18

Undercut detection and generation of core/ cavity in solid models

Mouhsine Serrar [a] *and Gary Gabriele* [b]

[a] *Motorola, Paging Products Group, 1500 Gateway Bld.,*
Boynton Beach, FL 33426, USA. E-mail: mserrar@pts.mot.com

[b] *Rensselaer Polytechnic Institute, Troy, New York 12180-3590, USA.*
E-mail: gabrig2@rpi.edu

Abstract

Existing undercut detection algorithms analyze the geometry of the part without considering the geometry of the mold halves. For this reason, these algorithms can only detect 'potential' undercuts. Actually, undercuts happen when the part can not be extracted from the core or cavity. Therefore, an accurate detection of undercuts must analyze the geometry of the core and cavity. Proposed algorithms are based on the generation of the core and cavity models. Undercut detection algorithms were implemented and successfully tested on industrial components.

Keywords

Undercut, mold, CAD, solid modeling, core, cavity

1 BACKGROUND

Undercuts make the extraction of a part from mold halves impossible or very difficult without adding sliding cores to the mold (Figure). Undercuts are typically protrusions or depressions in the molded part. In other cases, undercuts are caused by the overall shape of the molded part. One should avoid side cores because they increase the cost of the mold as well as the molding time (added time for operating the mechanism of the sliding cores).

2 MOTIVATION

Our motivation for automating undercut detection comes from the fact that undercuts can go undetected until late in the design process, especially when dealing with large models or complex parting surfaces. Automation of undercut detection will benefit the designer of the components as well as the mold maker by allowing the latter to assess 'designed-in' undercuts.

M. J. Pratt et al. (eds.), *Product Modeling for Computer Integrated Design and Manufacture*

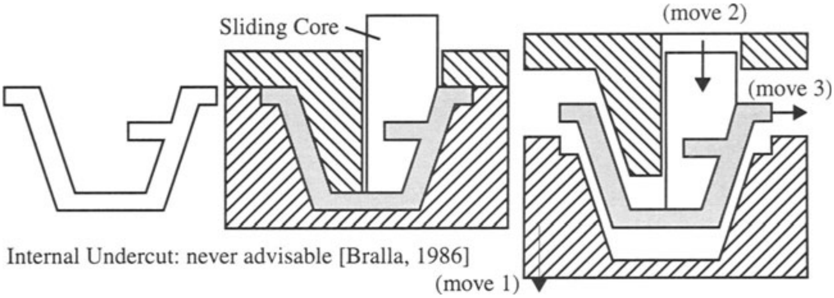

Figure 1: Example of an internal undercut.

3 REVIEW OF RELATED WORK

Liou et al. [Liou, 1991; Liou, 1990] worked on a 'design for die casting' system. Their procedure for detecting undercuts was limited to flat parting surfaces. First, they used a flat parting surface to split the part. Second, for each face, the angle between the face normal and the mold opening direction is computed. An angle greater than 90^0 indicates an undercut. The problem is that the procedure is not valid if the parting surface traverses a cavity in the part.

Rosen [Rosen, 1992] developed a system for manufacturability evaluation at the configuration stage of thin-walled parts. To detect undercuts, he first projects each wall's mid-plane surface along the mold opening direction. Second, he checks if the projected surfaces intersect. An intersection indicates a 'potential' undercut. This approach has limited application in detailed solid modeling because the large number of surfaces in solid models would make the approach computationally expensive. Also, this method does not use the parting surface information. For this reason, the method can only detects 'potential' undercuts and does not detect 'actual' undercuts.

Tan [Tan, 1988] described a scheme for detecting undercuts in solid models. Solid models are first converted to a polyhedral faceted form. Second, a list of 'visible' faces –faces with normal containing a positive vector component in the parting direction – is created. Next, 'visible' patches made of 'visible' faces adjacent to each other, are formed. If more than one 'visible' patch is found, then there is an undercut. If only one visible patch is formed, a further check is performed. The limitation of this approach is that it does not apply to models with non-drafted faces (faces parallel to the mold opening direction). Also, when dealing with complex models containing sculptured surfaces, a large number of faces is needed in order to accurately determine the parting edges and detect the undercuts.

