

ESSAI DE DEFINITION D'UN MONITEUR D'ANIMATION DE STRUCTURES

par M. G. COMPARETTI (1)

Résumé. — Cet exposé présente la technique utilisée dans un programme de construction d'images animées par un ordinateur.

La première partie montre de quelle manière l'image peut être décomposée en éléments qui ne sont soumis qu'à des déformations ou à des déplacements simples. Ces éléments sont liés entre eux par des lois dont l'ensemble constitue une structure. L'animation d'une image complexe peut être ainsi ramenée à la définition d'une structure et à la détermination, pour chaque élément, d'une forme, d'une position relative, d'un déplacement relatif et d'une déformation.

La seconde partie décrit la structure du moniteur. C'est un programme essentiellement composé de tables et des routines d'initialisation et d'exploitation de ces tables.

INTRODUCTION

Parmi les organes électroniques récents qui permettent de communiquer avec un ordinateur, les appareils qui traduisent directement les résultats sous forme d'images sont certainement appelés à un large et rapide développement.

Leur succès est dû pour une grande part à la facilité de lecture des résultats ainsi présentés. En effet, un gain considérable est souvent obtenu sur le temps d'interprétation de certains résultats fournis par l'ordinateur si ces derniers sont présentés sous la forme d'une courbe pour représenter une fonction mathématique, ou d'un film animé pour représenter l'évolution d'un phénomène plutôt que sous la forme unique d'un ou plusieurs tableaux de valeurs numériques.

Toutefois lorsque l'on dépasse le stade du tracé d'une simple courbe mathématique on rencontre l'obstacle de la difficulté de programmation d'une image à deux dimensions (ou trois dimensions dans le cas d'une projection stéréoscopique). Le programmeur préfère en effet consacrer son effort à la perfection de son calcul plutôt qu'à l'amélioration de la

(1) C.E.A., Centre d'Études de Limeil.

présentation des résultats. Ainsi l'absence d'outils commodes de programmation d'une image élaborée fait souvent renoncer à l'utilisation de cette possibilité.

Le but de l'étude qui suit est d'essayer de combler en partie cette lacune. Le moniteur d'animation présenté ici, utilisé pour la réalisation d'un film constitué de courbes mathématiques animées a réduit et surtout simplifié le travail de programmation dans une large mesure.

B. FRAUDET.

Rendre sensibles les propriétés d'une suite de fonctions par la visualisation de ses courbes représentatives successives, montrer les contours d'une surface en simulant un « travelling » autour de celle-ci, faciliter le travail d'un chorégraphe en lui permettant de visualiser à l'instant les figures du ballet qu'il conçoit ⁽¹⁾, ou bien confier à une machine le fastidieux calcul des intervalles d'un « cartoon » ⁽²⁾ sont quelques-unes des ambitions qu'a fait naître l'apparition des traceurs de courbes.

Or ces derniers ne savent généralement que placer un point ou joindre deux points choisis parmi un million de positions possibles disposées sur un carré.

C'est demander au calculateur associé à l'organe traceur un travail d'analyse particulièrement long et complexe. Il s'agit en effet de traduire par une suite d'ensembles de nombres entiers compris entre 0 et 1 024 des êtres aussi peu concrets qu'une famille de fonctions mathématiques ou que le personnage d'un cartoon.

Les pages qui suivent décrivent une tentative d'approche de la « chose animée », ainsi que la réalisation pratique d'un moniteur d'animation élémentaire.

PRINCIPE DE BASE DU MONITEUR

Supposons le problème résolu et considérons deux images successives extraites d'un film animé. Elles sont différentes et la seconde paraît être obtenue par de légères déformations de la première. Toutefois, la loi qui fait passer d'une image à l'autre semble, en général, très difficile à définir par des formules mathématiques.

Regardons plus en détail, nous constatons que seuls quelques éléments sont modifiés. Le plus souvent quelques éléments ont seulement changé de place. Deux types de modifications apparaissent alors :

1. — L'élément est indéformable, mais sa position par rapport au cadre de l'image a changé. Nous dirons dans la suite que l'élément a subi un « déplacement ».

(1) A. Michael NOLL, Bell Telephone Laboratories.

(2) Maurice CONSTANT, Waterloo Univ. Canada.

2. — L'élément reste en place, mais sa forme a varié ; nous dirons alors qu'il a subi une « déformation ».

Les deux sortes de modifications ci-dessus peuvent évidemment être décelées dans un même élément.

Remarquons tout de suite que la distinction entre déplacement et déformation est arbitraire. Ainsi la figure *B* (*fig. 1*) peut être considérée comme une déformation de la figure *A*, ou bien comme un déplacement du carré interne qui aurait tourné autour de son centre.

