
C.O.S. 2.0 - FROM THE OPTICAL IMPRESSION TO THE INLAY

G. SAUGER, W. PAULUS, U. ORTH, G. STEIN, I. LIEBALD, J. PFEIFFER
Siemens AG, 6140 Bensheim, Fabrikstraße 31, Germany

In December 1990 the new Cerec Operating System (C.O.S. 2.0) was released. Besides a complete redesign of the man machine interface many aspects of the algorithmic data handling were improved. A detailed analysis and description of the current C.O.S. data processing is given, including a glance at the drafting tools of C.O.S.

The shape of the Cerec inlay is calculated by the system on base of three dimensional surface data and input given by the dentist. The data processing, as it takes place during a Cerec session, includes acquiring of SD-surface information, input of characteristic 2D-lines, image enhancement, image interpretation and special algorithms to create a volume model of the restoration.

The edges of a cavity are detected automatically. As input information the algorithm needs bottom line and equator. Each segment of the bottom line is classified in order to find the segments, the edge detection algorithm has to work on. Prior to edge detection two invisible lines, centered around the given segments, are created. These lines deliver basic points for calculating straight lines through the height profile. On each cut an edge point is calculated. The resulting points for one edge line are connected by linear interpolation, weighted and smoothed by mean value filtering before they are shown.

C.O.S. 2.0 simplifies the design of a Cerec restoration. The dentist is encouraged to go into the field of greater and more complex restorations.

The Cerec method has been available to the dentist for a period of more than 4 years. Since that time the system has been improved in medical as well as technical aspects. One of the major improvements was the development of a new Operating System for the Cerec unit. Starting point was a careful analysis of the needs of Cerec dental practitioners. We found that the user interface was crucial. We decided to implement a graphical man machine interface that supports the

dentist in the CAD/CAM process by giving hints and warnings. The overall goal of C.O.S. is getting the ceramic restoration as fast as possible; dead ends and detours on the way to that goal have to be avoided. Any on-screen information presented to the clinician had to be seen on this demand.

The Cerec Operating System (C.O.S.) was created from scratch. The system has an overall size of about 320 Kbytes. The size of the code segment is about 240 Kbytes. The rest of the memory is used for the storage of user defined data (e.g. bottom line, equator, Window Manager etc.). C.O.S. 2.0 was implemented using the “high level”- programming language ‘C’. Time-consuming procedures were written in assembly language.

The Cerec application is designed as user-driven dialog system. Guiding lines for the development of C.O.S. 2.0 were the improvement of the user friendliness and the simplification of the overall CAD/CAM process, which is needed to get Cerec restorations.

The Man Machine Interface

During the last decade software companies have done a lot of research on the improvement of the interface between man and computer (1). The current results are very promising: Even today’s programs increase the productivity of the user predominantly. Applications with good man machine interfaces should allow the user to center his mind on problem solving. Actually the problem should not be the application itself. With the new Cerec Operating System the switches are thrown and we are heading for that direction too.

A rather important step on the way to a friendly man machine interface is the usage of graphical elements that are able to represent possible user actions. These paradigms (push buttons, check boxes, radio buttons etc.) are widespread in graphical user environments (2), because they actually can ease the recognition of the different functions. For C.O.S. 2.0 we implemented the subset of these elements (Fig. 1), that is needed for our purpose.

