# 2

# Active Vision based Regrasp Planning for Capture of a Deforming Object using Genetic Algorithms

Ashish Dutta, Goro Obinata and Shota Terachi
*Nagoya University*
*Japan*

## 1. Introduction

The ability to efficiently grasp an object is the basic need of any robotic system. This research aims to develop an active vision based regrasp planning algorithm for grasping a deforming 2D prismatic object using genetic algorithms (GA). The possible applications of the proposed method are in areas of grasping biological tissues or organs, partially occluded objects and objects whose boundaries change slowly. Most previous studies on robotic grasping mainly deal with formulating the necessary conditions for testing grasp points for static objects (Blake (1995), Chinellato et al. (2003), Galta et al. (2004), Mirtich et al. (1994)). Nguyen (1989) has suggested a strategy for constructing an optimal grasp using finger stiffness grasp potentials. A detailed review of multifinger grasping of rigid objects is presented in Bichi and Kumar (2000). There are few studies on grasping of deformable objects, such as Hirai et al. (2001) in which they present a control strategy for grasping and manipulation of a deformable object using a vision system. In this case the object deforms on application of fingertip forces, the deformation is recorded by a vision systems and based on the amount of deformation the object motion is controlled. Studies relating to searching and tracking of grasping configurations for deforming object are rare. Deforming objects are those that deform by themselves without application of external forces. Mishra et al. (2006) have proposed a method of finding the optimal grasp points for a slowly deforming object using a population based stochastic search strategy. Using this method it is possible to find the optimal grasp points satisfying force closure for 2D prismatic deforming objects. This method minimizes the distance between the intersection of fingertip normals and the object centre of gravity, and maximizes the area formed by the finger tip contact points. However their method fails in cases when the fingertip normals do not intersect at a point (as in case of a square object).

The problem of grasping deforming objects is a very challenging problem as the object shape changes with deformation. Hence the optimal grasp points have to be continuously found for each new shape. This process of recalculating the fingertip grasp points due to object shape change, slide or roll is called regrasping. The best method of determining the change in shape of an object is by using a vision system. A vision system not only captures the new shape but can also be used to track a moving object. The main objectives of this research are to use a vision system to capture the shape of a deforming object, divide the



Source: Vision Systems: Applications, ISBN 978-3-902613-01-1
Edited by: Goro Obinata and Ashish Dutta, pp. 608, I-Tech, Vienna, Austria, June 2007

object boundary into a number of discrete points (pixels) and then find the optimal grasp points satisfying form closure. As the object changes shape the new shape is continuously updated by the vision system and the optimal grasp points are found. Once the solution for the first frame is obtained this solution is used as the initial guess in subsequent cases for finding the optimal grasp points. This enables faster solutions for later frames recording the deformation of the object. It is assumed that the object deforms slowly, the contact between the fingertip and the object is frictionless and the fingers do not cause deformation of the object. Hence four fingers are required to grasp a prismatic object in 2D. Simulations were carried out on 200 synthetic shapes that deformed slowly and the optimal grasp points found. An experiment was conducted in which a deforming object was simulated by a piece of black cloth that was held from below and deformed. The shape change of the cloth was captured by a camera and for each shape the optimum gasp points were obtained. Experimental results prove that the proposed method can be used in real time to find the optimal grasp points for a deforming object. In section 2 the algorithm used for determining the optimal grasp points is explained. The procedure for obtaining the regrasp solutions is discussed in section 3. Simulation results are explained in section 4, while the experimental setup is given in section 5. The experimental results are shown in section 6 and conclusions are drawn in section 7.

## 2. Determining optimal grasp points using GA

This section describes the concept of form closure using accessibility angle and the algorithm used to determine the optimal form closure grasp points. Form closure is a purely geometric constraint under which an object cannot escape in any direction after it is grasped by a number of frictionless fingertips. The mathematical conditions for obtaining form closure of an object by a multifinger hand are as given below (Yoshikawa, 1996):

$$T = \begin{bmatrix} a_1 \ldots\ldots\ldots a_n \\ p_1 \times a_1 \ldots p_n \times a_n \end{bmatrix} \alpha = D^T \alpha \tag{1}$$