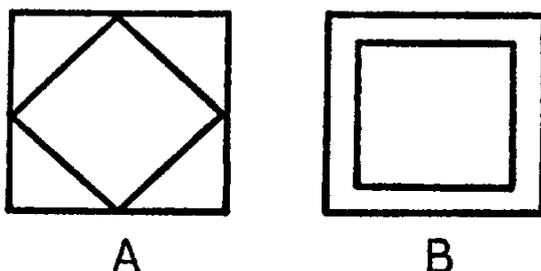


Figure 1

Considérée globalement, une image ne peut subir que des déformations. Par contre il est bien évident que l'on pourra trouver une décomposition telle que chaque élément ne subisse que des déplacements, devrait-t-on pousser la décomposition jusqu'à l'échelle atomique.

Par le terme « déplacement » nous comprendrons non seulement les translations et les rotations, mais aussi les homothéties. En effet lorsqu'un objet se déplace dans l'espace, son image change d'échelle.

Autre exemple (fig. 2)

1. — Les trois figures peuvent être déduites l'une de l'autre par une déformation bizarre.

2. — Il est préférable de les considérer comme un ensemble de deux carrés tournant à des vitesses différentes autour de leur centre commun. Le carré externe étant rigide, le carré interne ayant un sommet situé sur le carré externe.

Pour réaliser cette figure animée, il suffit dans le premier cas d'une seule fonction spécialisée dans le calcul des différentes positions d'une figure un peu complexe.

Dans le second cas plusieurs fonctions seront nécessaires :

1. — Une fonction dessinant la figure « carré » — utilisée deux fois.
2. — Une fonction réalisant la rotation d'une figure — utilisée deux fois ;
3. — Une fonction réalisant la liaison « sommet de *A* sur *B* » ;
4. — Une fonction capable de gérer les trois précédentes, c'est-à-dire de lancer leur exécution en leur fournissant les paramètres convenables.

L'ensemble paraît très complexe, mais chacune des fonctions est simple. De plus ce sont des fonctions standard susceptibles d'apparaître dans le traitement de figures très différentes.

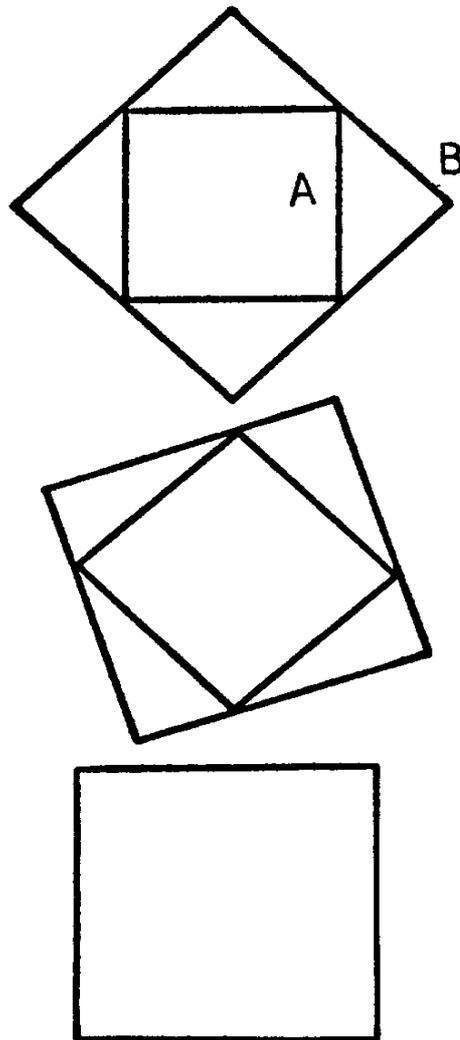


Figure 2

Considérons maintenant le personnage de la figure 3 (1). Plusieurs décompositions sont possibles. Choisissons la suivante qui comporte six membres :

- 1, le corps,
- 2, la tête,
- 3, le train (ensemble des 2 pattes),
- 4, l'œil,
- 5, la queue,
- 6, le bec.

(1) Extrait du film *Fantasy in ...* de L. VAN MAELDER.

