Optimization of Industrial, Vision-Based, Intuitively Generated Robot Point-Allocating Tasks Using Genetic Algorithms

Ambrocio Loredo-Flores, Emilio J. González-Galván, J. Jesús Cervantes-Sánchez, and Alvaro Martínez-Soto

***Abstract*—Current industrial robot-programming methods require, depending on the task to be developed, an elevated degree of technical ability and time from a human operator, in order to obtain a precise, nonoptimal result. This correspondence paper presents a methodology used to generate an optimal sequence of robot configurations that enable a precise point-allocating task applicable, for instance, to spot-welding, drilling, or electronic component placement maneuvers. The optimization process starts from a nonoptimal, initial sequence designated intuitively by a human operator using an easy-to-use interface. In this correspondence paper, intuitive programming is considered as the process of defining, in a computer graphics environment and with a limited user knowledge of robotics or the industrial task, the sequence of motions that enable the execution of a complex industrial robotic maneuver. Such an initial sequence is later followed by a robot, very precisely, using a vision-based, calibration-free, robot control method. Further robot path optimization is performed with a genetic algorithm approach. An industrial robot, which is part of the experimental setup, was used in order to validate the proposed procedure.**

***Index Terms*—Genetic algorithms, robot vision systems, simulated annealing, user interfaces.**

I. Introduction

One of the tendencies of current robot investigations is related to the development of more intuitive programming techniques. Intuition is defined as the immediate apprehension of truth, or supposed truth, in the absence of conscious rational processes [1]. An ideal intuitive system would be the one in which high-level instructions are given by a user. Under this approach, supplying these instructions would not require an excessive mental effort from a human operator, for instance, by using simple indications via a graphical interface or even by adopting different body postures that the system will later imitate. After these instructions are given, the robot is expected to perform its task in an autonomous fashion. For instance, a recent research effort associated with this idea is the programming-by-demonstration approach where a system observes what a human is doing. Later, the system generates robot programs to mimic the actions of the human [2].

Nowadays, most industrial robot programming is performed by a human operator using a teach pendant. The operator places the tool held by the robot over the surface of the workpiece, in a process known as online programming. As the life cycle of a product shortens, a frequent change in the program of the robot used during the manufacturing process is required. Therefore, a more flexible programming method, capable of easily adapting to the changing geometric conditions of the product, is essential. One approach has emphasized the development of methods for offline programming [3]. Such a programming method uses a computer in order to simulate a virtual environment where a manufacturing cell and a robotic system are graphically presented. Several works associated with the intuitive programming of robots [4], [5] involve the use of force/torque sensors. In another reference, a system for intuitive robot programming based on the operator's knowledge and experience is integrated with a computer system [6]. In this reference, an operator is integrated as a robot programmer in a programming process based on an advanced man–machine interaction that takes advantage of the operators's knowledge about the robot application.

The research presented herein proposes an alternative methodology for intuitive, industrial robot programming. This methodology requires a human operator to designate the path that the tool held by the robot must follow. The path designation is performed directly over images taken from the workpiece where the task will be performed. Other than this, the operator is not expected to have any training in robotics or in the particular industrial, point-allocating task that the robot will perform like spot-welding, drilling, etc. Later, a sequence of actual robot configurations is obtained using a vision-based, robot control method that does not require mechanical or vision-system calibration. The proposed vision-based method is known as camera space manipulation (CSM) [7]–[9]. This method has been successfully applied in tasks that involve a precise, 3-D robot positioning.

Once an initial sequence of robot configurations is defined, an optimization algorithm is used to improve such a sequence. The algorithm receives, from the vision-based methodology mentioned earlier, a sequence of internal robot configurations that enable an accurate placement of the tool over the workpiece. The optimization technique, based on a genetic algorithm (GA) approach, produces an ordered sequence of robot configurations that reduces, in the case of the research presented here, the amount of robot joint coordinate motion. Among others, one advantage of reducing the amount of joint coordinate motion is that it leads to a reduced wear of the robot. It also enables a reduction in the cycle time required to perform the task. The procedure presented herein takes into account the existing restrictions in the mobility of the tool among different regions of the workpiece, as well as the required 3-D location and orientation of the tool, demanded by the particular task.

The rest of this correspondence paper is organized as follows. Section II defines the way in which the sequence of robot configurations is defined in an intuitive fashion, using a calibration-free, vision-based method. In Section III, the optimization of the sequences of robot locations is described, including the basic structure of GAs and the particular objective function employed in the optimization algorithm. Experimental results are presented in Section IV and conclusions are included in Section V.

II. Intuitive Generation of Initial Sequence of Configurations

Fig. 1 shows a flow diagram with the general methodology for generating an initial sequence of robot configurations, in an intuitive manner. Such a diagram depicts the case in which a given path is followed only once over an arbitrary surface.

The first step in the flow diagram consists of placing the workpiece inside the robot's workspace. The workpiece does not have to be placed in a specific position. The only requirement is that the region over the surface of the workpiece where the robot is going to perform its task is in view of at least two of the cameras used to control the maneuver, as depicted schematically in Fig. 2. The user selects which of the images provided by the control cameras is the one with the best perspective of the workpiece. This image is loaded into a GUI like the one illustrated

Manuscript received December 15, 2006; revised May 25, 2007 and September 24, 2007. This work was supported in part by the National Council for Science and Technology of Mexico (CONACYT) and in part by the Fund for Research Support (FAI) of Universidad Autónoma de San Luis Potosí (UASLP). This paper was recommended by Associate Editor Y. Jin.

A. Loredo-Flores and E. J. González-Galván are with the Centro de Investigación y Estudios de Posgrado (CIEP), Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí, San Luis Potosí SLP 78290, Mexico (e-mail: aloredof@uaslp.mx; egonzale@uaslp.mx).

J. J. Cervantes-Sánchez is with the Facultad de Ingeniería Mecánica, Eléctrica y Electrónica (FIMEE), Departamento de Ingenieria. Mecánica, Universidad de Guanajuato, Salamanca Gto 36700, Mexico (e-mail: jecer@salamanca.ugto.mx).

A. Martínez-Soto is with Metalsa S. de R.L., Proeza Automotriz, San Luis Potosí SLP 78090, Mexico (e-mail: alvaro.martinez@metalsa.com.mx).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2008.923886

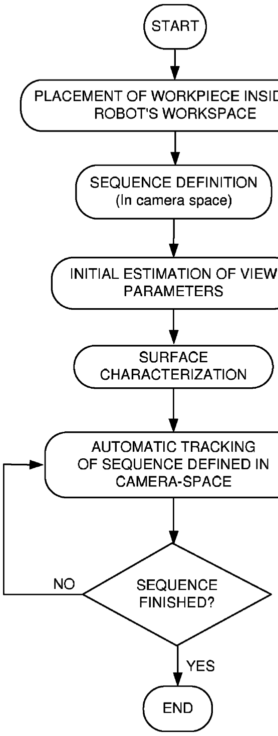

START
PLACEMENT OF WORKPIECE INSIDE OF ROBOT'S WORKSPACE
SEQUENCE DEFINITION (In camera space)
INITIAL ESTIMATION OF VIEW PARAMETERS
SURFACE CHARACTERIZATION
AUTOMATIC TRACKING OF SEQUENCE DEFINED IN CAMERA-SPACE
SEQUENCE FINISHED?
NO
YES
END

Fig. 1. Flow diagram of a user-designated, point-allocating task.