Extensive work on generating the optimal mold opening direction was found in the literature. Chen et al. [Chen, 1992a; Chen, 1992b], and Mochizuki [Mochizuki, 1992] approaches were to first determine all the pockets in the model –undercuts are necessarily pockets in the model– by performing a difference set operation (Boolean operation) of the solid model from the convex hull of the solid model. Next, for each pocket, they determined the set of mold opening directions which do not cause the pocket to be an undercut. Last, they determined the optimal mold opening direction which will minimize the number of potential undercuts. Chen et al. used an innovative approach: the visibility sphere. For each potential undercut, they used a visibility sphere to represent the set of mold opening directions which will eliminate the potential undercut. Similarly to Tan's approach, Chen and Mochizuki do not use the parting surface information, and thus can only detect 'potential' undercuts. Weinstein et al. [Weinstein, 1994] used an approach similar to Chen's and Mochizuki's and they proposed a method for evaluating the model as design features are added to the model.

Gunter [Gunter, 1990] proposed a search method which classifies potential parting lines using a set of design rules, e.g., "select parting line which minimize the number of undercuts". Similarly to Liou's approach [Liou, 1990], Gunter limited his system to flat parting lines and the proposed algorithm is not valid for parts with cavities.

Ravi et al. [Ravi, 1990] proposed a set of quantitative criteria to assess the castability of a given design. He proposed an undercut factor as being the total volume between undercut faces and the parting surface normalized with the total volume of the component.

Hui [Hui, 1992] proposed a system for selecting an optimal parting direction. He evaluated an undercut criterion by generating a mesh of points on the surface of the part and then counted the number of points hidden in the two directions of the parting direction. The disadvantage of this approach is that the accuracy of the evaluation depends on the quality of the mesh it uses. Also, this method detects 'potential' undercuts only.

There is a commercial CAD system [PTC, 1993] which offers an undercut detection utility. Before the utility is used, the user creates the mold halves and assembles them with the part. Then, the user specifies discrete 'moves' of each component of the assembly. While not made clear by the author, we believe that the system performs a Boolean intersection operation to check for interference between the components after each move (disassembly). An interference indicates an undercut. The disadvantage of this method is that the user is required to create the mold halves which is time consuming.

4 PROPOSED APPROACH FOR UNDERCUT DETECTION

In most of the reviewed work, undercut detection algorithms analyze the geometry of the part without considering the geometry of the mold halves. For this reason, these algorithms can only detect 'potential' undercuts. Actually, undercuts happen when the part can not be extracted from mold halves. Therefore, an accurate detection of undercuts must analyze the geometry of the mold halves. We are proposing two algorithms for detecting undercuts: the adjacency-based algorithm and the mold halves-based algorithm.

5 ADJACENCY-BASED METHOD

The adjacency-based method is founded on the boundary representation (B-Rep) of solid models. This method uses the B-Rep adjacency information to create two sets of faces: core and cavity patches. For each face, we evaluate the angle between the external normal and the mold half opening direction (for the corresponding mold half). An angle greater than 90^o indicates that the face is an undercut.

In this section, we first outline the adjacency-based algorithm, then we present the two main elements of the algorithm: selection of parting loops and determination of core/cavity patches. Finally, we present results from tests on a model of an industrial component.

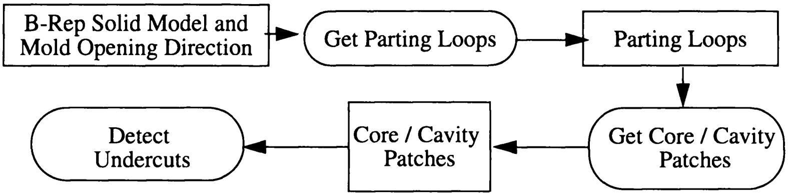

Figure 2: Undercut detection using the adjacency-based method.