Les relations logiques qui existent entre chacun des membres vont nous permettre de définir une structure. Il nous faut d'abord choisir le membre qui sera la base de notre structure. Si le personnage est pendu par la tête nous choisirons la tête ; s'il vole nous choisirons le corps. Nous voulons le faire marcher, choisissons le train. Nous avons les relations

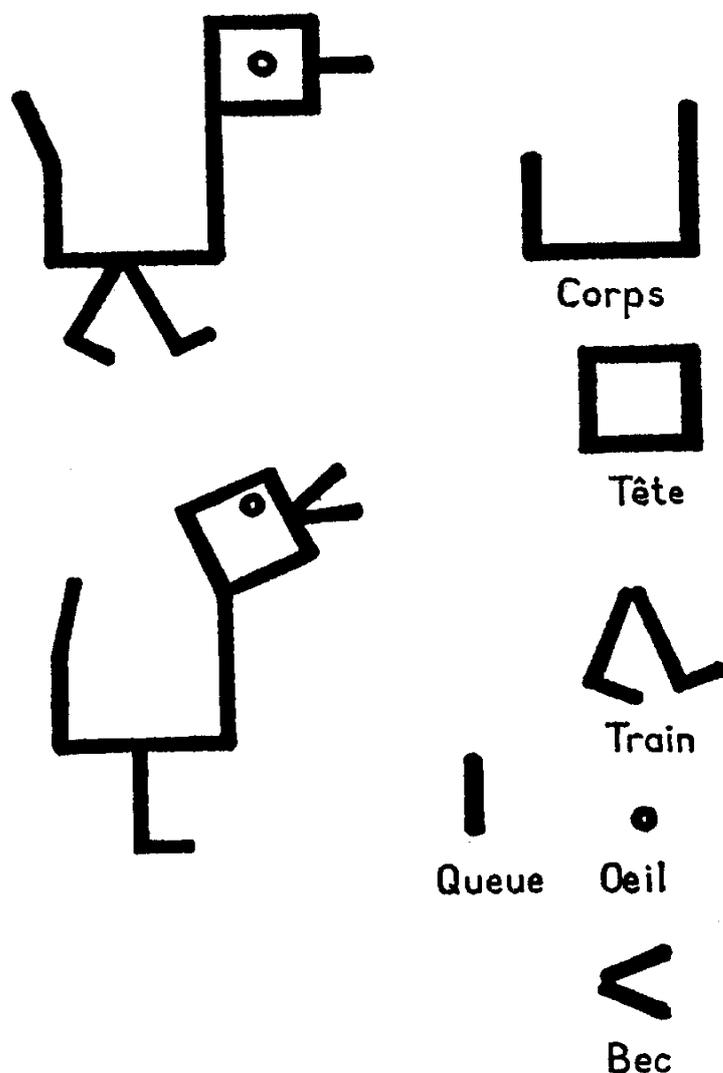


Figure 3

suivantes : le train s'appuie sur le sol, le corps est porté par le train, la queue et la tête sont sur le corps, l'œil et le bec sont sur la tête.

Nous avons défini la structure représentée sur la figure 4.

A partir de celle-ci, nous allons essayer de reconstruire notre personnage animé.

Prenons par exemple le membre CORPS. Il faut lui associer une forme. Cela sera réalisé par une fonction qui tracera la forme du corps, rapportée à un référentiel lié au membre CORPS. Nous choisissons ce référentiel que nous appellerons R_1 (fig. 5).

