A Robot System for Automatic Wire Crimping.

David Vernon,
Dept. Computer Science,
Trinity College,
Dublin,
Ireland.

Abstract.

This paper describes an automatic wire crimping process which has been developed in the context of a robotic and robot vision development system. This system comprises a Vicom Image Processor, used for image acquisition, a VAX 11/780, used for software development and on which all processing and analysis algorithms are presently implemented, and a Smart Arms 6R/600 D.C. servo-motor robot. The system architecture and interfaces are detailed and transportation to an industrial prototype is considered. The vision sub-system uses binary imaging techniques comprising grey-scale segmentation by thresholding, thinning, and subsequent image analysis. These techniques are summarised and illustrated. A robot control language, which has been developed specifically for this robot and application, is also described.

Introduction.

Virtually all electronic and electrical assemblies incorporate standard insulated electrical wire which must be properly prepared before being used in the unit being manufactured. Such preparation normally takes the form of cutting, stripping, and either solder tinning or crimping the wire ends. While high-volume automatic wire cutters and strippers are in common use, the final tinning or crimping has hitherto involved manual operation of solder baths or crimping presses. This paper describes a robot development system, which demonstrates the feasibility of automating this process using a five degree of freedom robot and visual sensing (see diagram 3). The use of robot manipulators is a solution that is increasingly being selected for situations of a batch nature, involving low-volume throughput with frequently changing workpiece characteristics [Mujtaba 1982, pp.5-6]. However, conventional first generation robots require that their workpieces be uniformly and uniquely presented [Simons 1982, p.20]; a condition most difficult to satisfy with flexible variable length wires. It is shown here that the use of robot vision offers a legitimate solution.

The thrust of the current approach has been to constrain the environment to facilitate relatively simple analysis techniques. In particular, it is assumed that the wires are almost flat, that the tray on which they rest provides a clearly visible contrasting background, and that the wires are layered no more than one or two deep. It is these types of restrictions that distinguish the robot vision approach and philosophy from the more general approaches of Computer Vision and Image Understanding [Pugh 1983, pp.4-5].

System Architecture.

The present system is configured as a development system and not as a final industrial target system; a possible prototype is considered later in the paper. This development system comprises a vidicon camera, a Vicom image processor, a Vax 11/780, and a Smart Arms 6R/600 Robot (see diagram 1).

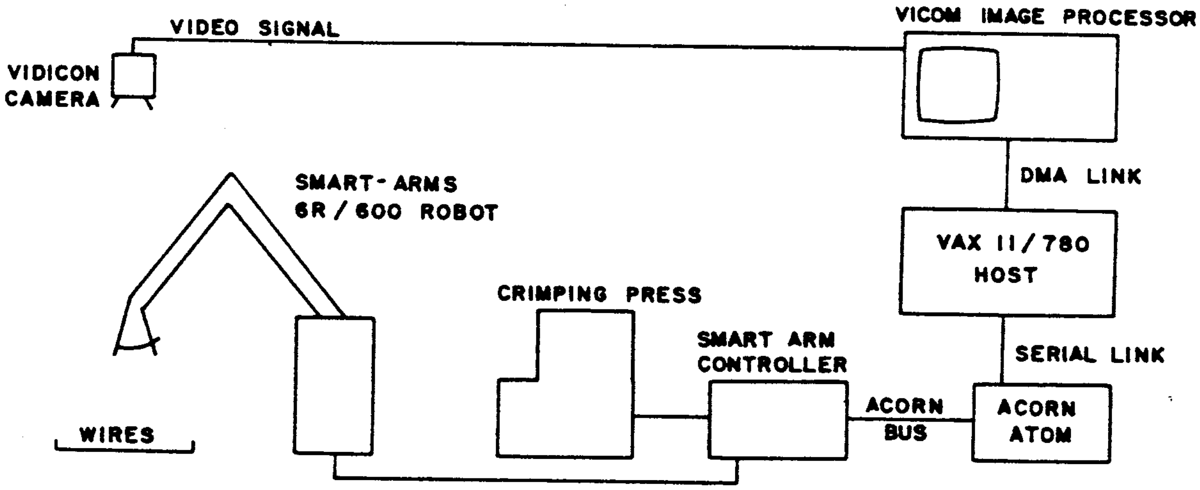

Diagram 1 - System Architecture.

The Vicom image processor is used solely for image acquisition, transferring images to the host Vax 11/780 via a DMA link. The images acquired by the Vicom have a resolution of 512x512 pixels with 128 levels of grey. All processing and analysis software, in particular the Robot Control Language (RCL) interpreter with related sensing routines, run on the Vax 11/780.