5.1 Algorithm outline

1. Input parting direction
   The parting direction can be input to the system, for instance, by asking the user to select a face normal to the parting direction. The system would get the parting direction by evaluating the normal to the selected face
2. Input parting loops
   Parting loops consist of an external parting loop and multiple internal parting loops. The external parting loop is misleadingly referred to in the industry as the parting line. An internal parting loop corresponds to through-all cuts (shut-off) in the component [Serrar, 1995a and b]. Parting loops information is necessary to determine if a given face of the model belongs to the core or cavity. An efficient method for selecting parting loops is critical to the usability of an undercut detection tool. Requiring designers to select each edge of a complex parting loop is too cumbersome and error prone. On the other hand, an efficient algorithm which fully automates the selection of parting loops does not exist in the literature. In this project, we developed a semi-automatic method for selecting parting loops. This method requires an 'acceptable' effort for selecting parting loops. This loop selection method is not limited by the complexity of the model. The loop selection method is presented in section 5.2.
3. Get cavity patches
   The goal is to group the faces of the model into two sets. One set corresponds to the core and the other to cavity. The method for grouping the patches is presented in section 5.3.
4. Determine mold opening-direction for the core and cavity
   These opening directions are either parallel or opposite to the parting direction.
5. For each patch:
   a. Evaluate outward normal in a 3 x 3 grid (or any other user-defined grid), defined in the 2-D parametric domain of the patch
   b. If angle between any outward normal to the patch and the corresponding mold half opening direction is more than 90^o, then the patch is causing an undercut
6. Output: highlight undercut patches and compute total area of undercut patches

5.2 Parting loop selection

We developed a semi-automatic method for selecting parting loops. This method takes advantage of the adjacency information (also called connectivity information), available in B-Rep solid models, to determine the branches (adjacent edges) of a given edge (Figure 3). In this method, the user first selects an edge from the parting loop. Then, the system finds the branches (adjacent edges), and successively highlights these branches until the user confirms which branch belongs to the parting loop. This process is repeated until the loop is closed. This interaction method is much more efficient and less error prone than manual selection of parting loops. This method was implemented and tested on industrial examples with satisfactory results.

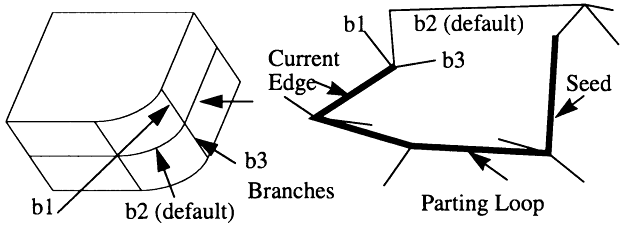

Figure 3: Parting-edges determination.

The parting-loop selection method would be more efficient if the CAD system allowed the user to store parting loop data along with the solid model. This would save designers from re-selecting parting loops every time they need to check for undercuts. This would also allow the mold maker and other members of the design team to evaluate the parting loops as designed by component designer.

5.3 Determining the core and cavity patches

On a solid model, external and internal parting loops define two sets of patches on each side of the external parting loop. The two sets correspond to the core and cavity, respectively. Our goal is to determine these two sets automatically. Our algorithm is based on the fact that patches of a given set are adjacent to each other and that these patches are bordered by the parting loops. The algorithm starts with a 'seed' patch on a given side of the external parting loop. Then, recursively, the algorithm traverses to patches adjacent to the current patch through the edges which do not belong to the parting loops (Figure 4). To avoid re-traversing the patches, we define a 'border-edges' list. This 'border- edges' list contains edges which are not valid for traversing to adjacent patches. In this case, a non-valid edge is either a parting edge or an edge already used to traverse to an adjacent patch. The 'border-edges' list is initialized with the edges of the parting loops. Then, every time an edge of a patch is used to get an adjacent patch, this edge is added to the 'border-edges' list.

Our algorithm assumes that every face in the B-Rep solid model is a connected 2-D domain. Specifically, faces consisting of two separate 'islands' are not allowed. This assumption is in accordance with the IGES and the STEP standards. Unfortunately, some CAD systems allow faces which are not connected. One way to 'fix' a solid model which contains non-connected faces is to export a solid model using a STEP interface, then, import the STEP solid model in to the CAD system. All the faces in the new solid model will be connected faces.

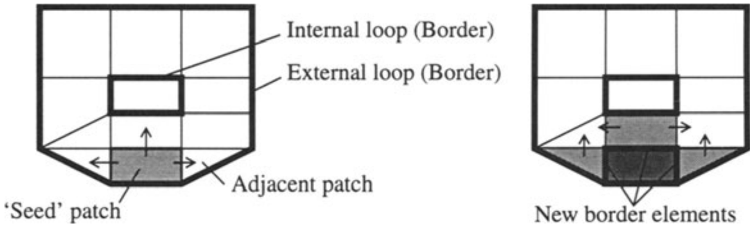

Figure 4: Cavity-patches determination.

