

# DESIGN, TRANSFORMATION AND ANIMATION OF HUMAN FACES

N.Magnenat-Thalmann, H.T.Minh, M.de Angelis, D.Thalmann

## Abstract

Creation of new human faces for synthetic actors is a tedious and painful task. The situation may be improved by introducing tools for the creation. Two approaches are discussed in this paper: modification and edition of an existing synthetic actor using local transformations; generation of new synthetic actors obtained by interpolation between two existing actors; creation of a synthetic actor by composition of different parts. This paper also describes the methods used in the facial animation of synthetic actors who change their personalities from one person to another. This means that our purpose is to transform one character into another, and also to transform the animation at the same time. The interpolation must be at several levels: the shape level, the parameter level, the expression level and the script level. For the animation, we introduce three levels of inbetweens: inbetween parameters, inbetween expressions and inbetween scripts. The method has been completely implemented and integrated into the Human Factory software.

**Keywords:** synthetic actor, computer animation, local transformation, inbetweening

## Introduction

The problem of constructing human characters from a geometrical point of view is mainly a problem of entering free-form shapes, as shown in Fig.1. Essentially, two general approaches have been used until now: digitizing methods and parametric-surface approaches. The first class of methods is time-consuming and suffers of a lack of creativity. The second class is convenient for creating human characters, as already shown by M.Nahas [1] except when these characters have to be like well-known personalities.

The most direct 3D digitizing technique [2] [3] is simply to enter the 3D coordinates using a 3D digitizer. Three types of such devices are now available: devices based on orthogonal magnetic fields, devices based on sound captors, devices based on laser light. Another common way of creating 3D objects is by 3D reconstruction from 2D information. Several techniques are possible: 3D reconstruction from 2D plans, 3D reconstruction from several photographs and 3D reconstruction from a set of serial cross sections, like tracing the contours from a topographic map. As described in another paper [4], the creation of new synthetic actors using a digitizing technique is a tedious and painful task.

The situation may be improved by introducing tools for the creation. Three approaches are possible and will be discussed in this paper:

1. Modification and edition of an existing synthetic actor using local transformations
2. Generation of new synthetic actors obtained by interpolation between two existing actors

3. creation of a synthetic actor by composition of different parts.

## **Local transformations of existing synthetic actors**

A local transformation (LT) is a transformation applied to a part of a figure and not the whole as a global transformation. Generally a local transformation consists in two steps:

1. selection of a region to be affected
2. selection of the transformation and its parameters

We may distinguish 5 ways of selecting a facet-based region :

1. by indicating the vertex numbers
2. by selecting the vertices inside a box; which means the selection of a region bounded by six 3D vectors, which are threshold values.
3. by percentage and angle on a circle; in this case the region is defined by two anchor points that are used as extremas and two percentage bounds. It selects a region which is like a slice of pie or a slice of a virtual cylinder.
4. by color selection, e.g. all vertices with a given hue value
5. by a set-theory operation between two regions already selected using one the previous methods.

Four methods of transformations may be found:

### **1. percentage to a vertex**

Each vertex is moved towards a reference vertex according to a specified percentage of the distance between both vertices.

### **2. guided translation**

A translation is first calculated to move a vertex A towards a vertex B according to a given percentage; then this translation is applied on all vertices included in the selected region. For instance, a percentage of 0 does not affect any vertex; a percentage of 1 makes all vertices move according to the distance between A and B in the direction A towards B.

### **3. scale according to a plane**

A scale is applied on the current figure. The amplitude of the scaling is proportional to the distance between a vertex and a specified plane.

### **4. variable translation**

The transformation is applied according to a variation degree and a possibility of an acceleration/deceleration [5] factor on the selected region. The variation degree is defined relatively to the distance between any vertex and the center of the region. A variation of 0 indicates that all vertices will be affected in the same way. A variation of 1 indicates that the center vertex will be the more affected and the peripheral vertices of the area won't be affected at all.

Results of local transformations are shown in Fig.2-6.

## Shape interpolation between human faces

Two solutions to this problem are possible:

1. Make facets and vertices in one object appear or disappear in order to obtain a good correspondence
2. Reorganize both figures by creating a new structure of facets common to both human faces.

Our approach is based on the second solution, because any inbetween human face should be animated as both original faces.

### Grid generation

The technique consists of extracting profiles of a digitized object from selected planes and generating a grid which corresponds to the original object. The designer provides an original figure (human face or body part) and a series of  $N$  planes; each plane is given by a point and the normal to the plane.

