Model Based Evolutionary Object Recognition System

Yohannes Kassahun and Gerald Sommer
Christian Albrechts University of Kiel, Institute of Computer Science, Olshausenstr. 40, 24098 Kiel, Germany

Abstract. In this paper we present a new model based object recognition system that recognizes an object and determines its pose simultaneously. The recognition is independent of translation, scale and rotation. In many evolutionary recognition systems, genetic algorithms are used either in searching the correspondence between a given model and the data or in searching the best geometric transformation that brings a large number of model points into alignment with the data. In this work, we make use of genetic algorithm to do the recognition and pose estimation task simultaneously. We have tested the algorithm on both artificial and real images and found the algorithm to perform very well.

1 Introduction

Object recognition is one of the most important, yet least understood, aspect of visual perception [12]. For many biological vision systems, the recognition and classification of objects is spontaneous, natural activity. In contrast, the recognition of common objects is still way beyond the capability of artificial systems, or any recognition system proposed so far. In this paper, we want to show that a model based evolutionary object recognition system is a step forward in solving this difficult problem.

The proposed system has the following features which makes it different from other recognition systems:

1. A common unifying approach for both recognition and pose estimation, treating both subtasks at the same time.

2. Finding a global optimal solution in space of solutions. The space of solutions is made up of the Cartesian product of objects to be recognized and their poses.

The presented system recognizes 2D plane objects in 3D and determines their 3D pose for which the model is memorized in the system. By pose, we mean the transformation needed to map an object model from its own inherent coordinate system into agreement with the sensory data [3]. The system extracts and saves contour points of objects from training images as their contour models.

2 Genetic Algorithms

Genetic Algorithms (Holland 1975) [6] are a popular form of evolutionary computation which operate on a population of artificial chromosomes by selectively reproducing the chromosomes of individuals with higher performance and applying random changes [10]. An artificial chromosome (genotype) is a sequence of symbols that encodes the characteristics of an individual (phenotype). The performance of an individual is evaluated using a criterion called fitness function. Higher fitness values indicate better performance.

A typical genetic algorithm starts with a population of randomly generated chromosomes. Each chromosome is decoded, one at a time, its fitness is evaluated, and three genetic operators: selective reproduction, crossover, and mutation are applied to generate a new population. This process is repeated until a desired individual is found, or until the best fitness value in the population stops increasing.

Selective reproduction guides the population toward areas that contain better solutions. Crossover and mutation operators maintain the variation between individuals so that children do not become identical copies of their parents. This variation between individuals helps the population to keep on improving from generation to generation.

3 Previous Work

There is a variety of approaches to object recognition that has been proposed. One can divide them in general in two groups: non-correspondence or global matching and correspondence or feature matching. A global matching involves finding a transformation that fits a model to an image without first determining the correspondence between individual parts or features of the model and the data. Several of this approaches base themselves on a simple geometric parameters such as area, perimeter, Euler number, moments of inertia, Fourier or other spatial frequency descriptions, tensor measures and so on. Representative examples include works of Hu, Murase and Nayar, and Zahn [9, 7, 14]. These methods are efficient but are sensitive to occlusion [3]. On the other hand, feature matching procedures try to find the correspondence between local features of a model and the data and then determining the transformation for a given correspondence between the model and the data. These methods are robust against occlusion but they are not as efficient as global matching procedures [3, 11, 12] since they have to solve the correspondence problem for every model that is going to be assumed.

Genetic algorithms are mostly employed in searching the best correspondence between a given model and the data or in searching the best geometric transformation that brings a large number of model points into alignment with the scene [1, 2]. In our implementation, we use genetic algorithm to search for a model that best fits a given scene and simultaneously determines the transformation that brings the model into alignment with the scene. The presented system belongs to the global matching group.

4 The System

Our system has three parts. The first part performs a simple visual grouping based on predefined colors of an object. The second part does the recognition and pose estimation of the perceived object and the third part is used to acquire new models of new objects that are not known to the system.

4.1 Visual Grouping

There is a considerable evidence that prior to the recognition, the early processing stages in the visual cortex are involved in grouping and segmentation operations on the base of image properties, such as proximity, collinearity, similarity of contrast, color, motion, texture, etc [12]. The grouping and segmentation process attempts to organize the image into coherent units, and to decide what parts of an image belong together.