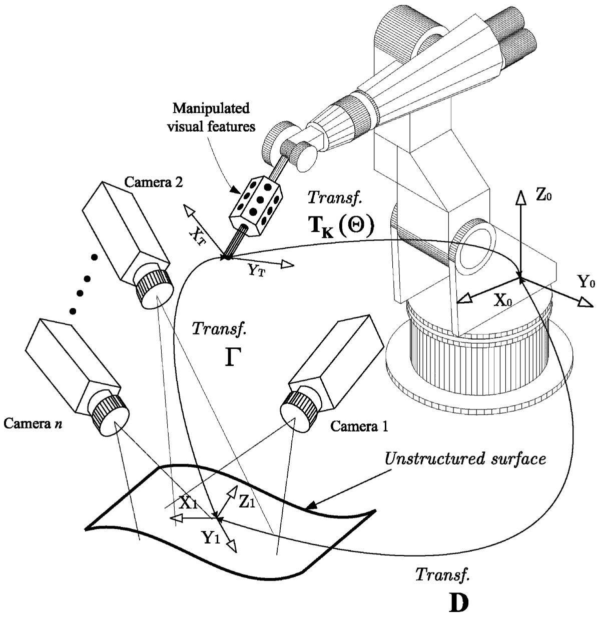
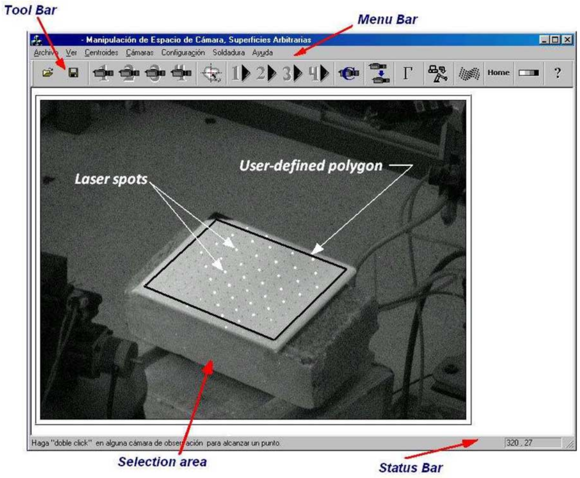


Fig. 2. Control cameras and reference frames associated with the maneuver.

in Fig. 3, which is used to help the user in designating the sequence in which the task will be carried out. A computer mouse is employed by the user in order to designate, point-to-point, the sequence over the image presented in the GUI. Such a sequence is likely not optimal. It may be optimal by coincidence as it depends exclusively on the ability and experience of the human operator.

Prior to the execution of the actual point-allocating task, the determination of six view parameters, independent for each camera, is required. The view parameters appear in the camera model used in our system, which is presented in (1). The evaluation of the view parameters defines a nonlinear, algebraic relationship between the robot configuration and the appearance, in camera space, of a number of manipulated visual features placed on the tool held by the robot, as explained in [9]. This relationship is referred to as camera space kinematics. In this correspondence paper, camera space refers to the 2-D images obtained from the control cameras used in the maneuver.



Haga "doble click" en alguna cámara de observación para alcanzar un punto.

320 , 27

Fig. 3. GUI showing laser spots projected over workpiece surface and user-defined polygon used to delimit the surface.

In the experiments reported herein, an orthographic camera model was used. Such a model, detailed in [9], establishes a relationship between the camera space location of a point (x_c, y_c) and the corresponding 3-D location (x, y, z) as

$$\begin{aligned} x_c &= f_x(x, y, z; \mathbf{C}) \\ y_c &= f_y(x, y, z; \mathbf{C}) \end{aligned} \tag{1}$$

where $\mathbf{C} = [C_1 \cdots C_6]^T$ are six view parameters. The evaluation of these parameters is performed independently for each camera, by using a nonlinear estimation process, described in detail in [9]. These view parameters are continuously updated using samples of both the 3-D location of manipulated features, obtained with the nominal kinematic model of the robot, and the corresponding camera space location of these features. These manipulated features are illustrated schematically in Fig. 2

The next step in the flow diagram of Fig. 1 is the surface-characterization process. This process is explained in detail in [10] and involves the projection of structured laser lighting in the form of a matrix array of laser spots, as illustrated in Fig. 3. Structured lighting has, among others, the advantage of not producing a permanent mark over the surface. Due to the large amount of laser spots that can be projected over the surface of the workpiece, a way to limit the amount of information associated with the location of such laser spots in camera space is desired. With this purpose, the user is required to define, also in camera space, a polygon that encircles a given region of interest, as depicted in Fig. 3. An algorithm used to restrict the laser spots that fall inside this user-generated region was implemented, based on the ideas presented in [11] and [12].

The estimation of the nominal, physical location of the projected laser spots, like those depicted in Fig. 3, is performed by taking into account the view parameters associated with each control camera. Such a linear estimation process is performed by minimizing the following scalar function, for the 3-D coordinates (x_i, y_i, z_i) corresponding to each spot

$$\phi_i = \sum_{j=1}^{n_c}\left\{\left[x_{c_i}^{(j)} - f_x(x_i, y_i, z_i; \mathbf{C})\right]^2 + \left[y_{c_i}^{(j)} - f_y(x_i, y_i, z_i; \mathbf{C})\right]^2\right\}. \tag{2}$$

The 3-D location of each spot is determined with respect to the system $X_0Y_0Z_0$ attached to the base of the robot, as presented in Fig. 2. In the previous expression, n_c represents the number of cameras that detect a given laser spot, whose camera space coordinates are given by $(x_{c_i}^{(j)}, y_{c_i}^{(j)})$.

The robot configuration that enables an accurate placement of a tool over the nonmanipulable surface is evaluated by using a nonlinear estimation scheme. Such an estimation scheme consists of the minimization of the following function, for the angular coordinates of the robot included in $\mathbf{\Theta}$

$$\begin{aligned}\gamma(\mathbf{\Theta}) &= \sum_{j=1}^{n_c}\sum_{i=1}^{n_t(j)}\left\{\left[x_{t_i}^{(j)} - f_x\left(r_{x_i}^{(j)}(\mathbf{\Theta}), r_{y_i}^{(j)}(\mathbf{\Theta}), r_{z_i}^{(j)}(\mathbf{\Theta}); \mathbf{C}^{(j)}\right)\right]^2\right.\\ &\quad \left. + \left[y_{t_i}^{(j)} - f_y\left(r_{x_i}^{(j)}(\mathbf{\Theta}), r_{y_i}^{(j)}(\mathbf{\Theta}), r_{z_i}^{(j)}(\mathbf{\Theta}); \mathbf{C}^{(j)}\right)\right]^2\right\}\end{aligned} \tag{3}$$

for n_t target points corresponding to each of the n_c control cameras that are part of the vision system. The ith target point for the jth sensor has a camera space location defined by $(x_{t_i}^{(j)}, y_{t_i}^{(j)})$, which corresponds to a physical location $(r_{x_i}^{(j)}, r_{y_i}^{(j)}, r_{z_i}^{(j)})$. In the case of the system described here, the target points correspond to the laser spots projected over the surface.

