On Line Visual-Grasping System Based on a Gripper with Two Flexible Fingers

J. Becedas † E. González † I. Payo † V. Feliu †

† Universidad de Castilla-La Mancha, Av. Camilo José Cela s/n, Ciudad Real, Spain (Tel: 0034 926295300; e-mail: †Elisabeth.Gonzalez, Jonathan.Becedas, Ismael.Payo, Vicente.Feliu †@uclm.es)
‡ University of Leicester, University Road, LE1 7RH Leicester, United Kingdom (Tel:+44 (0) 116 252 2872)

Abstract:
In this work we propose a novel gripper with two flexible fingers combined with a visual object recognition system for its implementation in industrial applications, such as manufacturing, assembly and manipulation, among others. This original system increases the manipulability and flexibility of the robots manipulators and improves the safety of the operations carried out with it. This combined system also makes the manipulation of very fragile or deformable objects possible as result of the flexible characteristic of the gripper fingers. The system is mounted at the end effector of a 6 DOF robot manipulator for object recognition and manipulation tasks. The object recognition software is developed and the force control law for the gripper is designed. Experiments are included to demonstrate the good performance of the system proposed.

Keywords: Shape Recognition, Visual Servoing, Grasping, Force Control, Flexible Robots

1. INTRODUCTION

Grasping is one of the most frequent subjects to deal with in robotics owing to the requirement of moving or manipulating different objects in repetitive industrial tasks, e.g. Kolluru et al. (1998). In spite of the various benefits of the grasping techniques developed to optimize industrial processes, their current limitations make them expensive and with low flexibility to changes in the design of the pieces or to the inclusion of new parts in the assembly task. Conventional combined vision and grasping systems cannot respond to changes in the product design, and a major part of the assembly line must be changed. Several solutions have been proposed to deal with this problem so far: the use of commercial components, reduced time of the product design and easy adaptation to the machines. Nevertheless, these guidelines can only be applied when specific tools are used for different assembly parts, and for each operation, thus increasing consequently the number of tools used in the assembling process, the assembling time, costs and complexity.

In the last decade, several techniques have been developed to deal with those problems. Sanz et al. (2005) described a visually guided system for robot grasping with a two-fingers gripper. The capabilities of this methodology to deal with objects of complex shapes including those with are unknown a priori, were studied although the methodology was only used to manipulate planar objects. In Kragic (2001) a visual servoing method was implemented for non-planar objects with a given wireframe model of the objects. The initialization process to estimate the object pose was manually performed by clicking onto the workspace corner points. Some years later this constraint was solved by Kyrki and Kragic (2005) who developed an automatic initialization process by using the Scale Invariant Feature Transform (SIFT). The main drawback of this research was the high computational cost. Recently, in Yamazaki et al. (2008), a method was developed with no constraint on the object shape. However, the system required the object’s exact location. Other more sophisticated methodologies have been developed to perform object recognition before the grasp planning. Effendi et al. (2008) used a framework setup similar to those of Jafari and Jarvis (2004) and Yamazaki et al. (2008) to achieve a human-centric solution. The difference is that their system utilizes stereo rather than monocular vision, and 2.5D rather than 3D object modeling is used, as in Yamazaki et al. (2004), and for instance, the cost of this system was more expensive than in the case of monocular vision based systems.

Another key point in the grasping system is the gripper. The possible changes in the work scene signify that the grasping system must be robust to possible errors in the grasping points estimation done by the vision system. Furthermore, the gripper performance is also very important when fragile objects of different stiffness are manipulated and hence a reliable force control is crucial. This problem can be overcome with the use of deformable or flexible fingers which improve the limited capabilities of robotic rigid fingers. Some researchers who are interested in this topic designed flexible grippers actuated by piezoelectric materials, e.g. Choi and Lee (1997), Tanaka et al. (1996) and Chonan et al. (1996), but this kind of actuators developed small forces and displacements. Other researchers used shape memory alloy (SMA) actuators to control the grasping force of a flexible gripper (e.g. Choi et al. (2001), Yang and Gu (2002) and Yang and Wang (2008)) but SMAs had...
certain drawbacks when they were applied to a high working bandwidth because of its slow thermal response.