Our visual grouping algorithm tries to group objects based upon predefined colors of an object. That means, we try to detect and group objects having certain predefined colors. The visual grouping is done as follows:

1. Read in an RGB color image.
2. Convert the image into its HSV (Hue, Saturation, Value) image format.
3. Generate a binary image with pixels marked for the object of interest with predefined colors.
4. Use graph-search or an equivalent algorithm to group pixels that come from the same object, and extract image regions (components).
5. Get the centroid and number of pixels of each of the components in the image.

4.2 Recognition and Model Acquisition Systems

The recognition system starts by generating a binary contour image from the components image using a standard contour following algorithm. Then it generates the complement image of the contour image and calculates the distance transform [5] of it. The distance transform assigns a distance value of zero to all contour points and a positive distance value to all non-contour points as the distance to the closest contour point depending on the distance metric used. In calculating the distance transform, we have assumed that the complement of the contour image is toroidal rather than planar. Figure 1 shows the steps involved in getting the distance transform of an input image to the recognition system. The distance transform will be used in determining the fitness value of an individual.

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

Figure 1: The steps involved in getting the distance transform taking letter A as an example. (a) The input image. (b) The components image which is the result of the visual grouping algorithm. (c) The contour image generated by the standard contour following algorithm. (d) The complement of the contour image. (e) The distance transform of the complement of the contour image.

In this work, we have used a chromosome (an individual) shown in figure 2. The chromosome has four genes. The first gene codes the index of an object in the database of the contour models, which are already acquired by the system. The second, third and fourth genes code the rotation of the object about z-axis, y-axis and x-axis respectively relative to the non-rotated model of the object. The centroid of the model is taken as the origin of the coordinate system and the z-axis is assumed to be perpendicular to the image plane of the camera. The length of the chromosome is determined by the number of contour models in the system and the number of bits used in representing the orientation of an object. The number of bits that code the rotation of an object about an axis determines the resolution of the rotation angle coded by the chromosome.

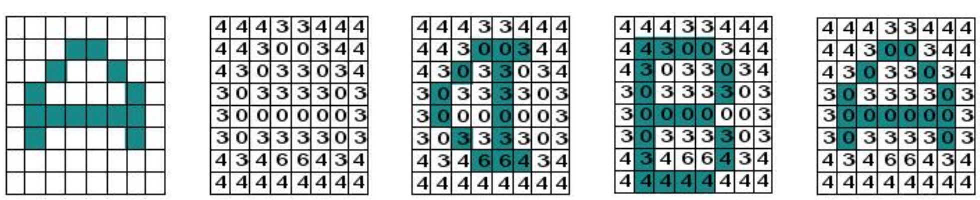
Index of an object.	Rotation about z-axis.	Rotation about y-axis.	Rotation about x-axis.

Figure 2: A chromosome: The first gene codes the object to be recognized. The second, third and fourth genes code the orientation of the object.

The fitness function, f, of a chromosome given by equation (1) is used to determine the fitness value of an individual. It is defined as the negative sum of the distance values, d_i, in the distance transform of the complement image of the contour image. The index i runs for all points of the translated, scaled and rotated contour model of an object coded by the chromosome. A zero value of the fitness function means a perfect fit of the model to a particular component in the image since the distance values for the contour points in the distance transform are zero.

$$f = -\sum_i d_i \tag{1}$$

Figure 3 shows a contour image of an 8×8 hypothetical input image, the distance transform of the complement of the contour image and contour models coded by three different chromosomes which are projected onto the distance transform. The fitness value of the chromosomes coding the contour models shown in figure 3(c), 3(d) and 3(e) are -31 , -39 and 0 respectively. Figure 3(e) shows a case where the model fits to the component perfectly.



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

Figure 3: Determination of the fitness value of a chromosome. (a) The contour image. (b) The distance transform of the complement of the contour image. (c), (d) and (e) Contour models coded by different chromosomes and projected onto the distance transform.

An individual is evaluated as follows:

1. Decode the individual. That is, get the type of the contour model with its orientation coded by the chromosome.

2. Calculate the scale factor between the rotated contour model and the component using,

$$s = \sqrt{\frac{A_i}{A_m}} \tag{2}$$

where A_i is the area of the component in the image and A_m is the area of the orthogonal projection of the rotated contour model onto the plane parallel to the image plane of the camera.