5.4 Test results

The adjacency-based algorithm was fully implemented and integrated with a commercial 'open' CAD system [PTC, 1994b] and tests on models of industrial components were successful (Figure 5):

- Number of faces in the solid model: 550
- Number of parting loops: 1 parting line and 12 internal loops (9 holes and 3 button-cuts)
- Input time (loops selection) (approximate): 10 min.
- Processing time (approximate): 60 s.

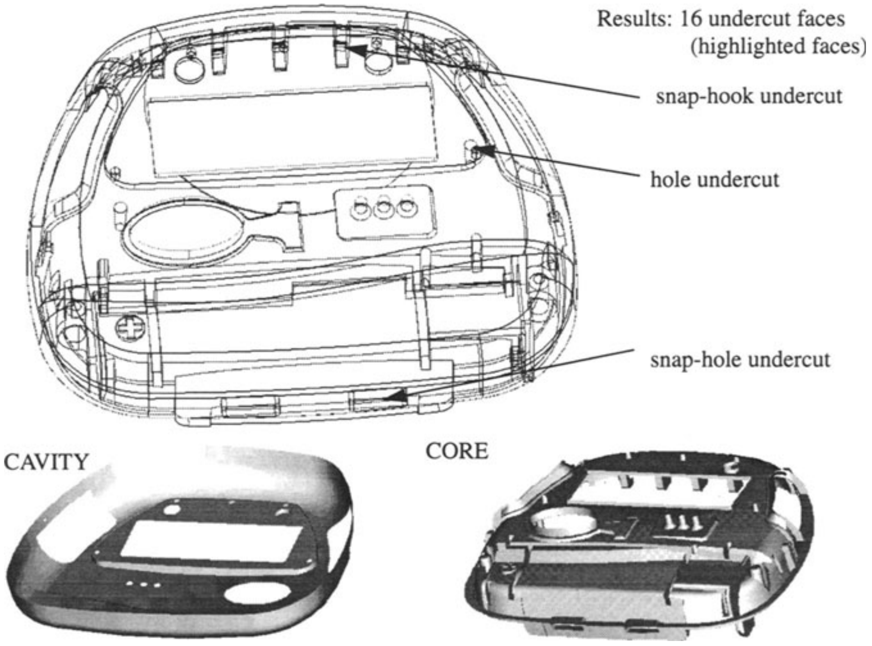

Figure 5: Undercut detection and core/cavity generation example.

6 MOLD HALVES-BASED METHOD

In the mold halves-based method, the procedure for detecting undercuts is applied to solid models of the mold halves. Automatic mold-halves generation is presented in [Serrar, 1995]. In this section we assume that mold-halves solid models are available.

A face in the cavity of a mold half is an undercut if the angle between the external normal and the mold half opening direction is less than 90°. This approach becomes more appealing than the adjacency-based method if solid models of the mold halves are readily available.

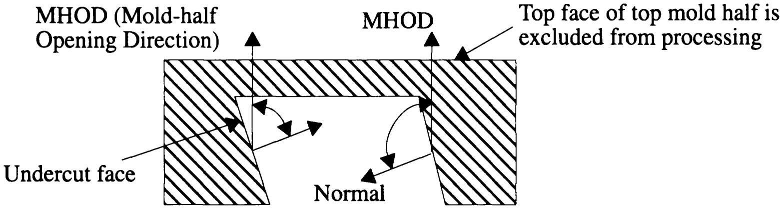

Figure 6: Undercut face detection.

6.1 Algorithm outline (Figure 7)

1. Generate mold halves
2. Generate faceted model of each mold half
3. Determine mold half opening direction (parallel or opposed to parting direction)
4. For each face in faceted model –except faces on top of top mold half and faces on bottom of bottom mold half–, determine if face is causing an undercut by computing the angle between the outward normal and the mold half opening direction. If computed angle is less than 90° (Figure 6), then the face is an undercut face.
5. Output total area of undercut faces and highlight undercut faces.

6.2 Faceting procedure and approximation error

In order to evaluate external normal vectors, this method takes advantage of the SLA interface available in most CAD systems. The SLA (Stereo-Lithography Apparatus) interface generates a list of triangles which approximate the surfaces on the solid model. Along with triangles, the SLA interface generates external normal vectors for each triangle. Before generating the SLA output, users specify the maximal chord-height allowed between original surfaces of the model and the new triangles [PTC, 1994a]. A smaller chord-height results in a better precision but on the other hand, more triangles are generated.

