



VERS UNE ARCHITECTURE BANALISÉE DES SYNTHÉTISEURS D'IMAGE
 An approach to a general purpose architecture of image synthesizers

FRANCIS MARTINEZ

Institut IMAG / Laboratoire ARTEMIS / B.P. 68 / 38402 SAINT MARTIN D'HERES cédex

RESUME

SUMMARY

Chaque semaine un nouveau terminal de synthèse d'image fait son apparition sur le marché et l'utilisateur a de plus en plus de difficultés à déterminer celui qui convient le mieux à ses applications. Deux types de configurations sont généralement opposées :

- Un terminal "bas de gamme" muni de logiciels permettant de satisfaire un grand nombre d'applications sans aucune possibilité d'interaction temps réel.
- Un terminal évolué adapté à une classe spécifique d'applications et proposant des fonctions temps-réel. Dans ce dernier cas l'architecture du synthétiseur fait appel à un nombre important d'unités de traitement cablées et la reconfiguration du processus interne du terminal est quasiment impossible.

L'architecture modulaire présentée dans cet article tente de résoudre ces problèmes en considérant séparément les informations de morphologie, de géométrie, d'aspect et d'éclairage nécessaires à la définition d'une image. Ces informations sont traitées et synthétisées par des unités de traitement banalisées, connectées en fond de panier sur un bus standardisé. Toutes ces unités sont cablées et opèrent à la vitesse du signal vidéo.

Cette architecture originale permet de nombreuses combinaisons permettant de répondre à une grande variété d'applications. Un prototype offrant des possibilités en temps réel (apposition de textures et calcul d'éclairage) est actuellement en cours de réalisation à l'Université de Grenoble.

Every week a new raster graphic device is available and the user has to select the best one according to his applications. This choice is not easy and two kinds of configuration generally face him.

- A low-cost terminal with a lot of software packages which satisfy many applications but doesn't allow any real-time interaction.
- A powerful terminal based on a specific application which allows many real-time facilities. In this case, the architecture of the synthesiser uses lot of wired units and modifications of the inner process is quite impossible.

The modular architecture discussed in this paper attempts to resolve these problems by separating morphology, geometry, aspect and lighting information which are handled apart by general purpose process units plugged on a standardized bus. All this units work at video rate.

This original architecture allows many different combinations according to the application requirements. A prototype system with some real-time capabilities (texturing and lighting process) is currently developed at the University of Grenoble.



VERS UNE ARCHITECTURE BANALISÉE DES SYNTHÉTISEURS D'IMAGE

An approach to a general purpose architecture of image synthesizers
Francis MARTINEZ

1. PRESENTATION

La mutation technologique de ces dernières années ainsi que l'accroissement des besoins en images synthétiques conduisent à une remise en cause périodique de l'architecture des terminaux dédiés à la synthèse d'images. La tendance actuelle est de répondre à la grande variété des performances exigées par une aussi grande variété de machines fondées sur des architectures spécialisées. L'éventail des possibilités en matière de qualité d'image (allant du dessin au trait aux images réalistes) ou encore en matière de dynamisme (allant de l'image entièrement statique à l'animation en temps réel) est ainsi pratiquement couvert par la gamme des terminaux disponibles.

Cependant, cette spécialisation conduit inévitablement à une diversification néfaste sous plusieurs aspects. En premier lieu, elle restreint notablement le champ d'application de chaque machine de synthèse, limitée par les possibilités d'une architecture figée et les performances "actuelles" de la technologie utilisée. En second lieu, la disparité et la spécificité des techniques entraîne une incompatibilité totale entre les différents terminaux, et l'absence de possibilités d'extension et d'adaptation enchaîne le plus souvent l'utilisateur à un constructeur voire même à un terminal donné.

