Abstract—The problem of image based visual servoing for robots working in a cluttered dynamic environment is addressed in this paper. It is assumed that the environment is observed by depth sensors which allow to measure the distance between any moving obstacle and the robot. Also an eye-in-hand camera is used to extract image features. The main idea is to control suitable image moments and to relax a certain number of robot’s degrees of freedom during the interaction phase. If an obstacle approaches the robot, the main visual servoing task is relaxed partially or completely, while the image features are kept in the camera field of view by controlling the image moments. Fuzzy rules are used to set the desired values of the image moments. Befonte that, the relaxed redundancy of the robot is exploited to avoid collisions. After removing the risk of collision, the main visual servoing task is resumed. The effectiveness of the algorithm is shown by several case studies on a KUKA LWR 4 robot arm.

I. INTRODUCTION

Classical image-based visual servoing (IBVS) is aimed at controlling the end-effector of a robot carrying a camera in such a way that some measurable quantities extracted from the image captured by the camera, denoted as image features, attain desired values. This allows, for example, a robotic hand to be aligned with an object to grasp, like the handle of a drawer or door whose position is uncertain or may change, especially in anthropic environments.

In the presence of moving obstacles, it is important for the robot controller to ensure suitable reaction capabilities beside main task performance. In this regard, two main operating modes are usually adopted, namely, avoiding undesired collisions, or handling the physical interaction. The latter case can be implemented by increasing the compliance of the robot and using suitable observers to estimate the external forces exerted on the robot body, [1], [2]. Physical interaction control relies on fast control loop rate, which is usually higher than that used to accomplish visual servoing tasks.

The selection of the visual features is very important in image based visual servoing and affects directly the interaction matrix, i.e., the Jacobian mapping the camera Cartesian velocity to the time derivative of the image features. A good choice of the features allows to obtain a large full rank domain for the interaction matrix and linear relationship between the mapped velocities. The best way to ensure this condition is to associate each camera degree of freedom with only one visual feature. However, such complete decoupling cannot be easily verified and only partial decoupling can be obtained under some conditions using image moments and invariants [3], [4].

The problem of path planning for IBVS of robotic arms with the aim of extending the robustness of classical control techniques to include image constraints (field of view limits, occlusions) and physical constraints (joint limits, singularities in robot Jacobian, collisions with obstacles, self-collisions) is addressed, e.g., in [5] and references therein. The main idea is to plan and generate off-line feasible image feature trajectories while accounting for the constraints, and then to control the robot along the planned trajectories. Generally, this results in a more robust servoing process with respect to violation of image and physical constraints. However, these techniques are not effective in the presence of moving obstacles (like humans), whose position is not known a priori.

An interesting approach to control the robot in the presence of constraints is the concept of task sequencing [6], [7], which exploits the functional kinematic redundancy of the robot. The key idea is to divide the task into several subtasks which, in the presence of constraints, are deactivated in sequence ordered according to suitable priority criteria. Further works exploiting redundancy in visual servoing tasks are [8], [9].

Although reactive control techniques are more suitable than off-line planning techniques to ensure a safe coexistence of robots and humans, reactive approaches purposely designed to address this problem have not been developed so far. Only some heuristic techniques can be found in the literature [10].

In this respect, a promising direction of research is that of combining visual servoing approaches which guarantee that the visual features remain in the field of view of the camera [11], [12], [13], [14] with collision avoidance techniques exploiting redundancy [15].

In this paper an image-based visual servoing algorithm with collision avoidance capabilities of dynamic obstacles is proposed. An eye-in-hand camera is used to extract image features which are used to control the end-effector motion. Moreover, the distance between any part of the robot with dynamic obstacle (human) is measured using other sensors.

It is assumed that the velocity controlled robot is executing a visual servoing task and the task can be partially relaxed when the obstacle is too close or come in contact with the
robot. To this aim, a combination of six image moments is selected as the visual features to control the six degrees of freedom of the camera such that the corresponding interaction matrix has a maximally decoupling structure. During the interaction phase, only the visual features related to the centroid and the variance of the image is regulated to keep the features in the camera field of view (FOV). Therefore, the dimension of the servoing task decreases without interruption and the released redundancy of the robot is used to keep the robot at a safe distance from obstacle(s) through a repulsive action. Moreover, fuzzy rules are designed to set the references for the image moments, depending on some features of the expected collision. The simulation results, performed on a 7-DOF KUKA LWR4 robot arm, confirm the effectiveness of the proposed algorithm.