According to (3), the task's final objective would be matching a set of targets in camera space, associated with a reference frame fixed on the tool, with the same number of targets projected over the surface. The relationship among the different coordinate systems associated with the maneuver is represented schematically in Fig. 2. As seen in this figure, consecutive intermediate robot configurations between an initial location to task culmination can be obtained by changing $\mathbf{\Gamma}$ from an initial value to a unit matrix when the task is finished.

The target points (laser spots) used in the estimation of the joint coordinates of the robot are referenced to a coordinate system placed over the work surface. As described earlier, a sequence of objective points can be defined in camera space by a user, using a GUI such as the one depicted in Fig. 3. Such objective points are used to define a coordinate system over the workpiece surface, depicted in Fig. 2. A matrix $\mathbf{D}$ is used to establish a relationship between this system and the one at the base of the robot. For the experiments reported herein, this transformation matrix is defined by translating the origin of the system at the base of the robot to the desired objective point. Next, its orientation is prescribed by rotating the translated system around an axis normal to the plane formed by the translated axis Z_0 and axis Z_1, so they become coincident.

A 6-DOF industrial robot like the one used in our experimental setup is redundant in the case of tasks like spot-welding or drilling. This means that an infinite number of robot configurations could be applied in order to successfully finish the task. In this context, a solution to the problem of determining a robot configuration in the presence of redundancies has been tackled by several researchers [13] and is not considered in this research. In the case of our experiments, the task is finished when the coordinate system attached to the tool coincides with the coordinate system placed over the unstructured surface, as depicted in Fig. 2. This approach eliminates the possibility of obtaining multiple solutions for the robot configuration.

With the procedure described earlier, an initial, nonoptimal sequence of robot configurations $(\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_n)$ that enable an accurate location of the tool with respect to surface is obtained. The use of the GUI allows for a simple and intuitive mean to define the robot configuration. Such an interface, which simply requires the positioning and clicking with a computer mouse, contrasts with the procedure used in industrial applications that involves the use of a teach pendant. In this case, the user can use the GUI to define the orientation of the coordinate system placed over the unstructured surface, avoiding the multiple solutions for the robot configuration described earlier. Another contrasting characteristic is the level of training that a user requires, from a basic level needed to operate the GUI, to a relatively advanced degree of training required to operate a teach pendant.

III. Optimization of Intuitive Robot Sequence

Research work on the optimization of robot sequences has focused on the optimization of mechanical assemblies [14]–[16], robot assemblies [17], [31], and sequences of robotic tasks [18].

The problem of determining the optimal sequence of robot configurations can be considered as an extension of the travel–salesperson problem (TSP) [19]. TSP is one of the most widely cited problems in combinatorial optimization. In this problem, a salesperson must travel once to a finite number of cities, returning to the initial city at the end of the trip. Given the distances between cities, the objective of the optimization problem consists of finding the optimal path that the salesperson must follow in order to minimize the traveled distance. An approach used to adapt the TSP to robotic systems consists of minimizing the cycle time that the robot requires for traveling to a number of configurations. Another approach considers the joint displacement of the robot, which is the case of the system described here.

During the past decades, several solutions to the TSP have been proposed. Some of these solutions, like simulated annealing (SA) [31] or GAs [20]–[23], have also been oriented toward reducing cycle time in the case of robotic systems. In the case of the research presented herein, a GA approach is used to find an optimal sequence of robot configurations in the sense that a minimal joint displacement is sought. It is important to mention that GAs do not guarantee a global minimum of the objective function. It is considered optimal in the sense that it is obtained from the convergence of a GA procedure, as explained later.

One important characteristic of the research proposed here is that the initial robot path is evaluated with the real system, using a vision-based method. In this context, in [22], an approach for optimizing the sequence for a welding application based on GA is presented. The initial sequence of robot configurations required for the application of the optimization algorithm is obtained through a commercial software, known as interactive graphics robot instruction program (IGRIP). A drawback of this approach is that, when applied to actual welding tasks, robot trajectories must be adjusted in order to compensate for the nonmodeled effects of real-world systems like link-bending, actuator flexibility, or deficient robot calibration [8]. For example, it is shown in [24] that an adjustment on the order of 7.2 mm is required, in the case of a spot-welding robot, when an offline program is introduced in a real-world system. A static deflection compensation is presented in [25] for the case of large glass-handling robots. A technique for estimating the accuracy of positioning is presented in [26], in the presence of elastic flexibility in the actuators. Such an adverse effect can be exacerbated when the robot carries a heavy load, as is the case of modern welding servo-guns, designed to increase productivity and quality of welding in automotive industries. If this mass changes, for instance, when a heavy load is grasped or released, the vision system would be capable of adjusting to this event. The adjustment is performed by taking samples of the manipulated features in both camera space and physical space, using the nominal kinematic model of the robot. The samples taken under the new load conditions are used to update the camera space kinematics described earlier. Other works, like the one presented in [23], are based on an inverse kinematic approach used to produce a first robot path. This approach has the same nonmodeled effect problem that requires robot path adjusting when the actual, real-world robotic tasks are performed.

In the research presented herein, since the initial path is accurately obtained using a noncalibrated, vision-based approach, no adjustment is required regarding the nonmodeled effects cited earlier. The optimization is based on a minimal robot joint path required to track the proposed path, using a GA approach. The following section provides additional details regarding the procedure developed for the optimization.

A. Genetic Algorithms

GAs are based on the genetic processes of biological organisms. The development of populations of living organisms takes place according to the principles of natural selection and survival of the fittest. By imitating this process, GAs can provide solutions for real-world problems if they are adequately coded [27]. Basic ideas of GA were presented in [28], which were later expanded in order to be applied to the problem of combinatorial optimization [29]. The GA process involves the creation of an initial population, aptitude evaluation, selection, and genetic operations required for reproduction. In the approach presented here, each chromosome of the population [30] is adapted as a path consisting of the sequence of robot configurations that reaches all points required by the task.

Like other optimization algorithms, GA starts by determining the objective function and control parameters, required to achieve convergence. The most important control parameters that affect the optimization process are explained [30] as follows.

1) *Population size*: The size of the population affects the number of chromosomes and then the amount of genetic material available for the search. A reduced population size covers a small area of the search space, resulting in a limited representative sample that reduces the possibility of finding a global minimum. On the other hand, a large population size increases processing time. In general, selecting the population size depends on the nature and complexity of the problem. In the algorithm developed here, several population sizes were tested. It was found experimentally that a population size of 500 was adequate in the case of a 6-DOF manipulator.
2) *Crossover rate*: This parameter affects the frequency in which the crossover operator is applied to the chromosomes of the population in order to generate a new one. This rate is a factor between 0.7 and 1.0.
3) *Mutation rate*: This parameter determines the probability that a gene's value in a chromosome would be changed. Mutation introduces new areas of the unexplored search space. However, the mutation rate should not be too high because it increases the randomness in the search. The mutation rate is usually less than 0.4.

There is no mathematical proof of the fact that the GA, once converged, will reach a global minimum. Besides, there is still debate on which is the best way to terminate the algorithm. In many cases, a maximum number of generations is established ahead of time, although this may be an ineffective measure as it is not possible to define *a priori* the number of generations required to achieve convergence. In the research presented herein, the condition to terminate the algorithm consists of defining a certain number of occasions in which a given optimal chromosome solution is repeated. Such a number of repetitions must be sufficiently large in order to ensure a consistent solution.