Face à cette situation nous proposons de jeter les bases d'une architecture "universelle" de synthétiseur d'image, permettant un haut degré de parallélisme, extensible et modulaire afin de répondre par simple programmation ou adjonction de modules, aux besoins spécifiques d'applications variées. Cette architecture supporte aussi bien des terminaux standards pour lesquels la couleur de chaque pixel est enregistrée dans une mémoire de trame, que des synthétiseurs temps-réel d'images réalistes prenant en charge, en parallèle, le calcul cablé de textures colorées, de l'éclairage d'objets tri-dimensionnels et de l'élimination des parties cachées.

Après avoir présenté les principes de base de la synthèse d'image, nous en déduisons les fondements d'une telle architecture, basée notamment sur la séparation et la "parallélisation" des informations graphiques.

2. LES CONCEPTS LIÉS A LA SYNTHÈSE D'UNE IMAGE.

Ce paragraphe est destiné à mettre en valeur les différents éléments nécessaires à la synthèse d'une image. Nous serons ainsi amenés à considérer successivement :

- les informations traitées,
- les opérateurs les concernant,
- les processus permettant de combiner ces opérateurs.

2.1. Les classes d'information

Les informations véhiculées par une image peuvent être partitionnées en six classes indépendantes [Mar 82]:

- Identité : le nom des objets
- Morphologie : leur forme intrinsèque
- Aspect : leur aspect intrinsèque (couleur, texture)
- Géométrie : leur situation dans la scène
- Eclairage : leur réaction à la lumière (réflexions)
- Structure : leurs liens hiérarchiques ou autres.

2.2. Les opérateurs élémentaires de synthèse

La synthèse d'image consiste à combiner les six classes d'informations initiales ci-dessus pour obtenir l'information finale exprimant la couleur en chacun des pixels de l'image (classe A).

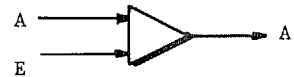
Pour effectuer cette synthèse il faut procéder par étapes successives à l'aide d'opérateurs élémentaires de calcul tels que ceux présentés ci-après [Mar 82].

Synthèse $M' = f(M,G)$:



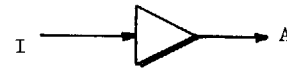
Application d'une transformation géométrique à une forme (M).

Synthèse $A' = f(A,E)$:



Modulation de l'aspect en fonction de l'éclairage.

Correspondance $A = f(I)$:

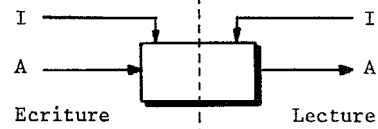


Délivre un aspect en fonction de l'identité.

Tous ces opérateurs sont des opérateurs de calcul synchrones ce qui signifie que les résultats sont exploités dès que les données nécessaires ont été fournies.

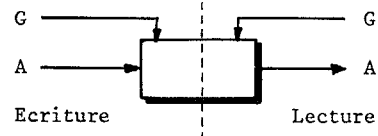
Cependant, la synthèse d'une image nécessite au moins un opérateur asynchrone servant de "tampon" entre deux processeurs différents (par exemple la logique cablée et le micro-processeur pilote). Cet asynchronisme est assuré généralement par un opérateur de mémorisation tel que ceux présentés ci-après.

Correspondance $A = f(I)$:



Cet opérateur est une simple "table de couleurs" délivrant une couleur (A) pour chaque objet exprimé à travers son identité (I).

Correspondance $A = f(G)$



Il s'agit ici d'une mémoire de trame (mémoire d'image) délivrant une couleur (A) pour chaque pixel, exprimé par sa position (G).

2.3. Les processus de synthèse d'image

Les performances d'un synthétiseur d'image (rapidité, qualité et complexité des images) dépendent directement du choix des opérateurs élémentaires et de leur ordonnancement. Nous avons montré lors de travaux précédents [Mar 82] que cet ordonnancement peut être déduit d'une façon quasi-systématique des performances requises. On peut illustrer ces propos en s'appuyant sur le processus d'un terminal "classique" schématisé sur la figure 1.