3. Translate the contour model to the centroid of the component. Then rotate it using the orientation angle obtained when decoding the chromosome and scale it using the scale factor calculated above.

4. Determine the fitness value of the individual using equation (1).

The position (centroid) of a component and the number of pixels of the component are determined by the visual grouping algorithm. The position of a contour model and the scale factor between the contour model and the component are determined while an individual (chromosome) is being evaluated. The search for the best model with best orientation that fits a given component in the image is made by the genetic algorithm. That is, the recognition and pose estimation problems are solved simultaneously by the recognition system. The main purpose of the genetic algorithm is to solve the optimization problem of finding the maximum value on the fitness landscape. Figure 4 shows an example of a fitness landscape of an input image containing letter W, which is rotated by 180 degrees about z-axis. The coordinate at which the global maximum value of the fitness function occurs gives the index of the object recognized with its rotation angles. The genetic algorithm uses its genetic operators to move on the fitness landscape.

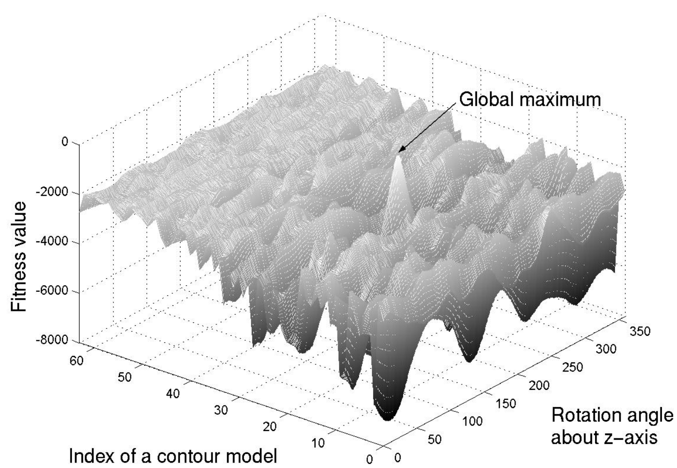


Figure 4: A fitness landscape of an input image containing letter W rotated by 180 degrees about z-axis taking the centroid of letter W as the origin of the coordinate system. The global maximum of the landscape occurs at the index of the contour model of letter W in the database of the contour models and at rotation angle of 180 degrees.

The algorithm of the recognition system has the following two important steps:

1. Perform a complete genetic run for a component until the best individual is found.
2. Decode the best individual and return the object recognized with its pose.

The recognition system can be easily parallelized in three different ways. First, we can have copies of the algorithm that operate in parallel on different components detected in the input image so that the recognition of all detected components can be done simultaneously. Second, we can run a parallel implementation of the algorithm for one component and apply it to the rest of the components sequentially. The algorithm can be parallelized since genetic algorithms let themselves easily parallelized [13]. This is specially useful if we want to increase the recognition rate of our system. Third, we can combine the above types of parallelizations and benefit an increase in both the recognition rate and speed of recognition.

The model acquisition system is used to acquire and save a new model for a new object for which the model does not exist in the system. It uses the above visual grouping algorithm to identify and locate the object. Then it uses a contour following algorithm to extract the contour points of the object and samples the contour points evenly so that the resulting model has the same number of points as the other models in the system. This is important because the fitness value of an individual depends on the number of contour points of a model coded by the chromosome (individual).

5 Experiments and Results

For all the experiments, we have set the parameters of the genetic algorithm as shown in table 1. The system is made to acquire models of 64 objects some of which are shown in figure 5. The objects include the English alphabets, digits, free hand drawn objects and some German traffic signs. The resolution of the system in coding the rotation angle of the object about an axis is 0.7045 degrees.

Table 1: Parameters of genetic algorithm used

Number of individuals in the population	500
Crossover probability	0.2
Mutation probability per bit	0.05
Selection scheme	Truncation selection
Number of bits per gene coding the index of an object	6
Number of bits per gene coding a rotation angle about z-axis	9
Number of bits per gene coding a rotation angle about y-axis	9
Number of bits per gene coding a rotation angle about x-axis	9

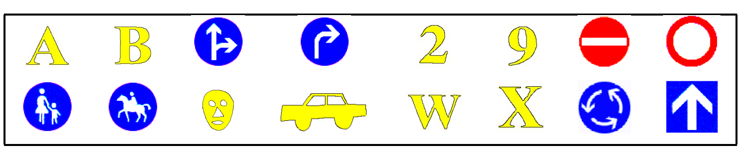


Figure 5: Some of the objects whose models are acquired by our system.