When dealing with highly curved surfaces, a few triangles may not be evaluated correctly, because of the error in approximating surface normal. Nevertheless, a few incorrect triangles should not prevent the designer from making a correct decision regarding the existence of the undercut.

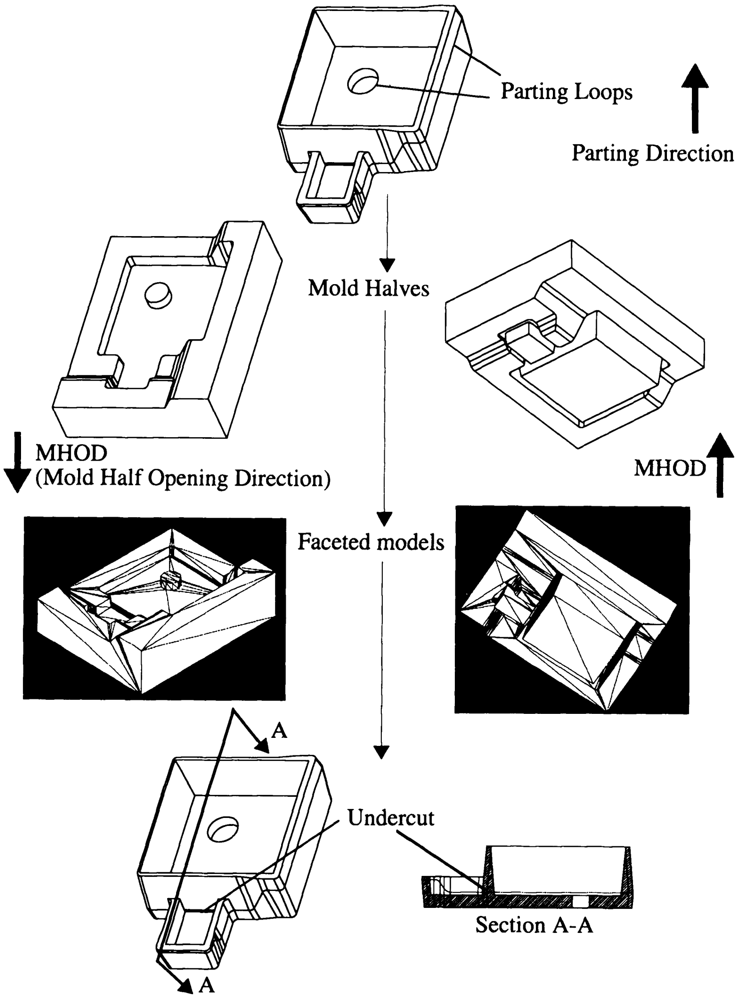

Figure 7: Mold halves-based algorithm.

7 DISCUSSION

7.1 Adjacency-based algorithm vs. mold halves-based algorithm

The adjacency-based method requires less implementation effort than the mold halves-based method. On the other hand, the adjacency-based method cannot check if the design of internal patches is causing an undercut. For example, the part in Figure 8 contains an undercut created by the internal parting-surface. This undercut cannot be detected using the adjacency-based method. The same undercut would be detected using the mold halves-based method.

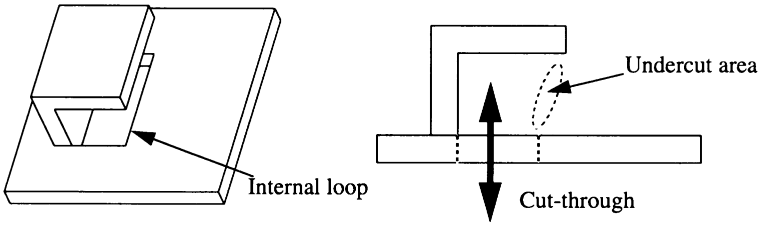

Figure 8: Internal parting-patches undercut.

7.2 User interface and 'parting-loop' features

The interface for selecting parting loops is a critical issue in automatic undercut detection. We have addressed this issue by developing a semi-automatic method for selecting parting loops. Nevertheless, users have to re-select parting loops every time they want to detect undercuts. For this reason, we need to be able to store parting loop data along with the solid model. This would be possible if the CAD system allows users to define 'parting-loop' features.