II. PRELIMINARIES

A. Camera Model and Image Jacobian

Consider a camera with coordinate frame \( F_c \). The camera views a collection of \( N \) visual features

\[
s(t) = [s_1^T, s_2^T, ..., s_N^T]^T. \tag{1}
\]

If the features are static in an inertial frame, the features’ velocity in the image plane is given as a function of the camera velocity \( v_c \) by the relationship

\[
\dot{s} = Lv_c, \tag{2}
\]

where \( L(t) \) is the interaction matrix and is constructed by concatenating a set of \( N \) sub-matrices \( L_k(s_k) \). For instance, for any point \( (x_k(t), y_k(t)) \) in the image plane, associated with 3D point with depth \( Z_k \) in \( F_c \), the interaction matrix can be obtained as \( L_k \). The interaction matrix related related to the most common geometric primitives can be found, e.g., in [3].

To follow a desired trajectory \( s_d(t) \) in the image plane, the control scheme usually is designed to ensure a decoupled exponential decrease of the error in the image plane. For instance for eye-in-hand system observing an static object, the camera command velocity is given as

\[
v_{c-corn} = \dot{L}^+ (s_d + \gamma(s_d - s)), \tag{3}
\]

where \( \dot{L}^+ (t) \) denotes the left pseudo inverse of the \textit{estimated} matrix \( L(t) \) and \( \gamma \) is a positive gain. Note that the depth of object points which is generally unknown in 2D visual servoing is necessary for calculation of \( L(t) \). However, this value can be estimated by a suitable observer during camera motion (see, for instance, [16]) or simply set as the desired depth value. The non-linearity in the interaction matrix explains the unpredictable behavior of the robot in 3D space even for a small displacement in image plane.

B. Image Moments and Moment Invariants

The image moments are very useful for providing an intuitive and meaningful representation of the object in the image plane. For a discrete set of \( N \) image points, the moments \( m_{ij} \) and centered moments \( \mu_{ij} \) (with respect to the object centroid \((x_g, y_g)\) in image plane) are defined by

\[
m_{ij} = \sum_{k=1}^{N} (x_k)^i (y_k)^j, \tag{4}
\]

and

\[
\mu_{ij} = \sum_{k=1}^{N} (x_k - x_g)^i (y_k - y_g)^j, \tag{5}
\]

respectively. The centered moments defined from (5) are known to be invariant to 2D translational motion if the image plan is parallel to the object. In the literature, many methods have been presented to derive moment invariants to scale and rotations. These invariants can be found in [4] and the references therein.

As for the classical geometric features, a linear map can be found in the form

\[
m_{ij} = L_{m_{ij}}v_c, \tag{6}
\]

between the time variation of moments and the relative camera-object velocity components. \( L_{m_{ij}} \) is the related interaction matrix in the form of

\[
L_{m_{ij}} = [m_{vx} m_{vy} m_{vz} m_{wx} m_{wy} m_{wz}](ij). \tag{7}
\]

The analytical form of the interaction matrix related to any image moments of order \( i + j \) can be computed under some conditions [3].

The moments based on the above definitions and their combinations can be effectively used in visual servoing as the features of the image, provided that they are well-defined and differentiable. For instance, with the selection \( s = (x_g, y_g, m_{00}, \mu_{02}, \mu_{20}, \alpha) \) with \( \alpha = \frac{1}{2} \tan^{-1}(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}) \) defining the object orientation, we may hope that the six DOF of the camera can be controlled effectively. However, the associated interaction matrix is not full rank for a symmetric object appearing in the image plan. For non-symmetric objects the interaction matrix is of full rank 6 but with high condition number.

III. CONTROL DEVELOPMENT

Let us assume that a (redundant) robot is performing a visual servoing task using an eye-in-hand camera. Depth sensors (e.g., Kinect) are employed to measure the distance of robot body and end-effector from any obstacle or humans. The visual servoing task is that to follow a desired trajectory \( s_d(t) \) in the image plane (IBVS). To this aim, the control law (3) can be adopted for the robot end-effector.