This RCL controls the Smart Arms robot via a RS232C serial link; the Acorn Atom robot supervisory microcomputer merely acts as a serial-link interface to the robots own motor controllers. The 6R/600 is a five degree of freedom D.C. servo-motor anthropomorphic robot and has been equipped with a special purpose gripper to allow it to grasp wires (see diagram 2). As mentioned above, it is assumed that the wires are almost flat and this gripper is designed to push the wire down onto the tray as it grasps it. Hence the sensing algorithms need only detect the x and y coordinates of the wire; the z component may be assumed to be the same as that of the tray.

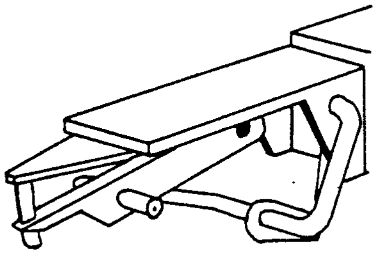

Diagram 2 - Robot End-Effector.

The Robot Control Language (RCL).

The 6R/600 robot is supplied by the manufacturers with a teach pendant to allow robot programming by training, that is, by stepping the robot through a sequence of positions and signaling the supervisory software (running on the Acorn Atom) that certain step positions are to be learned for subsequent replay. The robot is programmed in this instance in "joint space", i.e., by teaching a sequence of robot joint set-points (e.g. shoulder, elbow, wrist positions). Sophisticated visual sensing normally requires that the robot be controlled in a Cartesian frame of reference since the sensed position will usually be made available in the camera x-y reference frame (and then in the real-world Cartesian reference frame using a pre-determined camera model) [Paul 1981, p.128]. To facilitate such programming, the robot system has been modelled by determining the Cartesian-to-joint transformations, normally referred to as the inverse kinematic solution [Lee 1982, p.62]. This required a simple mechanical modification to extend the elbow joint motor linkage and some additional software to compensate for the fact that the elbow and shoulder joints are not independent on Smart Arms robots as they normally would be on conventional anthropomorphic manipulators.

Having determined the inverse kinematic solution, whereby the joint set-points may be computed from a required wrist position and end-effector orientation, a robot programming language was then designed to use this solution and allow the manipulator to be programmed by simply specifying the gripper position and orientation. This language was developed specially for this automatic wire crimping application and, while it supports this application explicitly through built-in robot vision primitives, it is completely general purpose in nature. It is an "explicit-level" or "robot-level" language [Lozano-Perez 1982, p.1] and thus requires the programmer to specify explicitly all positions and orientations through which the gripper must move. It offers the normal structured programming control statements (IF-THEN-ELSE, REPEAT-UNTIL, WHILE-DO, CASE) and robot control is effected through the use of built-in functional primitives like MOVE, GRASP, RELEASE, HOME.

The MOVE statements require gripper positioning information either in the form of wrist x, y, and z coordinates with gripper roll, pitch, an yaw, or in the form of a single **frame** parameter. These frames are simply data-structures which represent an orthogonal reference frame (x-y-z). By embedding such a frame in an object, e.g. gripper, wire, wooden block, nut, or bolt, and allowing the frame to be translated and rotated (again through the use of in-built functional primitives) this object/frame moves about in the robots reference frame. The frame that is assumed to be embedded in the gripper is normally labelled **T6**, it is referred to as **T5** in this application since the robot only has five degrees of freedom. The robot is effectively programmed by assigning to the frame T5 some sequence of translational and rotational operations and then using this T5 frame or gripper position/orientation as a MOVE statement parameter.

Additionally, these frames may describe not only the absolute position and orientation of an object with respect to the real world reference frame, but may also describe the relative position of one object to another.

To support this particular crimping application two visual sensing primitives, SNAP and VISION, have been incorporated into the language. SNAP causes the Vicom to acquire an image (of the wires) and transfer it to the Vax for subsequent analysis. The VISION primitive then effects all processing and analysis and returns two frame parameters defining **a)** the absolute position and orientation of the wire end and **b)** the position and orientation of the end-effector grasp point relative to the end position.

The position of the crimping press was determined empirically and the robot was then programmed to pick up the wire at the grasp point and introduce it to the crimp along a path coincident with a line parallel to the tangent at the end of the wire. This process continues as long as there are wires on the tray.

Image Processing and Analysis.