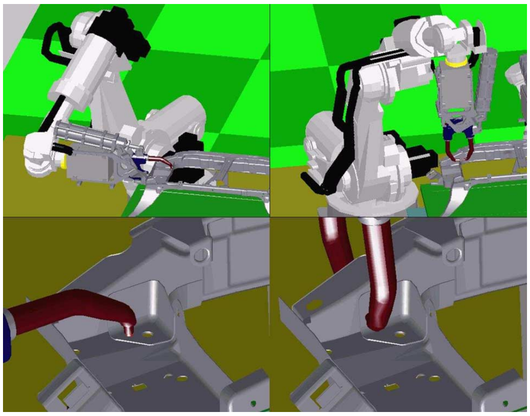


Fig. 4. Welding application involving a small 3-D displacement corresponding to a large robot joint motion. Two points with a small distance between them (bottom pictures). Corresponding motion of the robot (top pictures).

B. Objective Function

The joining of metallic parts with spot-welding is one of the most important processes in the assembly industry. The working principle in this process consists of the passing of electric current through the metals to be joined. Because of the high resistance in the spot where the joint is going to be performed, the temperature rises. This effect combined with a certain amount of pressure ensures the welding of two materials. Spot-welding machines can be stationary or, as is the case of the system presented herein, they can be mounted on a robot.

The typical criterion used in solving the TSP consists of optimizing the traveled distance of the salesman, by minimizing the scalar function f

$$f = \sum_{i=1}^{n} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \qquad (4)$$

for n cities where the 2-D coordinates of the ith city are given by (x_i, y_i).

In our case, a small 3-D distance between spots may require a large joint displacement in the robot. This is shown schematically in a case illustrated in Fig. 4. In this figure, pictures in the bottom show a spot-welding gun reaching two points that are close to each other. However, reaching these two points requires a large robot motion, which is represented in the figures in the top of Fig. 4. Therefore, an objective function that considers the joint displacement of the robot is preferred, as discussed in [22].

As mentioned earlier, a sequence of joint configurations $(\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_n)$ that enables an accurate placement of the welding tool with respect to the surface is available. This was obtained by using a calibration-free, vision-based method, with the help of a user who selects the maneuver objectives using a simple-to-use, intuitive GUI. For the optimization of the sequence, the following objective function is defined

$$\phi = \sum_{i=1}^{n} \sum_{j=1}^{m} \sqrt{(\theta_{i,j} - \theta_{i+1,j})^2} \qquad (5)$$

where n represents the number of joint configurations that enable a successful maneuver culmination, while m represents the number of degrees of freedom of the robot. In this case, the ith configuration is given by $\mathbf{\Theta}_i = [\theta_{i,1}, \ldots, \theta_{i,m}]^T$. If different regions exist in the pieces, for instance, in the case that a set of welding spots are separated by a wall, as illustrated in Fig. 4, or when a particular heat distribution is

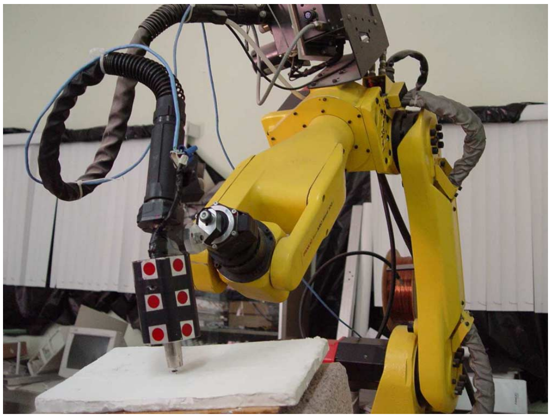


Fig. 5. Fanuc ArcMate 100i industrial manipulator.

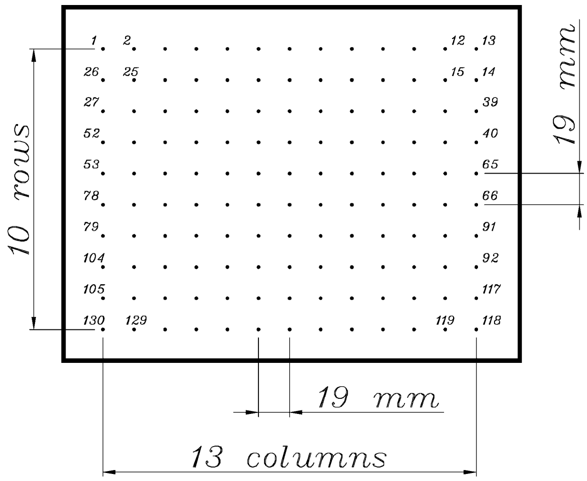


Fig. 6. Testing surface.

desired, the following objective function is proposed:

$$\phi = \sum_{i=1}^{n}\left[\sum_{j=1}^{m}\sqrt{\left(\theta_{i,j}-\theta_{i+1,j}\right)^2}+\lambda\left(\mu_i-\mu_{i+1}\right)^2\right]. \qquad (6)$$

The previous equation includes a value μ_i, associated with the region where a spot is placed. In this case, μ_i is given the same numerical value for spots located in the same region. Finally, λ represents a weight given to the restriction between regions. Both μ_i and λ can be defined according to the problem, depending on the physical restrictions among the different regions in which a given surface can be divided. For the experiments reported herein, a single region was considered, and therefore, only the expression in (5) was used.

IV. Experimental Work

In this section, a description of the experiments used to test the proposed procedure is presented. Such a procedure is intended for the intuitive user designation and optimization of a robotic task involving the placing of a tool over an arbitrary surface over a finite number of locations. An industrial Fanuc ArcMate 100i robot, like the one shown in Fig. 5, was used in the experiments. The control algorithms were developed in a PC where the communication between this device and the robot was performed using the transmission control protocol (TCP)/IP. A GUI, developed with MS Visual C++, was built and tested in the computer used in the experiments. Such an interface, depicted in Fig. 3, consists of a large selection area where an image of the workpiece is displayed. The user selects a number of target points in this area, which are later used as objective points in the vision-based algorithm used to control the robotic arm.

A flat surface made of clay was used in the experiments. A sheet of paper with a printed array of 10 × 13 dots, each with a diameter of 2 mm, as shown in Fig. 6, was placed over the surface made of clay. While the dots are not necessary to perform the task, they serve as a means of documenting the precision achieved. These dots are numbered according to a sequence that is also indicated in Fig. 6. The surface is made of clay in order to avoid damage to the system in case that the tool held by the robot hits the surface.

The surface characterization process was performed by using a commercial 5 mW, 635 nm, Lasiris laser diode. This device has an exchangeable head capable of producing a matrix array of 7 × 7 laser beams. With this laser diode, a large number of spots can be placed over the surface. Such a large number of laser spots, shown schematically in Fig. 3, enable an accurate representation of the surface, which in turns allows for an accurate placing of the tool. The centroid of each of the projected laser spots in each one of the images obtained with the Sony MPT304 cameras used in the experiments was obtained with a dedicated software, based on ideas presented in [32] and [33]. Images are digitized using a Data Translation DT3155 frame grabber, installed inside the PC compatible computer where the control algorithms were developed.

