A SIMULATION ENVIRONMENT FOR PASSENGER CAR SEAT OCCUPANT RECOGNITION

Eduardo A. Destéfanis (1), Luis R. Canali (1), Uwe Franke (2), Eberhard Kienzle (3)

(1) Grupo de Inv. en Informática para la Ingeniería, UTN, Fac. Regional Córdoba.
(2) Daimler-Chrysler AG. Forschungsinstitut Esslingen. FT3/AB.
Intelligente Fahrerassistenzsysteme
(3) Fachhochschule für Technik Esslingen. Fachbereich IT.

CC 36, Sucursal 16, 5016, Córdoba, Argentina
Fax: (+54)(351) 4681823 E-mail: destefan@sa.frc.utn.edu.ar

Keywords: pattern recognition, simulation, CAD systems.

Abstract: The use of air bags in presence of bad passengers and baby seats positions in cars' seats can injure or kill them in case of accident when this device is inflated. A proposed solution is the use of range sensors to detect passenger and baby seats risk positions. Such sensors allow to control the Airbag inflation. The present work is related to developing a CAD Companion Seat environment with the aim to allow a systematic analysis of the occupant detection problem.

1. INTRODUCTION.

Recent statistics have given more evidence about the risk involved when air bags are used in presence of unbelted passengers and rear-oriented forward-placed child seats. In last years occupant detection and recognition research has become more important due to the use of air bags is extending today to all personal cars, and the number of them is increasing with lateral, windows and back seats air bags.

Many solutions based on different sensors technologies have been proposed to solve this problem. One of them is the use of range sensors to detect passenger and baby seats dangerous positions. To construct and test different range sensors configurations can be eased by simulation. The opportunity to use a wide variety of passenger models is also increased by simulation. Simulation is carried out by using a general purpose CAD system which allows the user to define, store, modify and use different types of distance sensors. It allows the user to select a desired set of models related to companion seat, cockpit, child safety seats and other objects involved, to use an anthropometric model which ensures an accurate set of anthropometric measures, and to control nested loops to extract the desired amount of patterns.

2. CAD MODELS OF COMPANION SEAT ENVIRONMENT AND THEIR ASSOCIATED DEGREES OF FREEDOM.

2.1. Companion seat.

Two basic companion seat types exists: Mechanical and electromechanical. In mechanical seats the base of the seat can move along a rail. This rail has a fixed position and angle. In electromechanical seats the base level and angle can vary. Here only a mechanical seat model is considered. Results can be extended afterwards to electrical seat models. Companion seats have been defined whit independent models for backrest, base and head guard. Translation and rotation degrees of freedom (dof) can be controlled interactively. Companion seat plays a central role in pattern extraction by simulation. To extract an important amount of patterns a loop is performed. During it, the seat is moved from a back to a front position with a given step. Backrest angle and head guard position can also change with a given step. Passenger or child seats can be placed on companion seat and carried on it during this animation. One pattern is extracted and stored for each companion seat position.

2.2. Sensor.

Different types of sensors based on two technologies have been proposed and tested with different classification schemes in CAD systems and in a real car. The technologies involved in these sensors are: Photonic Mixer Device Sensor (Schwarte et al, 1997a,b) and Infrared Sensor (Spies H 1998). The overhead console has been selected to place the sensor considering the following aspects:

- Existing wiring going trough the car to the overhead console.
- Existing support fixtures and infrastructure at the same place.
- Existence of a sunshade protection in the middle and over the companion seat.
- Difficulty to locate the sensor at the center of the car's dome.
- Difficulty to obtain a better classification from other candidate points (e.g. between and over the doors, on front panel in front of the passenger, etc.).
- Ideal exposure of companion seat to the sensor.

From a geometrical point of view sensor models have been classified as line and matrix sensors. In line sensors all the emitters are arranged on a line, and all the rays lie on the same plane. This plane is here referred as line sensor plane. Line sensor can be defined with different number of rays, angles, range, etc. A set of CAD operations is used to orientate this plane. To do this, a direction vector is used as a geometrical reference.

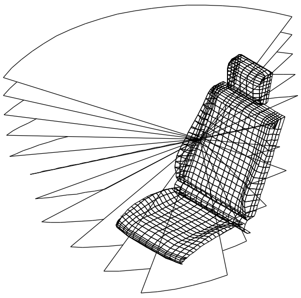

Fig. 1. Matrix sensor. Rays lie on different planes.

