A GENERAL ALGORITHM FOR 3-D SHAPE INTERPOLATION IN A FACET-BASED REPRESENTATION

Tong Minh Hong N.Magnenat-Thalmann D.Thalmann

MIRALab, HEC / IRO
Université de Montréal
Canada

ABSTRACT

This paper is concerned with interpolation between two objects defined using a faceted representation. The interpolation function is not examined, but the paper emphasizes the problem of correspondance between the two given key drawings. The problem is much more complex than interpolation between line drawings, because the two key drawings generally have a different total number of vertices, a different total number of facets and corresponding facets have a different number of vertices. A general algorithm is proposed based on a criterion of minimal dynamic displacement. There are numerous applications of the method: simulation of biological evolution, animation, portrait-robots.

RESUME

Cet article traite du problème de l'interpolation entre deux objets définis avec une représentation en facettes. La fonction d'interpolation n'est pas examinée; par contre, la correspondance entre les deux objets est étudié en détail. C'est un problème beaucoup plus complexe que l'interpolation entre des dessins en lignes, car les deux objets ont généralement un nombre différent de sommets, de facettes et les facettes correspondantes n'ont pas le même nombre de sommets. Un algorithme général est proposé; il est basé sur un critère de déplacement minimal dynamique. Les applications de la méthode sont très nombreuses: simulation d'évolution biologique, animation, portraits-robots.

KEYWORDS: Computer animation, Facets, Keyframe, Interpolation, Centroids

1. INTRODUCTION

Image-keyframe animation consists of the automatic generation of intermediate frames, called **inbetweens**, based on a set of key-frames supplied by the animator. The inbetweens are obtained by interpolating the keyframe images themselves. This technique is called **image-based keyframe animation** by Steketee and Badler (1985) and **shape interpolation** by Zeltzer (1985). The technique may be defined as follows: two key-frames are given in advance in 3D; the method consists of producing a series of inbetween images in such a way that the degree of the transformation is controlled by a real parameter varying from 0 to 1, as shown in Fig.1.

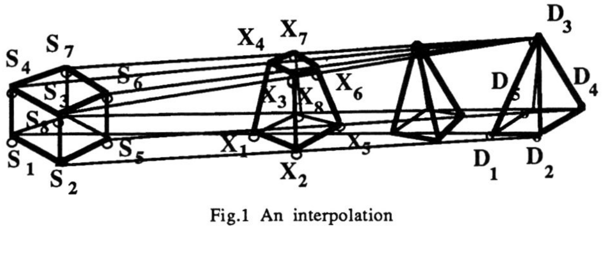

Fig.1 An interpolation

In two-dimensions, the technique was introduced in 1971 by Burtnyk and Wein. Their algorithm may be easily extended to three-dimensional linedrawings. However, the linear interpolation algorithm produces undesirable effects, such as lack of smoothness in motion, discontinuities in the speed of motion and distortions in rotations. Alternate methods have been proposed by Burtnyk and Wein (1976), Reeves (1981), Kochanek and Bartels (1984). However, according to Stekettee and Badler (1985), there is no totally satisfactory solution to the deviations between the interpolated image and the object being modeled.

We are concerned in this paper with interpolation between two objects defined using a faceted representation. We do not discuss the interpolation function, but emphasize the problem of correspondance between the two given key drawings. The problem is much more complex than interpolation between line drawings, because the two key drawings generally have a different total number of vertices, a different total number of facets and corresponding facets have a different number of vertices.

2. NOTATION

We call figure $\mathcal{F}$, a finite set of right-oriented polygonal facets:
$\mathcal{F} = \{ <p_i, \mathbf{N_i}>; i=1, m\}$

Each facet p_i is defined as $p_i = \{\mathbf{S_{ij}}; j = 1, n_i\}$ where $\mathbf{S_{ij}}$ are the vertices; $\mathbf{N_i}$ is the normal to the facet p_i.

Consider two figures $\mathcal{F}_S$ and $\mathcal{F}_D$ for which we wish to generate an inbetween figure $\mathcal{F}_I$ with a degree of transformation controlled by the real parameter $\lambda \in [0,1]$. The notation used in the paper is summarized in Table 1.

$\mathcal{F}_S$	source key figure (S: source)
$\mathcal{F}_D$	destination key drawing (D: destination)
$\mathcal{F}_I$	inbetween drawing (I: inbetween)
N_S	number of facets of $\mathcal{F}_S$
N_D	number of facets of $\mathcal{F}_D$
N_{Si}	number of vertices of the i-th facet of $\mathcal{F}_S$
N_{Dj}	number of vertices of the j-th facet of $\mathcal{F}_D$
$\mathbf{S_{ik}}$	k-th vertex of the i-th facet of $\mathcal{F}_S$
$\mathbf{D_{jk}}$	k-th vertex of the j-th facet of $\mathcal{F}_D$
ε_S	set of vertices of $\mathcal{F}_S$
ε_D	set of vertices of $\mathcal{F}_D$
P_{Si}	i-th facet of $\mathcal{F}_S$
P_{Dj}	j-th facet of $\mathcal{F}_D$
P_{Ik}	k-th facet of $\mathcal{F}_I$
C_{Si}	centroid of the i-th facet of $\mathcal{F}_S$
C_{Dj}	centroid of the j-th facet of $\mathcal{F}_D$