Each one of the 130 dots shown in Fig. 6 was used as an objective point for the vision-based method used for robot positioning, as described in Section II. The designation of each of these locations is performed by the user, using the GUI depicted in Fig. 3. In this case, a total of 130 robot configurations were available after this exercise. It is important to mention that it is not required to determine the configuration of each one of the 130 points in the surface, as only those involved in the path that we would like to optimize are required. The determination of the 130 robot configurations was performed in order to test the effectiveness and accuracy of the vision-based method for robot positioning. It is also important to mention that for the flat testing surface, the vision-based method converged for all the 130 target points. There are, however, instances in which this may not occur. For instance, in the case of complex, curved surfaces, problems like deficient image correspondence or obscured laser spots may be present. This may cause convergence problems in the estimation-based, vision-based method.

For each one of these positioning tasks, the error in camera space was found by considering the difference between the designated objective point and the location in the same space of the tip of the tool held by the robot. This location is determined by considering the known geometry of the tool, the nominal kinematic model of the robot given by matrix $\mathbf{T_K}(\mathbf{\Theta})$, depicted in Fig. 2, and the orthographic camera model in (1). These equations were evaluated with the most updated set of view parameters available for the camera. Typically, the error in camera space is less than 1 pixel. This fact implies that a large source of error in the final, 3-D positioning of the tip of the tool with respect to the surface, comes from the designation of the target point within the GUI. It can be mentioned that one effective measure to reduce this error consists of including an algorithm for enlarging the image presented to the user. Once this algorithm is in place, maneuver objectives in camera space will be defined with a precision of a fraction of a pixel. Without the application of this measure, the average error in the positioning of the tip of the manipulated tool with respect to the designated point on the surface is 1.50 mm for all the 130 tested points, with a standard deviation of 0.55 mm. In this case, the measurement of the physical error was performed by considering the angular configuration of the robot at the converged location, and the one required to reach the center of the objective point. Each configuration, combined with the kinematic

TABLE I
COMPARISON OF TOTAL CYCLE TIME BETWEEN INITIAL, USER-DESIGNATED PATH AND OPTIMAL, ACCORDING TO THE GA PROCEDURE

I	II	III	IV	V	VI	VII	$VIII$
1	3	47.836	2.546	47.836	2.546	0	0
2	3	47.346	2.578	47.346	2.578	0	0
3	3	49.172	2.67	49.172	2.67	0	0
4	7	66.047	3.536	66.047	3.536	0	0
5	7	53.382	2.867	49.797	2.779	7.20	3.17
6	7	60.982	3.408	60.982	3.408	0	0
7	10	66.974	3.443	61.654	3.364	8.63	2.35
8	10	71.858	4.101	65.624	3.771	9.50	8.75
9	10	66.745	3.625	61.105	3.402	9.23	6.55
10	15	81.212	4.547	70.132	3.937	15.80	15.49
11	15	100.716	5.436	78.623	4.293	28.10	26.62
12	15	88.24	4.492	75.764	4.094	16.47	9.72
13	21	111.021	5.951	96.194	5.419	15.41	9.82
14	21	94.487	6.199	84.138	6.13	12.30	1.13
15	21	112.082	6.163	98.751	5.595	13.50	10.15
16	27	127.771	7.542	111.982	6.172	14.10	22.20
17	27	132.777	7.829	117.294	6.732	13.20	16.30
18	27	133.833	7.979	117.528	6.783	13.87	17.63
19	35	158.134	8.913	125.503	7.123	26.00	25.13
20	35	165.619	9.462	136.021	7.773	21.76	21.73
21	35	165.759	9.679	125.969	7.646	31.59	26.59

I: path #; II: number of points in each path; III: angular displacement required from the robot in order to follow the sequence indicated by the user (in degree); IV: cycle time required (in seconds) to follow the path designated by the user; V: optimized angular displacement (in degrees) of the robot ; VI: optimized cycle time (in seconds); VII: reduction in angular displacement (in percent); VIII: reduction in cycle time (in percent).

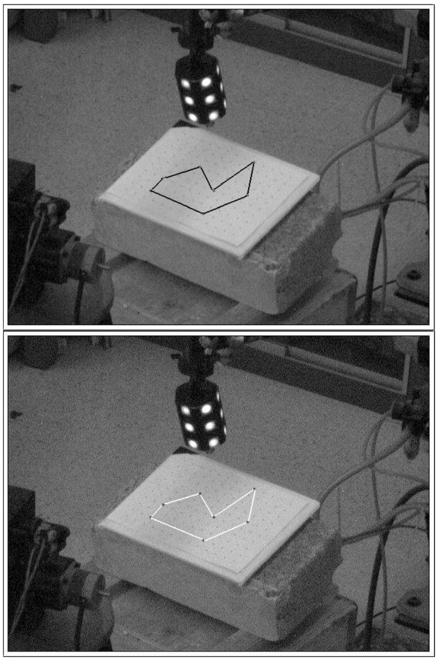


Fig. 7. Comparison of paths consisting of seven points. (Top) Path selected by the user. (Bottom) Optimized path. Note that this case corresponds to path 6 of Table I.

model of the robot, provides a measure for the physical error. As for the alignment error between the tool and the surface, it can be mentioned that a welding application may have a relatively large tolerance to this error. According to [37], this tolerance is in the order of 5–10°. For the experiments reported here, an error less that 5° was observed. This was estimated by using a square placed next to the tool and the surface, when the robot reaches its target configuration.

Once the 130 configurations of the robot are available, the next step of the experimental procedure designed to test the proposed methodology consisted of defining different paths, each with a different number of points. The user is required to do their best in designating the shortest path to complete the task. The designation is again performed using the GUI depicted in Fig. 3. In this case, the robot configurations used to optimize the path are those obtained from the vision-based procedure.

A GA is used with the purpose of optimizing the user-designated sequence. The parameters used for such a GA implementation are as follows: population size, 500; crossover rate, 0.8; mutation rate, 0.2. It will be shown in a typical case detailed later that the termination condition consisting of observing the repetition of a solution with a minimum objective function is adequate. The results of this testing procedure are included in Table I. In this case, 21 paths were tested. The paths consisted of 3, 7, 10, 15, 21, 27, and 35 points, as shown in column II. Each case was repeated three times, in order to give the user three opportunities to designate what he/she believes is an optimal path. Columns III and IV in Table I represent, respectively, the angular displacement required to move the robot from the initial to the final point of the path defined by the user and the corresponding cycle time (in seconds) required to track such a path, considering that the speed of the tool held by the robot is set at 160 mm/s. This speed is 8% of the maximum speed that the industrial robot is capable to develop, and is typical for some welding applications. Columns V and VI in Table I are the results obtained after applying the GA in order to optimize the path. In this case, column V represents the converged solution provided by the GA for the angular displacement of the robot (in degrees), while column VI represents the corresponding cycle time (in seconds) when the tool moves at a speed of 160 mm/s. The last two columns of Table I respectively, represent, the reduction (in percent) of angular displacement and cycle time when the optimized and the user-selected paths are compared.

As can be seen from the results presented in Table I, when the path consists of a small number of points, the user may be able to designate an optimal path, as was the case with paths 1, 2, 3, 4, and 6. However, as the number of points in each path increases, the user is less likely to find an optimum. As it can be seen from column VII, the percentage of reduction is not uniform, reflecting the erratic nature of the process for human designation of a short path, specially when this comprises a large number of points. This is another advantage of the proposed methodology, as it consistently achieves a reduced joint displacement, regardless of the ability or experience of the user.