5.1 Experiment on Artificial Images

In this experiment, we study the effect of random noise on the recognition rate and pose estimation of the system. Each input image is subjected to different levels of random corruption between 0% to 50%. The percentage of the noise levels shows the ratio of the number of pixels that are corrupted by the random noise to the total number of pixels in the image. A uniform random function generator is used to select a pixel in the input image. The color of the selected pixel is made to change to some other color which is not used in the visual grouping algorithm.

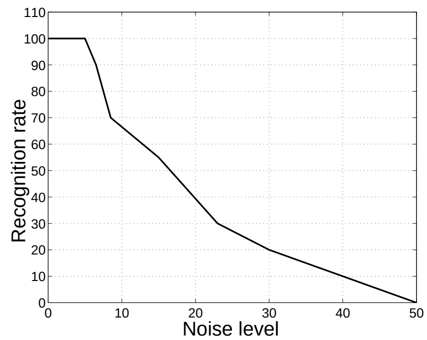


Figure 6: Recognition rate obtained for different random noise levels.

Figure 6 shows the result obtained for different noise levels added to the test images. As the test images, we have used translated, scaled and rotated versions of all images whose models are acquired by our system. The translation, scale and rotation parameters of each image is randomly selected. As can be seen from figure 6, the system has 100 percent recognition rate for noise level up to 5%. This makes the system robust to be used for real life applications.

In order to investigate the pose (rotation) estimation capability of our system, we have generated rotated versions of one of the German traffic signs shown in figure 7. The non-rotated version of this traffic sign was already acquired by our system. We have measured the average pose estimation error of 10 experiments, which are run for 0%, 1%, 5% and 10% noise levels and for all rotation angles shown in figure 7.

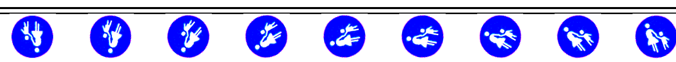
Image									
Angle	0	20	40	60	80	100	120	140	160
Image									
Angle	180	200	240	260	280	300	320	340	360

Figure 7: Rotated versions of the German traffic sign used to indicate a pedestrian path.

Each of the experiments are done by starting the recognition system with different random seed values. As can be seen from figure 8, the system is able to estimate the pose of an object with absolute maximum pose estimation error of 5.5 degrees for noise level of 10%. This makes the system again robust for estimating the pose of an object.

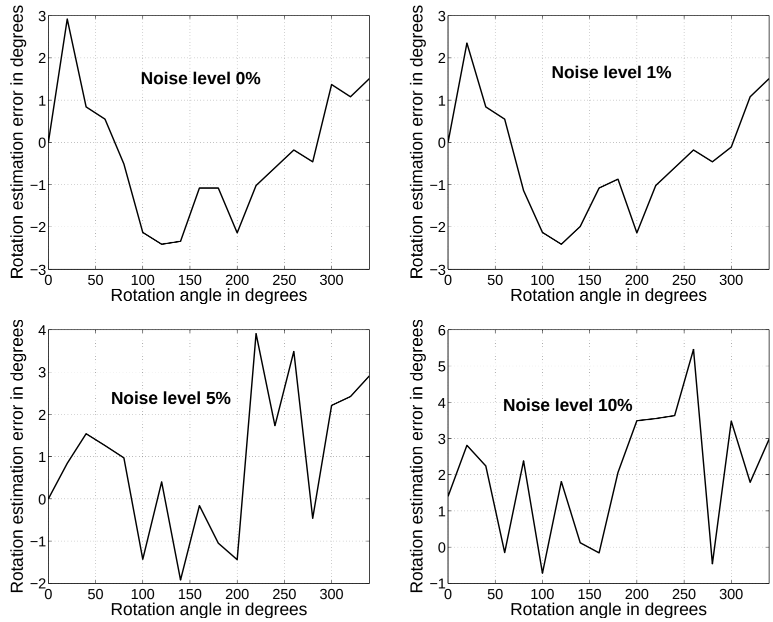


Figure 8: Pose estimation error for 0, 1, 5 and 10 percent noise levels.