Where T is the external forces and moment vector (total of six) acting at the centre of the object, $a_i$ is the unit normal directed into the object at the fingertip contact points, $p_i$ is the position vector of the fingertip contact points on the object, and $\alpha = [\alpha_1 \ldots\ldots \alpha_n]$ are the fingertip forces (n=total number of fingers). A necessary and sufficient condition for form closure are (i) rank D=6 and (ii) equation (1) has a solution such that $\alpha > 0$ (all forces are positive). Hence to obtain form closure in 3D we need seven contact points and in 2D we need four contact points. In this research we have proposed a geometrical method for finding the form closure grasps based on the concept of accessibility angle. The freedom angle ($\phi$) of a two dimensional objects is defined as the angular region along which it can be translated away from the contact. The concept of freedom angle is as shown in Figure 1(a). It shows an object grasped with three contact points, for each individual contact point we define the direction (range) along which the object can move away from the contact points. The three freedom angles are as marked in the figure. Figure 1(b) show that after combining all the freedom angles there is still an angle left (escape angle) from where the object can escape. Hence it can be derived that the object is not in form closure. Figure 2(a) show the same object with four contact points and the corresponding freedom angles. In figure 2(b) it can be seen that all the total 360˚ are covered and hence the object is in form closure. If 'x' represents the position vector at a point on the object surface then the freedom angle " $\phi_i$ "at that point is computed as:

$$\phi_i = \{\angle(x_{i+1} - x_i), \angle(x_i - x_{i-1})\}$$

$$\psi = \{\phi_1 \cap \phi_2 \cap .....\phi_n\}$$

The accessibility angle is the common angle between all the freedom angles. The accessibility angle ($\psi$) (Sharma et al. (2006)) is calculated as shown in Figure 2(b). An object is in form closure if the accessibility angle is the null set (or escape angle is zero). This means that there is no way the object can move away (translate or rotate) from the gripper points.

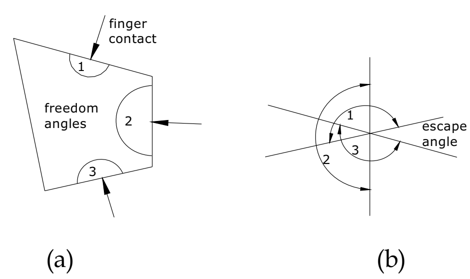



Figure 1. (a) The freedom angles showing the directions in which the object can move with respect to each individual finger contact, (b) direction in which the object is free to escape

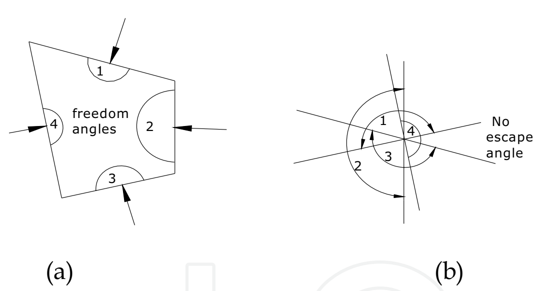



Figure 2. (a) The direction in which the object can move with respect to each finger contact, (b) the object cannot escape in any direction as there is no escape angle

Hence the method essentially searches for the best from closure grasp points by comparing all sets of four grasp points satisfying conditions of form closure. As the object boundary is made up of a very large number of points (pixels) and a good form closure grasp is desired this search is quite complex. Also as the search involves discrete points an efficient method to solve the problem is to use genetic algorithms.
GA is used to maximize an objective function subject to constraints. A traditional GA, like Gordy (1996), performs three operations on a population of genomes i.e. selection, crossover and mutation. The length of the binary string is equal to the number of discrete points on the object boundary. If a finger is present at a particular point then '1' is present or it is '0'. The binary string encoding the object boundary is as shown in Figure 3 .

0100000000010000000000100..........0000001000000

Figure 3. Binary string (1 means finger present at that location, 0 means no finger present)

Selection is performed to choose better individuals for cross-over. In Gordy (1996), selection is performed using the roulette wheel procedure. If an individual has better fitness, its probability of getting selected is more. In this selection process, cumulative sum of the fitness of all individuals in the population is calculated and normalized by dividing it with the total sum of the fitness of individuals in the population. A random number between 0 and 1 is chosen. If that number lies within the span of normalized cumulative sum of any individual, that individual is selected. An individual can be selected multiple times based on how fit it is. Once the number of individuals equal to the original population size is selected into the mating pool, a single point crossover is performed. A split point is randomly generated and contents of the two individuals are swapped about this split point. Post crossover, mutation is performed with a very low probability. Each individual is scanned through and a gene is randomly mutated if the probability is lower than the mutation probability. Thus, a new population of vectors is obtained and individual fitness is computed. Finally, eliticism is invoked by replacing the worst individual of the new population with the best from the previous population.