The designer also selects the number  $M$  of profiles. He/she also indicates a point  $A$  which is the approximate location of the starting point for each profile and a threshold value to suppress the connection between the initial and the final points for an open surface. The simplest algorithm could be to cut the object by a series of horizontal and vertical planes; however, this introduces ambiguities as shown in Fig.7. Basically, the method works in two steps: profile determination, followed by grid generation.

Consider  $F$  the object to be preprocessed and  $P_i$  ( $i=1$  to  $n$ ) a series of planes (where  $n$  depends on the accuracy); the larger the number of profiles, the more the grid will be accurate; however, there is a price to pay in memory. The types of element of the generated surface may be also selected; the two most popular surfaces are based on triangles and quadrilaterals. Note that the method works for surfaces with holes (e.g. eyes). The principle of the method is shown in Fig.8.

### First step: profile determination

- a) Determine the intersection points between the object  $F$  and the plane  $P_i$  as follows (see Fig.9 for notations):

A facet of  $F$  intersects the plane when the following relation is true for at least one of its edges (limited by vertices  $v_1$  and  $v_2$ ):  $((M_{v_1}-O_i) \cdot N_i) ((M_{v_2}-O_i) \cdot N_i) < 0$ . In this case the intersection point  $I$  is obtained as:

$$I = M_{v_1} + \frac{(M_{v_1}-O_i) \cdot N_i}{(M_{v_2}-O_i) \cdot N_i} (M_{v_2}-M_{v_1})$$

- b) Create object profiles by connecting the intersection points

### Second step: grid generation ( $m \times n$ )

Consider a series of profiles  $Pr_i$ ,  $i=1, \dots, n$

a) Find the length  $l_{\max}$  of the longest profile

b) Subdivide the length  $l_{\max}$  into  $(m-1)$  intervals  $l_j$ :  $l_{\max} = \sum_{j=1}^{m-1} l_j$

c) Find the grid points by applying the following principle:

Consider  $\{X_1, X_2, \dots, X_q\}$  the set of points of the  $i$ -th profile  $Pr_i$  and  $X_k$  the first point such

that  $\sum_{p=1}^{k-1} |X_{p+1} - X_p| \leq \sum_{j=1}^{m-1} l_j$

Let  $v_{i1} = X_1$  and  $V_{i,j+1} = X_{k-1} + \beta (X_k - X_{k-1})$

with  $\beta = \frac{\sum_{p=1}^{k-1} |X_{p+1} - X_p|}{|X_k - X_{k-1}|}$

For each  $j$  such that  $\sum_{p=1}^{k-1} |X_{p+1} - X_p| > l_j$ , we make  $V_{i,j+1} = X_q$ . The set of the  $V_{ij}$  then forms the  $m \times n$  grid. Fig.10 shows the meaning of the process.

Note that the algorithm works for surfaces with holes (e.g. eyes) by neglecting the  $l_j$  of the discontinuity portion.

Fig.11 shows a comparison between an original face and the generated grid.

## 2.2 Inbetween calculation

Consider now two human faces  $F_1$  and  $F_2$  obtained using the grid generation algorithm. Because the topology of two different human faces may be very different, the dimensions of the grids will probably be different. Assume two grids  $m_1 \times n_1$  and  $m_2 \times n_2$  with  $m_1 > m_2$  and  $n_1 > n_2$ . There are three steps in the algorithm:

### Step 1: search for correspondances

Establish a correspondance between the  $m_1$  and  $m_2$  profiles. (We assume  $m_1 > m_2$ . If not, we simply interchange the role of  $m_1$  and  $m_2$ ). For any profile  $i$  between 1 and  $m_1$  of  $F_1$ , the correspondance will be a profile  $f \in [1, m_2]$  of  $F_2$  such that:

$$f = \text{trunc} \left[ \frac{(m_2-1)i + m_1 - m_2}{m_1-1} \right].$$

The correspondance between the  $n_1$  and  $n_2$  parallel sections is found using a similar method. The  $j$ -th curve ( $j \in [1, n_1]$ ) will correspond to the  $k$ -th curve where:

$$k = \text{trunc} \left[ \frac{(m_2-1)i + m_1 - m_2}{m_1-1} \right].$$

Now, the correspondance between points is straightforward: a point  $v_{ij}$  belonging to the  $(m_1 \times n_1)$  grid of  $F_1$  corresponds to a point  $w_{fk}$ , with  $f$  and  $k$  determined as above.