It is possible to define a matrix sensor model by defining and inserting a selected number of line sensors planes at the same point. The direction vector of each plane allows to define the relative position between them. Matrix sensors has been used with the goal to determine the optimal number of rays to obtain a reliable classification.

A matrix sensor composed by eleven line sensors planes is shown in figure 1. In this case each plane has a different amount of rays. Angles between rays are the same in both row and column directions. The rays distribution of this sensor model is based on the constraint that it is necessary to measure the whole companion seat space for an early classification analysis. This space corresponds approximately to a spherical octant.

2.3 Child safety seats.

To obtain the patterns for child and baby seats they have been classified in three groups. Forward Facing Child Seats (FFCS), Rear Facing Infant Seats (RFIS) and baby seats. Some of these models have been digitized. Their possible positions and degrees of freedom have been studied in detail, with the aim to reproduce real world events.

Infant bodies have been considered with a given posture and anthropometric data. That is without dof information. This simplification contributes to reduce irrelevant degrees of freedom in the whole system.

Rear Facing Infant Seats:

Two examples are shown in figure 2. In RFIS infant body is usually hidden to the sensor by the child seat backrest. As a result of the special position of this child seat on the companion seat it is not possible for it to rotate with respect of the companion seat base (Other RFIS models not shown here can do it).

In figure 2b. RFIS backrest can rotate and move along RFIS base. Rotation constraints respect to the companion seat base are the same as in the previous example. Degrees of freedom for RFIS models are:

- Translation with respect its own base.
- Backrest rotation respect its own base.
- Translations and rotations (If possible) respect to the companion seat base.

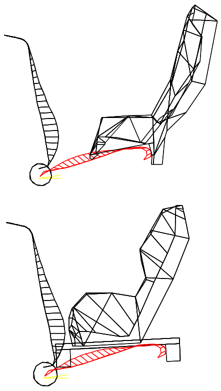

Fig. 2: Child seats models.

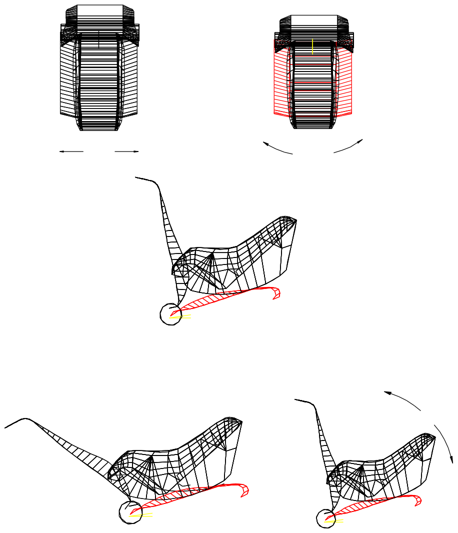

Fig. 3: Maxicosi baby seat model degrees of freedom.

The patterns are extracted considering a set of positions referred to RFIS dofs, the limit established by the board panel of the car, the opportunity to use a sunshade with some models and a set of positions and angles of the companion seat.

Forward Facing Child Seat:

Patterns for FFCS are obtained considering only companion seat backrest angles for which this model really fits.

Baby seat:

An example is shown in figure 3. This example is given to show how freedom degrees and fit problems have been taken under consideration during simulation with a baby seat. This seat is used as a rearward facing seat which can be used in the front or rear seat of the car. In this case, for different angles of the vertical part of the companion seat, different positions of the babyseat exist, as is shown in fig 3.

Baby seat out of position implies in this case translations and rotations, as it is shown in figure 3.

In this baby seat model a combination of dofs shown in figure 3 is also possible. Different reference distances between this baby seat model and a defined mechanical companion seat model have been measured. With these measures a table has been constructed and stored. When the program runs in a loop extracting a big amount of patterns that corresponds to a baby seat model, by looking into this table the program can automatically fit the baby seat in the right position for every situation. Since other baby seats models are similar to this model, it is possible to fit other baby seats too. This simple method has become important in this work since to obtain patterns one by one is extremely tedious and requires an unpredictable amount of machine time.

2.4. Cockpit and other objects.

Models for cockpits, and other objects related to the companion seat environment (Newspaper, hat, bag and so on) have been devised by using standard CAD methods for complex entities creation. This means they have been stored in a separate file and inserted into the current environment. Cockpit models must include a definition of two points. One point is the origin of a reference coordinate system to be used as the companion seat environment reference. The second point defines sensor location.