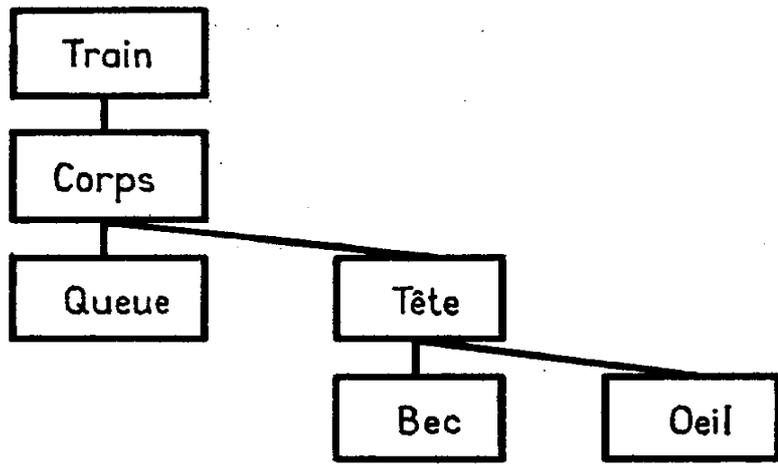


Figure 4
Structure

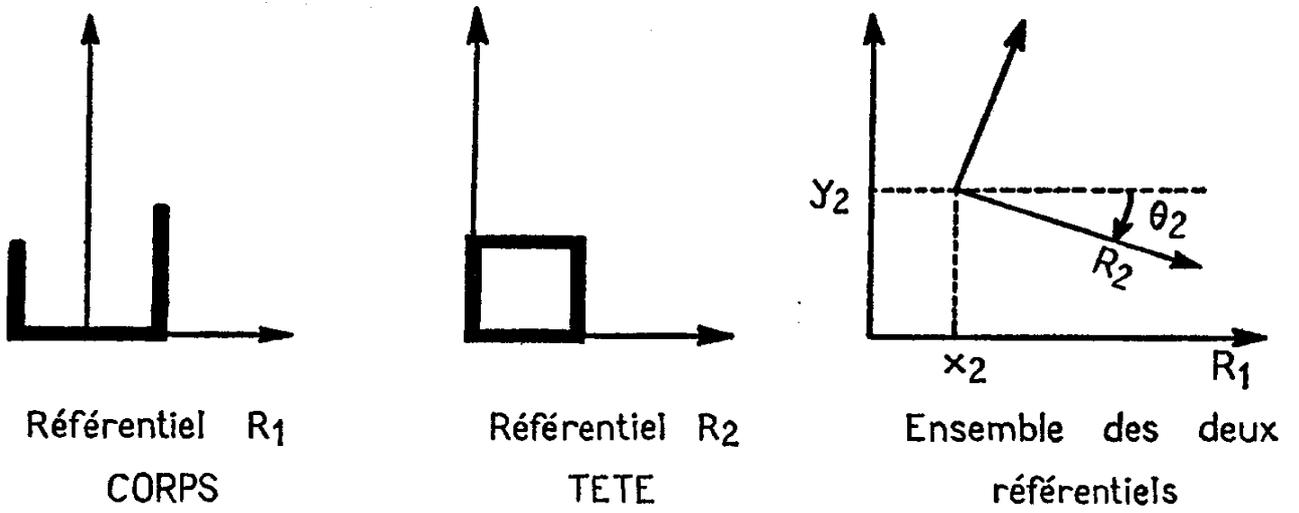


Figure 5
L'ensemble corps-tête

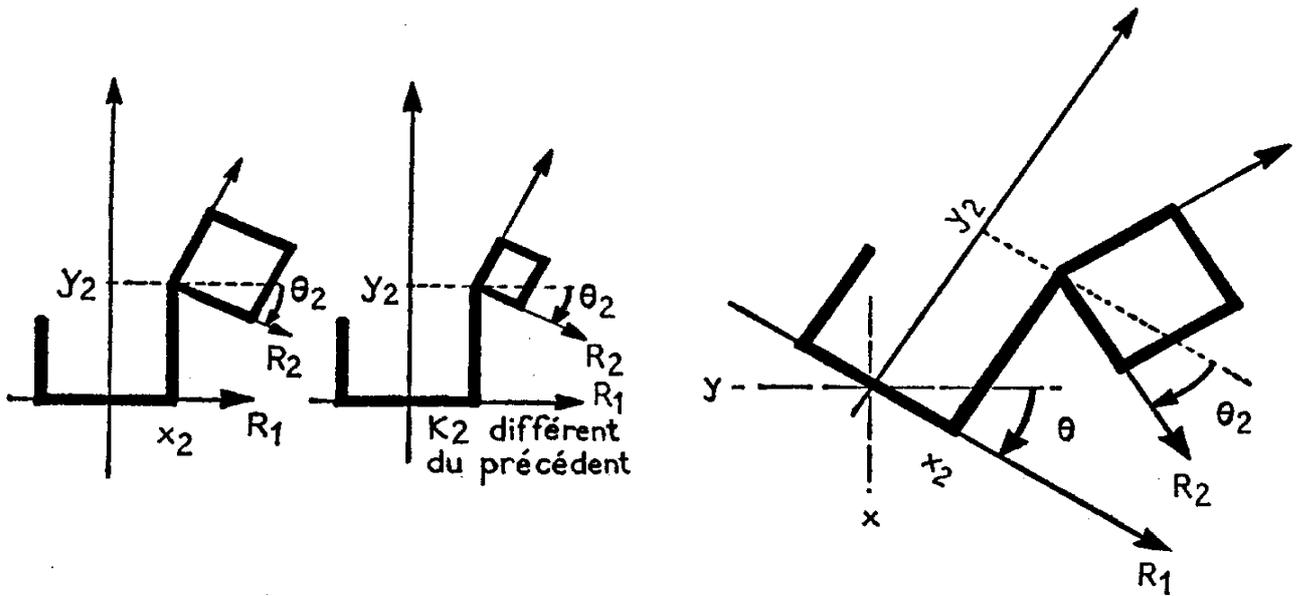


Figure 6
L'ensemble corps-tête

Une seconde fonction tracera une tête rapportée au référentiel R_2 . L'ensemble CORPS-TETE sera réalisé lorsque nous aurons défini la position du référentiel R_2 rapportée au référentiel père R_1 (fig. 6).