Table 1: Symbols and notation

For example, in Fig.1, we have:

$\varepsilon_S = \{\mathbf{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8}\}$

$\varepsilon_D = \{\mathbf{D_1, D_2, D_3, D_4, D_5}\}$

The problem first consists of establishing a logical correspondance $P_{Dj} = f(P_{Si})$ between the facets of $\mathcal{F}_S$ and $\mathcal{F}_D$; second a correspondance must be found between vertices $D_{jk} = \varphi(S_{ih})$ deriving from facets P_{Si}) and P_{Dj}, and finally the inbetween figure $\mathcal{F}_I$ has to be constructed: the vertices of the i-th facet of $\mathcal{F}_I$ are computed as:

$$\mathbf{B_{ih}} = \mathbf{S_{ih}} + \lambda\,(\varphi(\mathbf{S_{ih}}) - \mathbf{S_{ih}})$$

where $\mathbf{S_{ih}} \in p_{Si}$, $D_{jk} = \varphi(\mathbf{S_{ih}}) \in p_{Dj}$

$p_{Dj} = f\,(p_{Si})$ and $\lambda \in [0,1]$

As both figures do not generally have the same number of facets and/or vertices, a "logical correspondance" must be found. By denoting these facets (or vertices) by $\{X_i;\ i=1,m\}$ for $\mathcal{F}_S$ and $\{Y_j;\ i=1,n\}$ for $\mathcal{F}_D$ and assuming $m \geq n$, we define as a "logical correspondence" a technique which

- selects the n most representative elements $X_1', X_2', \ldots X_n'$ of the m elements of $\{X_i\}$ to be transformed into Y_j; $j=1,n$, when $\lambda \rightarrow 1$. The n elements $\{X_i'\}$ which are transformed into n elements $\{Y_j\}$ are called the **main elements**
- makes the (m-n) remaining elements of $\{X_i\}$ gradually disappear as $\lambda \rightarrow 1$. These (m-n) remaining elements are called **extra-elements.**

Note that the spatial position of a facet is described by its centroid; it means that $\mathbf{C_{Si}} = \frac{1}{N_{Si}} \sum_{k=1}^{N_{Si}} \mathbf{S_{ik}}$ will represent the i-th facet of $\mathcal{F}_S$ and $\mathbf{C_{Dj}} = \frac{1}{N_{Dj}} \sum_{k=1}^{N_{Dj}} \mathbf{D_{jk}}$ will represent the i-th facet of $\mathcal{F}_D$

This is a reasonable approach, because the way of numbering vertices and facets may vary. Sometimes the surface of the objects will experience distorsions. However, this is tolerable in any application when connectivity between facets is still satisfied for the inbetween figures.

3. THE ALGORITHM

The algorithm works as follows:

Step 1 Check whether $N_S \geq N_D$. Otherwise reverse the roles of $\mathcal{F}_S$ and $\mathcal{F}_D$.

Step 2 Perform a correspondance between facets

2.1) **Normalize** the two sets of centroids $\{\mathbf{C_{Si}}\}$ and $\{\mathbf{C_{Dj}}\}$ as described in Section 4.1. Normalizing in this paper means *translation + scale* before dealing with the correspondence of points.

2.2) Find the N_D most representative points of the N_S elements of $\{\mathbf{C_{Si}}\}$ and make them correspond with the points of $\{\mathbf{C_{Dj}}\}$ using the technique of **dynamic minimization** of distance described in Section 4.2. This means finding a subset $\{C'_{S1},\ldots\} \subset \{C_{Si}\}$ and a bijective function f_1 such that:

$$f_1:\ \{C'_{S1}\} \rightarrow \{C_{Dj}\} \quad i,j=1,n$$
$$C'_{S1} \rightarrow f_1(C'_{Si})$$

2.3) Find a mapping between the N_S-N_D remaining elements of $\{C_{Si}\}$ and those of $\{C_{Dj}\}$ using the technique of **image by neighborhood** (see Section 4.3). This means finding a mapping f_2 such that:

$$f_2:\ (\{C_{Si}\}\{C'_{Si}\} \rightarrow \{C_{Dj}\}$$
$$C_{Sk} \rightarrow f_2(C_{Sk})$$

2.4) Establish the facet correspondance as follows:

$$f:\ (\{C_{Si}\} \rightarrow \{C_{Dj}\}$$
$$C_{Si} \rightarrow f(C_{Si}) = \begin{cases} f_1(C_{Si}) \text{ if } C_{Si} \in \{C'_{Si}\} \\ f_2(C_{Sk}) \text{ if } C_{Si} \in \{C'_{Si}\}-\{C'_{Si}\} \end{cases}$$

Step 3: Make a correspondance between vertices

First case: $\mathbf{C_{Si} \rightarrow f(C_{Si})}$ **where $\mathbf{C_{Si}}$ is a main facet.**

3.1) Arrange all vertices of the facet j of $\mathcal{F}_D$ to **lie in the plane** of facet i of $\mathcal{F}_S$ (see Section 4.4)