2.5. Passenger.

A human model is needed to represent the passenger and his positions on the companion seat. To obtain this, an existing anthropometric program has been linked with the CAD environment. A human model generated with this program (Anthropos®), is shown in figure 4. Another way to use a human model in the CAD environment consists in developing a custom anthropometric program.

A method to compute human types called Anthropometric Data Reduction (ADR) gives an important reference to define human model degrees of freedom. This method has been developed in order to find a statistical approach, that allows the program user to avoid the need to select different types from an anthropometric table (Geuss, 1995b). This task demands a high level of expert knowledge. The base of this method, is to consider only a few group of human proportions as *relevant measures* in order to define human bodies. To detect which are these relevant measures, the statistical correlation of various body measurements has been observed. Researchers have found four relevant features: Height, proportion, corpulence and gender.

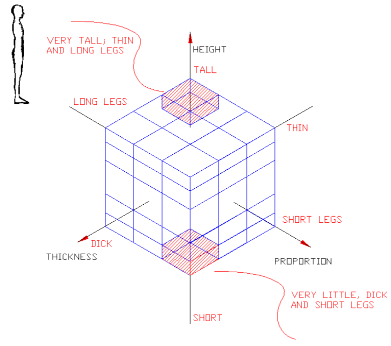

Fig. 3 Anthropometric feature space of ADR types..

For an initial set of human types the existing possibilities have been reduced by considering the .following cases:

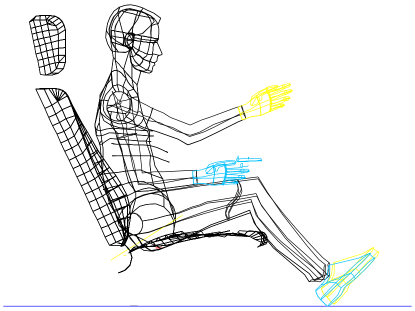

Fig. 4. Anthropos ® human model

- Height.: Very small - small - Medium - Big - Very big.
- Proportion: Long legs - Medium legs - Short legs.
- Corpulence: Thin - Normal - Thick.

By combining the features named corpulence and proportion nine combinations are obtained:

- Thin/long legs
- Normal/long legs
- Thick/long legs
- Thin/medium legs
- Normal/medium legs
- Thick/medium legs
- Thin/short legs
- Normal/short legs
- Thick/short legs

By combining them with the Height feature it is possible to obtain forty five combinations, and by considering Male / Female feature, the total amount of combinations is ninety.

The importance of this method to use it in occupant recognition simulation is to know which human measures are statistically relevant. Based on this *relevant measures* a *set of anthropometric models* has been defined. By discarding the Proportion feature for the present purpose, this set has been reduced. It contains thirty anthropometric model definitions. This analysis implies to consider three degrees of freedom for human model definition. This population has the property that it represents all possible human bodies with an equiprobable distribution.

In the other side a *set of human postures* has been defined too. By applying the set of human postures to each model in the anthropometric set a general population of patterns has been obtained for passengers.

How to define this set of human postures is a critical question. Statistics and experience related to certain postures must be considered. E.g. A passenger reading a newspaper. In the other side is possible to investigate ways to reduce degrees of freedom related with human postures.

One human posture can be defined in terms of positions of the more relevant skeleton members. Results have shown that patterns obtained with a matrix sensor oriented to the head guard are needed. In this case, legs are not considered to define postures degrees of freedom. Hands and fingers movements are not considered. They are not relevant in case of a sensor with an important amount of rays (100). Remaining dofs can be considered for head and neck (3), torso (3), hip (2), position respect the seat (1), clavicles(4), and arms (10). Start and end positions and restriction by objetcs must be considered too.

Even when only torso (3), hip (1), position (1) and arms (10) are only considered the number of degrees freedom remains too high. A method is needed to eliminate arms dofs or to find some structural feature capable of provide a simplified scheme. A random selection of postures to train a nnet is also possible.

3. PATTERN GENERATION.

3.1. Intersection points.

The problem to find sensor patterns is related with the problem to find distances between sensor focus and intersection points. Intersection points are defined by objects surface and ray lines. The CAD models used here have been based on a mesh surface description. Such surface was composed by four vertex polygons. An usual approach to find intersections between lines and four vertex polygons is to consider each polygon as composed by two planes. So the problem becomes to find intersection points between lines and planes. The mathematical procedure to resolve this problem can be outlined in the following way.