Les paramètres x_2, y_2, θ_2 et les deux fonctions de formes précédentes définissent parfaitement l'ensemble CORPS-TETE. Ajoutons un paramètre K_2 de rapport d'échelles.

Si nous connaissons les paramètres x, y, θ, K définissant la position absolue du membre CORPS (coordonnées absolues de R_1), nous saurons tracer l'image de l'ensemble CORPS-TETE.

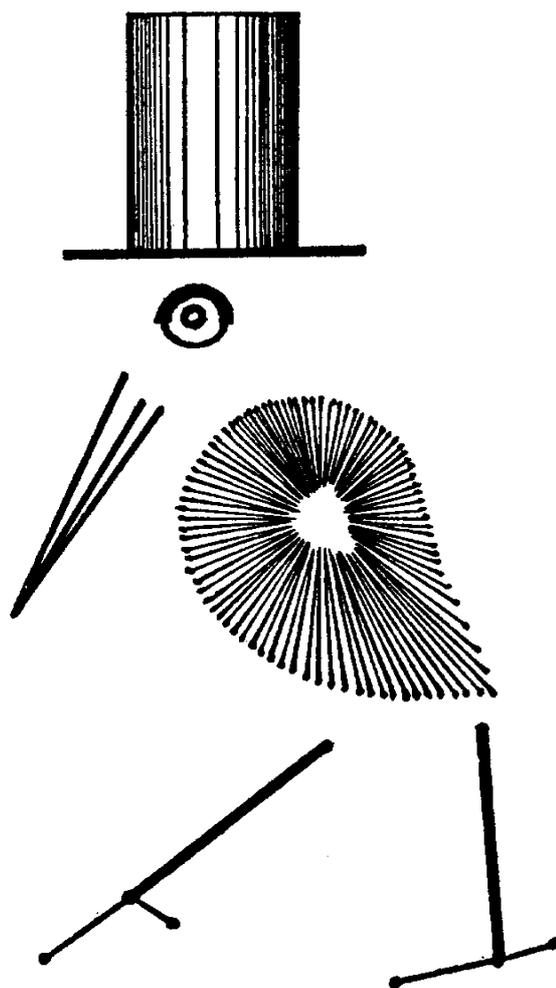


Figure 7

En résumé, pour tracer le membre TETE, élément rigide, il nous faut connaître :

- 1, la forme, rapportée au référentiel R_2 ,
- 2, la position de R_2 rapportée au référentiel R_1 du membre père,
- 3, la position absolue de R_1 .

L'exécution des trois fonctions suivantes sera nécessaire :

1. Calcul de la position relative actuelle de R_2/R_1 .
 2. Calcul de la position absolue de R_2
 3. Fabrication de la forme TÊTE rapportée à R_2
- } production de l'image absolue.

La figure 7 montre un personnage d'aspect différent de celui étudié.

L'animation en serait pourtant très voisine. On constate la présence de ce que l'on appellera plus loin un membre virtuel :

Il s'agit de la tête.

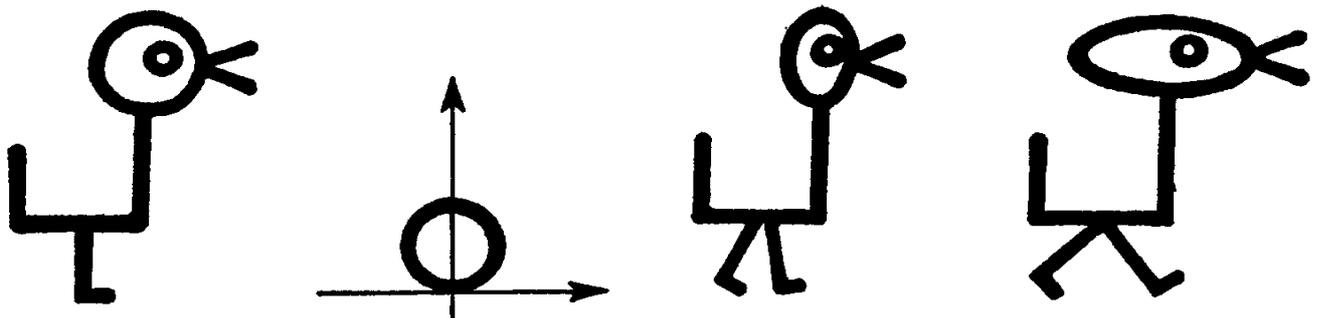


Figure 8

Ce membre existe logiquement. C'est lui qui supporte le chapeau, l'œil, le bec. Il doit apparaître dans la structure. Pourtant il ne produit aucune image.