### Step 2: Generation of the inbetween object.

Assuming  $m_1 \times n_1$  and  $m_2 \times n_2$ , an inbetween object  $F_I$  will have an  $(m_1 \times n_1)$  grid of points  $u_{ij}$ :

$$u_{ij} = v_{ij} + (W_{fk} - V_{ij}) \text{ with } 0 \leq t \leq 1$$

As our purpose is to animate inbetween human faces, it is essential to have a correspondance between similar physical regions. For this reason, we separate both human faces into 7 regions by cutting the 3D face using horizontal planes and applying the algorithm for each pair of regions. Regions must be selected so that their borders occur at the level of the mouth, and the eyes as shown in Fig.12.

Although, the algorithm processes surfaces with holes, it is easier for the animation process to consider a closed mouth and eyes. Therefore, it is easier if the extremities of the mouth and the eyes are on the borders of the regions.

Fig.13-14 show examples of interpolation between Marilyn Monroe and Humphrey Bogart. Fig.15-17 present a transformation from Marlyn to an extra-terrestrial creature.

### 3.2 A three-level approach to facial animation

Several models of computer-generated facial expressions have been described: parametrized models [6,7] and muscle-based models [8,9]. Our model [10] is based on the abstraction of muscle actions (AMA) instead of the muscles themselves. To animate an actor's face, the animator must first build a certain number of facial expressions specific to her/his character. The expressions are built by specifying face deformations caused by muscles: jaw opening, eye opening, face folds etc. These facial deformations are generally different from one person to another. For example, any person may be required to open their mouth to the maximum, but this maximum aperture is not the same for everybody. These expressions correspond to phonemes (speech expressions) and emotions (cry, laugh etc.). Once the animator has built the expressions, he/she animates the face by indicating to the computer some of these expressions at certain selected times. For example, "KID" will be pronounced by a character, indicating that the phoneme "K" is used at a given time, the phoneme "I" a short time later, and then the phoneme "D". Then the software will progressively transform the facial expression corresponding to the phonème "K" in order to obtain the facial expression corresponding to the phoneme "I", then to the phoneme "D".

From the considerations above, we may distinguish three levels of interaction for the animator:

The first and the lowest level is the level of control of facial parameters; the animator may decide how a basic deformation occurs on a specific synthetic actor.

At the second level, the animator creates specific expressions based on the facial parameters. For example, he/she creates phonemes, a smile, a loving look.

At the third level, the animator decides the timing of the animation by fixing some expressions at various times.

We now present an example to illustrate the three levels. At the first level, the animator may decide how the synthetic actress Marilyn Monroe may open her mouth; this means defining

the maximum horizontal and vertical openings, which are facial parameters. At the second level, the animator may decide the contribution of these facial parameters to an expression such as a smile. At the third level, the animator makes Marilyn smile at a certain time.

In other words, an animator may work only at the script level; however, in this case, he/she may only control the timing and the duration of expressions or combined expressions. He/she cannot create completely new expressions, except when they are combinations of others.

The parameter level is not necessary when basic facial parameters for a synthetic actor already exist, as for example, for Marilyn Monroe and Humphrey Bogart, created for the film *Rendez-vous à Montréal*. [4]. In this case, the animator may create any new scene involving the same actors. With the expression level, new expressions may be created using the facial parameters.

However, for a new synthetic actor, the parameter level is absolutely necessary, because all facial parameters must be defined for the actor. They are equivalent to the basic physical characteristics of the muscles and the bony structure of the actor's face.

Fig. 18-20 show examples of facial expressions.

## Animation techniques

### 3.6 Interpolating facial parameters and expressions between two different human faces

As our purpose is to transform one character into another one and also transform the animation at the same time, the interpolation must be at several levels:

the shape level, the parameter level, the expression level and the script level.

#### Shape level

There is a need for a shape interpolation, which has been already discussed.

