

# CA CREATIVE ALGORITHM

STÉPHANE DE GÉRANDO

RÉSUMÉ. CA - *Creative Algorithm* – est une recherche à la fois scientifique, technologique et artistique, devenant plus tard un projet de composition algorithmique pour la création d'un nouveau satellite du *Labyrinthe du temps* ou actuellement (2023) partiellement intégré à l'algorithme du grand cycle du LDT. Dans CA réside l'idée (classique) d'un tout unitaire polysensoriel émergeant d'un microcosme, ici à l'image d'« impulsions élémentaires », quantum de temps ou naissance de lignes de temps verticalement synchronisées /désynchronisées permettant de contrôler des polymorphies intriquées sonores, visuelles. La mise en relation commune de formalismes aussi bien probabilistes que déterministes est au centre de la génération des données du contrôle de la synthèse ou de programmes de transformation du matériau. Sous la direction scientifique de Stéphane de Gérando, la programmation de CA débute en juin 2007 en partenariat avec l'Ecole Nationale Supérieure d'Electronique, Informatique de Bordeaux et l'Institut International pour l'Innovation la Création Artistique et la Recherche. Revenons sur cette première phase de développement, avec à l'époque, une description des enjeux scientifiques et artistiques, une présentation générale des programmes informatiques (synthétiseurs puis processus de contrôle de la synthèse) et un premier bilan critique. Cette période de 2007 annonce l'émergence du *Labyrinthe du temps*, avec 16 années plus tard une des problématiques les plus actuelles qui reste à développer, la complexité de joindre l'invention algorithmique du microcosme et du macrosome d'une œuvre, dans une forme de continuité /discontinuité assumée.

## 1. INTRODUCTION

### Une dimension diachronique

Le projet CA s'inscrit dans une perspective diachronique et critique. L'art virtuel ou numérique paraît à la préhistoire d'une évolution. C'est au début des années 1960 que l'on commence à utiliser l'ordinateur à des fins artistiques. Les laboratoires Bell dans le New Jersey aux Etats-Unis regroupent par exemple différentes orientations de recherche, comme l'art numérique avec A. Michael Noll, l'animation numérique avec Edward E. Zajac, Frank Sinden et Kenneth C. Knowlton et la musique par ordinateur avec Max V. Mathews et John R. Pierce. Michael A. Noll exposera avec les Allemands Gerg Nees et Frieder Nake leurs premières œuvres lors de l'exposition « Computer-Generated Pictures » à la galerie Howard Wise de New York. Avec le développement de la vidéo ou des satellites de communication dans les années 1970, les artistes s'intéressent de plus en plus aux performances technologiques en direct et aux réseaux (1977, Douglas Davis, *The Las 9 Minutes*), puis un travail plus spécifique sur l'imagerie numérique avec les techniques d'acquisition, de simulation... jusqu'aux années 1980 rendant plus accessible l'utilisation grand public de l'informatique, avec le développement des micro-ordinateurs (IBM PC en 1981, les Macintosh en 1984), des logiciels de graphisme et de 3D. C'est aussi la période du développement de l'interactivité temps réel ou du tout algorithmique. Ces techniques se stabilisent dans les années 1990 avec une percée d'Internet et le World Wide Web en 1994. Le marché de l'art numérique se développe à partir de 2001, avec des galeries commerciales telles que le Walker Art Center ou le Whitney Museum of American Art ou des expositions comme en Allemagne Natürlichkünstlich (naturellement artificiel) (2001, Rostock, Mannheim et Berlin). La culture multimédia a cependant toujours aujourd'hui des difficultés à exister en dehors de ses réseaux spécifiques. Certains pays comme le Japon ou les Etats-Unis semblent être à une période charnière, profitant notamment du développement de cursus universitaires « appliqués » dans le domaine artistique avec des laboratoires de recherche dans le domaine du virtuel associé à l'Art. Le rapport Art-Science-Technologie de Jean-Claude Risset explique la naissance de ce contexte, rappelant le rôle en France du Ministère de la Culture pour tenter de financer certaines recherches, ce ministère se heurtant par ailleurs à des difficultés concernant structures et statuts<sup>1</sup>. Par conséquent, le sentiment d'une culture parallèle s'est développé, avec fraîcheur parfois et la naïveté de la découverte. L'évolution des outils technologiques grand public, les champs expérimentaux, le lien avec d'autres disciplines, la mixité des genres et des moyens d'expression,

---

<sup>1</sup> Jean-Claude Risset, Rapport Art-Science-Technologie, Ministère de la Culture, mars 1998.

autant de facteurs qui favorisent de nouvelles pratiques.