The two conditions needed to be satisfied in order to get a good grasp are: a) the fingertips must be capable of resisting all the external forces and moment acting on the object and b) the placement of the fingers should be such that the moment applied is minimum. The proposed objective function maximizes the moment that the fingertips can resist, by considering different combination of fingertip positions taking four discrete points at a time. The constraint uses accessibility angle to ensure that all the feasible solutions satisfy form closure. If the accessibility angle is zero it means that the object is in form closure. In case the constraint is not met, a very high penalty is placed on the function value that eliminates the non-feasible solutions. The objective function used is given by:

$$f = \left( \frac{|M_{cw}| + |M_{ccw}|}{N_{cw} - N_{ccw} + \varepsilon} \right) + \sum_{i=1}^{4} U_i V_i^T \tag{2}$$

The first part of the right side of equation (2) is the objective function while the second part is the constraints. $M_{cw}$ is the total clockwise moment and $M_{ccw}$ is the total anticlockwise moment applied by the fingers. These two terms ensure that the individual moments are maximized in both the clockwise and anticlockwise directions. This indirectly leads to minimum normal forces at the contact. $N_{cw}$ and $N_{ccw}$ are the number of fingers applying clockwise and anticlockwise moment. This ensures that the fingers are placed all around the object and do not get concentrated at one location. A term 'ε' having a small value (0.01) has been added to ensure that the denominator does not become zero when both the anticlockwise and clockwise moments are equal. The constraints used are $U=[u_i]$ and $V=[v_i]$ which are given as :

1. u1=0 If total number of contact points is four , else u1=1;
2. u2=1 If area formed by contact points equals zero, else u2=0;
3. u3=0 If Both clockwise and anticlockwise moments exist, else u3=1;
4. u4=0 If object is in form closure, else u4=1;

'$v_i$' = - $1x10^{20}$ (i=1…4), hence if the constraints are not met the function takes a very high value and that particular solution is rejected. The normal function values for feasible grasp points are approximately $6.5x10^3$ and hence the large negative value of '$v_i$' ensures that non-feasible solutions are rejected. In this way feasible solutions move towards feasible space and the non feasible solutions are eliminated.

## 3. Regrasp of deforming objects

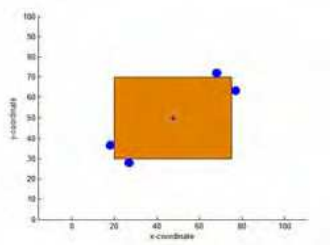
This section describes how regrasp solutions are obtained as the object deforms. The optimal grasp points depend on the geometry of the object and the solution for the first frame is obtained using a random guess as the initial solution in the GA routine. Hence this solution takes the largest time for convergence. Once the initial solution is obtained, it is used as the initial guess in the next search. As the object deforms the vision system obtains the next shape of the object in terms of pixel boundary points. These discrete points form the new GA design variable and the earlier solution is used as an initial guess. The object deforms very slowly and hence the shape changes slowly. This property ensures that the new grasp points are in the neighborhood of the earlier optimal grasp points and are not random. Hence it was found that the time for finding an optimal solution rapidly decreases in subsequent searches once an initial solution is found.

## 4. Simulation

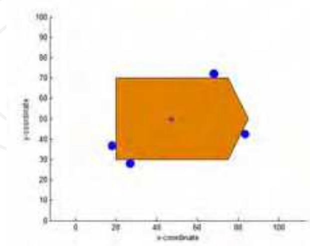
The proposed regrasp algorithm has been tested on 200 types of synthetic shapes that undergo slow deformation. Simulations were performed on a 1.86 GHz laptop computer with 512 RAM. We have assumed that the objects deforms slowly as the algorithm takes time (secs) to obtain a solution. An example of slow deformation is a rectangle that can slowly expands each side to become an octagon etc. However a rectangle cannot suddenly become a circle. This assumption is practical as an expanding object like a balloon does not change shape suddenly. The simulation was made in Matlab in which a closed object was constructed using straight lines segments. Each time a side of the object was expanded by dividing it into two or more segments and expanding it. In case of real objects the sides can be approximated by straight lines and hence this method can be used to approximately simulate deformable objects. A few sample cases of an object expanding are shown in Figure 4. As shown, an object (a) deforms to object (b), then (c) etc. by expanding one side at a time (all intermediate steps are not shown). The GA parameters used are:

1. Size of generation 60
2. Crossover 0.80
3. Mutation 0.12
4. Maximum number of iteration 5000
5. Maximum number of gains before stopping 1000

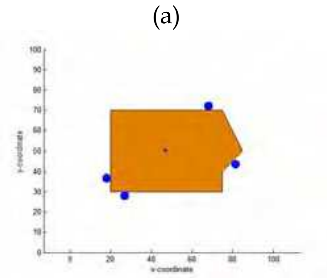
The time required to find the regrasp points was found in two ways for each object. In Case-I the time was found independently for each deforming object. There was no initial guess solution supplied to the algorithm. In Case II the time to get a solution was found by supplying the earlier solution as an initial guess to the algorithm. In both the cases for the same objects the four optimal finger positions were same but the time to get a solution was different, as shown in Table 1. It was seen that in Case II the time required to get each solution was very much less than in Case I. This can be explained by the fact that as the object deforms the optimal grasp points are not random but are related to the shape of the object.



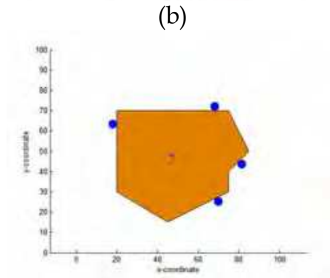
(a)



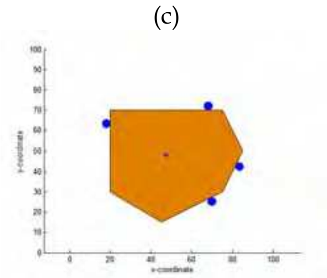
(b)



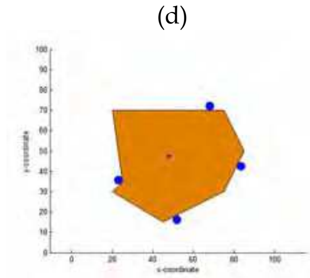
(c)



(d)



(e)



(f)

Figure.4. (a-f) Optimal grasp points for a slowly deforming object (the fingertip contact points are indicated by solid circles) x and y axis are in mm

| Object No. | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Case I Time(secs) | 53 | 75 | 61 | 67 | 74 | 73 |
| Case II Time (Sec) | 53 | 37 | 23 | 31 | 21 | 24 |

Table 1. Comparison of time taken for calculating optimal grasp points for Case I and II

## 5. Experimental details

The experimental system (as shown in Fig 5) consists of a vision camera, a slowly deforming object, a PC with image processing software and a laptop PC on which the GA based algorithm runs. The deforming object was a piece of black cloth that was deformed by holding it from below and deforming it. The image was captured by a black and white CCD camera model 'SB-6B' manufactured by Wireless Tsukamoto Co., Japan. The camera can capture frames at a rate of 30 fps and each frame has a resolution of 100x100 pixels. The number of pixels determines the total number of discrete points on the object boundary that are considered by the binary string in the GA algorithm. Hence increasing the number of pixels in a frame increases the resolution of the picture but it also increases the time required for computation as the length of the binary string will be longer. It was found that using 100x100 pixels per frame gave satisfactory results and the image was captured at intervals of 10 seconds. The sequence of images captured of the deforming object is shown in Figure 6. Thresholding was used to segment each image into the foreground and background based on different pixel intensities. The input was a grey scale image and the output was a binary image representing segmentation. The boundary of the segmented image was obtained by using edge detection as shown in Figure 7. After the edge is detected the coordinate of all the pixels with reference to a reference coordinate frame was found. These coordinates of the object boundary pixels are then passed on to the GA based algorithm for calculating the best grasp points.