In this work we propose an object recognition system based on Fourier Descriptors and Integral Invariants that is able to deal with non-planar objects using a monocular camera. Objects to be grasped are recognized thanks to an object model stored in a database. This characteristic allows the system to classify objects with different sizes and shapes. The advantages of this method with respect to the previously mentioned methods are the following: robustness to scene variations and low computational cost, which makes it ideal for real time applications. In addition, our approach does not require to know the object position in the scene, and the calibration process does not require high precision due to the gripper design, which increases the whole system performance. In addition, the grasping system is able of manipulate objects of different sizes, weights, shapes and stiffness to be used in industrial tasks in which the environmental conditions (light conditions, object background and occlusions) are controlled. The combination of the object recognition system with the flexible finger gripper provides a precise and safe grasping system. The force exerted on the object by the gripper is controlled by means of a Generalized Proportional Integral (GPI) regulator, based on the flatness characteristic of the system dynamics, see Becedas et al. (2009), which was originally applied for grasping tasks with a novel one flexible finger gripper in Becedas et al. (2008). In conclusion, the aim of the paper is the demonstration of a whole grasping framework, from object identification to grasping.

2. THE GRASPING SYSTEM PROPOSED

The proposed grasping system focuses on industrial applications, in which the environmental conditions (background, lights) and robot movements are controlled. The system is constituted by a vision system using a monocular camera and a gripper with two flexible fingers. Fig. 1 shows the designed gripper.

Fig. 1. Detailed picture of the grasping system.

The first requirement in our grasping system was to estimate the grasping points, i.e. the contact points between the fingers and the object. We used a monocular camera (2D views) and hence the on-line estimation of grasping points of 3D objects would require high computational cost and processing and execution times, since different points of view would be required to compute. To overcome this problem, we dealt with known objects (database), because in an industrial environment the objects to be grasped are perfectly known. Thus, the grasping points were a priori and off-line studied. They were computed in the system by means of a geometric transformation between the known object and its current pose in the scene, which was online detected and processed.

To simplify the number of views required to model an object, the robot end-effector was placed above the object and oriented perpendicular to the surface in which the object was located (table, bell). With these conditions, the object was related to object views, also known as footprints. Then, for each footprint the stable grasping points were defined. Note that in the scene, the object may appear rotated with regard to the footprint axis. The object recognition system was able of finding the database view that matched with the scene view and recognizing the rotation angle in order to apply the appropriate geometrical transformation.

Once the grasping points were defined, the robot end-effector was placed over the object and the gripper grasped the object with a desired force. Fig. 2 shows a general scheme of the proposed grasping system.

The grasping sequence had four main modules:

1. System Initialization: movement of the gripper to align the object center (view) with the image center.
3. Grasping Points Determination: Determination of the contact points between the fingers and the object located in the scene.

Fig. 2. Information flux of the grasping system.

2.1 System Initialization

The system initialization process controlled the robot end-effector position to obtain an object view-centered image. To develop this process, it was necessary to compute the relationship between the end-effector position variation and the image pixels coordinate variation by means of a calibration process. A proportional control law was then applied to the end-effector in order to fix the error between the image center and the current view-center. This process was similar to that used by image-based visual servoing systems (see Kragic and Christensen (2003)). Fig. 3 shows an example of the system initialization process. First, the object position in the image was computed (green box), along with its desirable position (red box), see Fig. 3(a). The end-effector was then moved in order to align the green box with the red box, Fig. 3(b).

2.2 Object Recognition

To deal with the object recognition by using 2D images from the monocular camera we computed the pose parameters by
using Fourier Descriptors. Persoon and Fu (1977) and González et al. (2008), because of their low computational cost, and stability to translation, rotation (ϕ), scale (λ) and displacement (δ) of the reference point (staring point). To solve the problems caused by their dependency to reflection symmetry and lack of geometric relations between the different parts of the shape, we implemented Integral Invariants, Manay et al. (2006).

The object database $B = \{O_1, O_2, \ldots, O_Q\}$ was defined by $Q$ objects. The object $O_q (1 \leq q \leq Q)$ could be viewed from the camera in $K_q$ positions related to their principal footprints. For each view, we stored a shape descriptor ($V_q^k, 1 \leq k \leq K_q$) and the grasping points ($G_q^k$). To extract the shape contour (silhouette), a suitable image pre-processing was applied to the image. This process consisted of filtering, thresholding and contour extraction. From this process we obtained the feature vector $V_q^k = [S_x, S_y]$, where $[S_x, S_y]$ are the silhouette cartesian coordinates. The next step was to regularize the shape contour descriptor (silhouette) by assuming that silhouette points ($V_q^k$) have been normalized ($\bar{V}_q^k$) in order to ensure that two consecutive points were always at the same Euclidean distance. Fourier Descriptors ($U_q^k$) and Integral Invariants ($I_q^k$) were then computed from $\bar{V}_q^k$.