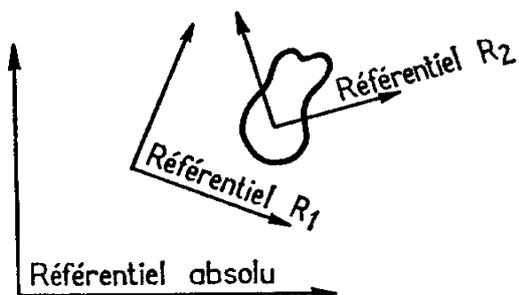
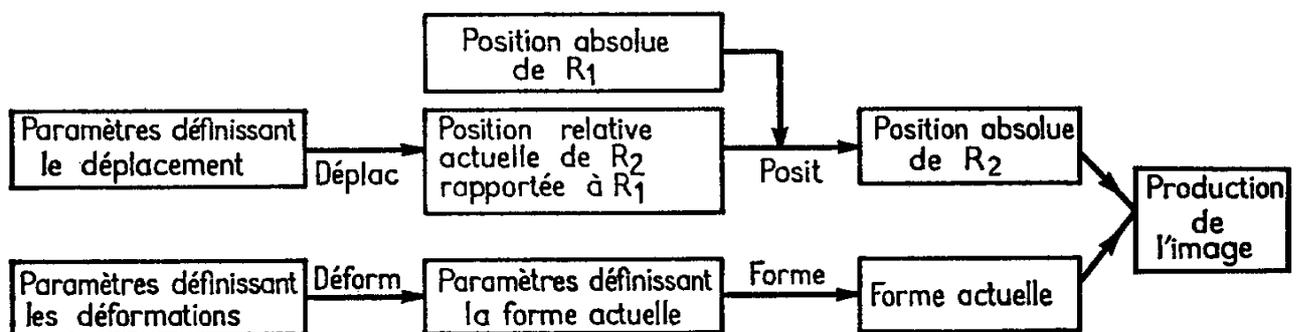


Figure 9

Les rectangles représentent des ensembles de valeurs de paramètres.
Les flèches symbolisent les fonctions utilisant ou calculant certaines de ces valeurs.

Supposons maintenant que la tête soit déformable (*fig. 8*).

Au lieu du carré rigide précédent, prenons une forme elliptique dépendant d'un paramètre P . Aux trois fonctions précédentes il faudra ajouter la fonction : calcul de la valeur actuelle du paramètre P .

Nous aurons donc, dans le cas général à exécuter quatre fonctions logiques. Nous leur donnerons dans ce qui suit les noms suivants : DEPLAC, DEFORM, POSIT, FORME

DEPLAC : Calcul de la position relative actuelle du référentiel R_2 rapportée au référentiel R_1 (*fig. 9*).

DEFORM : Calcul des paramètres définissant la forme actuelle.

POSIT : Calcul de la position absolue du référentiel R_2 .

FORME : Tracé de la forme.

On peut noter que les deux premières fonctions DEPLAC et DEFORM ont un rôle dynamique. Les deux dernières POSIT et FORME ont un rôle statique.

Dans le cas général le tracé d'un membre peut être représenté par le schéma de la figure 9.

LE MONITEUR D'ANIMATION DE STRUCTURES

Rôle

D'après ce que nous venons de voir, on peut résumer le travail de l'animateur humain à fournir trois types de données :

- Définir une structure.
- Définir la forme de chaque membre.
- Définir les mouvements relatifs ou les positions relatives des membres.

Le rôle du moniteur va être de produire le film animé à partir de ces données.

Fournir les données consistera principalement à remplir les cases de l'organigramme de la figure 10.

- Soit avec le nom d'un programme exécutant la fonction indiquée.
- Soit avec les valeurs des paramètres correspondants.

Remarque :

Une case fonction vide est ignorée.

Ainsi :

la case DEPLAC vide représente un membre fixe par rapport au référentiel,

la case POSIT vide représente un membre rapporté au même référentiel que le membre père,

la case DEFORM vide représente un membre rigide,

la case FORME vide représente un membre virtuel (c'est-à-dire ayant un rôle dans l'articulation de l'ensemble, mais ne produisant lui-même aucun tracé).