A. Keeping the Visual Features in the FOV

When an obstacle comes near the robot, collisions must be avoided. In some particular cases, the redundant degrees of freedom of the robot allow to avoid the collision by reconfiguring the robot through internal motions while keeping the servoing task. However, this is not always possible and, to avoid the obstacle, the task must be abandoned and cannot be easily resumed.
The solution proposed here for obstacle avoidance is that of relaxing the servoing task partially, while keeping the image features in the camera field of view by using a reduced number of degrees of freedom. This allows to have more degrees of freedom at disposal for obstacle avoidance and to recover the servoing task once the collision is avoided. To this end, the moments based features of order up to 2, including the centroid of the object’s image and the centered moments $\mu_{02}$ and $\mu_{20}$ are regulated during the interaction phase. Note that in comparison with the visual servoing task that is usually a 6 DOF task, this task involves 3 or 4 DOF and thus relaxes more degrees of freedom which may be exploited for obstacle avoidance.

The coordinate of the centroid point of a set of visual features in the image plane is given by

$$x_g = m_{10} / m_{00}, y_g = m_{01} / m_{00}. \quad (8)$$

For the case of $N$ point features, the time derivative of $s_g = [x_g, y_g]^T$ is given by

$$\dot{s}_g = J_g \dot{s}, \quad (9)$$

where $J_g$ is a $(2 \times 2N)$ matrix computed as

$$J_g = \frac{1}{N} [I_2, \ldots, I_2], \quad (10)$$

being $I$ the identity matrix. In this case also the moments $\mu_{02}$ and $\mu_{20}$ represent the variance of the feature points in the current image

$$s_v = \frac{1}{N} \sum_{k=1}^{N} \begin{bmatrix} (x_k - x_g)^2 \\ (y_k - y_g)^2 \end{bmatrix} = \frac{1}{m_{00}} \begin{bmatrix} \mu_{20} \\ \mu_{02} \end{bmatrix} \quad (11)$$

The Jacobian of the variance task function, $J_v(t)$ is a $(2 \times 2N)$ matrix given by

$$J_v = \frac{2}{N} \begin{bmatrix} x_1 - x_g & 0 & \cdots & 0 & x_N - x_g & 0 \\ 0 & y_1 - y_g & \cdots & 0 & y_N - y_g & 0 \end{bmatrix} \quad (12)$$

and relates the time derivative of $s_v$ to $\dot{s}$ through

$$\dot{s}_v = J_v(t) \dot{s}. \quad (13)$$

Note that $J_v(t)$ is singular when all the feature points are collinear. It is clear that the associated interaction matrix of these moments are obtained through multiplication of the above Jacobian matrices by the interaction matrix of the feature points. It can also be shown that the these task Jacobians are completely independent and thus the augmentation

$$J_{gv} = \begin{bmatrix} J_g \\ J_v \end{bmatrix} \quad (14)$$

does not introduce any singularities [13].

B. Selection of the Visual Features

The selection of visual features in 2D visual servoing is an important issues for both the discrete and continuous features. Indeed, because of the nonlinearity of the interaction matrix, the trajectory in 3D space related to a trajectory in the feature space may not be easily predicted. Thus, it is important to choose visual features that reduce this problem. Moreover, the well-conditioning of the interaction matrix is also important for the convergence of the servoing scheme. The best way to ensure that, is to associate each camera DOF with only one visual feature to obtain a decoupled behavior. On the other hand, since the depth information is necessary in interaction matrix and usually an estimation of interaction matrix is used in the control scheme, it is convenient to choose the visual features in such a way that the interaction matrix is almost constant around the desired pose, especially when the object and the image planes are parallel at the desired position. The coordinates $x_g$ and $y_g$ of the object image center, the orientation of the object $\alpha$ and the area $m_{00}$ of the object image are of particular interest since they have direct and intuitive link with the 3D. Indeed the coordinate $x_g$ and $y_g$ are closely related to $v_x$ and $\omega_y$, and $v_y$ and $\omega_x$, respectively. Furthermore, the area and the orientation of the object in the image scene are affected mainly by $v_x$ and $\omega_x$. On the other hand, based on the results of the previous section, the quantities $\mu_{20}$ and $\mu_{02}$ affects the variance of the features and can be useful for keeping the features in the FOV during any interaction.