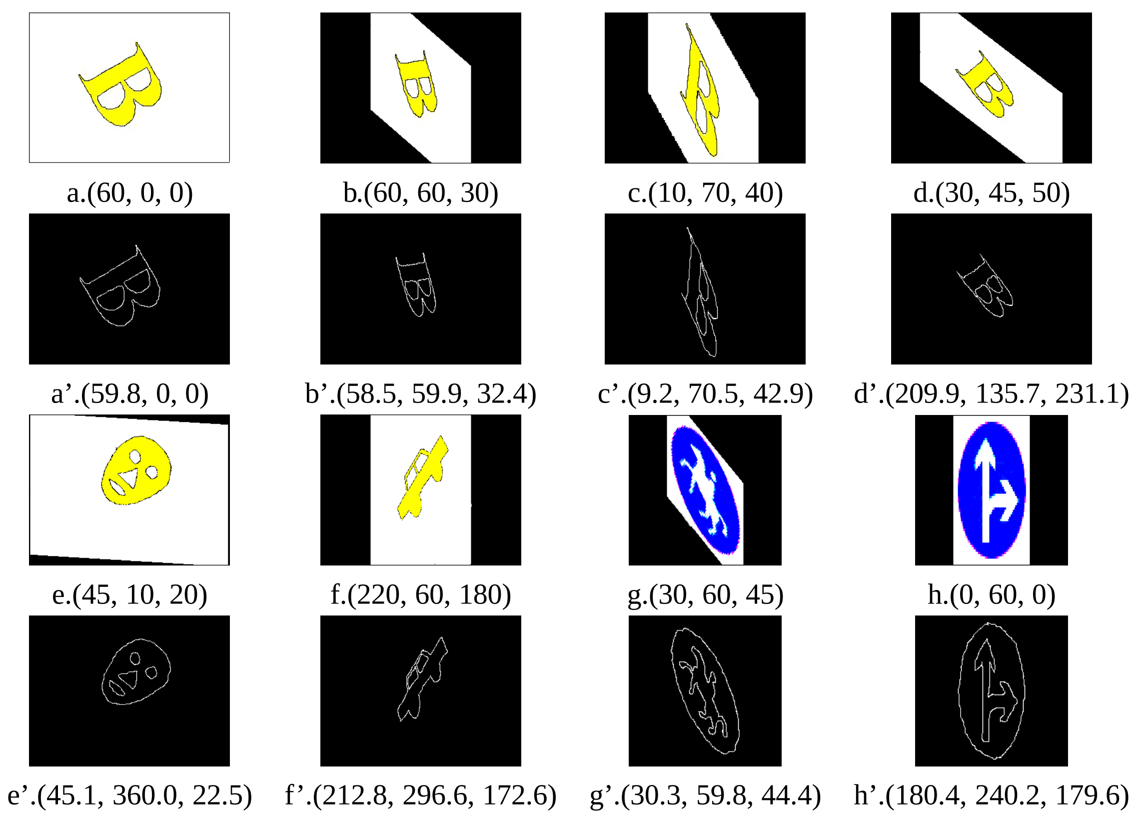


Figure 9: Sample recognition results obtained from experiments on artificial images. The numbers in the brackets below each of the images show the rotation angles of the images about z, y and x-axis respectively. The contour images show the recognition result obtained.

Figure 9 shows sample recognition result obtained on artificial images. As can be seen from the figure, the recognition system has found in (d), (e), (f) and (h) equivalent rotation angles, which will result in the same image projected onto the image plane of the camera.

5.2 Experiment on Real Images

Figure 10 shows sample recognition results obtained in recognizing real 2D images. For the experiments, we have used plane images printed on sheet of papers and some real German traffic signs.

For real images, we have obtained a recognition rate of 95%. This recognition rate can be increased even to a higher level if one uses a parallel implementation of the recognition system as stated in section 4, where different recognizers work on one component at the same time.

We have tested and implemented our system on a standard 800 MHz computer running the Linux operating system. On the system, our algorithm was able to process 2 input images per second if the image contains one component. The recognition time of the system increases linearly with the number of components detected in the input image.