3.2) **Normalize** the set of vertices $\{\mathbf{S_{ik}}\}$, $\{\mathbf{D_{jk}}\}$ (see Section 3.1), $\mathbf{S_{ik}} \in P_{Si}$ and $\mathbf{D_{jk}} \in P_{Dj}$

3.3) Find the N_{Dj} most representative elements of the N_{Si} elements of $\{S_{ik}\}$ and make them correspond with those of $\{D_{jk}\}$ if $N_{Si} \geq N_{Dj}$. This means:

find a subset $\{\mathbf{S'_{ik}}\} \subset \{\mathbf{S_{ik}}\}$ and a bijective function φ_1 using the method of dynamic minimization (see Section 4.2) such that φ_1: $(\{\mathbf{S'_{ik}}\} \rightarrow \{\mathbf{D_{jk}}\}$.

If $N_{Si} < N_{Dj}$, the opposite is true:

$$\varphi_1:\ (\{\mathbf{D'_{jk}}\} \subset \{\mathbf{D_{jk}}\}) \rightarrow \{\mathbf{S_{ik}}\}$$

3.4) Find a mapping between the (N_{Si} - N_{Dj}) remaining elements and those of $\{\mathbf{D_{jk}}\}$ using the **CODEX approach** (see Section 4.5). If $N_{Si} \geq N_{Dj}$: φ_2: $(\{\mathbf{S_{ik}}\}- \{\mathbf{S'_{ik}}\}) \rightarrow \{\mathbf{D_{jk}}\}$ otherwise φ_2: $(\{\mathbf{D_{jk}}\}-\{\mathbf{D'_{jk}}\}) \rightarrow \{\mathbf{S_{ik}}\}$

3.5) Obtain a correspondance between vertices:

Case A: $N_{Si} \geq N_{Dj}$:

$$\varphi_{ij}:\ (\{\mathbf{S_{ik}}\} \rightarrow \{\mathbf{D_{jk}}\}$$
$$\mathbf{S_{ik}} \rightarrow \varphi_{ij}(\mathbf{S_{ik}}) = \begin{cases} \varphi_1{:}(\mathbf{S_{ik}}) \text{ if } \mathbf{S_{ik}} \in \{\mathbf{S'_{ik}}\} \\ \varphi_2{:}(\mathbf{S_{ik}}) \text{ if } \mathbf{S_{ik}} \in \{\mathbf{S_{ik}}\}-\{\mathbf{S'_{ik}}\} \end{cases}$$

Case B: $N_{Si} < N_{Dj}$:

$$\varphi_{ij}:\ (\{\mathbf{D_{jk}}\} \rightarrow \{\mathbf{S_{ik}}\}$$
$$\mathbf{D_{jk}} \rightarrow \varphi_{ij}(\mathbf{D_{jk}}) = \begin{cases} \varphi_1{:}(\mathbf{D_{jk}}) \text{ if } \mathbf{D_{jk}} \in \{\mathbf{D'_{jk}}\} \\ \varphi_2{:}(\mathbf{D_{jk}}) \text{ if } \mathbf{D_{jk}} \in \{\mathbf{D_{jk}}\}-\{\mathbf{D'_{jk}}\} \end{cases}$$

The case treated at this step must be stored for use in step 4.

Second case: $f(C_{Si}) = C_{Dj}$ where C_{Si} is an extra-facet:

3.1b) Consider $\{S_{ik}\}$ the set of vertices of the i-th main facet of $\mathcal{F}_S$, corresponding to $\{D_{jk}\}$ the set of vertices of the j-th facet of $\mathcal{F}_D$.

If $N_{Si} \geq N_{Dj}$ {more vertices in the i-th main facet of $\mathcal{F}_S$ than in the j-th facet of $\mathcal{F}_D$}
then
find (see Section 4.3) **the neighborhoods** $\{S_{ik}\}$for $\{S_{hk}\}$and then find $\varphi_{hj}(S_{hk})=\varphi_{ij}(S_{ik})$.

If $N_{Si} < N_{Dj}$ {less vertices in the i-th main facet of $\mathcal{F}_S$ than in the j-th facet of $\mathcal{F}_D$}
then {we cannot find $\varphi_{hj}(S_{hk})$}
Solve the problem by temporarily replacing D_{jk} by $D_{jk} + \mu (\varphi_{ji}(D_{jk}) - D_{jk})$
$\mu \in [0,1[$
{note that this forces the j-th facet of $\mathcal{F}_D$ to transform into the k-th main facet of $\mathcal{F}_S$}
and find $\varphi_{hj}(S_{hk}) = D_{jq}$ if $\|S_{hk}-D_{jq}\| = \min_{p=1,N_{Dj}} \{\|S_{hk}-D_{jp}\|\}$

Step 4 Build inbetween facets p_{Ii} of $\mathcal{F}_I$

The vertices of $\mathcal{F}_I$ are computed as follows:

$B_{ik} = S_{ik} + \lambda (\varphi_{ij}(S_{ik})-S_{ik})$ if the interpolation is carried out between the facet i of $\mathcal{F}_S$ and the facet j of $\mathcal{F}_D$ (case A: $N_{Si} \geq N_{Dj}$)

$B_{ik} = D_{ik} + \lambda (\varphi_{ij}(D_{jk})-D_{jk})$ otherwise (case B: $N_{Si} < N_{Dj}$).

