbf{p}_{Ci}$, with $\mathbf{p}_{q_i+1} = \mathbf{p}_1$.

- Anti-chattering index:

$$Q_h = \begin{cases} 1 + \epsilon & \text{if } \Gamma = \Gamma_{opt} \\ 1 & \text{otherwise} \end{cases}$$

where ϵ is a positive constant and Γ_{opt} is the optimal set of points at the previous sample time.

- Point distribution indexes:

$$Q_e = 1 + \frac{2}{q} \left(\frac{2}{q} - 1 \right) \left| q_1 - \frac{q}{2} \right|$$

$$Q_d = \frac{q_1/d_1 + q_2/d_2}{q/\min\{d_1, d_2\}},$$

where q_i is the number of points assigned to the i -th camera, and d_i is the distance of the i -th camera from the object, $i = 1, 2$. The first index ensures an equal distribution of points among the cameras; the second index takes into account the distance of the cameras from the object.

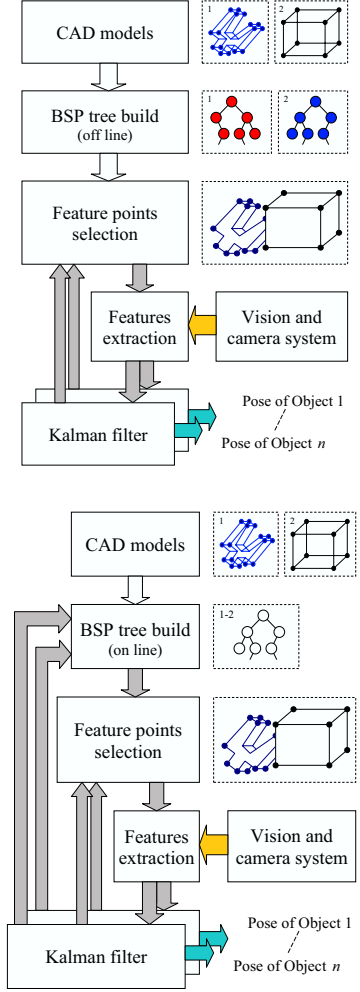


Figure 1: Functional chart of the estimation procedure. Top: non interposing objects. Bottom: interposing objects.

The above quality indexes represent only some of the possible choices for the case of two identical fixed cameras. Other examples are presented, e.g., in [6].

The cost function is chosen as

$$Q = Q_h \frac{Q_e Q_d}{q} \left(q_1 Q_{s1} Q_{a1} + q_2 Q_{s2} Q_{a2} \right)$$

and must be evaluated for all the possible combinations of the visible points on q positions. To determine the optimal set at each sample time, the initial optimal combination of points is first evaluated off-line. Then, only the combinations that modify at most one point for camera with respect to the current optimal combination are tested on line, thus achieving a considerable reduction of processing time.

4 Visual tracking of multiple objects

The visual tracking algorithm for the case of a single moving object is reported and experimentally tested in [8]. The estimation procedure assumes that a BSP

tree representation of the object is built off-line from the CAD model. A Kalman filter is used to estimate the corresponding pose with respect to the base frame at the next sample time.

This procedure can be extended to the case of multiple moving objects of known geometry. In fact, the algorithm can handle in an effective way both self occlusions and mutual occlusions. To this purpose two different situations shall be considered: the case of objects whose parts cannot be interposed, and the case of objects with interposing parts (e.g., a gripper grasping an object).

In the first case (see Fig. 1 (top)), it is assumed that a BSP tree representation of each object is built off-line from the CAD model. A different Kalman filter is required for each object to estimate the corresponding pose with respect to the base frame at the next sample time. The procedure for feature points selection has to be modified as follows:

- **Step 1:** For each camera, an ordered sequence of BSP tree representations is built, according to the estimated distance of the objects from the camera, from the farthest to the nearest.
- **Step 2:** For each camera, a visit algorithm is applied to each BSP tree of the sequence to find the set of all the feature points that are visible from the camera. In particular, for each BSP tree of the sequence, a current set of visible points is updated, by discarding the feature points of the previous objects occluded by the current object and by adding the visible feature points of the current object.
- **Step 3:** For each object, the resulting set of visible points is input to an algorithm for the selection of the set Γ_{opt} of the optimal feature points of the object for the n -camera system.
- **Step 4:** For each camera and each object, the location of the optimal feature points in the image plane at the next sample time is computed.
- **Step 5:** For each camera, a dynamic windowing algorithm is applied to select the parts of the image plane to be input to the feature extraction algorithm.

At this point, all the image windows of the optimal selected points are elaborated using a feature extraction algorithm. The computed coordinates of the points in the image plane, for each object, are input to the corresponding Kalman filter which provides the estimate of the actual object pose and the predicted pose at the next sample time used in Steps 1-2.