Given the initial assumptions about scene complexity, that the wires are well scattered, no more than one or two deep, and that the tray provides a visually contrasting background, then all the requisite information may be gleaned from the silhouette of the wire. In such circumstances, binary vision techniques are appropriate [Agin 1980, p.12]. A binary image of the wires, in which there are just two levels of grey, black and white, represents a significant reduction of complexity without appreciable reduction in information content. In this case, all black pixels correspond to wires and the white pixels to the tray background. This philosophy of image simplification without significant information loss characterises the overall approach taken in this implementation: the image is reduced to its simplest form before analysis by reduction in resolution, thresholding, and skeletonising. A similar approach has been taken in problems concerned with the analysis of paper pulp fibres [Kasvand 1978] and asbestos fibres [Dixon and Taylor 1978].

The original image acquired by the Vicom is a 512x512 pixel image, each pixel representing one of 128 levels of grey (see diagram 4). The first processing stage is a reduction in resolution to 128x128 pixels (see diagram 5) resulting in a reduction of the complexity of subsequent operation by a factor of sixteen. This grey-scale image is then converted to a binary image (see diagram 6) by a thresholding operation: each pixel value is compared to a suitably chosen threshold level, if the pixel has a value less than this threshold level it is labelled a wire and given a value corresponding to black, otherwise it becomes a white background pixel. Selection of an appropriate threshold is obviously critical, in this case threshold selection is effected automatically by choosing the value corresponding to the mean pixel grey-level of all those points on the boundary of the wire.

This binary image is then thinned to produce a skeletonised representation of the wires exactly one pixel wide (see diagram 7). Many thinning algorithms have been reported in the literature (for example, see a survey by [Tamura 1978]); this algorithm is based on continuous removal of border pixels so that one never breaks the object in two. Thinning ceases when no more pixels may be removed [Vernon 1984, pp.865-866].

With the image now in the form shown in diagram 7, the wire skeletons are analysed and it becomes a relatively simple task to detect valid grasp points. A suitable grasp point is defined in this instance to be one a fixed distance from a wire end, on a wire segment which is bordered at one extremity by a wire end and at the other by either a wire end or a wire crossing. This analysis phase is a two stage process; firstly the image is scanned in a methodical manner looking for wire points and secondly this detected wire is then followed in both directions to see if it is a valid wire segment. If so, then the position and orientation of the wire and grasp points are determined (see diagram 6), transformed to the real-world coordinates, and the two frames corresponding to the absolute position and orientation of the wire end and the relative position of the grasp point, are generated.

Process	Average Time (sec.)
Image acquisition	0.10
Transfer to Vax	3.87
Generate 128x128 image	4.95
Threshold selection	45.80
Thresholding	0.23
Thinning	5.86
Image analysis	0.28

Table 1 - Average Sensing Process Times.

Transportation to an Industrial Prototype.

The average sensing cycle time, as detailed in table 1, is 6.5 seconds. This is excessive and would require further optimisation for the system to be viable in an industrial environment. The single most expensive process in terms of computation time is the thinning algorithm. One possibility would be to implement this processing in hardware. This is a popular and valid approach and has been adopted for this type of low-level processing in a number of commercial systems (for example, see [Costlow 1982, Adaway 1983, Makhlin and Tinsdale 1983, and Tinham 1983]). However, one of the major constraints restricting the use of robot vision and visual inspection systems in industry is the current cost of the associated hardware, figures of the order of 4000 have been quoted as desirable [Stout and Thomas 1980, p.29] while many commercial vision systems cost in excess of 20000 [Sanderson 1983, p.106]. Low-cost vision systems may be functionally acceptable but are normally organised as input/output devices with restrictive image access times [Li et al. 1983, p.121]. This necessitates that the processing and analysis techniques must compensate for this additional overhead, a problem frequently solved by "windowing" techniques where the image is processed in just one local area of interest.

An additional argument against investing further effort in this present implementation is the current trend toward grey-scale image processing for industrial applications [Wallace 1983, p.178], a desirable trend which allows one to relax somewhat the constraints of high contrast backgrounds and scene clutter and complexity. Grey-scale processing is again more computationally expensive and some means of compensation must be explored and embodied in the resulting techniques.

Allowing for the economic constraints imposed on industrial vision systems and with these trends in mind, the industrial prototype is to be based upon this type of inexpensive frame grabber, specifically a Micro Sight 2 vision system, an IBM PC, together with more sophisticated vision software. This software has been designed to exploit the input/output image random access nature of the vision hardware and also enhance the user-friendliness of the system. The former is intended to reduce the computational overhead and is accomplished by implementing feature detection algorithms developed at TCD, which are able to recognise wires, and wire ends in particular, based purely on their outline shape. The outline is generated by simply following the wire boundary around the image, accessing only those points in the image which are necessary to the generation of that part of the contour. The latter objective of enhancing the user-friendliness of the system is effected by allowing the operator to teach the shape in question merely by pointing to it with a cursor. Subsequent image analysis is based on this taught shape. This approach has the added advantage of removing the hard-wired nature of the implementation and facilitates a more general purpose visual sensing environment where many types of shapes may be used.