4. ALGORITHM REFINEMENT

4.1 Normalization

Consider $\{X_i;\ i=1,...,m\}$ and $\{Y_j;\ i=1,...,n\}$ two sets of points; they are normalized if $\max\{\|X_i-G\|\} = \max\{\|Y_j-G\|\} = 1$ and $G = \frac{1}{m} \Sigma X_i = \frac{1}{n} \Sigma Y_j$

Using this process, it is easy to search the corresponding between facets and/or vertices between two figures which have different positions and dimensions. Fig.2 shows an example.

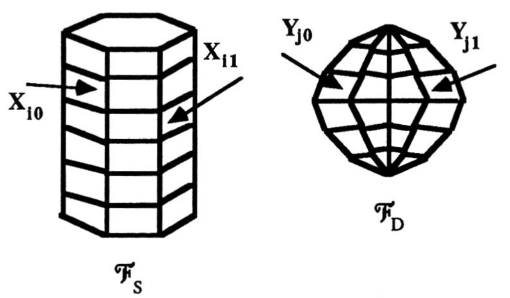

Fig.2 Correspondence between facets

After normalization of $\{X_i\}$ and $\{Y_j\}$, there is a better chance of having a correspondance between X_{i1} and Y_{j1}.

4.2 Minimal dynamic distance

Let $\{X_i;\ i=1,...,m\}$ and $\{Y_j;\ j=1,...,n\}$ with $m \geq n$. The technique is useful for selecting the elements $X'_1, X'_2, ..., X'_n$ that are the most representative among m elements of $\{X_i\}$ and for making them correspond with the elements in $\{Y_j\}$.

The subset $\{X'_i\} \subset \{X_i\}$ and the bijection g: $\{X'_i\} \rightarrow \{Y_j\}$ defined for minimal dynamic distance are such that $\Sigma \|X_i - g(X_i)\|^{\mu}$, $X_i \in \{X'_i\}$ is minimal (μ=1 or 2)

In other words, by considering $I=\{1,...,m\}$ and $J=\{1,...,n\}$, the problem to be solved consists of finding a subset $K \subset I$ and a bijection b: $K \subset I \rightarrow J$ in such a way as $\Sigma \|X_i - Y_{b(i)}\|^{\mu}$ is minimal.

To do this, we have to consider an arbitrary subset K and an application b: $\{1,...,m\} \rightarrow \{0,...,n\}$ satisfying the three following conditions:

(i) card(K) = n
(ii) if $i \in \{1,...,m\}$-K then b(i)=0
(iii) b: $K \rightarrow \{1,...,n\}$ should be a bijection

and to apply the following algorithm:

repeat
  SWAP:=FALSE
  for i varying from 1 to m
    for j varying from 1 to n
      Let $k = b^{-1}(j)$
      if b(i) = 0 {X_i has no image}
      then
        $D:=\|X_k - Y_j\|^{\mu} - \|X_i - Y_j\|^{\mu}$
      else {X_i has an image}
        $D:=\|X_i-Y_{b(i)}\|^{\mu}+\|X_k-Y_j\|^{\mu}-\|X_i-Y_j\|^{\mu}-\|X_k-Y_{b(i)}\|^{\mu}$
        if D = 0 and b(i)≠0
        then
          $D_1:=\min\{\|X_i-Y_{b(i)}\|^{\mu},\|X_k-Y_j\|^{\mu}\}$
            $+\|X_i-X_k+Y_j-Y_{b(i)}\|^{\mu}$
          $D_2:=\min\{\|X_i-Y_j\|^{\mu},\|X_k-Y_{b(i)}\|^{\mu}\}$
            $+\|X_i-X_k+Y_{b(i)}-Y_j\|^{\mu}$
          $D := D_1-D_2$
      if D > 0
      then
        SWAP:=TRUE
        if b(i) = 0
        then
          b(i) ← j
          b(k) ← 0
        else
          b(k) ← b(i)
          b(i) ← j
until not SWAP

The process is repeated until we obtain a minimal summation $\Sigma \|X_i - Y_{b(i)}\|^{\mu}$. The sets $\{X_i\}$ and $\{Y_j\}$ are finite sets. Therefore $\Sigma \|X_i - Y_{b(i)}\|$ is finite and is lower bounded. This implies that the termination criterion will be satisfied when we reach the lowest summation $\Sigma \|X_i - Y_{b(i)}\|^{\mu}$; in this case, any other swapping between 2 correspondences cannot occur anymore, which is indicated by SWAP:=FALSE.

To emphasize the advantages of this technique, consider the example in Fig.3.