The following pseudo-code shows the main steps in the object recognition process:

```plaintext
//Shape recognition
img: captured scene image
η: Similarity threshold
V = get_scene_silhouette(img);
Ṽ = regularize(V);
I = get_integral_invariant(Ṽ);
U = fft(Ṽ);
for q=1:Q
  for k=1:k_q
    measure(q,k)=get_FD_similarity(U_q^k, U);
  end
end
(obj, view) = index(measure ≤ η);
for h=1:length(obj);
  // Compute pose the U_{view(h)} with respect to U;
  (angle(h), disp(h)) = get_pose(U, U_{view(h)});
  // Displace disp(h) positions the invariant integral vector first element;
  temp = displace(U_{view(h)}, disp(h));
  dist(h) = Euclidean_distance(temp, I);
end
```

An example of the shape recognition process is shown in Fig. 4. Fig. 4(a) depicts the shape identification process and Fig. 4(b) shows the output of the pose estimation process.

### 2.3 Grasping Points Determination

The grasping points determination process was set manually, in an off-line process. For each footprint in the dataset, we located the gripper at a position where the grasp could be stable. Then, several grasping tests were developed to check if under small variations in the grasping points coordinates, the grasping execution was successful. If with the initial grasping points the grasping was unstable, the process was repeated until a stable solution was found.

Thus, for each view ($V_q^k$) in the dataset, the robot parameters to grasp the object are $G_q^k = (X, Y, Z, β, γ, α)$, where $X, Y, Z$ are the cartesian coordinates and $β, γ, α$ are the yaw, pitch and roll angles of the robot end effector. Due to the features of our system yaw and pitch are fixed to achieve a perpendicular orientation, only two parameters have been stored to characterize the grasping points: the $Z_q^k$ coordinate of grasping point (in object coordinate world) and the $α_q^k$ roll angle required to grasp the object.

Let consider that the robot parameters to grasp the scene object are $\bar{G} = (\bar{X}, \bar{Y}, \bar{Z}, \bar{β}, \bar{γ}, \bar{α})$. According to the coordinate system from Fig. 5, $\bar{Z}$ and $\bar{α}$ parameters should be computed as follows:

$$\bar{Z} = Z_q^k + Z_t + l$$  \hspace{1cm} (1)
$$\bar{α} = α_q^k - \bar{ϕ}$$  \hspace{1cm} (2)

where $l$ is the finger length and $Z_t$ is the scene Z coordinate in the robot world coordinates.

![Fig. 5. Coordinate system.](image-url)
The gripper presented is made up of two flexible spring steel fingers actuated by a DC motor. This improves the grasping performance due to the conforming ability of the flexible fingers. This flexibility also increases the safety of the gripper, thanks to the capability of absorbing energy in the impact with the object surface.

Dynamics of the system A DC Motor is used to close and open the flexible gripper. A servo-amplifier controls the input current to the motor by means of an internally PI current controller. This electrical dynamics can be disregarded because this is faster than the mechanical dynamics of the motor. Thus the servo-amplifier is considered as a constant gain between the current motor and the voltage supplied to the servo-amplifier from a computer. In addition, the motor and the flexible finger dynamics are decoupled by means of a compensation term which affects the voltage supplied to the motor as previously done in Feliu et al. (1992). Thus, the dynamics of the flexible link and that of the motor can be separately treated. We finally obtain the following dynamics for the actuator:

\[ Au_c = \dot{\theta}_m + B\theta_m + \xi_0 \]  

where \( A = \frac{J_n}{l}, B = \frac{\mu}{l} \) and \( \xi_0 = \frac{\Gamma_F}{l} \) is the disturbance owing to the Coulomb friction. \( J \) is the motor inertia, \( k \) is the electromechanical constant of the motor, \( n \) is the reduction ratio of the gear, \( \nu \) is the viscous friction of the motor, \( \theta_m \) is the motor angle at the output of the gear box, \( \Gamma_F \) is the unknown torque produced by the Coulomb friction in the system and \( u_c \) is the voltage supplied to the servo-amplifier.