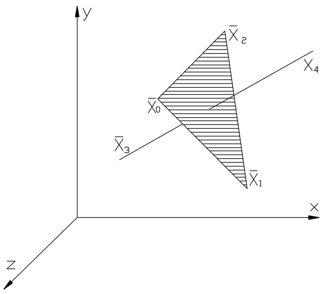

Fig.5.Intersection of a plane and a ray

Fig. 5 shows a plane defined by three points $\bar{x}_0, \bar{x}_1, \bar{x}_2$ and a line defined by two points respectively. Where

$$\bar{x}_i = (x_{ix}, x_{iy}, x_{iz}) \quad \text{i}=1, 2, 3 \qquad (1)$$

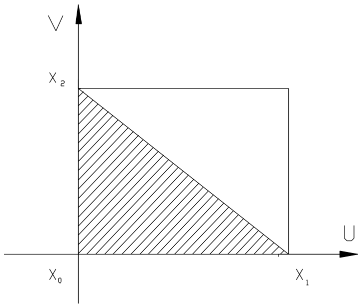

Fig. 6.Definition of a plane

A parametric equation of a plane can be given as (Fig. 6)

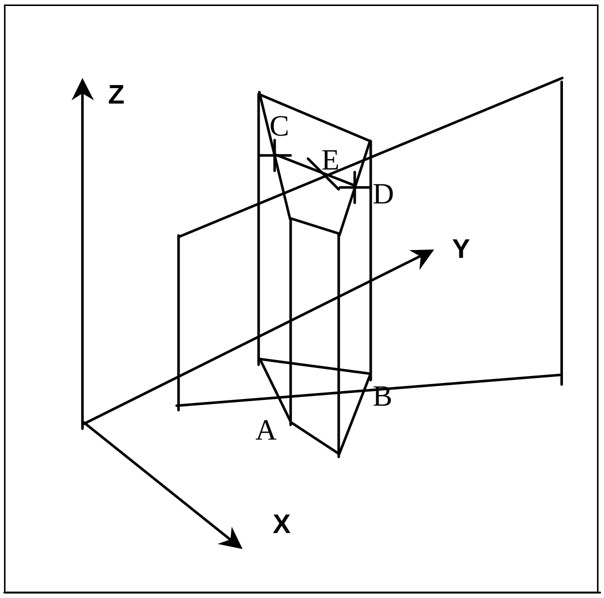

Fig.7 Polygon and ray intersection

$$\bar{x} = u(\bar{x}_1 - \bar{x}_0) + v(\bar{x}_2 - \bar{x}_0) + \bar{x}_0 \qquad (2)$$

with the following conditions

$$u + v < 1 \qquad (3)$$
$$0 < u \qquad (4)$$
$$0 < v \qquad (5)$$

The equation of a line is

$$\bar{x} = t(\bar{x}_4 - \bar{x}_3) + \bar{x}_3 \qquad (6)$$
$$0 < t < 1 \qquad (7)$$

From (2) and (6)

$$u(\bar{x}_1 - \bar{x}_0) + v(\bar{x}_2 - \bar{x}_0) + \bar{x}_0 = t(\bar{x}_4 - \bar{x}_3) + \bar{x}_3 \qquad (8)$$

and

$$u(\bar{x}_1 - \bar{x}_0) + v(\bar{x}_2 - \bar{x}_0) - -t(\bar{x}_4 - \bar{x}_3) = \bar{x}_3 - \bar{x}_0 \qquad (9)$$

From here is possible to compute u, v, t and after, by replacing in (2) and (6) the solution can be obtained. The solution gives an intersection point only when values of u and v satisfy (3), (4),

(5) and (7). Pathological cases can be singled out by controlling a null determinant.

3.2. A simple method to compute intersections.

To find intersection points between rays and polygons, a simple method has been and used. It consists in to project both, ray and polygon on to a reference plane, and to find the intersections between the projected polygon sides and the ray projection (Points A and B, fig. 7). The program is encoded to find only two boundary intersection points. (Intersection trough polygon vertex are a pathological case). The third coordinate for 2D intersection points is then computed based in the original 3D vertix of the polygon. That gives the 3D intersection points with the polygon boundary (Points C and D). This two points define a line. The intersection of this line with the ray line gives the intersection between polygon and ray (Point E). This intersection can be obtained by line projection too.