On tente modestement dans le cadre de ce projet de retrouver certaines filiations, de réfléchir au sujet des liens entre histoire, outils techniques et imaginaires, de proposer des réflexions théoriques toujours en relation avec la question artistique.

Prenons un exemple délicat, le cas de l'étude de la relation entre la fréquence d'un son et de la lumière, sujet qui intéresse nécessairement l'art virtuel poly-sensoriel. Naturellement dans le domaine des sciences, le sujet a été traité, d'Aristote à Kepler, du père Mersenne à Descartes, Huyghens, Hooke, Newton, Fresnel... Dans le domaine artistique, au XVI<sup>e</sup> siècle, Arcimboldo aurait créé un clavecin de couleurs partant des proportions harmoniques pythagoriciennes notamment. C'est le début d'une longue liste d'instruments tentant de lier les fréquences d'un son et de la lumière, du clavecin oculaire<sup>2</sup> jusqu'à des systèmes comme le Reflectodyne, Chromara, Vampire... Pourtant, malgré ces travaux à travers les siècles, il ne semble pas actuellement y avoir de consensus scientifico-artistique définitif et clair à ce sujet. Dans le prolongement de cet exemple, les solutions ponctuelles selon les expérimentations ou les personnalités qui les proposent donnent toujours aujourd'hui le sentiment d'être à l'époque des pionniers. Le manque de résultat scientifique en lien avec la création artistique pose une véritable difficulté. Par ailleurs, les laboratoires de recherche en réalité virtuelle n'ont pas toujours les moyens et le temps d'approfondir ces questions. De manière plus générale, les liens entre science et art restent délicats, appartenant à des logiques, des processus de validation, des cultures institutionnelles, sociologiques et économiques différentes, l'enjeu étant la communication de ces logiques sans les annuler, dans un esprit à la fois de rigueur et d'ouverture.

### **L'origine de CA**

Collaborant en 2006 avec un artiste peintre, Pierre Chaveau<sup>3</sup>, nous avons travaillé sur une version multimédia de *Binaurale*, à l'origine une œuvre électronique. Une des particularités du travail de Pierre Chaveau est de créer des peintures en mouvement, grâce à la projection de couleurs primaires sur ses toiles. Il parle lui-même de partition rythmique qui contrôle l'intensité des trois faisceaux lumineux permettant une recomposition du spectre chromatique des couleurs grâce à la synthèse additive : alors que la toile est peinte sur un support « traditionnel », ses formes et couleurs se métamorphosent suivant l'éclairage projeté. Dans le cadre de *Binaurale*, il a réalisé une peinture à partir de la forme d'onde de l'œuvre électronique, puis nous avons réalisé une vidéo à partir du travail du peintre, le tout créant une installation à trois dimensions simultanées avec la musique. Par la suite, nous avons composé *Blue bird* œuvre multimédia créée au Musée d'Aquitaine à Bordeaux et réalisée pour une projection monumentale, en extérieur et en intérieur. Au même moment, nous commençons à travailler sur le concept de tableau virtuel, parfois même sans musique, image d'un objet à déformation lente jouant sur les seuils différentiels de perception.

### **Les premiers objectifs liés au développement de CA**

Décrivons trois objectifs qui ont guidé les premières étapes de la programmation de CA.

1. Un positionnement scientifique, artistique et technologique  
CA se nourrit d'années de pratique de la création et de la recherche à la fois esthétiques et technologiques, aussi bien sur des questions d'évolution de langage, d'utilisation de certains concepts en relation avec l'invention et la perception d'une œuvre ou le choix de certaines techniques, comme celle qui permet d'inventer le son ou l'image « de l'intérieur » (modèle additif), sans rapport à des modèles prédéterminés (cf. liste de nos

---

<sup>2</sup> oil-lamp device, colored-liquid clavichord, Pyrophone, color organ, vacuum-tube Instrument, sound-color, Chromola, Clavilux, Sarabet, Piano Optophonique, optophone, Sonchromatoscope, colour Keyboard, Chromatophon, Lichtrequisit, Kinetic Light Machine, color projector, Musicolour Machine, Musiscope, Vortex Concerts, Sonovision, Lumidyne.