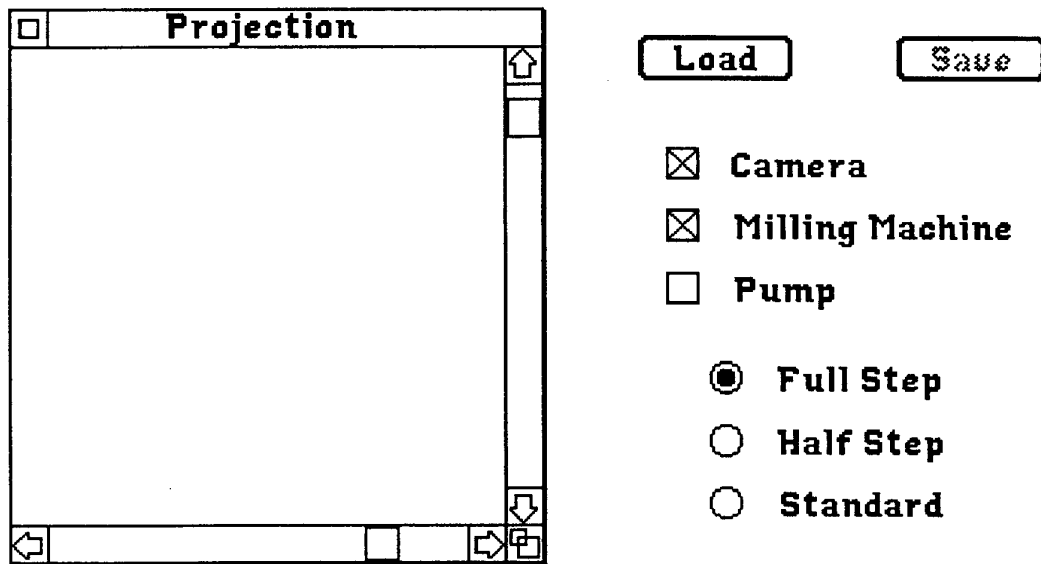


Fig. 1: Graphical elements of the C.O.S. Man Machine Interface

To reduce the struggle of learning, the system can be adapted to the native language. Throughout the design process the user may receive messages, that give hints or warnings (Fig. 2). C.O.S. proposes the next step that has to be done. The user is able to change proposals, but if he follows the system it shows him the straight way to the restoration.

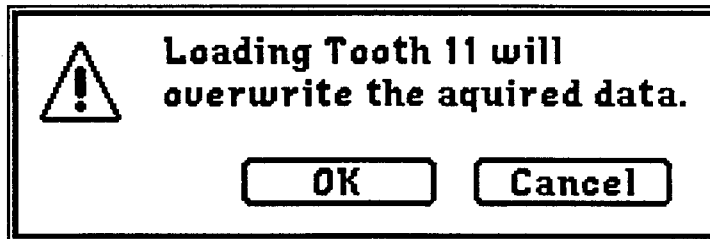


Fig. 2: One of the warnings given by C.O.S. during the design process.

Whenever the user is drawing or editing parts of the inlay, he can judge the results immediately. The results, shown on screen, are identical the changes in the restoration (WYSIWYG “What you see is what you get”).

The Tooth-Editor

A major part of C.O.S. is the definition of the restoration. This is done by a drafting tool that uses the image of the cavity as background for drafting. We decided to create an interactive tool for that purpose. Especially with the cavity as background the user can judge his drawings instantaneously (Fig. 3b).