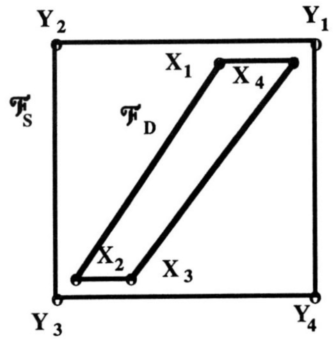

Fig.3 An example

If we search the correspondance by using a minimal distance criterion ($Y_j = f(X_i)$ if $\|X_i - Y_j\| = \min\{\|X_i - Y_k\|\}$) and not on minimal summation $\sum \|X_i - Y_j\|$ ($\mu=1$), the results are strongly dependent on the way the vertices are numbered in the two key figures. X1 which is the first numbered vertex implies that f(X1)=Y1 because $\|X_1 - Y_1\| = \min\{\|X_1 - Y_k\|\}$). This is incorrect as a logical correspondance should give:

$$X_1 \rightarrow Y_2 \quad X_2 \rightarrow Y_3 \quad X_3 \rightarrow Y_4 \quad \text{and} \quad X_4 \rightarrow Y_1$$

These are exactly the results obtained with the method of minimal dynamic distance, because $\|X_1 - Y_2\| + \|X_2 - Y_3\| + \|X_3 - Y_4\| + \|X_4 - Y_1\|$ is the minimal summation of $\sum \|X_i - Y_j\|$.

Consider now Fig.4.

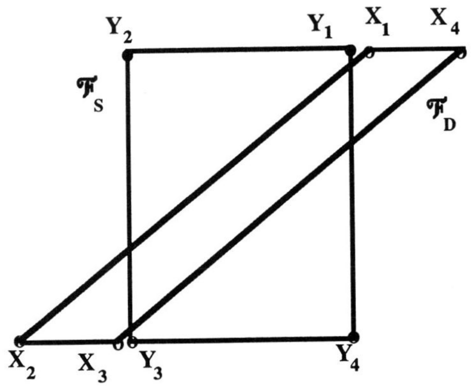

Fig.4 Another example

In this example, we have $\|X_1 - Y_1\| + \|X_4 - Y_2\| = \|X_1 - Y_2\| + \|X_4 - Y_1\|$ which provides two different choices:

(1) $X_1 \rightarrow Y_1$, $X_4 \rightarrow Y_2$, **etc.**

(2) $X_1 \rightarrow Y_2$, $X_4 \rightarrow Y_1$, **etc.**

Consider (1), if $X_1 \rightarrow Y_1$ then X_4 will be translated from X_1Y_1 and the new position will be $X_4 + X_1Y_1$. To ensure that X_1 and X_4 converge to Y_1 and Y_4, the following displacement is required:

$$\|X_1 - Y_1\| + \|X_4 + X_1Y_1Y_2\| = 1 + 9 = 10.$$

If we first consider $X_4 \rightarrow Y_2$ and look at the new position of X_1, the displacement is:

$$\|X_4 - Y_2\| + \|X_1 + X_4Y_2 - Y_1\| = 10 + 9 = 19.$$

In order to have X_1 and X_4 converging to Y_1 and Y_4, the minimal dynamic distances are as:

$$D_1 = \min\{\|X_1 - Y_1\|, \|X_4 - Y_2\|\} + \|X_4 + X_1Y_1 - Y_2\| = \min\{1, 10\} + 9 = 10$$

Similarly, considering (2), we obtain:

$$D_2 = \min\{\|X_1 - Y_2\|, \|X_4 - Y_1\|\} + \|X_4 + X_1Y_2 - Y_1\| = \min\{8, 3\} + 5 = 8$$

We have $D = D_1 - D_2 > 0 \Rightarrow$ **the correspondance obtained by (2) is the best.**

Comments:

- the best correspondance is the correspondance which guarantees $D = \min\{D_1, D_2\}$
- For $\mu=2$ in the expression $\sum \|X_i - Y_j\|^{\mu}$, we obtain a dynamic minimization corresponding to the least-squares method. The method is less satisfactory than the method with $\mu=1$, because $\sum \|X_i - Y_j\|$ can be considered as the energy required to move all points $\{X_i\}$ from $\mathcal{F}_S$ to form the new figure $\mathcal{F}_D$ with $\{Y_j\}$ points. However, it is much less expensive in terms of CPU, because there is no square root operation
- CPU time of the algorithm is approximately equals to $m \cdot n^2/10$;

4.3 Technique of images by neighborhood

Consider a mapping b: $K \subset \{1,\ldots,m\} \rightarrow \{1,\ldots,n\}$, obtained as defined in the dynamic summation of the distances (see Section 4.2), our goal is to find the images for the (m-n) remaining elements of $\{1,\ldots,m\}-K$.

Consider S: $(\{1,\ldots,m\}-K) \rightarrow \{1,\ldots,n\}$ defined as follows:

$$\forall\ s \in (\{1,\ldots,m\}-K);\ S(s) = b(p)$$
$$\text{if } \|X_s - X_p\| = \min\{\|X_s - X_k\|\} \quad k \neq l,\ k \in K$$

p is called a **neighborhood** of i.