#### Parameter level

For facial parameters, consider for example, the maximum vertical angle allowed for the eye motion. Assume that the angle is  $V_{MAX}=29^\circ$  for the first character and  $V_{MAX}=34^\circ$  for the second character. Now the command `MOVE_RIGHT_EYE_VERTICAL 0.4`, means the right eye is moved by an angle which is 0.4 times  $V_{MAX}$ ; for the first actor, this is  $11.6^\circ$ , while for the second, it is  $13.6^\circ$ . For an inbetween actor at  $l=0.5$ , it should be  $0.4*(29+0.5*(34-29))$ . Finally, this means that the basic parameters of the AMA procedures must be calculated by linear interpolation. More formally, for an inbetween actor  $A_I$  at the percentage  $l$  between two actors  $A_1$  and  $A_2$ , the basic parameter  $p_I$  is computed as:

$$p_I = p_{1l} + (p_{2l}-p_{1l})$$

#### Expression level

If the actor  $A_1$  is smiling (expression  $E_1$ ) and the actor  $A_2$  is crying (expression  $E_2$ ), any inbetween actor  $A_I$  at the percentage  $l$  must have an inbetween expression  $E_I$ . The expression  $E_I$  is obtained as the collection  $\{FP_{kI}\}$  where each  $FP_{kI}$  is calculated as the result of the corresponding AMA procedure using basic parameters  $p_{jI}$  obtained themselves as above by a linear interpolation:

$$p_{jI} = p_{j1l} + (p_{j2l}-p_{j1l})$$

#### Script level

A unique script may be provided for an actor  $A_1$  transformed into another actor  $A_2$ . In this case the action is performed on all inbetween actors. A more general and interesting case is

when the actor  $A_1$  plays a role according to a script  $S_1$  and the actor  $A_2$  plays another role according to the script  $S_2$ . An inbetween actor  $A_I$  should play a role according to an inbetween script  $S_I$ . This means that the expressions of the inbetween actors are calculated as inbetween expressions according to the current value of the percentage between actors. Fig.21 shows a complete example.

## Acknowledgements

The authors would like Ross Racine and Denis Rambaud for their collaboration. The research was supported by the Natural Sciences and Engineering Council of Canada, and the FCAR foundation (Government of Quebec).

## References

- [1] Nahas M, Huitric H and Saintourens M (1987) **Animation of a B-spline Figure**, The Visual Computer, Vol.3, No4
- [2] Smith AR (1983) Digital filmmaking. Abacus 1(1):28-45, Springer-Verlag
- [3] Blum R (1979) **Representing three-dimensional objects in your computer**. BYTE, May Issue, pp14-29
- [4] Magnenat-Thalmann N, Thalmann D (1987) **The Direction of Synthetic Actors in the film Rendez-vous à Montréal**, IEEE Computer Graphics and Applications, Vol. 7, No 12.
- [5] Magnenat-Thalmann N, Thalmann D (1985) **Computer Animation: Theory and Practice**, Springer, Tokyo New York Berlin Heidelberg, p.49.
- [6] Parke F.I. (1982) **Parameterized Models for Facial Animation**, IEEE Computer Graphics and Applications, Vol.2, No9, pp.61-68.
- [7] Pearce A, Wyvill B, Wyvill G and Hill D (1986) **Speech and expression: a Computer Solution to Face Animation**, Proc. Graphics Interface '86, pp.136-140.
- [8] Platt S, Badler N (1981) **Animating Facial Expressions**, Proc. SIGGRAPH '81, pp.245-252.
- [9] Waters K (1987) **A Muscle Model for Animating Three-Dimensional Facial Expression**, Proc. SIGGRAPH '87, Vol.21, No4, pp.17-24.
- [10] Magnenat-Thalmann N, Thalmann D (1987) **Abstract Muscle Action Procedure for Human face Animation**, The Visual Computer, Vol.3, No4, pp.290-297.

## Captions

- Fig.1 Human characters are essentially based on free-form surfaces
- Fig.2 Original synthetic actress
- Fig.3-6 Local transformations to the actress face
- Fig.7 Ambiguities in selecting intersection points
- Fig.8 Principle of grid generation
- Fig.9 Determination of intersection point
- Fig.10 Grid generation process
- Fig.11 Comparison between an original face and a grid obtained using the algorithm  
Left half face is original and right half face is the generated grid
- Fig.12 Selection of regions
- Fig.13 Four views Marilyn Monroe - Humphrey Bogart: 0%, 20%, 40%, 50%
- Fig.14 Four views Marilyn Monroe - Humphrey Bogart: 60%, 80%, 90%, 100%
- Fig.15-17 Transformation from Marilyn to an extra-terrestrial creature (From the film GALAXY Sweetheart, directed by N.Magnenat-Thalmann and D.Thalmann)
- Fig.18-20 Examples of facial expressions
- Fig.21 Animation of inbetween synthetic actors  
Four views Marilyn Monroe - Humphrey Bogart: 0%, 33%, 66% 100%  
Expression interpolation: from smile to opened mouth and eyebrow motion