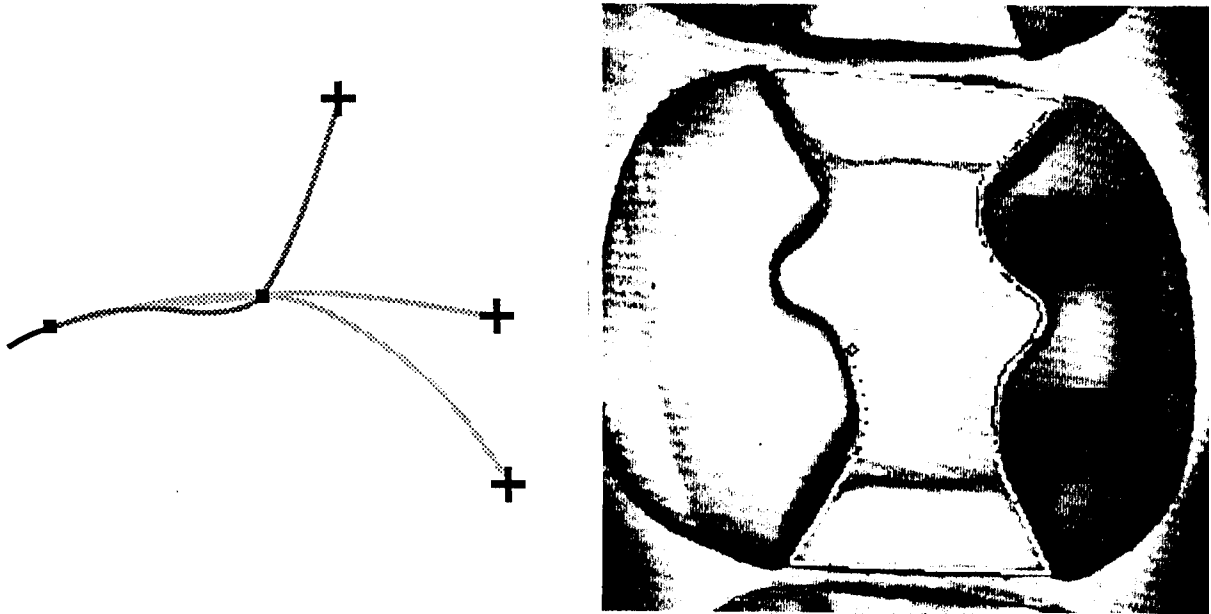


Fig. 3: The principle of the drafting tool (a) and its realization (b)

After entering the tool, a line seems to flow out of the cursor, when the user moves the trackball. By clicking the pedal the user fixes the given line to that position. Still, moving the cursor (Fig. 3a) can change the line between the last two clicked points. The rest of the line is already fixed and may be changed later by the editing tool. The 'Delete' button, located on the soft-key panel, also can remove a fixed point,.

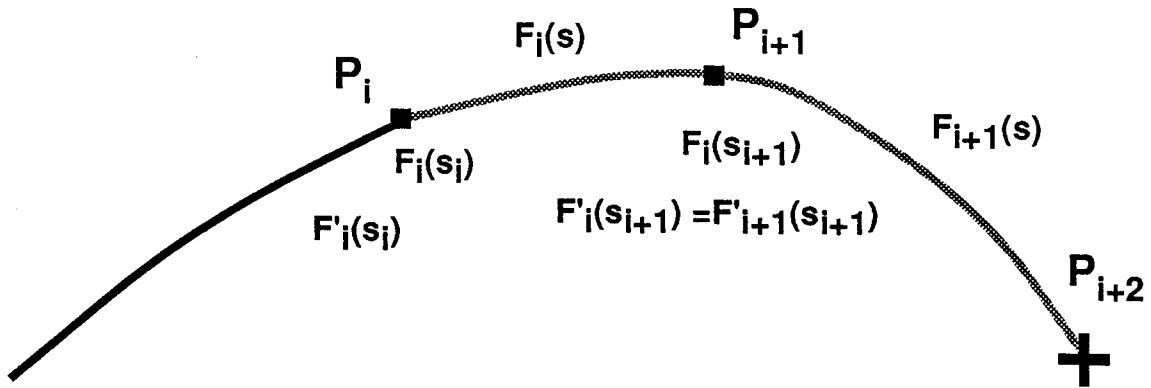


Fig. 4: Interpolating parabolas

To calculate the line segment between the user defined points, we use an interpolation with cubic parabolas (Hermit Interpolation) (See also Ref. 3). The internal representation of all the lines is independent of a special coordinate system. To achieve this, we use a path length description, which is denoted by s in formula (1). To determine the actual slope of any parabola, you need to know all the a_i . This is done by 4 linear independent equations.

$$F_i(s) = a_0*(s-s_0)^3 + a_1*(s-s_0)^2 + a_2*(s-s_0) + a_3 \quad (1)$$

The user-defined Points P_j, P_{j+1} and the first derivative at that positions (Fig. 4) form these equations. Neighboring segments are connected in a smooth way: The derivatives of both segments are equal at the connection. Double clicking ends up the interpolation process and a new one may start.

Edge Detection

C.O.S. 2.0 supplies an algorithm that is able to detect the edges of a given cavity. The algorithm actually does not depend on the direction of the edge.