In the case of multiple objects with interposing parts, the above procedure may fail because the objects



Figure 2: COMAU Smart3-S robot carrying the moving object and SONY 8500CE cameras.

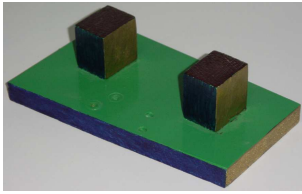


Figure 3: Object used in the second experiment.

cannot be correctly ordered with respect to the distance from the camera. This problem can be overcome, at the expense of computation time though, by adopting the solution represented in the functional chart of Fig. 1 (bottom). As before, a Kalman filter is used for each object to estimate the corresponding pose with respect to the base frame at the next sample time. Differently from the previous case, a unique BSP tree representation of all the objects is built on line, using the CAD model and the estimation provided by the Kalman filters. Hence, for each camera, the visit algorithm of the tree is applied once to find the set of all the visible points. Then, Steps 3, 4, and 5 follow.

Notice that the procedures described above can be applied also to the case of objects moving among obstacles of known geometry.

5 Experiments

The experimental set-up is composed by a PC with Pentium IV 1.7GHz processor equipped with two MATROX Genesis boards, two SONY 8500CE B/W cameras, and a COMAU Smart3-S robot. The MATROX boards are used as frame grabbers and for a partial image processing (e.g., window extraction from the image). The PC host is also used to realize the whole BSP structures management, the pre-selection algorithm, window processing, the selection algorithm and Kalman filtering.

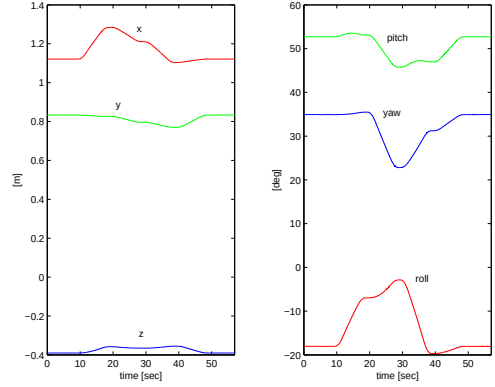


Figure 4: Trajectory of the moving object. Left: Position trajectory. Right: Orientation trajectory.

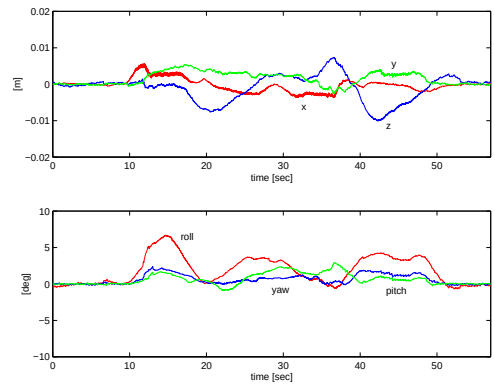


Figure 5: Estimation errors in the first experiment. Top: Position errors. Bottom: Orientation errors.

The robot is used to move one object in the visual space of the cameras (see Fig. 2), while the object of Fig. 3 is placed on a known position of the workspace; thus the position and the orientation of the moving object, with respect to the base frame of the robot, can be computed from joint position measurements via the direct kinematic equation.

The cameras (disposed as in Fig. 2) have been calibrated with respect to the base frame of the robot using the calibration procedure presented in [9], where the robot is exploited to place a calibration pattern in some known pose of the visible space of the camera. The resolution of each camera is 576×763 pixels and the nominal focal length of the lenses is 16 mm. The sampling time used for estimation is limited by the camera frame rate, which is about 26 fps. No particular illumination equipment has been used.

All the algorithms for BSP structure management, image processing and pose estimation have been implemented in ANSI C. The features points are the corners of the object, which can be extracted with high robustness in various environmental conditions. The moving object and the fixed object used in

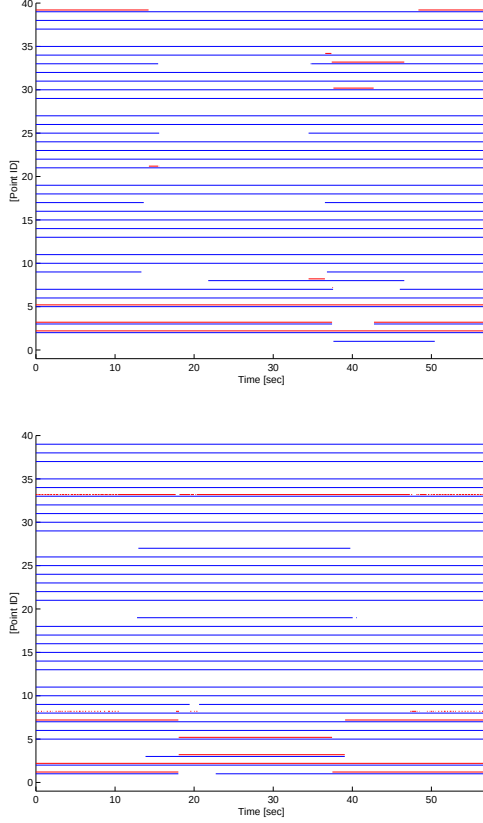


Figure 6: Visible and selected points for camera 1 (top) and camera 2 (bottom) in the first experiment.