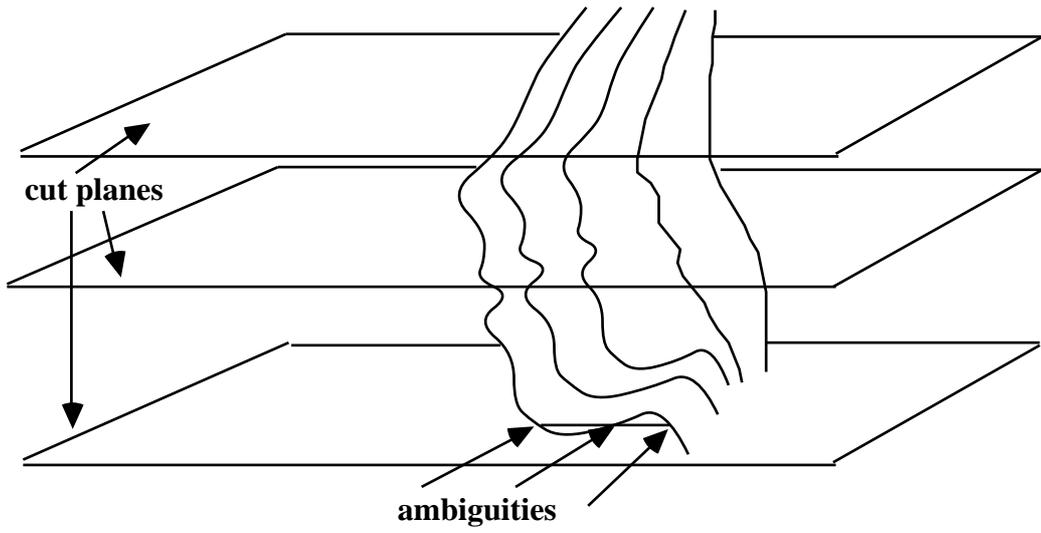


Fig.7

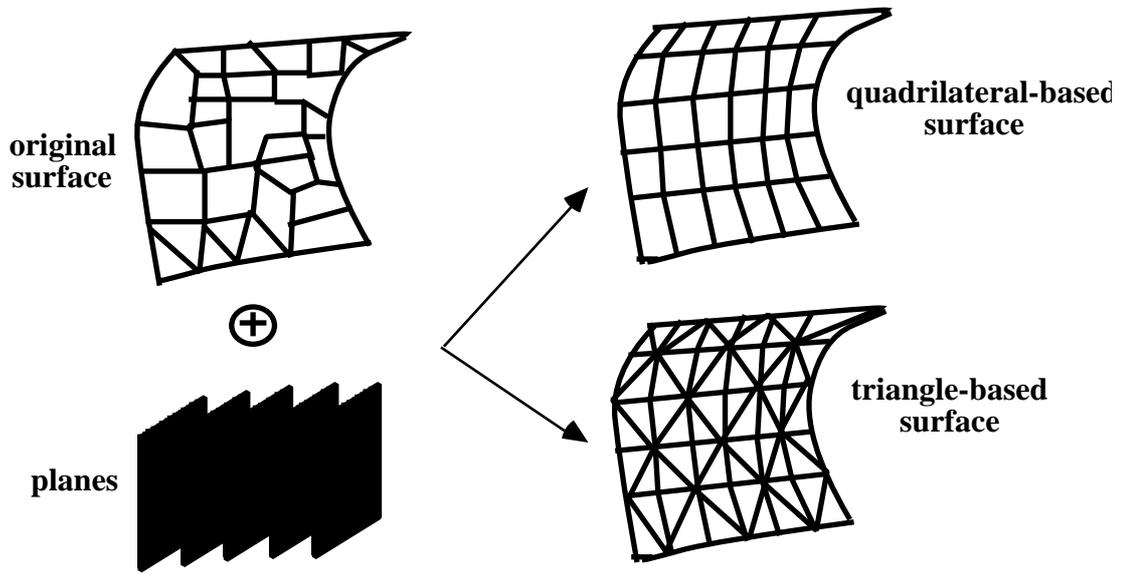


Fig.8

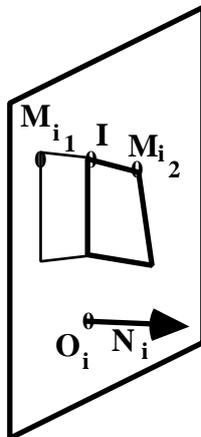