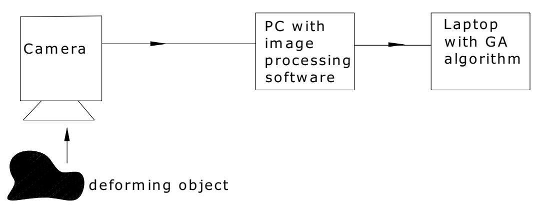
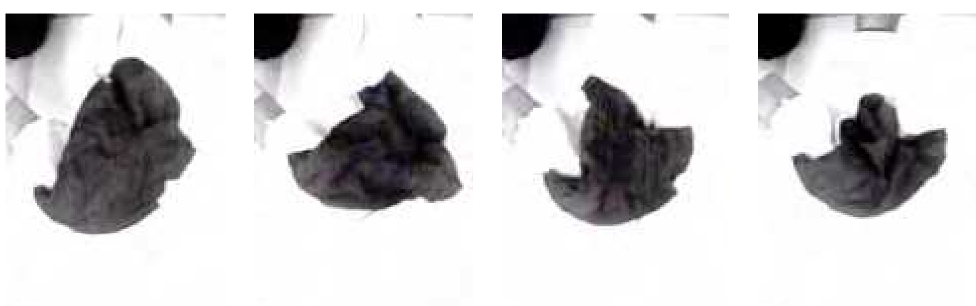
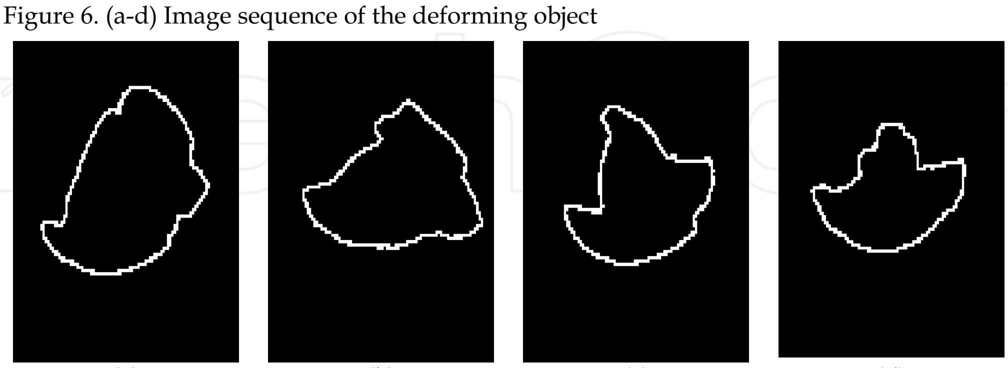



Figure 5. The experimental setup



(a) (b) (c) (d)

Figure 6. (a-d) Image sequence of the deforming object



(a) (b) (c) (d)

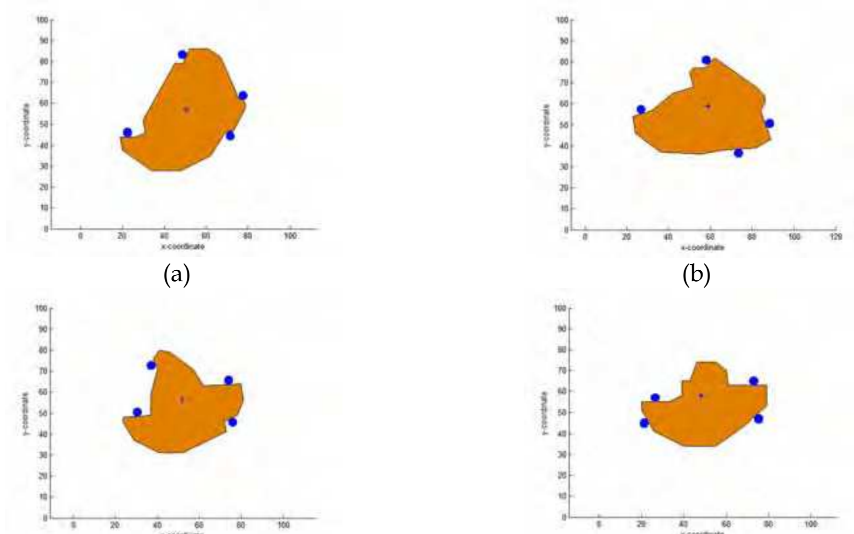
Figure 7. (a-d) The edge of the deformed objects

## 6. Experimental results

The pixel coordinates of the boundary of the deforming object as obtained by the image processing software was input to the GA based grasping algorithm. Computations were performed on a 1.86 GHz laptop computer with 512 RAM. The results of the experiments are as shown in Figure 8. Each figure corresponds to the frame obtained by the vision camera in Figure 6. The GA parameters used in the algorithm are same as those used in the simulations. The optimal grasp points for the first frame were obtained by using a random initial guess solution in the GA algorithm. Subsequent solutions were obtained by using the previous solutions as the initial guess. Table 2 shows the time required to get each solution and it is again seen that the first frame required the most time.

| Object No. | (a) | (b) | (c) | (d) |
|---|---|---|---|---|
| Time (secs) | 46 | 22 | 27 | 25 |

Table 2. Time required for computing the grasp points



(a) (b) (c) (d)

Figure 8. (a-d) The optimal grasp points for each of the deforming objects (x and y coordinates in mm)