In order to decouple $\omega_y$ from $v_x$, and $\omega_x$ from $v_y$, two combination of moments of order 3 have been proposed in [3]. These combinations are obtained based on the following invariants to translational motions and rotational motion around the optical axis, since the visual features to control these motions is already available.

$$I_1 = (\mu_{20} - \mu_{02})^2 + (2\mu_{11})^2,$$
$$I_2 = (\mu_{03} - 3\mu_{21})^2 + (\mu_{30} - 3\mu_{12})^2,$$
$$I_3 = \mu_{20} - \mu_{02}. \quad (15)$$

The combinations are different for the case of symmetric and non-symmetric observed objects. Here, we recall the supplementary features for non-symmetric case which are given by

$$p_x = I_1 / I_3^2,$$
$$p_y = m_{00} I_2 / I_3^3. \quad (16)$$

For centered symmetrical objects these choices produce singularity in the interaction matrix. That’s why, for symmetric objects, these features are different. The related equations are not presented here and can be found in [3]. Indeed, when the object is parallel to the image plane, the interaction matrix related to these features has non zero values only for the columns associated with $\omega_x$ and $\omega_y$. However, as we will see in the simulations, in other configurations the values of the other columns are negligible and the results are satisfactory in the sense of decoupling even when the object is not parallel to the image plane during servoing task.
C. Obstacle Avoidance

The distance vector between the obstacle $O$ and any point $P$ on the robot, $D(P, O)$, is assumed to be known by depth sensor or laser sensor. As soon as the distance $|D(P, O)|$ is found to be less than $|D(P, O)|_{\min} = \rho_1$, a repulsive vector is produced and used to modify on-line the current trajectory. The measure of proximity of the obstacle to the robot is defined by a smooth function,

$$\nu(P, O) = \frac{1}{1 + e^{(||D(P, O)| - \rho_2)^2/(\rho_1 - 1)}} \alpha,$$

(17)

which is equal to one when $|D(P, O)|_{\min} < \rho_2$ and is zero for $|D(P, O)|_{\min} > \rho_1$ for appropriate shape factor $\alpha$.

A simple and effective repulsive vector is given by

$$V_{rep} = \nu_1 V_{max} \frac{D(P, O)}{|D(P, O)|},$$

(18)

where, $\nu_1$ is given by (17) with appropriate parameters. This vector has the same direction as $D(P, O)$ with $V_{max}$ as the maximum admissible speed in task space. Note that this repulsive action is only given based on the distance between obstacle $O$ and control point $P$ on the robot body. The factor $\nu_1$ allows to further modulate the repulsive action depending on the distance of the obstacle from the robot body. It is more effective to include also the direction and velocity of the obstacle which can be extracted by observing the time variation of the repulsive vector, i.e., $\dot{V}_{rep}$. This information can be used, for instance, to escape the collision by moving the robot perpendicular to the direction of the obstacle trajectory. The algorithm is called Pivot algorithm [15] and modifies the direction of the repulsive vector according to its variation. The repulsive velocity is then reflected in the joint space through the transpose of the Jacobian $J_P$ associated with the point $P$, i.e.,

$$\dot{q}_{rep} = J_P^T V_{rep}.$$  

(19)

D. Combined Action

Now, assume that a robot is doing a visual servoing task and that the command velocity for the end-effector is calculated by (3). As soon as an obstacle approaches the robot, the repulsive vector $\dot{q}_{rep}$ is used to update the trajectory. However, applying this repulsive action definitely affects the assigned visual servoing task for the end-effector and may cause the robot to lose the visual features during obstacle avoidance.

In order to keep the visual features in the field of view during the repulsive action, the combination $s = (x_g, y_g, \mu_0, \mu_2, P_x, P_y)^T$ is selected as the features vector. During the interaction phase, the first four features which regulate the centroid and the variance of the image is kept and the last two features are released. This increases the dimension of the null-space which is exploited to handle the interaction. However, for symmetric object, it can be shown that the interaction matrix associated with the first four features is singular when the camera plane and the objects are parallel. That’s why we propose to use $s' = (x_g, y_g, \mu_0, \mu_2, P_x, P_y, \alpha)^T$ for symmetric objects. In this case, the last three features are released during the interaction and produce a three dimensional redundant space. Based on the above selections, the feature vector and the interaction matrix can be partitioned as

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \hat{L} = \begin{bmatrix} \hat{L}_1 \\ \hat{L}_2 \end{bmatrix}.$$  