The benefit obtained from using this simple method is related with the fact that it needs to compute 2D lines intersections five times and it is necessary to find z coordinates three times. These operations involve about 40 flops.

When using the mathematical approach to solve this problem, as has been explained before, the four sided polygon must be separated in two triangles. After it is necessary to check both triangles for intersections. To do this it is necessary to find each inverse matrix (Formula 2.30). Only to compute each inverse matrix in the best case about 40 flops are required.

3.2. 3D Transformations.

Every CAD program needs to perform a set of 2D and/or 3D transformations. These operations allow to make translations, scalings, rotations, reflections and so on. Such operations are based in matrix algebra, and are performed trough matrix products, It is important to unify the matrixes that correspond to each transformation in one global transformation matrix that performs the desired transformation operation. This requirement is given in order to reduce computing costs. (Hearn and Baker, 1988). Since system coordinates are given by a set of points, it is possible to use these transformations in order to obtain a relative coordinate system (RCS) based in a main coordinate system. The main coordinate system is usually designed as World Coordinate System (WCS).

A quick algorithm has been developed in this work to find the polygon set that is composed by polygon candidates to be intersected by each sensor ray. As a first step, such algorithm requires to transform WCS into a given RCS, with the condition that the end point of the current ray defines the origin of the current RCS (xr, yr, zr), and ray origin (Sensor location) defines the Z axis direction of the current RCS system. As a second step, one point of each polygon must be transformed from WCS in current RCS with the goal to check his proximity with zr axis (The current sensor ray). Other steps of this algorithm are described afterward. The basic 3D transformations used in this algorithm are the following::

Translation: A point is translated from (x,y,z) to (x’, y’, z’) with the operation:

$$[\,x\;y\;z\;1\,] = [x\;y\;z\;1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 1 \end{bmatrix} \qquad Tx, ty, tz \in R \tag{10}$$

Tx, Ty, Tz correspond to translations distances, that is x’ = x + Tx y’ = y + Ty z’ = z + Tz, Every 3D object in CAD database is translated by translating his defining points.

Rotation: To rotate an object it is necessary to define a rotational axis, and the desired rotational angle. In 2D the rotational axis is ever perpendicular to xy plane. In 3D a rotational axis can have any direction in space. The simplest 3D case is given when rotational axis are parallel to WCS axis. That gives the possibility of compute rotations around every axis in 3D space trough chaining rotation matrixes that correspond to rotations around WCS axis. A common convention is that positive values of rotational angles produces counterclockwise rotations. That arises from the consideration that the origin is viewed from positive values of a WCS axis. A 2D rotation is given by the following transform equations for compute rotations around x and y axes.

$$[\,x'\;y'\;z'\;1\,] = [x\;y\;z\;1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos q & \sin q & 0 \\ 0 & -\sin q & \cos q & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$[\,x'\ y'\ z'\ 1\,] = [x\ y\ z\ 1]\begin{bmatrix}\cos q & 0 & -\sin q & 0\\ 0 & 1 & 0 & 0\\ \sin q & 0 & \cos q & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{12}$$

Coordinate system transformation matrix: In order to use the line defined by sensor ray to define a RCS which z' axis is collinear with the ray defining line, and to translate polygon vertexes into this coordinate system, it is necessary to find the related transformation matrix. Extreme points of a given segment (z') define the vector:

$$V = (x2 - x1,\ y2 - y1,\ z2 - z1) \tag{13}$$

The following translation matrix translates the WCS origin in to RCS origin:

$$T = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ -x1 & -y1 & -z1 & 1\end{bmatrix} \tag{14}$$

After translation it is necessary to align the RCS with z': To do this, two steps are considered, first a rotation around x is performed, so u lies in xz plane. Second, a rotation around y is performed to align u with z. Rotation matrices around x and y axes can be written as:

$$R_x(a) = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & c/d & b/d & 0\\ 0 & -b/d & c/d & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{15}$$

$$Ry(b) = \begin{bmatrix}d & 0 & a & 0\\ 0 & 1 & 0 & 0\\ -a & 0 & d & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{16}$$

After transformation matrices (14, 15 and 16) z axis is aligned with z'. By using
$R(\theta) = T \,.\, Rx(\alpha) \,.\, Ry(\beta)$
a matrix is obtained to translate points of interest from one coordinate system into another. So in terms of computing costs each point [x y z 1] translation requires 28 fps. For a 7000 piece Anthropos polygon definition the figure is:

$7000 \times 28 = 196000 \cong 200000$ flops.