VERS UNE ARCHITECTURE BANALISÉE DES SYNTHÉTISEURS D'IMAGE

An approach to a general purpose architecture of image synthesizers

Francis MARTINEZ

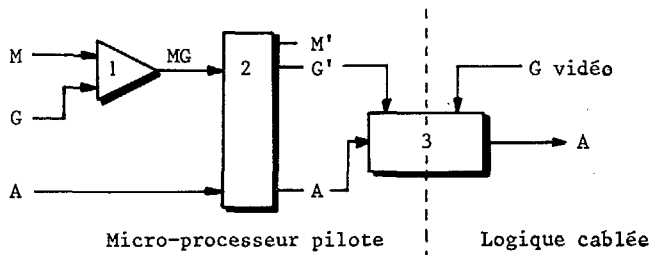


Figure 1

Si ce terminal traite des tâches polygonales uniformes, le rôle des opérateurs est le suivant :

- Opérateur 1 :** Application des transformations géométriques (translations, rotations, homothéties) aux sommets de chaque polygone. Le résultat est un nouveau polygone.
- Opérateur 2 :** Décomposition du polygone en pixels. L'objet élémentaire considéré à présent est donc le pixel pour lequel la morphologie (M') est implicite, la géométrie (G') exprime les coordonnées et l'aspect (A) est celui du polygone initial.
- Opérateur 3 :** Il s'agit de la mémoire de trame dans laquelle est mémorisée la couleur de chaque pixel.

A partir de ce cas très simple, on peut envisager une infinité de variantes adaptés aux besoins spécifiques des applications. Par exemple si l'on désire favoriser l'interactivité temps réel au niveau des couleurs on ajoute "après" la mémoire de trame une table de correspondance (Opérateur 4 de la Figure 2).

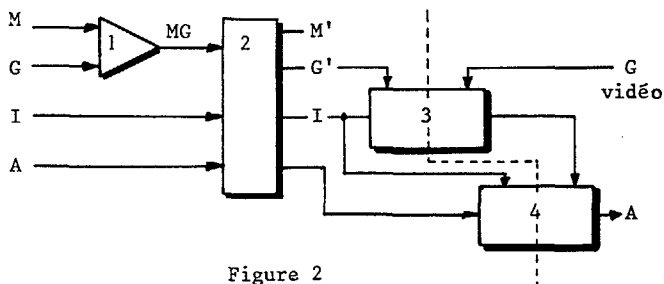


Figure 2

Si l'on désire présenter des surfaces gauches comportant toutes les nuances dues à l'éclairage tout en conservant l'interactivité temps réel sur les couleurs on peut enregistrer dans une seconde mémoire de trame (opérateur 5) les informations d'éclairage relatives à chaque pixel (réflexion diffuse et spéculaire). Le processus est alors celui de la figure 3 dans lequel l'opérateur 6 effectue la modulation de la couleur en fonction du modèle d'éclairage.

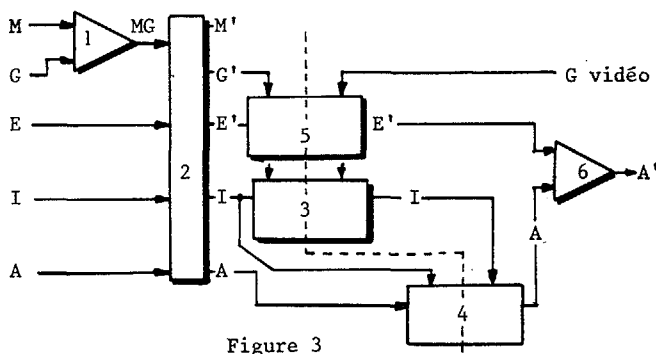


Figure 3

3. INCIDENCE AU NIVEAU DE L'ARCHITECTURE D'UN SYNTHÉTISEUR

Les considérations évoquées ci-dessus ont une incidence directe sur l'architecture interne d'un synthétiseur d'image. Nous nous intéresserons ici plus particulièrement à la partie câblée qui conditionne très largement les performances d'ensemble.