In a more formal way, if g: $\{X'_i\} \subset \{X_i\}$ is a bijection with $\#\{X_i\}=m$, $\#\{X'_i\}=m$, $\#\{Y_i\}=n$, then the application h: $(\{X_i\}-\{X'_i\}) \rightarrow \{Y_j\}$ will be in such a way as:

$$\text{for } X_s \in (\{X_i\}-\{X'_i\}),\ h(X_s) = g(X_p)$$
$$\text{if } \|X_s - X_p\| = \min\{\|X_s - X_k\|\} \quad k \neq l,\ k \in K$$

4.4 Facet rotation

Consider the example in Fig.5; it shows that facets can be reversed.

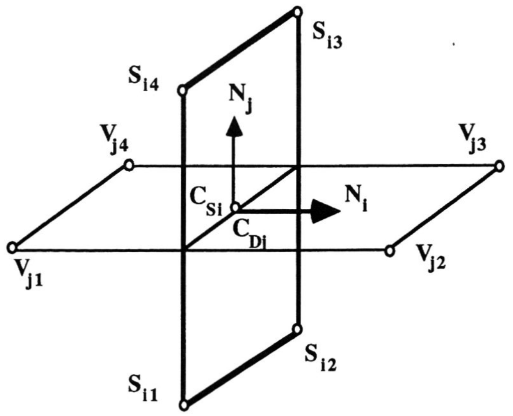

Fig.5 Facet rotation

We have

$$\|S_{i1} - V_{j2}\| = \|S_{i1} - V_{j1}\|$$
$$\|S_{i2} - V_{j3}\| = \|S_{i2} - V_{j4}\|$$
$$\|S_{i3} - V_{j4}\| = \|S_{i3} - V_{j3}\|$$
$$\|S_{i4} - V_{j1}\| = \|S_{i4} - V_{j2}\|$$

which may generate various choices.

If we choose

$$S_{i1} \rightarrow V_{j1}$$
$$S_{i2} \rightarrow V_{j4}$$
$$S_{i3} \rightarrow V_{j3}$$
$$S_{i4} \rightarrow V_{j2}$$

we obtain a reversed facet when $\lambda \rightarrow 1$.

It is then necessary to move all vertices of the j-th facet of $\mathcal{F}_D$ into the plane of the i-th facet of $\mathcal{F}_S$. The transformation T providing a satisfactory orientation of both polygons is:

$$T: \{V_{jh}\} \rightarrow \text{(plane containing the i-th facet of } \mathcal{F}_S)$$
$$V_{jh} \rightarrow T(V_{jh}) = Rot(\alpha, N_k, C_{Dj})\,[V_{jh}]$$

where Rot (α, N_k, C_{Dj}) is a rotation of an angle a about N_k relatively to the point C_{Dj} and $N_k = \frac{N_j \times N_i}{\|N_j \times N_i\|}$

This transformation is particularly important for the case of Fig.6 (see also Fig.12):

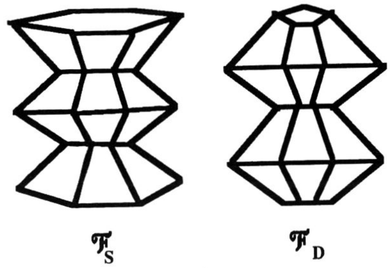

Fig.6 Convex and non-convex objects

4.5 The CODEX technique

CODEX (**C**omparative **D**istance at the **Ex**tremities) is another approach for processing the (m-n) remaining elements after the establishment of a bijection g: $\{X'_i\} \subset \{X_i\} \rightarrow \{Y_j\}$ according to the technique of dynamic summation of the distances. The method involves finding a mapping h: $\{X_i\}-\{X'_i\} \rightarrow \{Y_j\}$ defined in such a way that:

$\forall X_h \in \{X_i\} - \{X'_i\}$ (or $\forall$ h $\in$ I-K)

i) when $s > i_{min} = \min\{i \in K\} < s < i_{max} = \max\{i \in K\}$, we may always find i_1 and $i_2 \in K$ such that $i_1 < s < i_2$ where $i_1 = \max\{i \in K \mid i < h\}$ and $i_2 = \min\{i \in K \mid i > h\}$.

$g(X_{i1})$ when $\|X_s - X_{i1}\| \leq \|X_s - X_{i2}\|$

We may then define $h(X_s)$ =

$g(X_{i2})$ when $\|X_s - X_{i2}\| < \|X_s - X_{i1}\|$

ii) when $s < i_{min} = \min\{i \in K\}$ or $s > i_{max} = \max\{i \in K\}$,

$g(X_{imin})$ when $\|X_s - X_{imin}\| \leq \|X_s - X_{imax}\|$

We may then define $h(X_h)$ =

$g(X_{imax})$ when $\|X_s - X_{imax}\| < \|X_s - X_{imin}\|$

To explain the advantage of the CODEX technique over the technique of images by neighborhood, consider the example in Fig.7.

