2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Safety for a robot arm moving amidst humans by using panoramic vision

Enric Cervera, Nicolas Garcia-Aracil, Ester Martínez, Leo Nomdedeu, and Angel P. del Pobil

Abstract— **This paper describes how the use of panoramic cameras can dramatically simplify safety issues for a robot arm moving in close proximity to human beings, since they can simultaneously observe a 360° field of view. We present in this context an approach to visual servoing in which both the manipulator as well as any other moving object are tracked. Reliability and robustness are enhanced by adaptative background modelling and global illumination change detection.**

I. INTRODUCTION

In recent years, research in robotics has focused on developing autonomous robot systems capable of performing some of our daily tasks. As the robot is continuously interacting with its environment, its dependability is based on adequately performing its tasks and guaranteeing the safety of all elements around it, mainly when they are human beings. The last issue is specially necessary when the robot is moving in unknown, dynamic environments and it is performing manipulation tasks (such as picking up and carrying items or opening and closing doors) because the system is larger and most sophisticated and the damage caused to the objects around it can be considerable.

Although a variety of sensors have been developed to prevent or detect collisions with robot manipulators such as cages, laser fencing or visual acoustic signals, they are not suitable for one that operates in human-populated, everyday environments. An alternative device is a fisheye camera. It provides panoramic vision whereby rich information is obtained, which has proven to be useful for autonomous robot navigation and surveillance. We propose a mobile manipulator which incorporates a visual system composed of N (> 1) fisheye cameras mounted on the robot base, pointing upwards to the ceiling, to guarantee the safety in its whole workspace. Figs. 1 depicts our experimental setup, which consists of a mobile Nomadic XR4000 base, a Mitsubishi PA10 arm, and two fisheye cameras (Foculus FO124TC IEEE1394 cameras, with FUJINON-YV2.2X1.4A-2 lenses which provide 185°field of view).

The complete workspace can be covered with only two cameras placed at both sides of the manipulator. Nevertheless, additional cameras allow the straightforward recovery of 3D information, increasing the robustness and thus the dependability of the setup. The software system is scalable to any number of cameras, since a distributed, agent-based approach is used [1].

E. Cervera, E. Martinez, L. Nomdedeu and A. P. del Pobil are with the Robotic Intelligence Lab, Jaume-I University, Spain. N. Garcia-Aracil is with the Virtual Reality and Robotics Lab, Miguel-Hernandez University, Spain. Corresponding author: ecervera@icc.uji.es

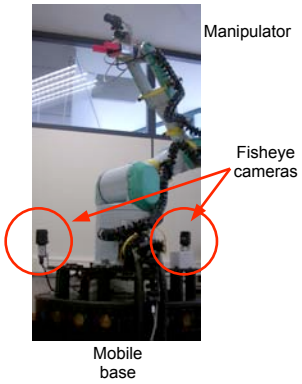

Fig. 1. Experimental setup: external view of the arm and cameras.

We present two different visual applications to cope with the safety problem. On the one hand, a visual servoing approach based on fisheye cameras which can track the motion of the manipulator. On the other hand, an adaptative background modelling combined with a global illumination change detection has been developed to track any moving object in the robot workspace and its surrounding area.

II. FISHEYE VISION FOR DEPENDABLE MANIPULATION

Visual control for central catadioptric cameras [2], [3] has been extensively studied and many applications have been demonstrated (e.g. obstacle recognition [4], formation control [5], occupancy grids [6]), but little or no attention has been devoted to fisheye cameras. The vision acquired by a fisheye lens is somehow similar to human vision from the point of view of resolution distribution. However, fisheye lenses introduce radial distortion which is difficult to remove, and they lack a single view point, having instead a locus of viewpoints [7].

Nevertheless, a visual servoing interaction matrix can be derived for a fisheye projection. Using simple projection equations, an interaction matrix can be obtained, which takes into account image data, and the distance to the object.

Tracking is simplified, since the background consists mainly of the ceiling of rooms (homogeneous colors, semi-structured, no obstacles, few moving objects). People tracking is limited to the border of the fisheye image (efficient). Motion detection is first used to track people approaching to the robot.

The goal of the system is to augment the dependability of the visually-controlled arm by redundantly covering the whole workspace of the manipulator with several fisheye

cameras. In the next section, we will develop the theoretical fisheye control model needed in such setup for visually guided tasks.