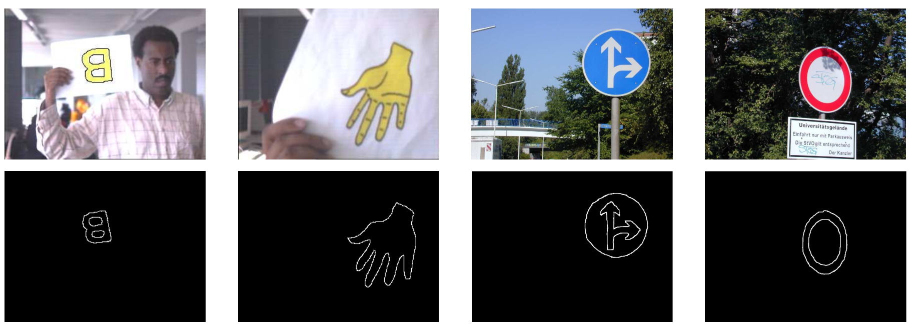


Figure 10: Sample recognition results obtained in recognizing real 2D images. The contour images below each of the images show the type of the object that is recognized.

6 Conclusion and Outlook

In this paper, we have presented a new evolutionary object recognition system, which is independent of translation, rotation and scaling, for recognizing 2D plane objects in 3D and determining their 3D poses simultaneously. From our experiment, we have concluded that one can use the power of genetic algorithms to search for the best fit in the space of objects and their pose. The presented system is suitable for applications requiring the recognition of navigation symbols such as traffic signs.

The system has one limitation. It is not robust with respect to occlusions. If the object is occluded, the system may not recognize the object correctly. In the future, we have a plan of designing an evolutionary object recognition system that handles the occlusion problem. This can be done by redesigning the visual grouping algorithm to use other features such as object boundaries.

This work can be extended to the recognition and pose estimation of 3-D objects. For 3D objects instead of having a database of 2-D contour models, we will have a database of 3-D wire models. This will be a matter of future results.

Acknowledgment

This work is sponsored by the German Academic Exchange Service (DAAD) under grant code R-413-A-01-46029 and the VISATEC project under the EC contract no. IST-2001-3422, which are duly acknowledged. We would like also to thank Dr. Josef Pauli for giving us important suggestions and for commenting our paper. We thank also Oliver Granert for having important discussions with him.

References

[1] Bebis G., Louis S., Varol Y. and Yfantis A. Genetic object recognition using combinations of views. IEEE Trans. Evolutionary Computation. 2(6):132-146 (2002)

[2] Beveridge, R. and Riseman, M. How easy is matching 2D line models using local search? EEE Trans. Pattern Analysis and Machine Intelligence. 6(1)(1997)

[3] Eric, W. and Grimson, L. Object Recognition by Computer. The MIT press (1990)

[4] Foley, D., van Dam, A., Feiner, K. and Hughes, F. Computer Graphics. Principles and Practice. Addison-Wesley Publishing Company (1996)

[5] Haralick M. and Shapiro G. Computer and Robot Vision, Volume II. Addison-Wesley Publishing Company (1993)

[6] Holland, J.H. Adaptation in Natural and Artificial Systems. Univ. of Michigan Press, Ann Arbor, Mich.(1975)

[7] Hu, M.K. Visual pattern recognition by moments invariants. RE Trans.Inform. Theory, 8:179-187 (1962)

[8] Louis, S., Bebis, G., Uthiram, S. and Varol, Y. Genetic search for object identification. Evolutionary programming 199-208 (1998)

[9] Murase, H. and Nayar, S. Visual learning and recognition of 3D objects from appearance. In Int. Journ. of Comp. Vision, 14: 5-24 (1995)

[10] Nolfi, S. and Floreano, D. Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-Organizing Machines. The MIT Press (2000)

[11] Rosenhahn B. Pose Estimation Revisited. PhD Thesis, Christian Albrechts University of Kiel. (2003)

[12] Ullman, S. High-level Vision. The MIT press (1996)

[13] Whitley, D. A Genetic Algorithm Tutorial. Colorado University. Technical Report cs-93-103. (1993)

[14] Zahn, C.T. and Roskies R.Z. Fourier descriptors for plane closed curves. IEEE Trans. Computers, 21(3): 269-281 (1972)