3.4. Algorithm to find polygons candidates to intersections.

The intersection algorithm involves a few main tasks, related to each ray of the sensor.

1. Obtain an *initial points set* (By selecting one vertex from each polygon).
2. Find the *candidate points set* to one ray from the *initial points set.*
3. Compute intersections for polygons associated with each point in the *candidate points set.* An *intersection points set* is obtained.
4. Found the *minimal distance* from point to emitter from *intersection points set.*

These steps must be repeated for each ray to find the current pattern.

An initial problem is how to select quickly the polygon group that is near enough to one ray. For this purpose it is sufficient to consider only one polygon vertex for each polygon. In this way a “cloud” of points (xr, yr, zr) in 3D space is obtained. This *initial points set* represents all polygons that compose objects in the current environment. From this set, it is necessary to identify which points are near enough to one selected ray.

To reach this goal a method based on translating the polygon vertexes from a coordinate system to another has been used. (The computing cost arises by the need to translate all the tested points from World Coordinate System to Relative Coordinate System).

In 3D space, it is common to consider a reference coordinate system and a relative coordinate system to define positions and other points of interest. In the cockpit environment the rays of the sensor are located in a World Coordinate System. (X, Y, Z). Each of these rays can be used to define his own relative coordinate system (Xr, Yr, Zr), e.g. by defining the Z axis, this situation is shown in figure 7.

It is possible to define a box centered on the origin point of this relative coordinate system defined by one ray. The base of this box defines a square lying in Xr Yr plane (Zr = 0).

From the *initial points set,* mentioned before only points which coordinates (xr, yr, 0) lies inside the box are the points of interest (*Candidate points set*).

After these c*andidate points set* have been found, it is necessary to test if polygons that correspond to these points intersect the ray or not. The polygons that intersects the ray are stored in a buffer (or marked). By this way a *intersection points set* is obtained.

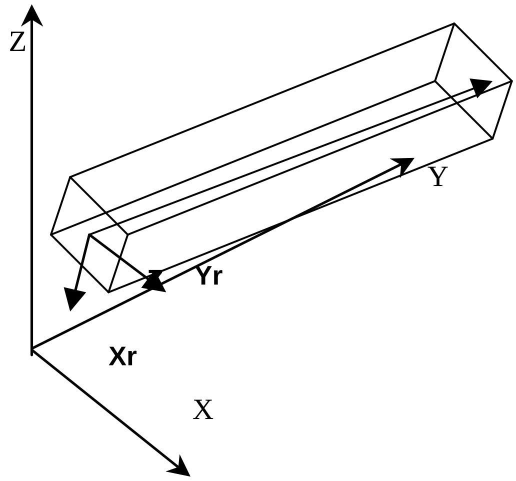

Fig. 8.A box centered in Z_r axis in RCS space.

After it the *nearest intersection point* (to sensor location) is selected.

By repeating this process for each ray a *nearest intersection point set* is obtained.

By computing distance between each point in *nearest intersection point set* and sensor location point, a *pattern* is obtained.

3.5. Software modules.

A function has been encoded performing the tasks referred above, read polygons files, find intersections and store patterns. Nested loops are performed to allow the user to obtain thousands of patterns for different man-model postures and companion seat positions, running the program only once. Modules are outlined in figure 9.