Summary.

This system demonstrates the feasibility of automating a common industrial assembly process and current development is oriented toward more reliable and useful techniques which will be of practical value with low-cost vision systems.

Some problems have been encountered with the repeatability and the accuracy of the robot. This is due in part to the inaccuracies of the measurements used when modelling the robot, and in part to gearbox backlash and dead-zone effects. Bearing in mind the small dimensions of the objects being manipulated, a 1.5 mm wire diameter in most cases, and the fact that this is a low-cost laboratory manipulator, not an industrial version, its performance is acceptable.

The RCL developed to support this application has given rise to a useful general purpose programming environment for the robot system, and future versions will include an interface to a CAD/CAE system to facilitate graphic simulation of robot motion during training and programming and will also include user-trainable robot vision capabilities.

References.

Adaway, B. 1983. "Vision for Robots", Systems International, April, pp.38-39.

Agin, G. 1980. "Computer Vision Systems for Industrial Inspection and Assembly", Computer, Vol. 13, No. 5, pp.11-20.

Costlow, T. 1982. "Plug-in Boards Add Vision Functions to Range of Robots", Electronics, June 30th., pp.47-48.

Dixon, R.N. and Taylor, C.J. 1978. "Automated Asbestos Fibre Counting", in "Machine-Aided Image Analysis", W. Gardener (Ed.), pp.178-185.

Kasvand, T. 1978. "Experiments on Automatic Extraction of Paper Pulp Fibres", Proceedings of the Fourth International Joint Conference on Pattern Recognition, pp.958-960.

Lee, C.G.S. 1982. "Robot Arm Kinematics, Dynamics, and Control", Computer, Vol 15, No. 12, pp.62-80.

Li, H.F., Tsang, C.M., and Cheung, Y.S. 1983. "A Low-Cost Real-Time Imaging and Processing System", Software and Microsystems, Vol. 2, No. 5, pp.121-129.

Lozano-Perez, T. 1982. "Robot Programming", AI Memo No. 698, MIT AI Lab.

Mujtaba, M. 1982. "Motion Sequencing of Manipulators", Report No. STAN-CS-82-917, Stanford University.

Paul, R. 1981. "Robot Manipulators: Mathematics, Programming, and Control", MIT Press, Cambridge, Massachusetts.

Pugh, A. 1983. "Second Generation Robots" in "Robot Vision", A. Pugh (Ed.), IFS (Publications) Ltd, U.K., pp.3-10.

Sanderson, R.J. 1983. "Machine Vision Systems: A Summary and Forecast", Tech Tran Corporation, Illinois.

Simons, G.L. 1980. "Robots in Industry", NCC Publications, Manchester, England.

Stout, K. and Thomas, A.F. 1980. "How Robots are Improving their Eyesight", The Production Engineer, Vol. 59, No. 3, pp.29-34.

Tinham, B. 1983. "Fast Image Analysis", Systems International, April, pp.43-44.

Vernon, D. 1984. "Robot Vision in Automated Electrical Wire Crimping", Premier Colloque Image, Biarritz, Vol. 2., pp.863-868.

Wallace, A.M. 1983. "Grey-Scale Image Processing for Industrial Applications", Image and Vision Computing, Vol. 1, No. 4, pp.178-188.

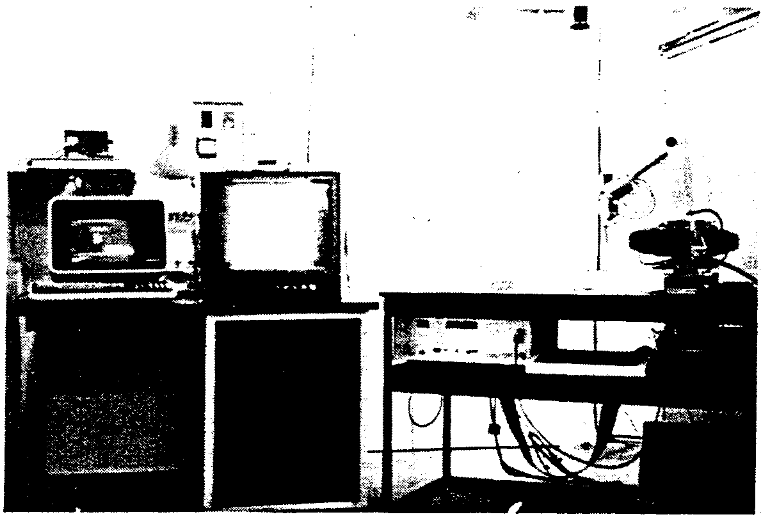

Diagram 3 - Development System.