As shown in Table I, an optimized joint displacement produces a reduction in cycle time in all the studied cases. The optimization of joint displacement allows for a reduced wear of the robot. When the last two columns of Table I are compared, it can be seen that there is no apparent correlation between optimal angular displacement, as defined in this correspondence paper, and the corresponding reduction in cycle time. Future work will consider experimental values for relative joint speeds of the robot, with the purpose of obtaining minimal cycle time, instead of minimum joint displacements, as considered in the developments herein.

Figs. 7–12 depict a comparison between user-defined and optimized paths for different number of points for each path. Consider, for instance, Fig. 10. This figure shows a typical path composed of 21 points placed over an arbitrary surface. Fig. 13 shows the performance of the GA for this 21-point path. This graph shows the total robot joint displacement required to complete the path, corresponding to each generation. With the purpose of observing the performance of the algorithm, a number of iterations is established ahead of time, in this case, 200 generations. As can be seen from the graph, a minimum value is obtained at approximately the 100th generation. Such a value is consistently repeated and is considered optimal, according to the GA procedure. As mentioned earlier, it may not represent a global

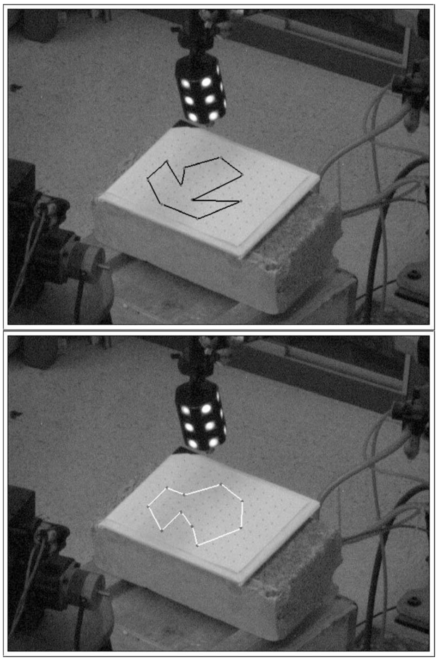


Fig. 8. Comparison of paths consisting of ten points. (Top) Path selected by the user. (Bottom) Optimized path.

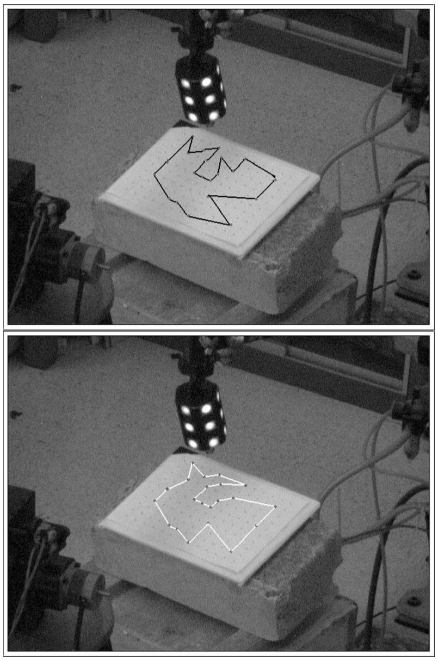


Fig. 10. Comparison of paths consisting of 21 points. (Top) Path selected by the user. (Bottom) Optimized path. *Note*: This path is also presented in Fig. 15.

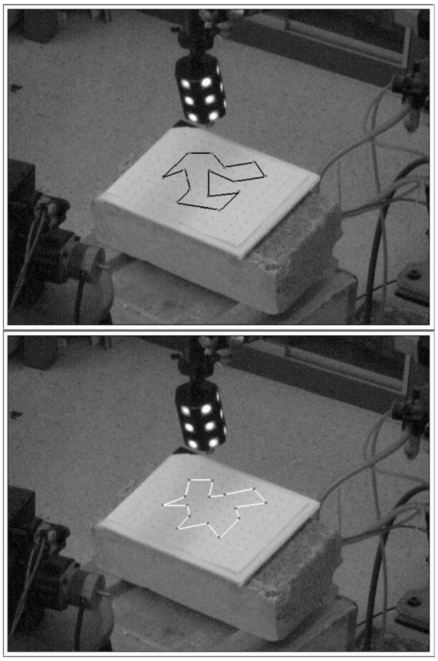


Fig. 9. Comparison of paths consisting of 15 points. (Top) Path selected by the user. (Bottom) Optimized path.

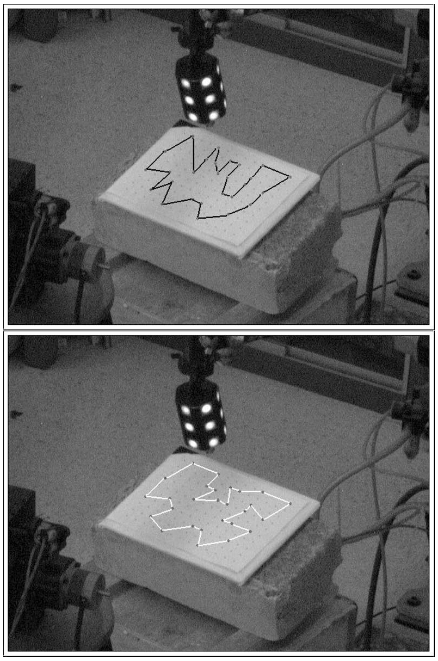


Fig. 11. Comparison of paths consisting of 27 points. (Top) Path selected by the user. (Bottom) Optimized path.

minimum, but the GA employed has provided consistent values for reduced joint displacements. This is shown for different paths in Table I. Fig. 15 shows the optimal sequence, which was also traced over the surface employed in the experiments, as shown in Fig. 10. Table II shows the robot configurations for this 21-point path. In this table, the point # in the second column corresponds to the point numbering sequence depicted in Fig. 6. This is also the case for the numbers indicated in Fig. 15. It is important to notice that the initial path used for the GA optimization process does not correspond to the initial, user-designated path. In this case, as presented in [27], opposed to other combinatorial optimization algorithms, GA convergence is not sensitive to the initial population, as long as this population is sufficiently large and chosen in an aleatory fashion. Therefore, the angular displacement corresponding to the first generation in Fig. 13 does not correspond to the angular displacement in Table I, path 15, column III.

Other optimization techniques also exist besides GA. For instance, SA [34] is also another combinatorial optimization technique that has been successfully applied to the TSP. In order to compare with the results obtained with the GA approach, such a technique was implemented and tested with the typical 21-point path cited earlier. The

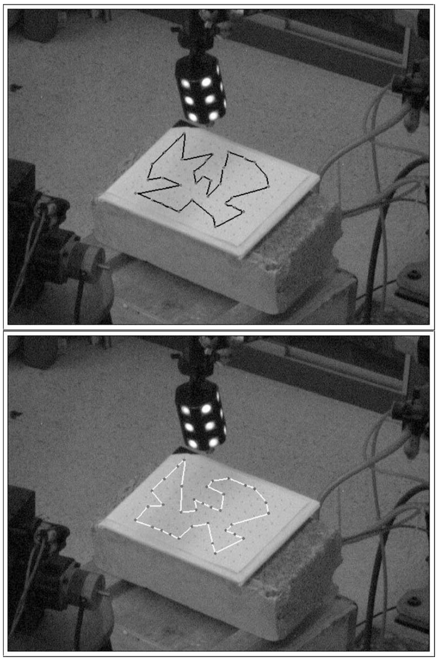


Fig. 12. Comparison of paths consisting of 35 points. (Top) Path selected by the user. (Bottom) Optimized path.