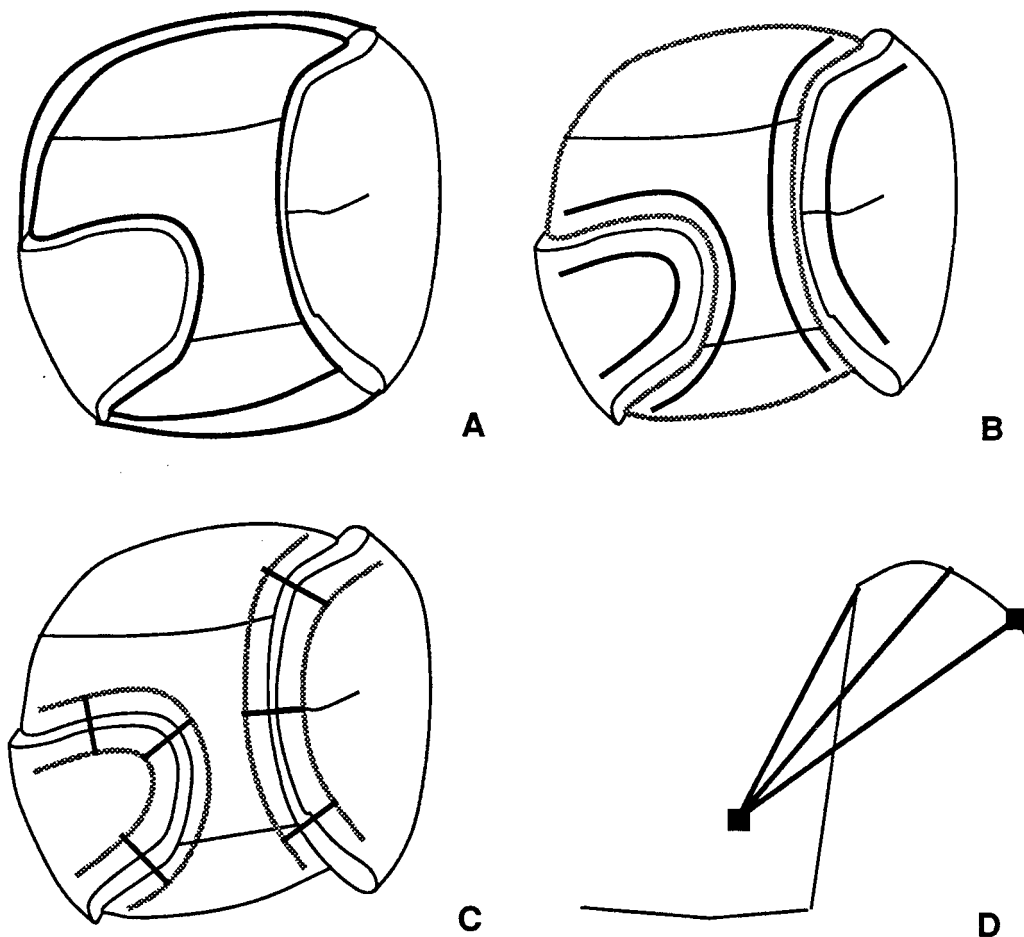


Fig. 5: The edge detection algorithm

Prior to edge detection the measured height data are filtered by a combination of median and mean value filter. This kind of filter allows on one hand the suppression of noise but on the other hand it maintains clear sharp edges.

The C.O.S. 2.0 edge detection algorithm needs a bottom line and equator segments as input (Fig. 5a). The bottom line may have - according to the inlay - more than one segment. For each segment the algorithm calculates the distance to the nearest equator segment. If the distance of this segment is greater than 0.8 mm then the edge detection algorithm will search an edge for that segment. The procedure starts by calculating two invisible lines, one shifted to the center of the inlay by about 0.5 mm and the other shifted about 1.0 mm away from the center (Fig. 5b). These lines deliver the basic points for calculating straight lines through the height profile (Fig. 5c). On each cut an edge point is found. This point is defined by the horizon one may see standing at the foot line (Fig. 5d).