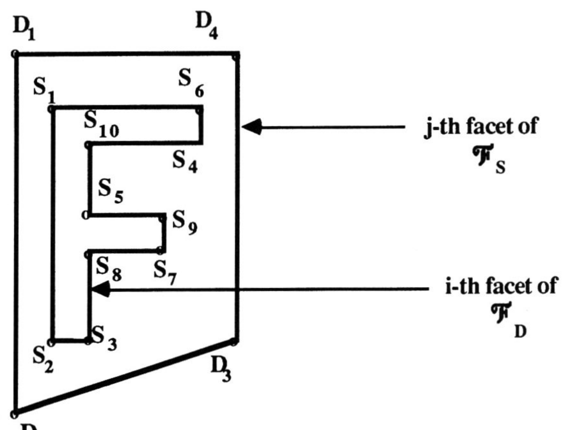

Fig.7 Example to show that the algorithm is able to deal with a random numbering

Using the principle of least squares, we obtain the bijective function f: $P_{Si} \rightarrow P_{Dj}$ such that:

$$X_1 = S_1 \quad \rightarrow \quad f(X_1) = Y_3 = D_1$$
$$X_2 = S_2 \quad \rightarrow \quad f(X_2) = Y_4 = D_2$$
$$X_5 = S_7 \quad \rightarrow \quad f(X_5) = Y_1 = D_3$$
$$X_{10} = S_6 \quad \rightarrow \quad f(X_{10}) = Y_2 = D_4$$

By applying the techniques of images by neighborhood, we obtain the correspondance shown in Fig.8.

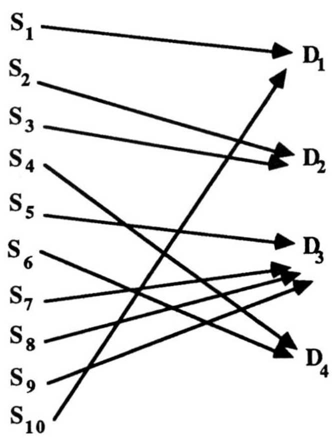

Fig.8 Vertex correspondence implying the inbetween configuration shown in Fig.9.

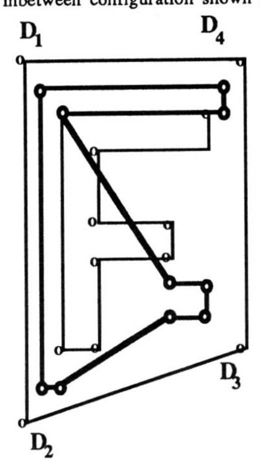

Fig.9 Inbetween configuration

When $\lambda \rightarrow 1$, we do not obtain a quadrilateral for $\mathcal{F}_D$.

By applying the CODEX technique, we obtain the correspondance in Fig.10.

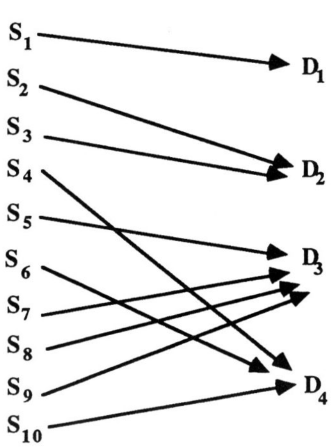

Fig.10 Vertex correspondence using the DICEX approach implying the inbetween configuration shown in Fig.11.

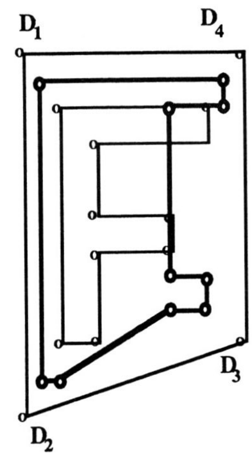

Fig.11 Inbetween configuration using the DICEX approach

When $\lambda \rightarrow 1$, figure $\mathcal{F}_S$ will be transformed into $\mathcal{F}_D$

CONCLUSION

We have presented an algorithm for solving the problem of interpolation between two facet-based graphical objects. The algorithm is based on the criterion of minimal dynamic displacement. This algorithm can transform any facet-based figure into any other facet-based figure; it provides good results in most cases. However, if the shapes of the two objects to be interpolated are too different, the results may not be aesthetically pleasing.

This algorithm has been implemented and introduced into the MIRANIM director-oriented animation system [Magnenat-Thalmann et al. 1985] and in the Multiple Track Animator System MUTAN [Fortin et al. 1983]. There are numerous applications of the method: simulation of biological evolution, animation, portrait-robots.

Fig.12-16 show applications of our algorithms.

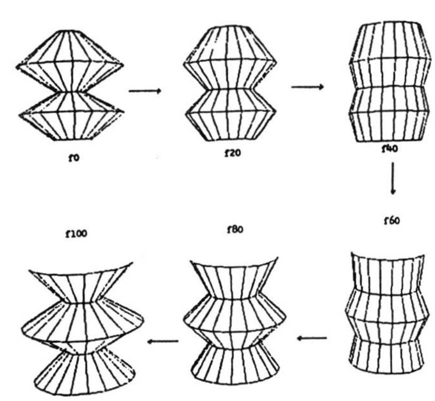

Fig.12 Case of Section 4.4 (figures are clockwise arranged)