8 CONCLUSIONS

The originality of our approach for detecting undercuts is that we are analyzing the geometry of the mold-halves instead of analyzing the geometry of the part. This approach allows the detection of 'actual' undercuts instead of 'potential' undercuts such as in [Rosen, 1992] and [Hui, 1992]. Algorithms in [Tan, 1988; Liou, 1991; Gunter, 1990] can not detect some undercuts which are common in industrial components. Other algorithms [Chen, 1992a; Mochizuki, 1992] are based on the visibility-sphere concept which is computationally-expensive for detailed solid models.

Compared to existing algorithms, the proposed algorithms are more efficient because they take advantage of the adjacency information available in B-Rep solid models. We believe it is the first time that an undercut detection algorithm was implemented and tested successfully on models of industrial components.

9 REFERENCES

Bralla, J. G., 1986. *Handbook of Product Design For Manufacturing, A Practical Guide to Low-Cost Production.* New York: McGraw-Hill Book Company.

Chen, L.-L. and Chou, S.-Y., 1992a. "Optimal Parting Directions By Partial Visibility." *PED-Vol. 59, Concurrent Engineering, ASME,* 17-26.

Chen, L. L., Chou, S. Y., and Woo, T. C., 1992b. *Optimal Parting Directions for Mold and Die Design.* Department of Industrial and Manufacturing Systems Engineering, Iowa State University (submitted to J. of CAD). Technical Report 92-105.

Ganter, M. A. and Tuss, L. L., 1990. "Computer-Assisted Parting Line Development for Cast Pattern Production." *Transactions of the American Foundrymen's Society*, 795-800.

Hui, K. C. and Tan, a. S. T., 1992. "Mould design with sweep operations - a heuristic search approach." *Computer-Aided Design*, Vol. 24, No. 2.

Liou, S.-Y., 1990. "Diecast: A Knowledge-Based Approach to Die Casting Design." Ph.D. dissertation, The Ohio State University.

Liou, S. Y. and Miller, R. A., 1991. "Design for die casting." *Int. J. Computer Integrated Manufacturing*, Vol. 4 , No. 2, 83-86.

Mochizuki, T. and Yuhara, N., 1992. "Methods of Extracting Potential Undercut and Determining Optimum Withdrawal Direction for Mold Designing." *Int. J. Japan Soc. Prec. Eng.*, Vol. 26, No. 1, 68-73.

PTC, 1993. *Pro/ENGINEER Mold Design User Guide v.13*. Parametric Technologies Corporation, Waltham, MA.

PTC, 1994a. *Interface Guide, Pro/ENGINEER v.13*. Parametric Technologies Corporation, Waltham, MA.

PTC, 1994b. *Pro/DEVELOP User Guide, Pro/ENGINEER v.13*. Parametric Technologies Corporation, Waltham, MA.

Ravi, B. and Srinivasan, M. N., 1990a. "Computer Aided Parting Surface Generation." In *Proceedings of Manufacturing International '90 in Atlanta, GA, Part 4: Advances in Materials and Automation*, 125-129.

Ravi, B. and Srinivasan, M. N., 1990b. "Decision criteria for computer-aided parting surface design." *Computer-Aided Design*, Vol. 22, No. 1, 11-18.

Rosen, D. W., 1992. "A Feature-Based Representation to Support the Design Process and the Manufacturability Evaluation of Mechanical Components." Ph.D. Thesis, University of Massachusetts, Amherst, MA.

Serrar, M. and Gabriele, G., 1995a. "Automatic Generation of Parting Surfaces and Mold Halves." In Proc. of the ASME Computers in Engineering Conf. in Boston, MA, 75-82.

Serrar, M., 1995b. "Applying Moldability Evaluation to Solid Models." Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, NY.

Tan, S. T. et al., 1988. "A method for generation of parting surface for injection moulds." In *Conference on computer aided production engineering in Edinburgh*, 401-408.

Weinstein, M. and Manoochehri, S., 1994. "Geometric Influence of a Molded Part on the Draw Direction and Parting Line Locations." *DE-Vol. 69-1, Advances in Design Automation ASME*, Vol. 1, 585-592.

10 BIOGRAPHY

Mouhsine Serrar is a staff engineer at Motorola, Paging Products Group. His interests are in design automation, solid modeling, and in Open CAD technology. Serrar received his bachelor from Ecole Mohammadia in Rabat, Morocco (1988), his MS from Ecole Polytechnique in Montreal, Canada (1991), and his Ph.D. in mechanical engineering from Rensselaer Polytechnic Institute, Troy, USA (1995).