III. FISHEYE VISUAL CONTROL MODEL

The fisheye projection is based on the principle that, in the ideal case, the distance between an image point and the principal point is linearly dependent on the angle of incidence of the ray of the corresponding object point.

For a 180 degree fisheye lens, an object point's ray with an angle of incidence of 90 degrees is projected onto the outer border of the circular fisheye image [8]. The relation between the angle of incidence and the resulting distance of the image point from the principal point is constant for the whole image. Consequently, the following ratio equation holds for the fisheye projection:

$$\frac{\alpha}{r} = \frac{\pi}{2R} \tag{1}$$

where α is the angle of incidence, $r = \sqrt{u^2 + v^2}$ is the distance from the image point to the optical axis, R is the image radius, and (u, v) are the image coordinates.

For the sake of simplicity, we will assume that there is no distortion and the principal point is centered. The interaction matrix can then be fully derived (2). We use the method proposed by Espiau et al. [9]: first the motion equations of a 3D point are transformed to its spherical coordinates. Then, a simple relationship between the spherical coordinates and the fisheye image projection is used to compute the interaction matrix of an image point.

The resulting matrix depends on the image coordinates (u, v), the distance from the image point to the optical axis r and the angle of incidence α computed in (1). As in standard pinhole visual servoing, 3D information is needed. In this case, the distance d to the 3D point. Not surprisingly, this information is only needed in the translational part of the matrix, not in the rotational part.

A simpler formulation is obtained if the image points are expressed in polar coordinates (3). Instead of (u, v), the feature vector consists of (r, θ) for each point. Not only the computations are simplified, but the behavior during the control task is different. In visual servoing, the features are expected to go through a straight trajectory from their current to desired values. If image Cartesian coordinates are used, the point will move along a straight line. If image polar coordinates are used, the point describes an arc. The behavior of the 3D trajectory of the end-effector is different too, as will be shown in the experiments.

The approach can be easily extended to multiple cameras, by stacking the interaction matrix of each camera multiplied by the appropriate screw transformation matrix [10].

The dependability of the resulting visual control is increased with respect to other approaches [11] due to the following reasons:

- Few calibration parameters: compared to the pinhole model, fisheye cameras are simpler to calibrate. Distortion is included in the projection model.
- Wide field of view: the image covers the whole workspace of the manipulator, thus the features will always remain inside. As a minor drawback to take into account, the resolution decreases in the border of the image, thus feature detection is harder yet feasible.
- Camera redundancy: multiple cameras overcome the problem of occlusion of the arm and other objects in the scene. Feature redundancy also adds robustness to the control law.

IV. SURVEILLANCE APPLICATION

As in most of surveillance applications, the visual system we present for tracking humans or other objects can perform two main tasks:

- moving object recognition and segmentation from the surrounding environment
- obtaining information about the proximity of the detected objects to the robot system. This information is necessary to make correct decisions about robot movements in order to achieve the robot goal without causing any damage

Research in human and object recognition and segmentation has taken a number of forms. However, it is important to pay attention to a key issue in vision applications in which dependability is based on the ability of adapting to several changes in the system environment. So, it must cope with two different kinds of changes:

- minor dynamic factors, such as, for example, blinking of computer screens, shadows, mirror images on the glass windows, curtains movement or waving trees, as well as changes induced by camera motion, sensor noise, non-uniform attenuation or atmospheric absorption, among other things
- sudden changes in illumination such as switching on/off a light or opening/closing a window

Different approaches have tried to adapt to these dynamic factors, but they fail when a sudden change in illumination occurs or when they are building an initial background model if someone or something appears in the scene [12], [13]. The novel algorithm we propose here works at two levels to overcome these problems:

- pixel level, in which an adaptative background model is used to classify pixels as foreground or background. This model associates a statistical distribution defined by its mean color value and its variance, to each pixel of the image
- frame level, whereby the raw classification based on the background model is improved and the model is adapted when a global change in illumination occurs. Moreover, it allows to obtain the initial background model without any restrictions