(20)

The above choices of $s$ or $s'$ produce maximum decoupling of the interaction matrix as mentioned in section III.B. Finally, for a robot with a velocity controller in the joint space, the command velocity is given by

$$\dot{q}_{com} = (\hat{L} J)^T \begin{bmatrix} \dot{s}_{com-1} \\ (1 - \nu_2) \dot{s}_{com-2} + \nu_2 (\hat{L}_2 J) (\hat{L}_1 J)^T \dot{s}_{com-1} \end{bmatrix} + \nu_2 N \dot{q}_{rep},$$

(21)

where, $J$ is the robot’s Jacobian matrix, and $\nu_2$ is given by (17) with appropriate parameters. The quantities $\dot{s}_{com-1}$ and $\dot{s}_{com-2}$ are command feature velocities related to the first and second partition of $s$ or $s'$. These command velocities are set as

$$\dot{s}_{com-1} = \dot{s}_{des-1} + \gamma_1 (s_{des-1} - s_1),$$

(22)

$$\dot{s}_{com-2} = \dot{s}_{des-2} + \gamma_2 (s_{des-2} - s_2),$$

with $\gamma_1$ and $\gamma_2$ as positive gains. In (21), $N$ is the projection matrix given by

$$N = I - (\hat{L}_1 J)^T (\hat{L}_1 J).$$

(23)

Note that, in the absence of interaction, i.e., $\nu_2 = 0$, from (21) the command joint space velocity is reduced to

$$\dot{q}_{com} = (\hat{L} J)^T \begin{bmatrix} \dot{s}_{com-1} \\ \dot{s}_{com-2} \end{bmatrix}.$$  

(24)

When an obstacle comes near the robot, $\nu_2$ increases to 1 and (21) decreases to

$$\dot{q}_{com} = (\hat{L}_1 J)^T \dot{s}_{com-1} + N \dot{q}_{rep},$$

(25)

and the control of the centroid and of the variance with a continuous transition from full visual servoing is ensured. For more information about the continuous transition among multiple tasks and during multi-priority control, please refer to [2], [17].

The interaction matrix $\hat{L}$ can be chosen as a constant matrix at the desired pose, $L_{|s=s_{des}}$, which presents excellent decoupling properties. However the following choice seems more appropriate:

$$\hat{L}_{s} = \frac{1}{2} (L_{|s=s_{des}} + L_{|s(t)}).$$  

(26)

In the case that the feature vectors are different from the above selection $s$ or $s'$, a similar algorithm can be used. In that case the visual servoing task is attenuated by a factor of proximity and the centroid and variance features are selected and controlled to keep the scene in the FOV.
During the interaction phase, the desired trajectory for the centroid and the variance features can be left constant or may be set as the values reached at the last capture of the visual servoing task. However, these values can be chosen wisely based on the location of the nearby obstacle as explained in the next Section.

The proposed formulation can be used for interaction on the robot body as well as on the end-effector of the robot using the Jacobian matrix of the interaction point $P$ which is assumed to be known by monitoring the robot scene with depth sensors. Note that the above analysis was performed assuming the absence of occlusion.

E. Fuzzy Based Reactive Planning of the Desired Variance

As mentioned above, the desired values of the centroid position and the image variances during the interaction phase are the important parameters that mainly affect the behavior of the robot. In other words, the centroid of the object in the image plane is controlled by the former, and the latter affects the scale factor in the image plane. The desired value for the centroid is usually set to zero to keep the centroid of the visual features in the center of the image plane. The choice of the desired value of the variance can be more complex. To increase the autonomy of the algorithm during the interaction, a *Mamdani*-type fuzzy reactive planning algorithm is proposed to set this desired value intelligently. The obstacle data which are available through the depth sensors are used as the inputs to a fuzzy planner. The rules are defined so that the robot body always keeps a safe distance from obstacle trajectory. Some robot-obstacle situations and the corresponding variation of the desired value of the variance are shown in Fig. 1 for a typical manipulation task. The fuzzy planner changes the desired value based on the situations using predefined rules. When the robot is in the situation that changing the desired variance value is not useful to avoid the obstacle, then the desired variance is left unchanged.