3.1. La nature des modules

Nous nommerons *module* un processeur câblé (généralement regroupé sur une carte) associé à un opérateur élémentaire du processus. Outre les modules de calcul, les opérateurs de mémorisation revêtent une forme différente selon que l'information qui joue le rôle d'adresse est mono-dimensionnelle (cas des tables de correspondance) ou bi-dimensionnelle (cas des mémoires de trame). Trois types de modules peuvent donc être présents dans un système. A ceci il convient cependant d'ajouter deux modules spéciaux extérieurs au processus de synthèse : le *module de communication* chargé de l'écriture des mémoires et de la configuration des calculs ; le *module d'affichage vidéo* chargé de la génération du signal vidéo. Tous ces modules doivent être capables de fonctionner à la vitesse du signal vidéo soit dans une période de 75 ns environ dans le cas du standard T.V. Si le traitement à effectuer dépasse le temps imparti (cas des mémoires de trame et des calculs complexes) une architecture pipe-line permet de décomposer ce traitement en N sous-espaces. Le retard introduit par le module est alors de N cycles.

3.2. La nature des bus

Les bus internes du synthétiseur câblé véhiculent toutes les informations présentes dans le processus de synthèse. Ainsi, si l'on se reporte à la figure 3, la partie câblée comporterait 5 bus véhiculant en synchronisme les informations :

G vidéo, E' , I , A , et A').

Le débit de ces bus est également celui du signal vidéo et leur "largeur" dépend de l'information transmise soit, par exemple 20 bits pour G vidéo (1024×1024), 8 bits pour E' (intensité), 12 bits pour I (numéro d'objet), 18 bits pour A et A' (6 bits par composante RVB).

Il faut, en outre, prévoir l'écriture des informations dans les modules de mémorisation. Si la partie micro-programmée comporte plusieurs processeurs pilotes indépendants fonctionnant en parallèle, il est intéressant de prévoir plusieurs bus d'écriture. Dans le cas où tous les opérateurs micro-programmés sont confiés à un seul micro-processeur, un seul bus de données est nécessaire. Ce bus doit avoir la largeur de la plus grande de toutes les informations à transmettre. Certaines informations jouent le rôle d'adresse (G' et I dans le processus de la figure 3) et la largeur du bus d'adresse est également celle de la plus grande de ces informations.

3.3. La synchronisation du processus

Si l'on se reporte encore une fois au processus de la figure 3, il est clair que les informations (A) et (E') doivent parvenir en synchronisme aux entrées de l'opérateur 6. Or les chemins parcourus sont différents et les opérateurs traversés peuvent induire des retards également différents. Il est donc indispensable d'introduire dans le plus court chemin un opérateur neutre introduisant un retard de compensation.



VERS UNE ARCHITECTURE BANALISÉE DES SYNTHÉTISEURS D'IMAGE

An approach to a general purpose architecture of image synthesizers
Francis MARTINEZ

4. LA BANALISATION DE L'ARCHITECTURE

Les terminaux actuels sont généralement conçus selon un processus donné et obéissent donc aux principes ci-dessus. Ces terminaux sont parfaitement adaptés à un type d'application mais deviennent inefficaces voire même inutilisables lorsque l'on désire les introduire dans un processus différent de celui pour lequel ils ont été conçus. Ainsi si la mémoire de trame est destinée à recevoir la couleur de chaque pixel, on ne peut l'utiliser pour enregistrer un numéro d'objet, ou une profondeur pour l'algorithme du Z-buffer, ou encore un vecteur normal pour les surfaces gauches.

Notre objectif est de proposer une architecture modulaire, banalisée, et permettant une reconfiguration complète de la logique câblée en vue de s'adapter à une majorité de processus de synthèse. Nous exposons ci-après les problèmes posés par ce type d'approche et les solutions qui ont été retenues pour la conception d'un prototype en cours de réalisation au Laboratoire ARTEMIS de l'Institut IMAG de Grenoble. Nous reprendrons successivement les trois points évoqués ci-dessus : les modules les bus et enfin la configuration et la synchronisation du processus.