Thus, when a human or another moving object enters the room where the robot is, it is detected by means of the background model at pixel level. It is possible because each pixel belonging to the moving object has an intensity value which does not fit to the background model. The method is

$$\mathbf{L}_{uv} = \begin{pmatrix} \frac{u^2\cos\alpha + rv^2\csc\alpha}{dr^2} & \frac{uv(\cos\alpha - r\csc\alpha)}{dr^2} & \frac{-u\sin\alpha}{dr} & \frac{uv(r\cot\alpha - 1)}{r^2} & \frac{u^2 + rv^2\cot\alpha}{r^2} & -v \\ \frac{uv(\cos\alpha - r\csc\alpha)}{dr^2} & \frac{v^2\cos\alpha + ru^2\csc\alpha}{dr^2} & \frac{-v\sin\alpha}{dr} & \frac{-v^2 - ru^2\cot\alpha}{r^2} & \frac{uv(1 - r\cot\alpha)}{r^2} & u \end{pmatrix} \quad (2)$$

$$\mathbf{L}_{r\theta} = \begin{pmatrix} \frac{\cos^2 r\cos\theta}{z} & \frac{\cos^2 r\sin\theta}{z} & -\frac{\cos r\sin r}{z} & -\sin\theta & \cos\theta & 0 \\ -\frac{\cot r\sin\theta}{z} & \frac{\cot r\cos\theta}{z} & 0 & -\cot r\cos\theta & -\cot r\sin\theta & 1 \end{pmatrix} \quad (3)$$

similar to the one presented in [14], but the constraint to be satisfied is:

$$|n_{i,j} - \mu_{i,j}| > k * \sigma_{i,j} \quad (4)$$

where $n_{i,j}$ represents the value of the pixel (i, j) in the current frame, $\mu_{i,j}$ and $\sigma_{i,j}$ are the mean and standard deviation values calculated by the background model respectively and k is a constant the value of which depends on the point distribution.

Then, the obtained binary image is refined by using a combination of substraction techniques at frame level. Moreover, two consecutive morphological operations are applied to erase isolated points or lines caused by the dynamic factors mentioned above. The next step is to update the statistical model with the values of the pixels classified as background in order to adapt it to some small changes which do not represent targets. On the other hand, a pattern of each moving object is built once each detected connected component has been labeled. This pattern allows the system to track the moving objects because it can match an object in two consecutive frames even when it suffers a partial or whole occlusion, since the system maintains information about a detected object during several frames in order to recognize it if it reappears.

It must be taken into account that the built pattern for each detected object has a different orientation depending on its position inside the scene because the raw image is a circular omnidirectional one. So, several rotations would be necessary in order to correctly match the patterns of the same object in two different frames. However, a transformation from the circular omnidirectional image to a perspective one is applied, as can be seen in Fig. 2. Therefore, all patterns have the same orientation and its comparison is easier, faster and more dependable.

Fig. 2. An omnidirectional image, its corresponding panoramic image and the pattern of the detected object

At the same time, a process for sudden illumination change detection is performed at frame level. This step is necessary because the model is based on intensity values and a change in illumination produces a variation of them. A new adaptative background model is build when an event of this type occurs, because if it was not done, the application would detect background pixel like moving objects.

V. EXPERIMENTAL RESULTS

The experiments presented in this section start with a visual servoing simulation and then human tracking results are presented in the second part.

A simulation with two cameras has been performed. The end-effector frame traverses the simulated workspace from its initial position to its destination in the opposite side. The whole displacement amounts to 2m translation and a large rotation (90°X, 180°Z). Such task cannot be handled with pinhole cameras, but it does not pose any problem to the large field of view of fisheye lenses.

Two cameras attached to the base observe the target which is made of 5 points, attached to the end-effector. The image trajectories depicted in Fig. 3 show very straight line trajectories of the image points in the fisheye view. Such trajectories are expected since the visual servoing control law imposes a feature *motion* in the negative direction of the first derivative.

The classical problem is that such control in the image space will consequently produce unpredictable behavior in the Cartesian space, i.e. the 3D trajectory of the end-effector. As a result, the robot might go out of its workspace, trying to move a joint further from its range, or colliding with an obstacle of the environment, or a person.

Polar image coordinates do not impose such a straight line trajectory of the image point (Fig. 4). Instead, both the radius and the angle of each feature *move* linearly. As a result, the image points describe arcs in the image plane. Features are not likely to go out of the field of view, though. The radius is bounded by the initial and final values, which are obviously within the image range. The change in the angle does indeed keep the features visible.

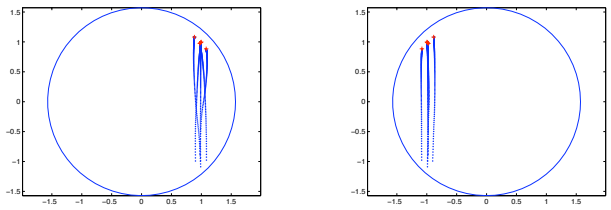

Fig. 3. (u, v) features: image trajectories on the left and right cameras.