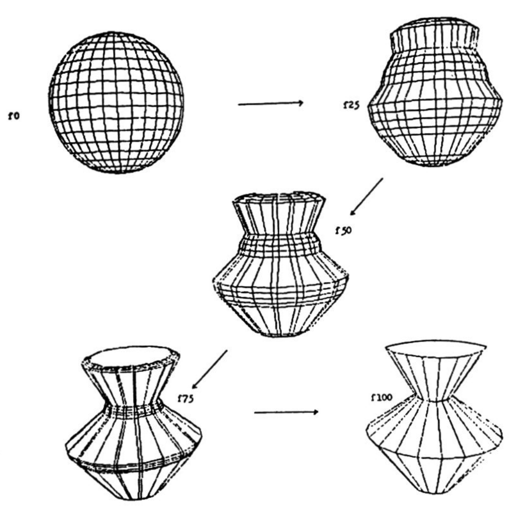

Fig.13 From a sphere to a revolution surface

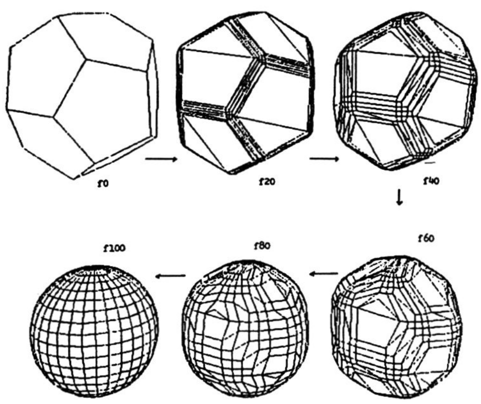

Fig.14 From a dodecahedron to a sphere

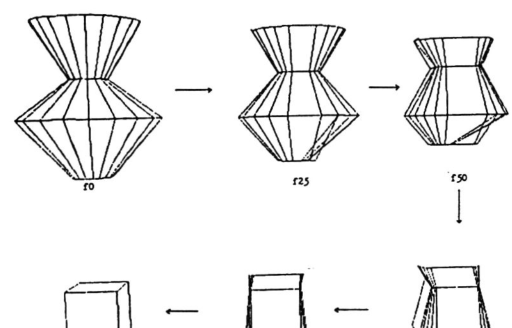

Fig.15 From a revolution surface to a cube

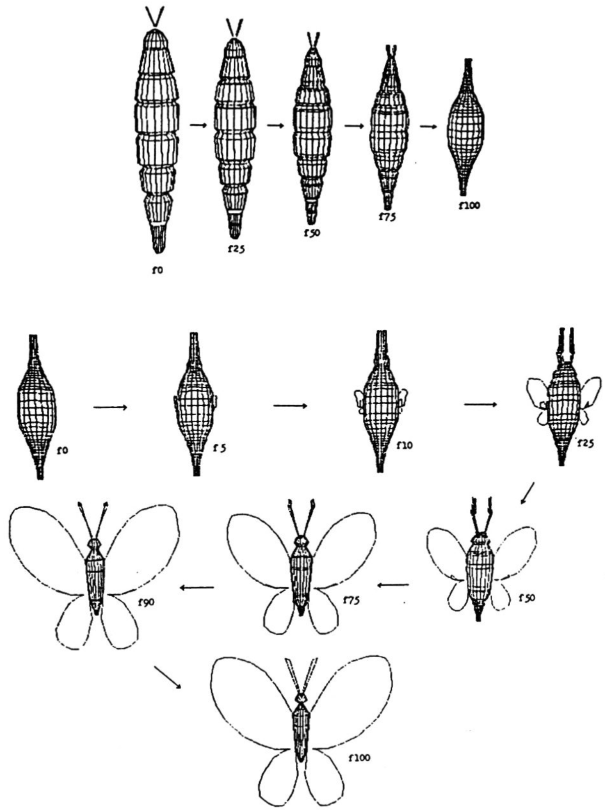

Fig.16 a-b. Butterfly metamorphosis

ACKNOWLEDGEMENTS

The research was supported by the Natural Sciences and Engineering Council of Canada, and the FCAR foundation (Government of Quebec).

REFERENCES

Burtnyk N, Wein M (1971) Computer-generated Key-frame Animation, Journal of SMPTE, 80, pp.149-153.
Burtnyk N, Wein M (1976) Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation, Comm. ACM, Vol.19, No10, pp.564-569.
Fortin D, Lamy JFL, Thalmann D (1983) A Multiple Track Animator System for Motion Synchronisation, Proc. ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion: Representation and Perception, Toronto, pp.180-186.
Kochanek D, Bartels R (1984) Interpolating Splines with Local Tension, Continuity and Bias Tension, Proc. SIGGRAPH '84, pp.33-41.
Magnenat-Thalmann N, Thalmann D (1985) Computer Animation: Theory and Practice, Springer, Tokyo New York Berlin Heidelberg.
Reeves W (1981) Intbetweening for Computer Animation Utilizing Moving Point Constraints, Proc. SIGGRAPH '81, Vol.15, No3, pp.263-269.
Steketee SN, Badler NI (1985) Parametric Keyframe Interpolation Incorporating Kinetic Adjustment and Phrasing Control, Proc. SIGGRAPH '85, pp. 255-262.
Zeltzer D (1985) Towards an Integrated View of 3D Computer Animation, The Visual Computer, Springer, Vol.1, No4, pp.249-259.