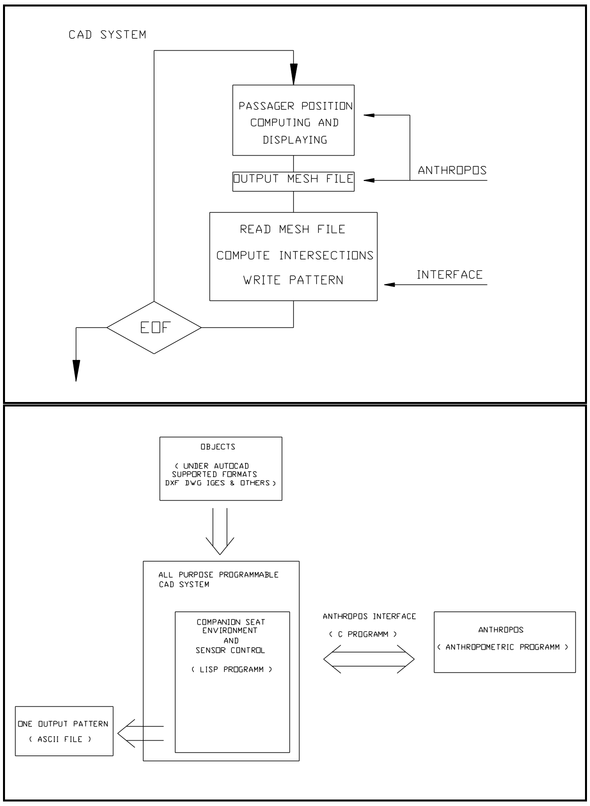

Fig. 9: upper: Anthropometric-CAD interface module. Lower: System modules.

The parameters defining each passenger position are internal of the anthropometric program and are not available for to use them in other programs. Because of this, the anthropometric program has been modified to write an output file containing a polygon set corresponding to each passenger position (Mesh model) after each mesh model is computed. Then, the interface program between the anthropometric program and the CAD environment, reads the polygon definition, compute intersections, and stores intersections points into the CAD database.

4. RESULTS AND FUTURE TRENDS.

An infrared sensor composed by eight fibers lying on the same plane has been constructed by Temic GMBH. This sensor has been tested in a real cockpit and by CAD simulation with different classification schemes. These experiences and results related with the classification problem are described elsewhere. As a result of these experiences a different sensor is actually under development. It is based on the so called Photo Mixer Device (PMD). Based on PMD technology, is possible to build in a few years a low cost 3D matrix sensor. While this new device is under development a theoretical geometry for this future sensor has been simulated. Thousands of patterns have been obtained to use them in training, testing and optimization of different classification schemes using the system described in this paper.

This sensor model has been defined using about a hundred rays lying on different planes, but results obtained using the present simulation system have shown for the given set of patterns that this large amount of rays is not required and a reduced set of rays can carry out the desired classification. This result is valid only for the given set of patterns that have been obtained from simulation.

Based on this experience, it seems interesting to repeat this experiment in real world to confirm or not that hundred rays or a smaller set are enough to perform a reliable classification. Real word tests are also important in order to find passenger positions that cannot be recognized. At this point a problem appears since such a sensor composed by a hundred fibers, and with a cost proper for use it in the car industry has not been yet developed. Due to this, this real world test must be carried out in laboratory by using a laser camera, and selecting from deep images a set of distances corresponding to a point subset in the image.

5. ACKNOWLEDGMENTS.

Special thanks are given to authorities of Universidad Tecnológica Nacional, for their unconditional supported of new possibilities leading to new projects, to Dr. Carlos O. Nicasio, for his wise counsel that has been of great value in this work, to Prof. Dr. Ing. Ulrich Kressel, Ing. Bernd Klepser and Doctor candidate Ing. Bernd Schneider for their assistance and helpful work related to design and test different classification schemes that have been of great value in this work.

The authors want to thank Daimler Chrysler AG, that through Dr. Hahn and Dr. Metzler have been extremely generous in supporting an important part of this work and providing the best place to do it. Thanks to Fachhochschule Für Technik Esslingen and the personal of GIII group in Universidad Tecnologica Nacional Facultad Regional Córdoba for their valuable support.

5 .REFERENCES.

1. Geuss H. (1995). *“Entwicklung eines anthropometrischen Messverfahrens für das CAD-Menschmodell RAMSIS”.* FAT Schriftenreihe Nr. 123. (Geu95a).
2. Geuss H. (1995). *”Verfahren zur Generierung neuartiger konstruktionrelevanter Typologien”.* Zeitschrift für Arbeitwissenschaft. Dortmund. Verlag Dr. Otto Schmidt, Köln. (Geu95b).
3. Schwarte R. et al. (1997) *“New Optical four-quadrant phase-detector integrated into a photogate array for small and precise 3D-cameras”.* Presentation on Photonics West/Electronic Imaging, Symposium of SPIE, San Jose. (Sch97b)
4. Schwarte R. et al. (1997) *“Schnelle und Einfache optische Formerfassung mit einem neuartigen Korrelations-Photodetektor-Array”.* Vortrag auf der DGZfP-GMA-Fachtagung in Langen, “Optische Formerfassung”. (Sch97a)
5. Spies H. (1998) Multi-Distance-Measurement System for Occupant and Child-asiento Detection. Temic. Internal paper. (Spi98)