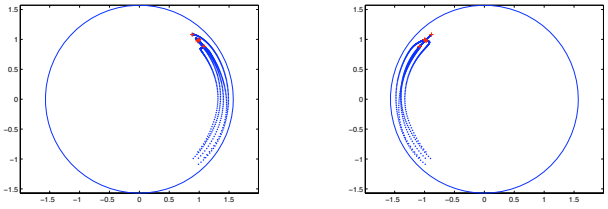

Fig. 4. (r, θ) features: image trajectories on the left and right cameras.

The choice of features has direct consequences in the behavior of the end-effector in 3D space, as depicted in (Fig. 5). In this figure, the location of the left and right cameras is represented by two Cartesian frames at $(-1, 0, 0)$ and $(1, 0, 0)$. The trajectories of the end-effector are depicted in blue for the (u, v) representation, and in red for the (r, θ) representation. Interestingly, the latter is more close to a straight line path than the former. Intuitively, if the image points are constrained to straight line trajectories in the image space, the end-effector needs to move away from the camera, since the points approach the center of the image. However, if the image points move along arcs, the end-effector does not need to perform such motion.

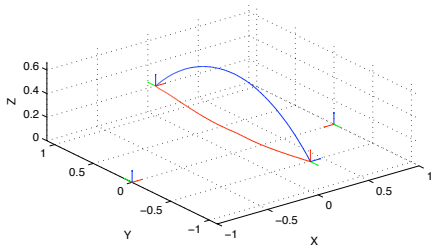

Fig. 5. End-effector trajectory: using (u, v) features in blue, using (r, θ) features in red.

Quantitatively, in the example, the maximum distance from the blue path to a straight linear path is 0.42 m. On the other hand, such distance from the red path is only 0.08 m. The risk of collision of the arm with surrounding objects or agents is dramatically decreased in the latter case.

Feature errors for (u, v) and (r, θ) representations are depicted in Figs. 6 and 7 respectively. In this particular task, the error is mostly owed to v and θ, since the initial and final u and r values are very similar. Thus, the error plots of the former features exhibit the classic exponentially decreasing curve. The latter do not exhibit the same patter at the beginning of the task, but one should take into account that the error magnitude is significantly lower, thus there are no visible consequences neither in the image space nor in the Cartesian space.

The kinematic screw is depicted in Fig. 8 for the (u, v) representation and in Fig. 9 for the (r, θ) representation. Velocities are smooth, and the coupling between translation and

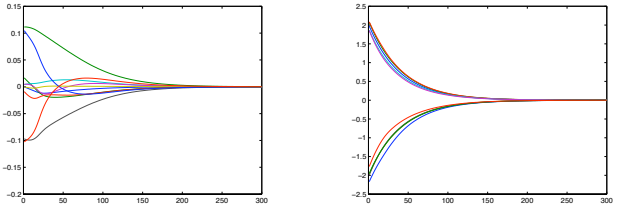

Fig. 6. (u, v) features: error.

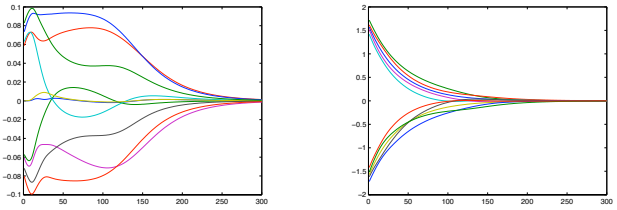

Fig. 7. (r, θ) features: error.

rotation does not cause any misbehavior either in the image or Cartesian trajectories. There are no significant differences between the velocity patters of each representation.

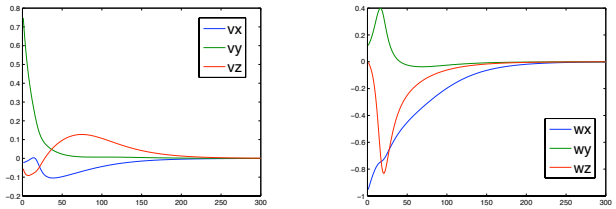

Fig. 8. (u, v) features: Kinematic screw, linear (left) and angular (right) velocities.