### 6.1 Real time application

One of the potential uses of the proposed method is an application in which an autonomous multifinger robot with a vision camera has to capture a deforming object. In such applications the time from image capture to obtaining the optimal grasp points has be done in real time (in a few seconds). As shown earlier, the time required to get the first solution was the highest as it depended on parameters like, initial guess solution, total number of iterations and the total iterations before stopping if gains are not exceeded. Hence faster solutions can be obtained by dynamically tuning these parameters. Figure 9 shows two solutions for the same object obtained by varying the GA parameters. The final objective function values indicated that solution (a) with function value of $6.8\text{x}10^3$ (iteration 5000 and number of gains before stop 200) is better than solution (b) with function value $6.1\text{x}10^3$ (iteration 1000, number of gains before stop 100). The solutions were obtained in 6 seconds and 2 seconds respectively. Hence it is possible to obtain faster solutions in real time by dynamically tuning the GA parameters based on required function value or number of iterations, and also using a faster computer for running the algorithm. It is however not clear how the function value varies with different shapes and parameter values. In future, we hope to study how to adjust the GA parameters dynamically to obtain the fastest solutions in real time.

(a) (b)

Figure 9. (a-b) Finger points for the same object for different functional values

## 7. Conclusion

The main contributions of this research are an effective vision based method to compute the optimal grasp points for a 2D prismatic object using GA has been proposed. The simulation and experimental results prove that it is possible to apply the algorithm in practical cases to find the optimal grasp points. In future we hope to integrate the method in a multifinger robotic hand to grasp different types of deforming objects autonomously.

## 8. References

Bicchi, A. & Kumar, V. (2000). Robot Grasping and Control: A review, *Proceedigns of the IEEE International Conference on Robotics and Automation*, pp. 348-353, ISSN 1050 4729.

Blake, A. (1995). A symmetric theory of planar grasp, *The International Journal of Robotics Research*, vol. 14, no. 5, pp. 425-444, ISSN 0278-3649.

Chinellato, E., Fisher, R.B., Morales, A. & del Pobil, A. P. (2003). Ranking planar grasp configurations for a three finger hand, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1133-1138, ISSN 1050 4729.

Gatla, C., Lumia, R., Wood, J. & Starr, G.(2004). An efficient method to compute three fingered planar object grasps using active contour models, *Proceedigns of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3674-3679, ISBN 07803-8463-6.

Gordy, M. (1996) A Matlab routine for function maximization using Genetic Algorithm. *Matlab Codes: GA*.

Hirai, S., Tsuboi, T. & Wada, T. (2001) Robust grasping manipulation of deformable objects, *Proceedings of the IEEE International Conference on Assembly and Task Planning,* pp. 411-416, ISBN 07803-7004.

Sharma, P., Saxena, A. & Dutta, A. (2006). Multi agent form closure capture of a generic 2D polygonal object based on projective path planning, *Proceedings of the ASME 2006 International Design Engineering Technical Conferences*, pp.1-8, ISBN 07918-3784.
Mishra T., Guha, P., Dutta, A. & Venkatesh K. S. (2006). Efficient continuous re-grasp planning for moving and deforming planar objects, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2472 – 2477, ISSN 1050 4729.
Mirtich, B. & Canny, J. (1994). Easily computable optimum grasps in 2D and 3D, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 739-747.
Nguyen, V.D. (1989). Constructing stable force-closure grasps, *International Journal of Robotics Research*, vol. 8, no. 1, pp. 26-37, 0278-3649.
Yoshikawa, T. (1996). Passive and active closures by constraining mechanisms, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1477-1484, ISBN 07803-2988.

**Vision Systems: Applications**

Edited by Goro Obinata and Ashish Dutta

ISBN 978-3-902613-01-1
Hard cover, 608 pages
**Publisher** I-Tech Education and Publishing
**Published online** 01, June, 2007
**Published in print edition** June, 2007

Computer Vision is the most important key in developing autonomous navigation systems for interaction with the environment. It also leads us to marvel at the functioning of our own vision system. In this book we have collected the latest applications of vision research from around the world. It contains both the conventional research areas like mobile robot navigation and map building, and more recent applications such as, micro vision, etc.The fist seven chapters contain the newer applications of vision like micro vision, grasping using vision, behavior based perception, inspection of railways and humanitarian demining. The later chapters deal with applications of vision in mobile robot navigation, camera calibration, object detection in vision search, map building, etc.



**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821