In details, the fuzzy subsystem uses four different inputs in order to decide how to change the variance. These are: the obstacle distance from the manipulator, the obstacle relative position to the manipulator configuration, the link exposed to the interaction and the actual robot configuration are the information required to make a proper decision. These data are computed using the vector connecting a moving control point $P$ on the robot to the obstacle. The control point moves along the robot body repeatedly, to find the the possible position of collision.

Based on the fuzzy input data, the fuzzy planner decides how to change the desired variance. The membership functions related to the main robot configurations are shown in Fig. 2, in which the 3rd and the 4th situations correspond to Fig. 1. Consider the first configuration depicted in the top of this figure. According to the obstacle position which is above or below the link, the fuzzy planner will increase and decrease the desired variance respectively. The relative obstacle-link position membership functions are shown in Fig. 3. Increasing the desired variance will decrease camera distance with respect to the scene in order to move the robot away from obstacle trajectory, and vice versa. The amount of variance variation depends on the obstacle distance from the manipulator (Fig. 4). If the obstacle is not below or above the link and they are in the same level relative to each other, zero membership function in Fig. 3 is dominant, and the fuzzy planner does not change the variance and allows the repulsive vector to move the robot to a safe position. This is true, except for the case where the obstacle position is very near to the manipulator according to the membership function shown in Fig. 4. For other configurations similar rules are designed. Their explanations are skipped for brevity. The output membership functions are illustrated in Fig. 5.
A. Case Study I: Interaction on the Robot’s Body

The snapshots of this experiment are illustrated in Fig. 6. The parameters of the command velocity (21) are chosen as follows. For the visual servoing task the parameters $\gamma_1$ and $\gamma_2$ in (22) are chosen as $\gamma_1 = \gamma_2 = 3I_3$. Moreover, $V_{\text{max}}$ in (18) is set $V_{\text{max}} = 5m/s$. The distance between the risky control point $P$ on the robot body and the obstacle as well as the parameters $\nu_1$ and $\nu_2$ are shown in Fig. 7. As soon as the obstacle moves near to the robot, the parameter $\nu_2$ increases rapidly to 1 and the visual servoing task is partially released. The desired centroid values are set to its last values before the interaction phase, which in this experiment are zero. The desired variance is computed by the fuzzy planner as shown in Fig. 8. In fact, as it was mentioned before, the desired centroid values control the offset of the image center in the image plane and the desired variance value affects mainly on the distance of the camera frame with respect to the image scene. Thus, for smaller values of the desired variance, higher retraction of the camera is achieved during the interaction.

B. Case Study II: Interaction on the End-effector

In the second case study, the interaction happens on the end-effector. Here, the visual servoing task is the same as in the first case study but the obstacle passes near to the end-effector assuming no occlusions. The main parameters of the velocity controller have not been changed to show the effectiveness of the algorithm for different interaction points. The results are depicted in Figs. 9-11. It can be seen that the safety distance between the end-effector and the obstacle is always respected, and can be controlled by $\nu_1$, $\nu_2$ and $V_{\text{max}}$. The retraction of the end-effector during the interaction phase as a result of the variation of the desired value of the variance is clearly seen by comparison of the robot configurations at the times $t = 1.5$ and $t = 4$ in Fig. 9. The effect of $V_{\text{rep}}$ can
be seen by comparison of these snapshots with the snapshot related to \( t = 7.5 \).

Fig. 9: Case 2: snapshot of the system during interaction on the end-effector.

During the interaction, these values are controlled to keep the visual features in the field of view and the rest of the visual features are released smoothly. Consequently, the null-space dimension increases and is exploited to effectively keep a safe distance from the obstacle. In addition, a fuzzy planner was used in order to change the desired variance value so that the available degrees of freedom can be better exploited. The proposed approach was tested in several simulations on a KUKA LWR robot arm in the presence of a moving obstacle.

ACKNOWLEDGMENT

The first author acknowledges the support given by the office of vice chancellor of the research and technology in University of Isfahan under research project number 930402.

REFERENCES


V. CONCLUSION

The control of the interaction during an image based visual servoing for a robot working in dynamic cluttered environments was considered in the paper. The main concerns in this scenario are the performance of the main visual servoing task, keeping the visual feature in the field of view as well as a safe obstacle-robot distance. To cope with these three objectives, a special combination of the visual features extracted from the image moments including the centroid and the variance values of the visual features is selected.