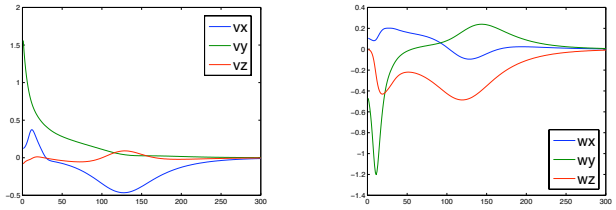

Fig. 9. (r, θ) features: Kinematic screw, linear (left) and angular (right) velocities.

In our human tracking experiments, the fisheye camera is located in the center of our laboratory room. The lab contains some of the small dynamic factors named above (e.g. blinking of computer screens or variations in illumination due to the different time of the day or switch on/off a light). Images are acquired in 24-bit RGB model with a $640x480$ resolution. Detection results for multiple moving humans and its corresponding cylindrical panoramic images under different illuminations are shown in Fig. 10

A first experiment involving the robot arm and a person is shown in Fig. 11. The task is simplified by using only

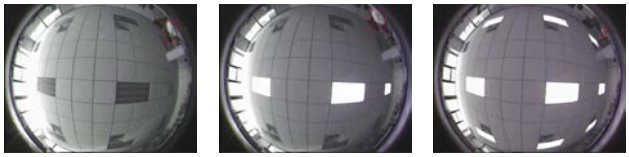

Fig. 10. Resulting detection images under different illuminations.

one fisheye camera. The goal is to detect an approaching person, and move the arm to point the end-effector towards that person, whose intention is supposed to act interactively with the arm.

The end-effector of the arm is tracked by a color mark. Since only two feature values are available (r, θ), the motion of the arm is restricted to a rotation about a vertical axis. This simple motion is enough to accomplish the task.

The approaching person is detected when he is still far from the workspace of the robot. The arm starts to move, in order to align its orientation to the angle of the detected feature. While the arm is moving, the motion filter is programmed to discard the moving features of the arm (ego-motion). As can be seen in the third row of the sequence, the visual feedback approach converges to the correct orientation of the approaching person.

VI. FUTURE WORK

We have presented two different applications to guarantee the safety in the robot workspace. They separately give the system some information about the manipulator position with respect to the reference system and the proximity of the detected moving objects to the system. So, we are working to combine both of them in order to make correct decisions about the robot movements not only to achieve the robot goals but also not to cause any damage. Moreover, future work will include the definition of such high-level behaviors for the safe interaction between the arm and the surrounding people.

VII. ACKNOWLEDGMENTS

The authors gratefully thank support from the EU 6th Framework Programme (IST-045269), the Spanish Ministry of Education-CICYT (DPI2005-08203-C02-01, FPI grant DPI2004-01920), and Fundació Caixa Castelló - Bancaixa (P1-1B2005-28).

REFERENCES

[1] E. Cervera, "Distributed visual servoing: a cross-platform agent-based implementation," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 319–324.

[2] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," in *Proc. of the European Conf. on Computer Vision*, 2000, pp. 445–461.

[3] Y. Mezouar and E. Malis, "Robustness of central catadioptric image-based visual servoing to uncertainties on 3D parameters," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004, pp. 1389–1394.

[4] H. Koyasu, J. Miura, and Y. Shirai, "Recognizing moving obstacles for robot navigation using real-time omnidirectional stereo vision," *J. of Robotics and Mechatronics*, vol. 14, no. 2, pp. 147–156, 2002.

[5] R. Vidal, O. Shakernia, and S. Sastry, "Formation control of non-holonimic mobile robots with omnidirectional visual servoing and motion segmentation," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 584–589.

[6] F. Correa and J. Okamoto, "Omnidirectional stereovision system for occupancy grid," in *Proc. of the Int. Conf. on Advanced Robotics*, 2005, pp. 628–634.

[7] M. Grossberg and S. Nayar, "A general imaging model and a method for finding its parameters," in *Proc. of the IEEE Int. Conf. on Computer Vision*, vol. 2, 2001, pp. 108–115.

[8] E. Schwalbe, "Geometric modelling and calibration of fisheye lens camera systems," in *Proc. of the 2nd Panoramic Photogrammetry Workshop*, 2005, pp. 273–285.

[9] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[10] E. Malis, F. Chaumette, and S. Boudet, "Multi-cameras visual servoing," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 3181–3188.

[11] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," *The confluence of Vision and Control, LNCIS series, Springer Verlag*, 1998.

[12] L. Shapiro and G. Stockman, *Computer Vision*, U. S. River, Ed. Prentice Hall, 2001.