We consider a very lightweight flexible beam clamped at one end to a gear assembly and the other end is touching an object as shown in Fig. 6. Small deflections are considered, and the mass of the beam is assumed to be concentrated at the tip, therefore only the first vibration mode is considered, Feliu et al. (1992), Becedas et al. (2009). Assuming a rigid object, the coupling torque between the beam and the motor is represented as follows:

\[ \Gamma_m = c(\theta_m - \theta_t) \]  

where \( \theta_t \) is the beam tip angle and \( c = 3EI/l \) is the rotational stiffness of the beam. \( E \) and \( l \) are, respectively, the Young’s modulus and the inertia moment of the beam and \( l \) is the length of the beam. The tip force is related to the coupling torque as follows:

\[ F = \frac{\Gamma_m}{l} \]  

Assuming a static and rigid object (\( \theta_t = \text{constant} \)) and taking the first and second derivative with regard to time of expression (4), the velocity and acceleration of the motor at the output of the reduction gear are obtained:

\[ \dot{\theta}_m = \frac{\Gamma_m}{c}; \quad \ddot{\theta}_m = \frac{\Gamma_m}{c} \]  

and by substituting these expressions in (3), we obtain a new equation for the dynamic model of the complete system (see Becedas et al. (2010b)):

\[ A\dot{u}_c = \dot{\Gamma}_m + B\dot{\Gamma}_m + \xi \]  

where \( \xi = c\xi_0 \) is an input disturbance to the system.

Force control The control scheme is found to be robust with regard to the effects of the constant perturbations in the system dynamics and does not require the use of time derivative measurements to feedback the control law: the force control law only uses feedback from the torque at the root of the flexible finger, Becedas et al. (2008).

Since no high accuracy is required in free movement, an open loop control is proposed. This consists of applying a constant voltage to the DC motor \( V^*(t) \), which produces a torque higher than that of the Coulomb friction.

The closed loop control to regulate the grasping force on the object is designed to be a GPI control based on the flatness characteristic of the system dynamics. This methodology was introduced in flexible robotics in Becedas et al. (2009) to cancel structural vibrations on a beam and control the position of the beam tip in Becedas et al. (2010b) to control the force applied by a flexible robot. A first approach in the application of this methodology in grasping tasks can be found in Becedas et al. (2008).

The flatness relation of the system is given by the following equation:

\[ u_c = \frac{1}{A_c} \Gamma_m + \frac{B}{A_c} \Gamma_m \]  

The closed loop control law based on GPI control is the following:

\[ u_c(s) = u_c^*(s) + \frac{\alpha_2 s^2 + \alpha_1 s + \alpha_0}{s(s + \alpha_3)}(\Gamma_m^*(s) - \Gamma_m(s)) \]  

(9)

with the gains \( \alpha_0 = \frac{\alpha_1}{\alpha_2}, \alpha_1 = 4\alpha_2 \) and \( \alpha_3 = 4p - B \) if the poles are chosen to be in a common location of the negative real axis \( p \) (see Becedas et al. (2010b) for more information about the design of the control law).

Impact detection algorithm To detect the exact instant at which the control system has to change from the open loop control algorithm (free movement) to the closed loop control algorithm (constrained movement) when the contact with an object is produced, an on line impact detection mechanism to switch from one to another controller is needed:

\[ |\Gamma_m^*(t) - \Gamma_m| > \mu \]  

(10)

\( \mu \) depends on the maximum trajectory tracking error of free motion and also takes the torque sensor noise level into account, Becedas et al. (2010b):

\[ \mu = \zeta(max(|\Gamma_m^*(t)|) - min(|\Gamma_m^*(t)|)) + \eta + min(|\Gamma_m^*(t)|) \]  

(11)

where \( \eta \) represents the maximum torque sensor noise level in absolute value and \( \zeta = 0.08 \) was experimentally obtained after experimental tests with trajectories of different amplitudes.

3. EXPERIMENTATION

The effectiveness of the system performance has been demonstrated by grasping a sequence of different objects, and a pos-
sible industrial application has been developed in which fragile objects have to be classified and manipulated.

3.1 Experimental setup

The experiments are carried out with a 6 DOF Staubli RX-90 robot manipulator. The manipulator maximum payload is 11 [kg]. The communication with the robotic arm is via a CS7-M controller. The gripper is mounted at the end effector of the robot. The camera is a webcam Philips model SPC350, and is an integrated part of the gripper.