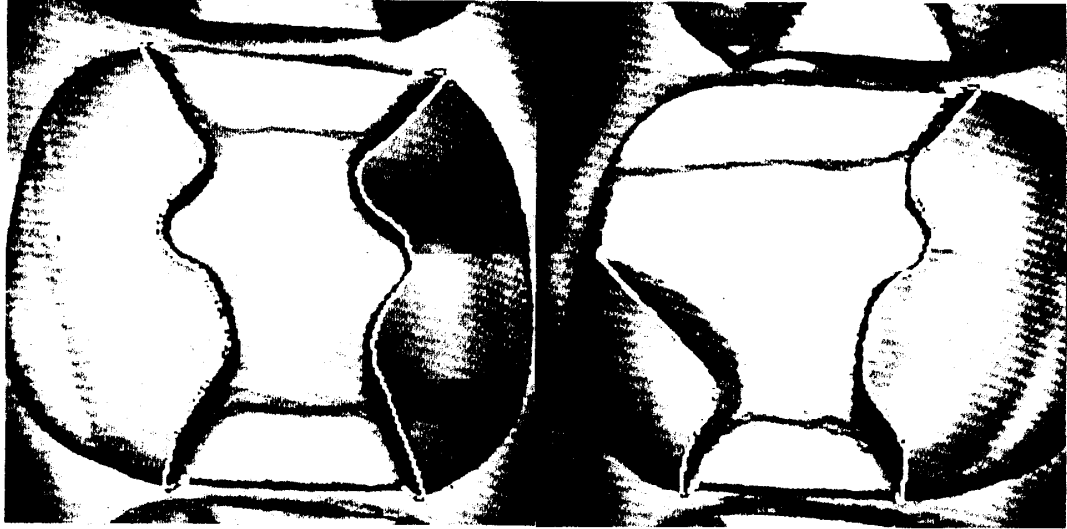


Fig. 6: Results of the edge detector for some dental scenes

The resulting points for one edge segment are connected by linear interpolation and smoothed by a mean value weighted filter before they are presented to the dentist. This procedure is justified in most cases (Fig. 6). It is obvious that a clear preparation is very helpful for the edge detector.

The Volume Model

At the end of the design process, the user has supplied the bottom line, the equator segments, the edge lines and the cusp segments of the current tooth (Fig. 7a) that has to be reconstructed. Using these lines and the measured height data C.O.S. calculates a Cartesian representation of the inlay. As bottom of the restoration the measured height data are used directly (Fig. 7b). The occlusal area between the edge lines and the cusp segments is linearly interpolated on a slice by slice basis (Fig. 7c). Cubic parabolas (Fig. 7d) define all other surfaces of the inlay. There are some geometrical limitations that arise from the milling disc. These limitations are considered at that stage.

At the end of these calculations the milling process is started. During this process the Cartesian representation is transferred to a special cylindrical representation that can be used to control the milling machine.

Fig. 7: Building the volume model of a restoration

The user-friendly man-machine interface of C.O.S. 2.0 simplifies the design of a Cerec restoration. The dentist is encouraged to go into the field of greater and more complex restorations.

The only image enhancement that can be done by C.O.S. is the filtering of measured height data. This fact makes it crucial to supply good quality data. Powdering of the tooth, good calibration of the camera and a clearly focused cavity is essential for a good fitting restoration.

The edge detector works well with clearly cut cavities. Any distortion from that need will cause a bigger editing effort.

C.O.S. 2.0 is the result of many helping hands and thinking minds. We would like to give our applications to all of them.

The development of C.O.S. 2.0 is part of the Cerec R&D activities of the Siemens AG.

1. Shneiderman, B., Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley Publishing Company, Reading, Massachusetts 1987
2. Brown, C. M., Human Computer Interface Design Guidelines, Ablex Publishing Company, Norwood New Jersey 1986
3. Spath H., Spline Algorithmen zur Konstruktion glatter Kurven und Flächen, Oldenbourg, Munchen 1986