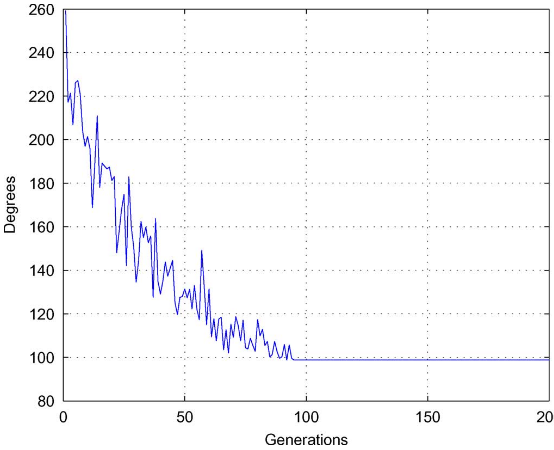


Fig. 13. Typical behavior of the GA solution.

implementation was based on the developments presented in [34], with the following parameters associated with this algorithm: initial temperature $T = 6000$; length of Markov chains $L = 4000$; and cooling parameter $a = 0.97$. The condition to terminate the algorithm is similar to the GA, i.e., the algorithm terminates when the objective function is repeated a certain number of times, within a certain tolerance. The behavior of this algorithm is schematically shown in Fig. 14, considering a total of 400 iterations. In this case, the initial joint displacement corresponds to the path designated by the user, i.e., the angular displacement corresponding to the first iteration in Fig. 14 corresponds to the angular displacement in Table I, path 15, column III. The converged solution coincides with the converged solution obtained from the GA approach shown in Table II for the 21-point path. However, as seen in Fig. 14, the solution is reached at approximately the 350th iteration, as opposed to 100 generations required for the GA to converge. This is due to the fact that GA performs its search in a parallel fashion [27], as opposed to SA [34], which searches for a solution in a single direction. Considering these facts, a GA approach is preferred over the SA approach in order to optimize the path originally set by the user. Note that the term *iteration* is associated with SA while the term *generation* is associated with GA; however, both terms are equivalent.

TABLE II
OPTIMAL SEQUENCE OF INTERNAL JOINT-CONFIGURATIONS ESTIMATED WITH THE VISION-BASED METHOD

i	Point #	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	2	-15.773	12.887	-42.689	20.365	48.186	-16.120
2	29	-14.106	16.280	-43.153	18.152	48.265	-14.458
3	54	-14.980	20.143	-42.749	19.406	48.088	-15.446
4	80	-14.642	23.602	-42.196	19.129	47.482	-15.375
5	109	-10.880	26.428	-41.725	14.229	46.228	-11.888
6	110	-9.740	26.315	-41.832	12.653	46.134	-10.708
7	123	-7.487	27.941	-41.213	9.659	45.200	-8.562
8	86	-7.533	22.733	-42.639	9.492	46.613	-8.278
9	116	-2.862	26.253	-41.857	3.118	45.455	-3.622
10	64	-2.607	19.248	-43.284	2.695	46.869	-3.262
11	15	-2.403	13.906	-43.451	2.310	47.005	-2.950
12	34	-7.683	15.588	-43.450	9.570	47.432	-8.242
13	46	-8.918	17.443	-43.216	11.268	47.364	-9.499
14	59	-8.839	19.240	-43.257	11.155	47.393	-9.411
15	58	-10.080	19.428	-43.180	12.832	47.506	-10.634
16	57	-11.323	19.454	-43.077	14.510	47.621	-11.862
17	31	-11.554	15.842	-43.364	14.750	47.941	-11.984
18	21	-10.361	14.009	-43.123	13.125	47.459	-10.829
19	8	-7.749	11.918	-42.942	9.641	46.905	-8.326
20	5	-11.768	12.121	-42.830	15.067	47.425	-12.282
21	23	-12.978	14.405	-43.052	16.613	47.880	-13.353

Result corresponds to path 15, consisting of 21 points, indicated in Table I.

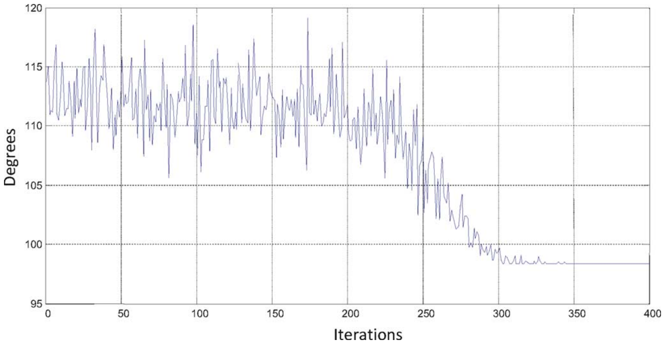


Fig. 14. Typical behavior of an SA solution.

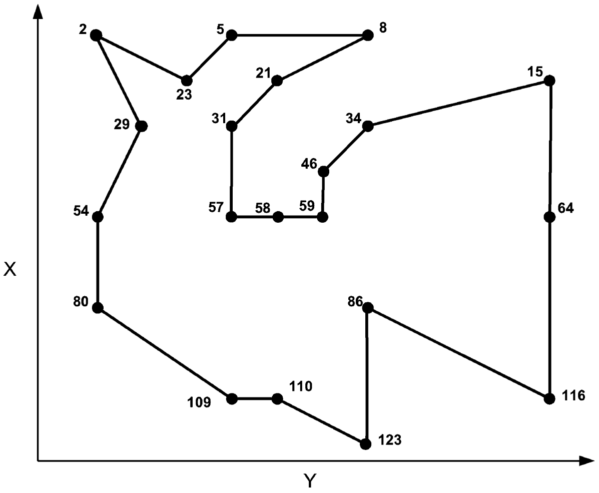


Fig. 15. Optimal sequence of 21 spots located in a plane.

Based on the experimental work cited before, it was determined that the methodology described herein requires a minimal input from the user in order to achieve an industrial task. This feature facilitates a remote task designation. This is among other reasons, due, to the fact that the human designation of tasks is accomplished in an asynchronous fashion with respect to the actual execution of the robotic maneuver. The designation is made through a graphical interface that shows the image provided by a single selection camera, which can be sent via Internet to the remote site where the user is located.

V. Conclusion

The most important contribution of this research consists of the development of a system that combines a calibration-free, vision-based approach with a GA procedure designed to produce an optimal sequence of joint configurations that enables a successful culmination of a task that involves a finite, discrete sequence of locations. The resulting sequence of robot configurations is optimal in the sense that it is obtained from the convergence of a GA procedure. The combination of the two procedures described earlier is initiated with the input from a user who designates, in an intuitive fashion, a starting sequence. The system can be applied in industrial tasks, for example, spot-welding application, drilling, etc.