The gripper has two flexible fingers. Each finger is a rectangular sheet made of spring steel with width of 1.3 [cm], a length of 13 [cm] and a thickness of 0.7 [mm]. One end of the flexible fingers is clamped to a 11 [W] DC motor by means of a worm gearbox. A servoamplifier supplies the input voltage to the DC motor. This accepts control inputs in the range of [−10, 10] [V]. The parameters used for the torque control were estimated as Becedas et al. (2010a). These are the product $A_c = 17 \left[ \frac{N^2}{V \cdot kg} \right]$, $B = 0.8 \left[ \frac{N \cdot s}{(kg \cdot m)} \right]$ and $\xi = 3.4 \left[ \frac{N^2}{kg} \right]$. The maximum torque to apply in the gripper is limited to 0.65 $Nm$ in order to fulfill the small deformations hypothesis, and the minimum torque $4 \cdot 10^{-3} Nm$, to overcome the effects of the sensor noise, which is of $2 \cdot 10^{-3} Nm$ amplitude. The maximum load to manipulate with the gripper is 225 $g$ weight. More information regarding the gripper in the design, control, robustness, stability, asymmetrical grasping and grasping when the object contacts in an intermediate position of the fingers can be found in Becedas et al. (2011).

The sensor system is integrated solely by a pair of strain gauges, with a gauge factor of 2.16 and a resistance of 120.2 [Ω], situated in the coupling of one of the flexible fingers. The amplitude of the sensor noise was $\eta = 2 \cdot 10^{-3}$. Only one finger was sensed because of the mechanical system’s symmetry. Finally, the sample time chosen for real time tasks was 2 [ms].

The controller gains (see (9)) were designed to place the closed loop poles in $−50$ in the real axis. Their values were $\alpha_0 = 3.7 \cdot 10^5$, $\alpha_1 = 2.9 \cdot 10^4$, $\alpha_2 = 872.9$ and $\alpha_3 = 199.2$.

The objects to be manipulated were located in an aleatory position of the delimited workspace $0.65 \times 0.80 \: [m^2]$. Finally, the software used to develop the force control and visual pattern recognition was Matlab.

3.2 Grasping System Evaluation

We used a set of 40 different free form objects to evaluate several grasping system parameters: recognition rate, computational costs and grasping stability. Fig. 7 shows a sample of the objects used in the experiments.

![Fig. 7. Sample of objects used in the experiments.](image)

<table>
<thead>
<tr>
<th>Object</th>
<th>Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Initialization</td>
<td>1.40</td>
</tr>
<tr>
<td>Object Recognition</td>
<td>0.45</td>
</tr>
<tr>
<td>Computing Grasping Points</td>
<td>0.01</td>
</tr>
<tr>
<td>Grasping Execution</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1. Grasping system execution times

Recognition Rate We used a non-cluttered background (black). The objects were placed on the table in a random position. The total number of tests were 120, and four fails were obtained (a recognition average of 96.6%). Those fails appeared because three database objects presented views with the same appearance.

Grasping System Execution Times In Table 1 we detail the system execution times on a Pentium IV 1.2 GHz processor needed by each of the stages of the whole grasping system. Note that the Object Recognition and Computing Grasping Point stages required shorter times than those stages which require mechanical movements. On the one hand, the System Initialization is a stage in which the robot manipulator had to move to stay centered above the object to be grasped and the execution times depended on the object scene position. On the other hand, the Grasping Execution time was restricted to the computing of a desired trajectory based on Bezier polynomials, which was chosen to be smooth in order to gradually increase the contact force when the grasping task was carried out. We used 2 s trajectories to better show the behaviour of the system. This trajectory can be shortened but care must be taken not to saturate the DC motor servo-amplifier with the control effort. In conclusion, these execution times were small enough to develop real time applications in industry as will be shown in Section 3.3.

Grasping Task The grasping target object used to show the good performance of the grasping force control strategy proposed is a crystal bottle. Fig. 8 depicts a sequence of key movements in the grasping task. This experiment is focused on the grasping stability and force control performance. The minimum and maximum values of the torque profile applied to grasp the bottle are, respectively, $\min(\Gamma_m(t)) = 0.01 \: [N m]$ and $\max(\Gamma_m(t)) = 0.3 \: [N m]$. The detection threshold was computed through the equation (11), which is $\mu = 35.2 \cdot 10^{-3} \: [N m]$ (these parameters are included in Table 2 to facilitate the comparison with other grasped objects).