the experiments have 40 and 26 corners respectively, which are all used as feature points.

To evaluate the capability of the proposed algorithm in coping with multi-object tracking, two experiments have been realized. In the first experiment, the case of a single moving object is considered; in the second experiment, a fixed object is added to produce mutual occlusion. The time history of the trajectory used for the moving object is represented in Fig. 4. The maximum linear velocity and angular velocity are about 3 cm/s and 3 deg/s respectively.

The time history of the estimation errors for the first experiment is shown in Fig. 5. As it was expected, all the errors are of the same order of magnitude, thanks to the use of two cameras.

In Fig. 6 the output of the selection algorithm, for the two cameras, is reported. For each of the 40 feature points, two horizontal lines are considered: a point on the bottom line indicates that the feature point was classified as visible by the pre-selection algorithm at a particular sample time; a point on the top line indicates that the visible feature point was chosen by the selection algorithm. In Fig. 7 the total number of the visible, selected and localized points are shown. Notice that $\sum_i q_i = 8$ feature points are selected at each sample time, in order to

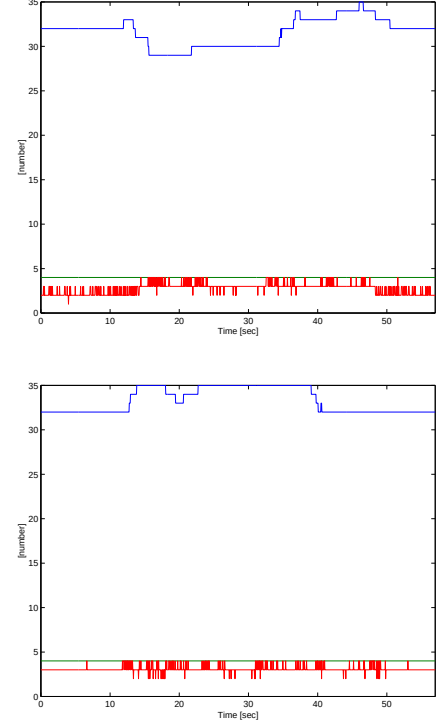


Figure 7: Number of visible (blue), selected (green) and localized (red) points for camera 1 (top) and camera 2 (bottom) in the first experiment.

guarantee at least five or six measurements in the case of fault of the extraction algorithm for some of the points. Remarkably, 4 feature points for camera are always chosen, due to the symmetric disposition of the cameras with respect to the object.

In the second experiment, the same trajectory used in the first experiment is used for the moving object, while a fixed object (see Fig. 3) is placed in the pose $(\mathbf{o}_o = [1.102 \ -0.504 \ 0.794]^T \text{m}, \varphi_o = [-85.7 \ 59.2 \ 48.7]^T \text{deg})$, which partially and temporarily occludes the moving object.

The time history of the estimation errors, reported Fig. 8, shows that the tracking accuracy of the system remains satisfactory also in the presence of occlusions. The occurrence of occlusions is evidenced in Fig. 9 where the output of the selection algorithm, for the two cameras, is reported, as well as in Fig. 10, where the total number of the visible, selected and localized points is shown. Notice that the selected feature points are not always equally distributed among the two cameras because, when occlusions occur, the disposition and the number of the points available for each camera is not always symmetric.

6 Conclusion

The problem of visual tracking of multiple objects has been considered in this paper. Special attention has been devoted to the case when two or more ob-

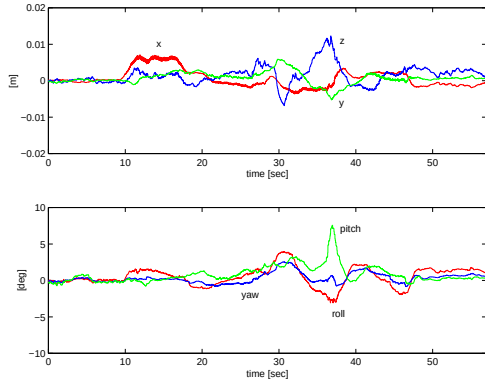


Figure 8: Estimation errors in the second experiment. Top: Position errors. Bottom: Orientation errors.

jects overlap with respect to the visual system causing occlusion. Experiments have been presented for the case of two cameras and two objects.