<sup>3</sup> Diplômé ès-Lettres, Arts Plastiques et Sciences de l'art de la Sorbonne et de l'École Nationale Supérieure des Beaux-Arts de Paris (ENSBA), Pierre Chaveau déploie son activité comme peintre, illustrateur et auteur. En 1990, il invente l'« écriture lumière » qui explore le lien organique entre la couleur et la lumière et interroge les rapports avec le texte, la danse et particulièrement la musique. Depuis 1971, son œuvre a été régulièrement exposée en Europe, aux Etats-Unis et en Extrême-Orient.

publications et réalisations artistiques). Il peut être intéressant de lire certains de nos travaux pour mieux comprendre les objectifs fixés ou les choix opérés dans le cadre de CA.

2. Un processus unique d'engendrement de l'œuvre  
 La problématique originelle est d'inventer un même processus de création du son, de l'image [...], une sorte d'ADN unique support de l'information pluri-sensorielle de l'œuvre, un « élément » simple ou fondamental propre à engendrer de la complexité, du microcosme au macrocosme de CA.  
 Il ne s'agit par exemple en aucun cas d'assujettir et de hiérarchiser l'invention de l'image en rapport avec le son ou inversement, comme historiquement on en a globalement l'habitude dans le domaine artistique.  
 Unicité, matrice biologique d'un organisme vivant, « théorie du tout » liant échelle quantique et cosmologique, naissance de l'univers etc., autant d'images qui nous renvoient d'une certaine manière au mythe de l'unité par opposition au thème de la pluralité, signification sociale, réunification du temporel et du spirituel, expression d'un déterminisme logique et culturel et problématique d'un art total.
3. Une œuvre autonome, éphémère, sans trajectoire déterminée ni début ni fin, toujours différente  
 Si le programme est définitif en fonction de chaque étape de son évolution, l'œuvre n'est quant à elle jamais la même. A chaque évaluation du programme, CA se veut en effet différente. L'équilibre des processus statistiques et déterministes et l'auto-modification des paramètres de contrôle de la synthèse permettent d'exprimer la permanence d'un temps présent, sans passé ni avenir, chaque instant de l'œuvre s'effaçant à jamais, incarnation du « temps réel » dont la trajectoire se voudrait idéalement chaque fois imprévisible.

## 2. DEVELOPPEMENT ET PROGRAMMATION

Dans cette partie, après une brève introduction au sujet de MAX-MSP-JITTER et de l'interface utilisateur de CA, nous présentons la programmation des synthétiseurs image et son puis les procédures de contrôle de la synthèse avec quatre principaux programmes : le cœur du système – l'invention et le contrôle du temps (1), la synchronisation /désynchronisation des paramètres de contrôle de la synthèse (2), le processus itératif d'auto-modification des tables (tirage probabiliste de grandeurs numériques) (3) et le régulateur (4).

### Le choix de l'environnement MAX-MSP-JITTER

La programmation sera réalisée dans l'environnement MAX-MSP-JITTER. L'architecture de l'environnement lui-même et les bibliothèques associées et la programmation graphique par module fonctionnel font de MAX l'un des plus puissants logiciels actuels pour ce type de développement multimédia temps réel (installations, œuvres interactives...). Cette programmation peut faire l'économie d'une écriture de lignes de code grâce à l'association d'objets graphiques dont l'ordre de droite à gauche et de haut en bas dirige l'algorithme. Ces objets associés par des connexions<sup>4</sup> se déclenchent par la réception de « bang » ou de « click », régie par un contrôle du temps<sup>5</sup>, créant ainsi un processus séquentiel à la base des algorithmes. Ces mêmes objets nécessitent des arguments<sup>6</sup> soit définis directement après le nom de l'objet, soit en les définissant aux entrées<sup>7</sup> de cet objet, et possèdent également une ou plusieurs sorties<sup>6</sup>.

<sup>4</sup> Voir annexe pour les fils de connexion de l'interface utilisateur.

<sup>5</sup> Ici on parlera plus de durées (temps existant entre deux clicks exprimé en milliseconde).

<sup>6</sup> Bang, entiers, flottants, listes, booléens, ...

<sup>7</sup> Un objet peut posséder plusieurs entrées (1, 2, 5, ...) chacune ayant un rôle bien défini, certaines étant même facultatives.

## CA et son interface utilisateur

Pour l'interface utilisateur, nous avons choisi de présenter CA en trois zones distinctes : une première zone dédiée à la synthèse sonore, une deuxième consacrée à la synthèse de l'image et une troisième réunissant les programmes de contrôle de l'image et du son.