The controlled force applied to the crystal bottle is depicted in Fig. 9. The control effort (voltage supplied to the servoamplifier) is depicted in Fig. 10. Note that during the entire manipulation process the control effort was very small, always under 4 [V] while the saturation voltage of the servoamplifier was ±10 [V].

When the robot manipulator was in the desired location to grasp the object, the gripper was given the order to grasp it. In a first instance, an open loop control was applied to close the gripper: a voltage of 0.4 [V] was injected into the DC motor to overcome the static friction and close the fingers. When the fingers contacted the surface of the object, the impact detection algorithm detected the collision and switched the system to the closed loop GPI force control. The instant at which the collision was detected was at 0.604 [s]. Notice that in the very time in which the collision torque was higher than the impact detection threshold, the force control started to regulate the
Table 2. Torque amplitude and impact detection threshold of the objects used in the experiments

| Object       | max($|\Gamma^* m(t)|$) [Nm] | min($|\Gamma^* m(t)|$) [Nm] | $\mu$ [Nm] |
|--------------|-------------------------------|-------------------------------|------------|
| Bottle       | 0.3                           | 0.01                          | $35.2 \cdot 10^{-3}$ |
| Toy          | 0.5                           | 0.01                          | $51.2 \cdot 10^{-3}$ |
| Piece of pottery | 0.1                    | 0.01                          | $19.2 \cdot 10^{-3}$ |
| Banana       | 0.6                           | 0.01                          | $59.2 \cdot 10^{-3}$ |
| Quail egg    | 0.1                           | 0.01                          | $19.2 \cdot 10^{-3}$ |
| Hen egg      | 0.3                           | 0.01                          | $35.2 \cdot 10^{-3}$ |

force applied in the contact. The force was then increased with the Bezier trajectory until 0.3 [Nm] was reached, value which was maintained until the grasping task was completed. To open the gripper, the system again switched to open loop control (at time 9 [s]), which in this case injected a voltage into the DC motor of $-2$ [V] to open fast the gripper.

3.3 Examples of possible industrial applications

**Grasping a Sequence of Objects** The first experiment consisted of the independent grasping of a real time sequence of three different objects: a plastic toy, a piece of pottery and a banana. The control parameters are depicted in Table 2. The objects were introduced into the robot workspace in a sequential manner. First, the toy was introduced; the system recognized it, grasped it, elevated it to show the grasping stability and finally deposited it in the original location. The first sequence of three photograms shown in picture Fig.11 depicts the task carried out with the plastic toy. Secondly, the toy was retired from the workspace and the piece of pottery was introduced in the scene. The system proceeded, as in the previous case, to recognize it and repeat the grasping task. This second grasping task is depicted in the second sequence of photograms in Fig.11. Finally, the piece of pottery was retired again and a banana was introduced to repeat the process, third sequence of photograms in Fig. 11.

**Egg Packaging** This application is an example of object recognition and decision making of the robotic system proposed. Two different objects are in the workspace. The system had to recognize each one and make a decision: packaging each object in its correct box. To show the feasibility of the system
a quail egg and a hen egg, were tested. We thus demonstrate that the system can distinguish among similar shape objects but with some different characteristic (in this case the size). In this experiment the two eggs were both introduced in the scene at the same time. The system recognized both and made the decision of picking up the quail egg, manipulate it and place it in the quail egg box (photograms 1-4 of Fig. 12). Once that this task was finished, the robotic arm moved to the location of the hen egg and manipulated it to place it in the hen egg box (photograms 5-9 of Fig. 12).

4. CONCLUSIONS AND FUTURE WORKS

This paper presents a novel grasping system which combines a gripper with two flexible fingers and a monocular camera vision system. The camera signal is used to estimate the grasping points of a known object, meanwhile, the gripper allows the manipulation of free form objects with different textures, weights, and made of different materials. This grasping flexibility is possible thanks to the characteristic of absorbing energy in the impact with the flexible finger using only one sensor: a pair of strain gauges placed at the base of one finger. This system has been tested experimentally in a 6 DOF robot manipulator to prove its effectiveness performing a sequential manipulation of objects with different materials even when they are very fragile as is the case of eggs. One limitation of the grasping system presented here is its inability to identify ambiguous objects. Our future works will involve the implementation of an active recognition system to solve this problem.

REFERENCES