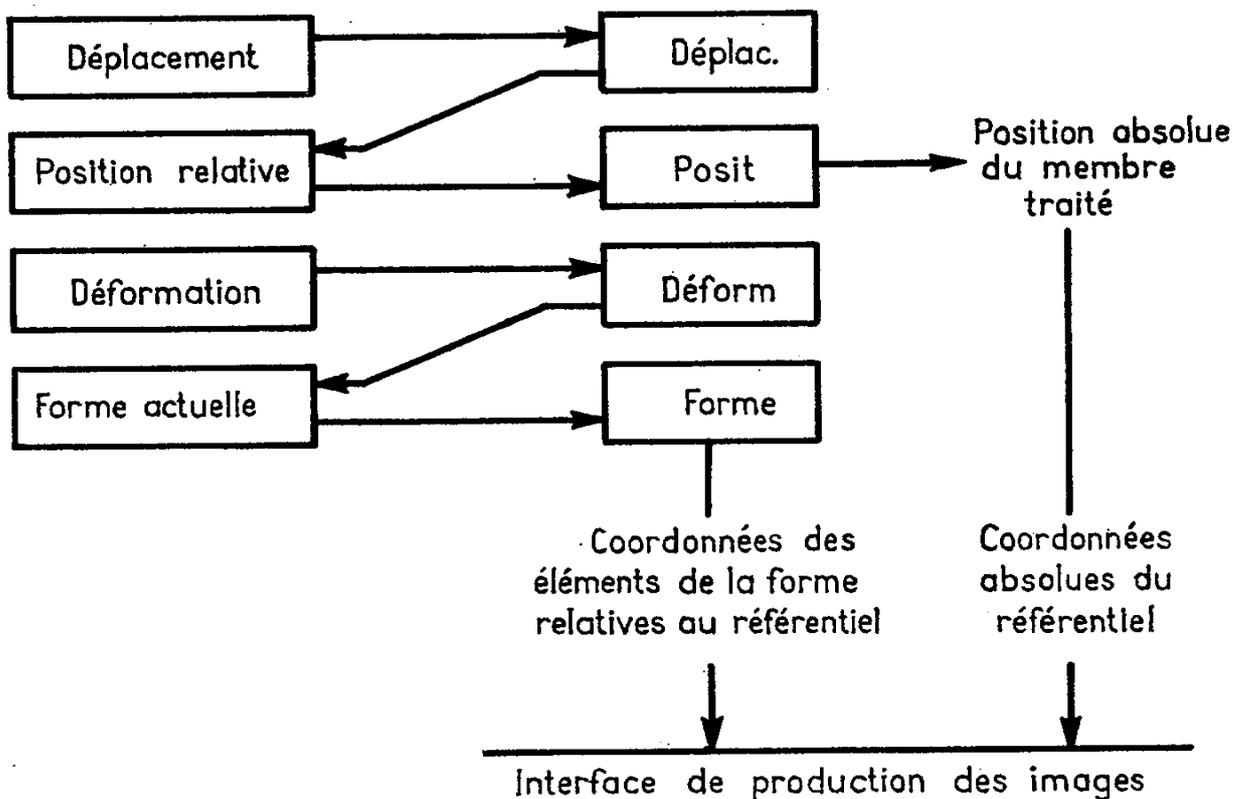


Figure 10

Structure du moniteur

Il se compose de trois parties essentielles.

1. — Module BLDCHN

Il est chargé de la définition des structures. Les données qui lui sont fournies sont codifiées et entrées dans une table appelée table CHAINE.

Celle-ci possède alors suffisamment d'information pour permettre un balayage dans l'ordre hiérarchique des structures qu'elle contient, ainsi que la modification dynamique de ces dernières.

2. — Module BLDFON

Il gère deux tables :

La table FONCTIONS qui contient les noms des fonctions précédemment définies. La table PARAMETRES qui permet de retrouver les paramètres associés aux fonctions précédentes.

A chaque membre de structure entré dans la table CHAINE est associée une entrée dans la table FONCTIONS et une entrée dans la table PARAMETRES.

Chacune de ces entrées possède quatre cases correspondant aux quatre fonctions DEPLAC, POSIT, DÉFORM, FORME.

Le module BLDFON a pour rôle d'initialiser ces cases à partir des données qui lui sont fournies.

3. — Module GOON

Son rôle est de produire les images en exploitant les tables CHAINE, FONCTIONS et PARAMETRES.

Il travaille à partir des données suivantes :

— Type du support matériel de l'image (actuellement papier ou film 35 mm).

— Valeurs extrêmes à donner à la variable temps.

— Intervalle de temps séparant deux images (action sur la vitesse d'animation).

— Nombre de fois que doit être dupliquée chaque image sur le film (action sur la finesse de l'animation).