4.1. Les modules banalisés

Il est clair que la banalisation de l'architecture passe d'abord par celle des modules : les mémoires de trame, les tables de correspondances et les modules de calcul. Si les derniers restent spécifiques à un opérateur donné, les deux premiers peuvent mémoriser des informations quelconques et un type unique de mémoire de trame et de table de correspondance a été développé. Chaque module, quel qu'il soit, fait actuellement l'objet d'une carte "double europe" et obéit au même schéma de connexion sur le fond de panier.

4.2. Les bus banalisés

Le nombre et la largeur des bus à prévoir dépend des processus que devra supporter le synthétiseur. Il est bien entendu que la plus grande souplesse est donnée par un grand nombre de bus les plus larges possibles. Par ailleurs tous ces bus doivent être situés dans le fond de panier afin de pouvoir alimenter ou collecter tous les modules présents. En appliquant strictement ces principes un synthétiseur banalisé susceptible d'être configuré selon des processus variés devrait comporter en fond de panier un nombre de lignes prohibitif (environ 180 lignes pour le processus simple de la figure 3).

Une première remarque permet de constater que le bus "G vidéo" ne s'adresse qu'aux mémoires de trame et qu'il peut être supprimé si l'on introduit dans chaque mémoire de trame les compteurs (X,Y) effectuant le balayage. Cette option permet en outre d'effectuer la translation câblée des différentes mémoires.

En second lieu l'expérience permet de vérifier qu'un grand nombre de bus réduits est plus intéressant qu'un nombre limité de bus plus larges. Il est alors possible, pour les applications exigeant une plus grande précision, de coupler deux bus pour véhiculer l'information.

Le test d'un grand nombre de configurations répondant à des besoins divers a montré qu'une largeur de 12 bits est un bon compromis et que le nombre de bus peut être limité à 8 ; soit 96 lignes en tout. Ces lignes occupent la partie basse du fond de panier qui est schématisé sur la figure 4.

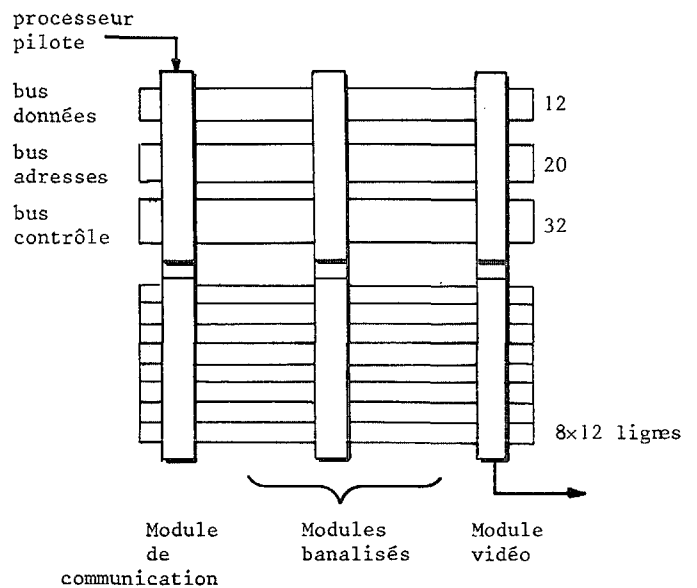


Figure 4

Compte-tenu des données ci-dessus les modules doivent répondre aux spécifications ci-après :

- Mémoire de trame : plan de $1024 \times 1024 \times 12$ bits, la sortie peut être aiguillée sur l'un quelconque des 8 bus.
- Table de correspondance : mémoire de 4096×12 bits, l'entrée et la sortie peuvent être aiguillées sur l'un quelconque des 8 bus.

4.3. Configuration et synchronisation du processus

La source et la destination de chaque module sont programmées selon le processus à suivre. Chaque module contient, à cet effet, un registre indiquant quels sont les bus concernés. Ce registre peut être modifié à tout moment par le processeur pilote. Le logiciel doit, par conséquent, être capable de gérer l'occupation des bus pour l'ensemble des processus envisageables. Cette gestion soulève un certain nombre de problèmes délicats et constitue l'un de nos axes de recherche actuels.