[13] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294–307, March 2005.

[14] H. Liu, W. Pi, and H. Zha, "Motion detection for multiple moving targets by using an omnidirectional camera," in *Proc. IEEE Int. Conf. on Robotics, Intelligent Systems and Signal Processing*, Chansgsha, China, October 2003, pp. 422–426.

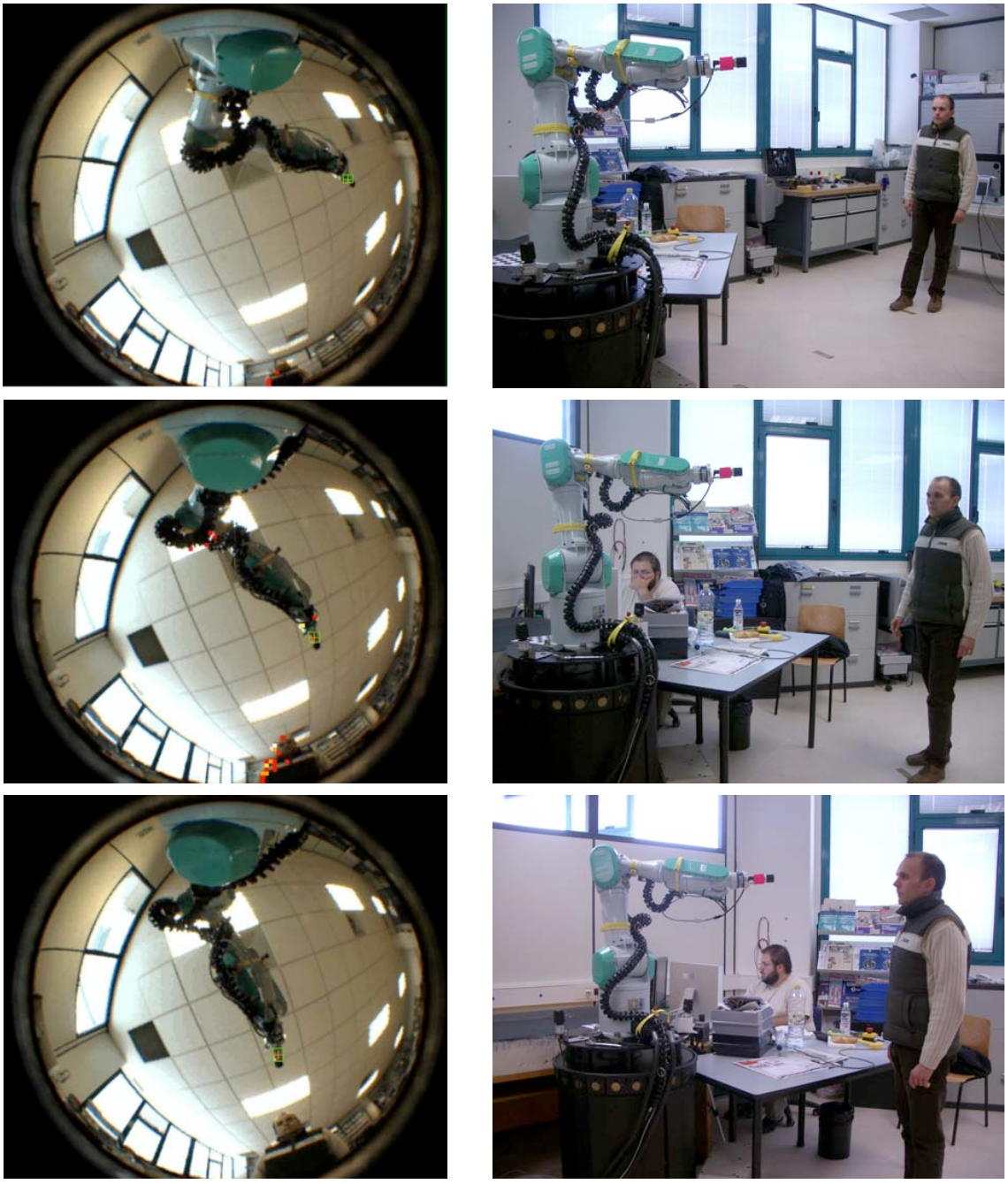

Fig. 11. Image sequence of an experiment with one camera. The end-effector has a color mark, which is segmented and plotted with a green cross and bounding box in the image. Motion detection is plotted with red and orange pixels.