Acknowledgments. This work was supported by MIUR and ASI.

References

- [1] S. Hutchinson, G.D. Hager, and P.I. Corke, “A tutorial on visual servo control,” *IEEE Trans. on Robotics and Automation*, vol. 12, pp. 651–670, 1996.
- [2] J.W. Wilson, “Relative end-effector control using Cartesian position based visual servoing,” *IEEE Trans. on Robotics and Automation*, vol. 12, pp. 684–696, 1996.
- [3] V. Lippiello, B. Siciliano, and L. Villani, “Position and orientation estimation based on Kalman filtering of stereo images,” *2001 IEEE Int. Conf. on Control Applications*, pp. 702–707, 2001.
- [4] V. Lippiello, B. Siciliano, and L. Villani, “Objects motion estimation via BSP tree modeling and Kalman filtering of stereo images,” *2002 IEEE Int. Conf. on Robotics and Automation*, pp. 2968–2973, 2002.
- [5] T.W. Drummond and R. Cipolla, “Real-time tracking of complex structures with on-line camera calibration,” *British Machine Vision Conf.*, pp. 574–583, 1999.
- [6] F. Janabi-Sharifi and W.J. Wilson, “Automatic selection of image features for visual servoing,” *IEEE Trans. on Robotics and Automation*, vol. 13, pp. 890–903, 1997.
- [7] J. Wang and J.W. Wilson, “3D relative position and orientation estimation using Kalman filter for robot control,” *1992 IEEE Int. Conf. on Robotics and Automation*, pp. 2638–2645, 1992.
- [8] V. Lippiello, B. Siciliano, and L. Villani, “Robust visual tracking using a fixed multi-camera system,” *2003 IEEE Int. Conf. on Robotics and Automation*, 2003.
- [9] J. Weng, P. Cohen, and M. Herniou, “Camera calibration with distortion models and accuracy evaluation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 965–980, 1992.

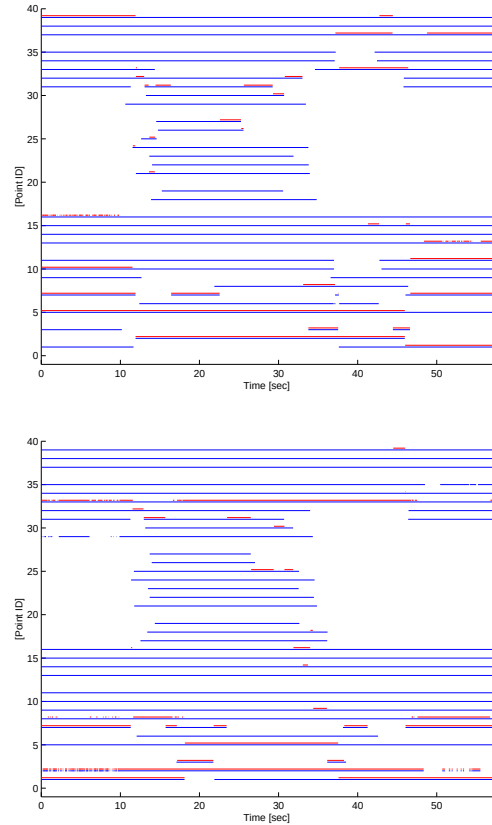


Figure 9: Visible and selected points for camera 1 (top) and camera 2 (bottom) in the second experiment.

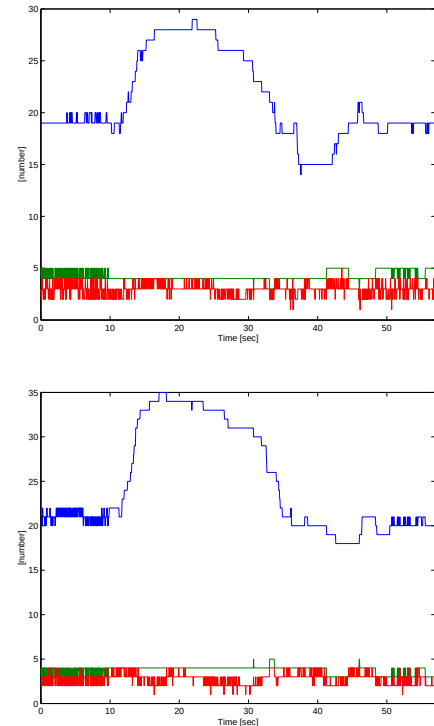


Figure 10: Number of visible (blue), selected (green) and localized (red) points for camera 1 (top) and camera 2 (bottom) in the second experiment.