Fig.9

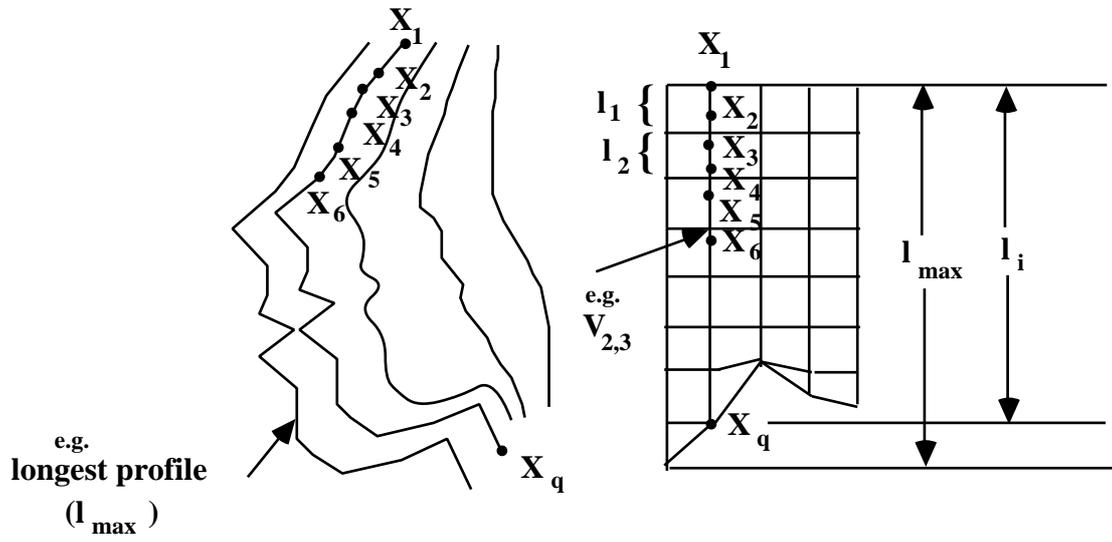


Fig.10

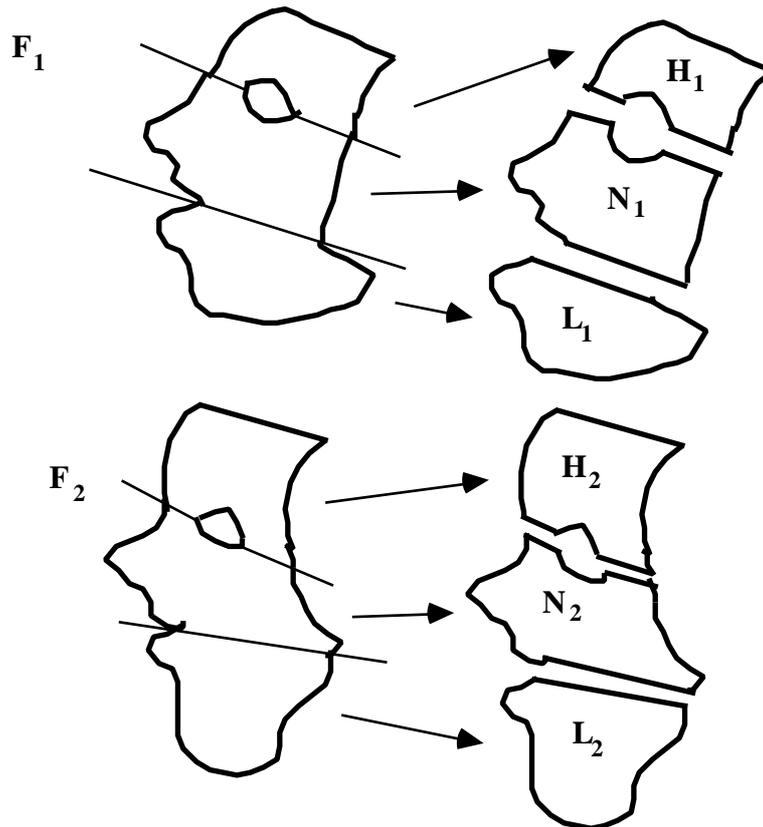


Fig.12