Langage

Un langage élémentaire et provisoire a été utilisé pour la mise au point du moniteur et pour les premières réalisations.

Analysé et traité par un interpréteur, ce langage est composé d'instructions qui permettent soit de sélectionner un module, soit de charger les tables CHAINE, FONCTION ou PARAMETRE avec les valeurs désirées, soit de contrôler la production des images.

D'une utilisation simple, il offre toutefois l'inconvénient majeur de ne refléter en rien l'évolution visuelle des images à produire.

EXEMPLE DE PROGRAMMATION DU MONITEUR

Animation du personnage cité comme exemple dans ce qui précède (fig. 11).

Les termes soulignés représentent les variables du langage (noms de membre, nom de programme, ou valeur numérique).

A) *Définition de la structure.* — Chargement de la table CHAINE :

— *BLDCHN (sélection du module BLDCHN)

— STRUCTURE OISEAU

— LIER TRAIN A OISEAU

— LIER CORPS A TRAIN

- LIER QUEUE A CORPS
- LIER TETE A CORPS
- LIER BEC A TETE
- LIER ŒIL A TETE

B) *Chargement des tables* FONCTION et PARAMETRE :

- *BLDFON (sélection du module BLDFON)

1. Définition des formes :

- FORME TRAIN AVEC SUBF1 ALLOC 1 (allocation d'une mémoire de travail)
- FORME CORPS AVEC SUBF2
- FORME QUEUE AVEC SUBF3
- FORME TETE AVEC SUBF4
- FORME BEC AVEC SUBF5 ALLOC 1 (idem)
- FORME ŒIL AVEC SUBF6

Les noms SUBF1, ..., SUBF6 sont des noms de programmes chargés de dessiner, la forme des membres correspondants. SUBF1 et SUBF5, représentant des formes variables (le TRAIN et le BEC) nécessitent la donnée d'un paramètre pour lequel une mémoire est allouée.

2. Définition des positions relatives :

- POSIT TRAIN AVEC SUBP1 ALLOC 4 LIRE
- carte contenant les valeurs des paramètres abscisse, ordonnée, angle et échelle du membre TRAIN
- POSIT CORPS AVEC SUBP1 ALLOC 4 LIRE
-
- POSIT ŒIL AVEC SUBP1 ALLOC 4 LIRE
- carte donnée contenant les valeurs des paramètres.

Ici le même programme SUBP1 peut être utilisé pour tous les membres.

3. Définition des déformations pour les membres variables TRAIN et BEC :

- DEFORM TRAIN AVEC SUBDF1 ALLOC 3 LIRE
- carte donnée
- DEFORM BEC AVEC SUBDF5 ALLOC 4 LIRE
- carte donnée

Les programmes SUBDF1 et SUBDF5 calculent la valeur du paramètre définissant la forme actuelle pour les membres TRAIN et BEC.

4. Définition des déplacements pour les membres TRAIN et QUEUE, qui bougent :

- DEPLAC TRAIN AVEC TRANSX ALLOC 1 LIRE
- carte donnée
- DEPLAC QUEUE AVEC ROT ALLOC 4 LIRE
- carte donnée

Le programme TRANSX modifie les paramètres de position, image par image de façon à provoquer une translation parallèle à l'axe des abscisses.

Le programme ROT provoque une rotation.

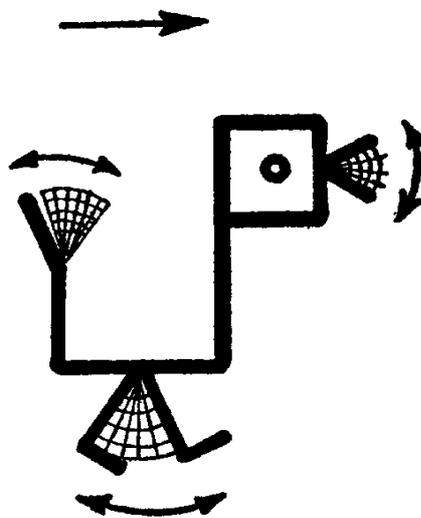


Figure 11

C) *Calcul et production des images :*

- *GOON (sélection du module GOON)
- SUR FILM TIRER OISEAU IMAGE 1 DUPLIQUE 50
- SUR FILM TIRER OISEAU IMAGE 1 à 200 DUPLIQUE 2
PAS 2
- SUR FILM TIRER OISEAU IMAGE 200 DUPLIQUE 50

On obtient ainsi :

- 50 images de plan fixe de la première image,
- 200 images d'animation identiques deux à deux,
- 50 images de plan fixe de la dernière image.