One of the most important advantages of the method is that, once the optimal sequence is determined, no further adjustment is required when applied to an actual industrial task. This contrasts with other systems for the optimal determination of robotic sequences, based on a simulation platform or offline programming. The approach used with the system described here can be considered as the one compatible with a supervised autonomy [35] approach, enabling the designation of tasks in a remote fashion, for instance, using the Internet. The system described herein greatly facilitates the programming of industrial manufacturing tasks. These can be performed without alteration of currently available, commercial robotic system, as was the case of the robotic device used in our experiments. Such an advantage is not available for other vision-based systems, like visual servoing [36], due to the slow communication between the computer and the commercial system, and the time required for the analysis of the images obtained with the cameras used to control the maneuver.

Acknowledgment

The authors would like to thank the reviewers of this correspondence paper for their valuable comments.

References

[1] *New Webster's Dictionary and Thesaurus of the English Language*, Lexicon, Danbury, CT, 1992.

[2] M. Ruchanurucks, S. Kudoh, K. Ogawara, T. Shiratori, and K. Ikeuchi, "Humanoid robot painter: Visual perception and high-level planning," in *Proc. 2007 IEEE Int. Congr. Robot. Autom.*, pp. 3028–3033.

[3] C. K. Ahn and M. C. Lee, "An off-line teaching by vision information for robotic assembly task," in *Proc. 26th Annu. Conf. IEEE Ind. Electron. Soc.*, 2000, pp. 2171–2176.

[4] M. H. Choi and S. J. Kim, "A force/moment direction sensor and its use for human–robot interface in robot teaching," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2000, pp. 2222–2227.

[5] M. H. Choi and W. W. Lee, "A force/moment sensor for intuitive robot teaching application," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, pp. 4011–4016.

[6] C. Skourup and J. Pretlove, "Intuitive robot programming based on operators' implicit knowledge," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2001, vol. 2, pp. 702–705.

[7] M. J. Seelinger, E. J. González-Galván, M. Robinson, and S. B. Skaar, "Towards a robotic plasma spraying operation using vision," *IEEE Robot. Autom. Mag.*, vol. 5, no. 4, pp. 33–36, Dec. 1998.

[8] S. B. Skaar, W. H. Brockman, and W. S. Jang, "Three dimensional camera space manipulation," *Int. J. Robot. Res.*, vol. 9, no. 4, pp. 22–39, 1990.

[9] E. J. Gonzalez-Galvan and S. B. Skaar, "Efficient camera space manipulation using moments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1996, pp. 3407–3412.

[10] E. J. González-Galván, S. R. Cruz-Ramírez, M. J. Seelinger, and J. J. Cervantes-Sánchez, "An efficient multi-camera, multi-target scheme for the three-dimensional control of robots using uncalibrated vision," *Robot. Comput.-Integr. Manuf.*, vol. 19, no. 5, pp. 387–400, 2003.

[11] R. Sedgewick, *Algorithms in C++*. Reading, MA: Addison-Wesley, 1995, pp. 379–388.

[12] D. Stücker, "Elementary geometric methods: Line segment intersection and inclusion in a polygon," Dept. Comput. Sci., Univ. Oldenburg, Oldenburg, Germany, Tech. Rep., 1999.

[13] K. Anderson and J. Angeles, "Kinematic inversion of robotic manipulators in the presence of redundancies," *Int. J. Robot. Res.*, vol. 8, no. 6, pp. 80–97, 1989.

[14] A. Swaminathan and K. S. Barber, "APE: An experience-based assembly sequence planner for mechanical assembly," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1995, pp. 1278–1238.

[15] J. C. Latombe, R. H. Wilson, and F. Cazals, "Assembly sequencing with toleranced parts," *Comput. Aided Des.*, vol. 29, no. 2, pp. 159–174, 1997.

[16] Y. Lee and E. T. Sanii, "Process selection and operation sequencing based on feature extraction," *Eng. Des. Autom.*, vol. 3, no. 4, pp. 307–320, 1997.

[17] C. K. Shin, D. S. Hong, and H. S. Cho, "Disassemblability analysis for generating robotic assembly sequences," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1995, pp. 1284–1289.

[18] O. Maimon, "The robot task-sequencing planning problem," *IEEE Trans. Robot. Autom.*, vol. 6, no. 6, pp. 760–765, Dec. 1990.

[19] E. Lawer, J. Lenstra, A. H. G. Rinnoy-Kan, and D. B. Shmoys, Eds., *The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Chichester, U.K.: Wiley, 1985, pp. 37–85.

[20] H.-S. Hwang, "An improved model for vehicle routing problem with time-constraint based on genetic algorithm," *Comput. Ind. Eng.*, vol. 42, pp. 361–369, 2002.

[21] S. Chatterjee, C. Carrera, and L. A. Lynch, "Genetic algorithms and traveling salesman problems," *Eur. J. Oper. Res.*, vol. 93, no. 3, pp. 490–510, 1996.

[22] K.-Y. Kim, D.-W. Kim, and B. O. Nnaji, "Robot arc welding task sequencing using genetic algorithms," *IIE Trans.*, vol. 34, no. 10, pp. 865–880, 2002.

[23] P. T. Zacharia and N. A. Aspragathos, "Optimal robot task scheduling based on genetic algorithms," *Robot. Comput.-Integr. Manuf.*, vol. 21, pp. 67–79, 2005.

[24] H. Nakamura, K. Yamamoto, T. Itaya, and T. Koyama, "Development of off-line programming system for spot welding robot," in *Proc. 2nd IEEE Int. Workshop Emerging Technol. Factory Autom. Des. Oper. Intell. Factories*, Sep. 27–29, 1993, pp. 223–228.

[25] P. J. Cho, D. Il Kim, and H. G. Kim, "Real-time static deflection compensation of an LCD glass-handling robot," *Mechatronics*, vol. 17, no. 4/5, pp. 191–198, 2007.

[26] T. V. Zavrazhina, "A control for spatial motions of a robot in a rectangular Cartesian coordinate system," *J. Comput. Syst. Sci. Int.*, vol. 45, no. 4, pp. 658–670, 2006.

[27] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[28] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Michigan Univ. Press, 1975 (Cambridge, MA: MIT Press, 1992, 2nd ed.).

[29] D. W. Coit, A. E. Smith, and D. Tate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems," *INFORMS J. Comput.*, vol. 8, no. 2, pp. 173–182, 1996.

[30] A. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*. New York: Springer-Verlag, 1992, pp. 168–176.

[31] J. H. Park and H. Asada, "Sequence optimization for high speed robotic assembly using simulated annealing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 3441–3446.

[32] K. R. Castleman, *Digital Image Processing.* Upper Saddle River. NJ: Prentice-Hall, 1996, pp. 447–486.

[33] B. Jähne, *Digital Image Processing: Concepts, Algorithms and Scientific Application*, 4th ed. New York: Springer-Verlag, 1997, pp. 453–487.

[34] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[35] P. G. Backes, "Supervised autonomy for space telerobotics," in *Teleoperation and Robotics in Space. Progress in Astronautics and Aeronautics,* vol. 161, S. B. Skaar and C. F. Ruoff, Eds. Washington, DC: Amer. Ins. Aeronaut. Astronaut. (AIAA), 1994.

[36] A. Crétual and F. Chaumette, "Visual servoing based on image motion," *Int. J. Robot. Res.*, vol. 20, no. 11, pp. 857–877, 2001.

[37] F. Cisneros, Application engineer. Yaskawa-Motoman robotics, private communication, Mar. 2000.