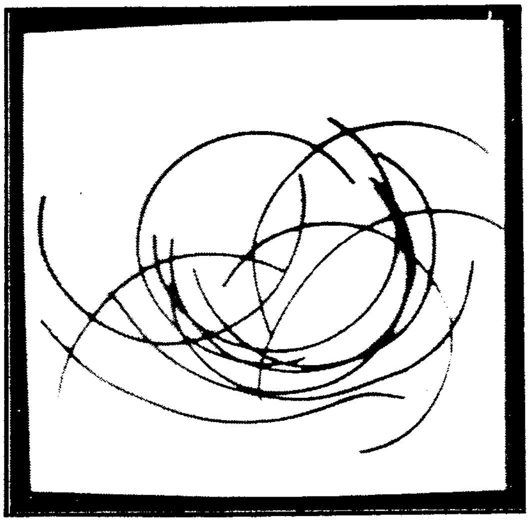

Diagram 4 - 512x512 Image.

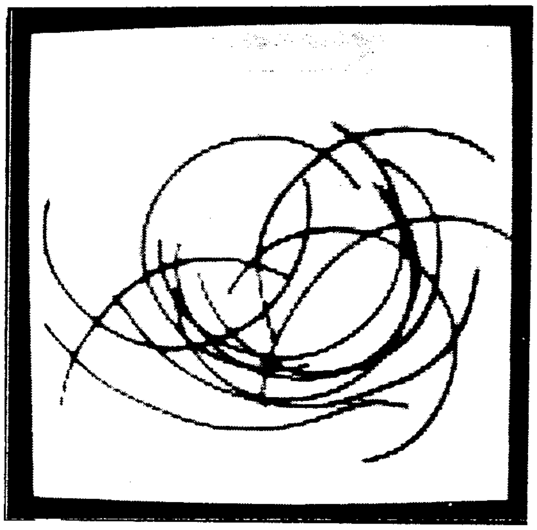

Diagram 5 - 128x128 Image.

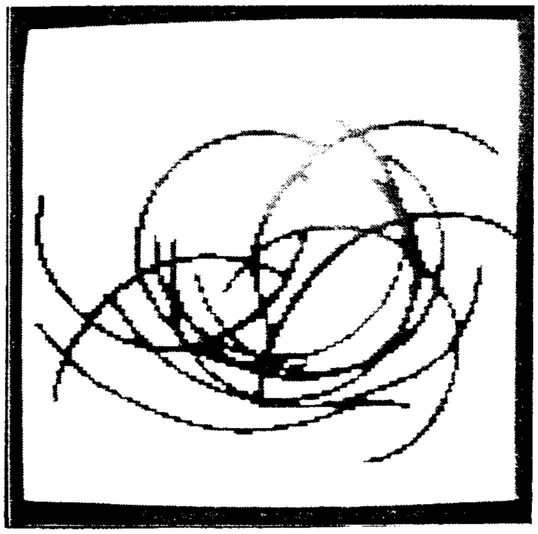

Diagram 6 - Binary Image.

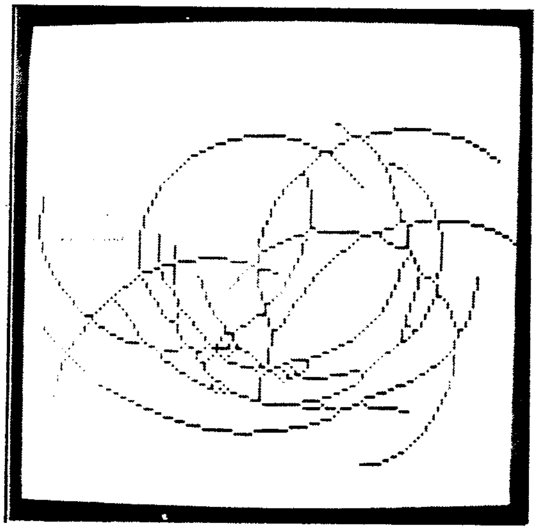

Diagram 7 - Thinned Image.