Un autre problème crucial est celui de la synchronisation des informations. Il n'est plus possible ici d'introduire un module de retard fixe puisque la durée et la localisation de ce retard dépendent du processus de synthèse. Nous avons résolu ce problème en associant à chaque bus une ligne indiquant la présence de données valides sur le bus ce qui correspond à la partie visible d'une ligne vidéo. Ce signal est répercuté sur le bus de sortie d'un module au même titre que les données et avec le même retard. Lorsqu'un module de calcul nécessite la synchronisation de deux informations on adjoint à chaque bus d'entrée une file d'attente asynchrone intégrée dont le remplissage est commandé par le signal de validité du bus. Les deux fils d'attente sont ensuite explorés simultanément pour effectuer les calculs.

À titre d'illustration la figure 5 schématise la configuration nécessaire au processus de la figure 3. Quatre bus banalisés seulement ont été représentés. Ces mêmes modules permettent quantités de configurations allant du terminal classique de la figure 1 au synthétiseur évolué comportant des possibilités d'interaction en temps réel sur les couleurs, sur l'éclairage ou incluant l'élimination des parties cachées à travers l'algorithme du Z-buffer.



VERS UNE ARCHITECTURE BANALISÉE DES SYNTHÉTISEURS D'IMAGE
An approach to a general purpose architecture of image synthesizers
Francis MARTINEZ

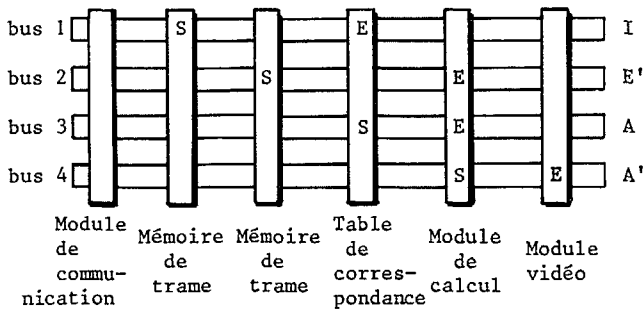


Figure 6

5. CONCLUSION

Le prototype en cours de réalisation est un système expérimental destiné à valider la faisabilité des concepts proposés. Les modules disponibles à l'heure actuelle outre les modules vidéo et communication sont : deux mémoires de trames de $512 \times 512 \times 12$ bits avec translation, fenêtrage et possibilité d'invalider une ou plusieurs informations en temps réel ; deux tables de correspondance (4096×12 bits) et un module de calcul d'éclairage. D'autres modules déjà réalisés lors d'une étude précédente [Fer 81] [MaF 82] seront adaptés à cette nouvelle architecture. Il s'agit d'un module de calcul de textures et d'un module permettant le calcul des coefficients de réflexion spéculaire et diffuse en fonction du vecteur normal pour chacun des pixels.

Muni de ces modules ce système permettra de répondre à un grand nombre d'applications allant du graphique bas de gamme jusqu'à la synthèse en temps réel d'images réalistes. L'adaptation à des applications très spécifiques pourra également être réalisée par la simple adjonction d'un module de calcul dédié à cet usage.

Nous pensons que ce type d'architecture peut conduire dans un avenir proche à une unification des synthétiseurs d'images autour d'un bus standard tel que celui qui est proposé dans ces pages. L'utilisateur pourrait alors construire, de manière modulaire, le synthétiseur répondant à ses besoins propres à l'aide des "cartes modules" proposées par différents constructeurs.

REFERENCES

- Fer 81 : F. Nunes Ferreira
Conception et réalisation d'un système interactif pour la synthèse d'images réalistes : HELIOS.
Thèse de Docteur-Ingénieur, Grenoble, 9-81.
- MaF 82 : F. Martinez, F. Nunes Ferreira
HELIOS : Terminal vidéo interactif pour la synthèse d'images réalistes.
Le Nouvel Automatismes, N° 30, 5-82.
- Mar 82 : F. Martinez
Vers une approche systématique de la synthèse d'image : Aspects logiciel et matériel.
Thèse d'Etat, Grenoble, 11-82.