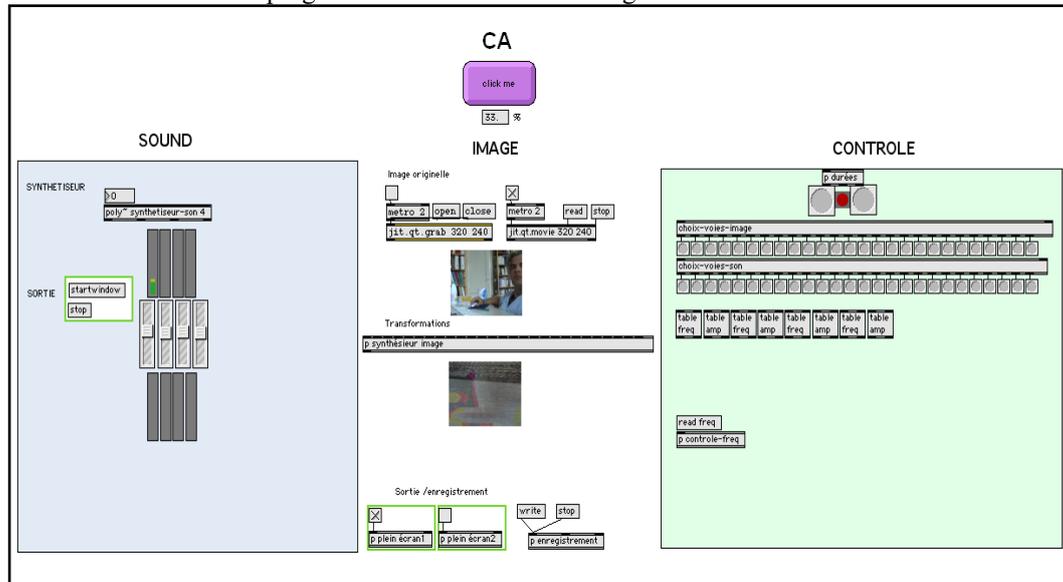


Figure 1 : Interface utilisateur de CA réalisée dans Max-MSP-JITTER

Lorsque CA est évalué (bouton « click me »), des durées sont créées à partir d'un « cœur rythmique » qui délivre des « bangs » à l'origine de l'organisation de l'ensemble des paramètres de contrôle de la synthèse image et son. Nous y reviendrons par la suite.

### Les synthétiseurs image et son

Pour répondre au concept d'unité, notre volonté est à terme d'utiliser la synthèse additive de la lumière et du son en créant des analogies pour le contrôle de ces deux synthétiseurs grâce à l'utilisation d'un même type de modèle de synthèse. Mais lors des premiers tests concernant un contrôle systématique de chaque pixel à travers un mixage des couleurs sur l'ensemble de la matrice, nous avons été confrontés à des problèmes de calcul différant l'utilisation du modèle additif dans le cadre de l'image.

#### 1. Le synthétiseur son

Nous souhaitons utiliser nos précédentes recherches dans le cadre de la synthèse sonore afin de réaliser un synthétiseur polyphonique hybride, sur la base d'un modèle additif lié à d'autres techniques de synthèse comme des modulations (amplitudes, fréquences, phases), des programmes de traitement (filtrages, réverbération, spatialisation...), des synthèses par formes d'onde mixées... Il s'agit ici de contrôler 16 paramètres pour chaque oscillateur de la synthèse additive (sinusoïdale ou autres).

##### Oscillateur de base

- p1 : forme d'onde
- p2 : amplitude (avec enveloppe)
- p3 : phase

##### Modulation de fréquence

- p4 : fréquence du vibrato
- p5 : amplitude du vibrato
- p6 : fréquence du jitter
- p7 : amplitude du jitter

##### Modulation d'amplitude

- p8 : fréquence du vibrato
- p9 : amplitude du vibrato
- p10 : coefficient d'amplitude
- p11 : fréquence du jitter

p12 : amplitude du jitter  
 p13 : coefficient d'amplitude  
 Réverbération  
 p14: durée réverbération  
 p15 : mixage  
 Spatialisation  
 p16 : panoramique

Si l'on compte par exemple 50 oscillateurs pour un type de son, cela revient à devoir produire 800 paramètres dans des durées très courtes, en deçà de 20 millisecondes pour un contrôle granulaire du son. Voici le sous-programme représentant ce synthétiseur avec la liste des paramètres de contrôle sur chaque oscillateur :

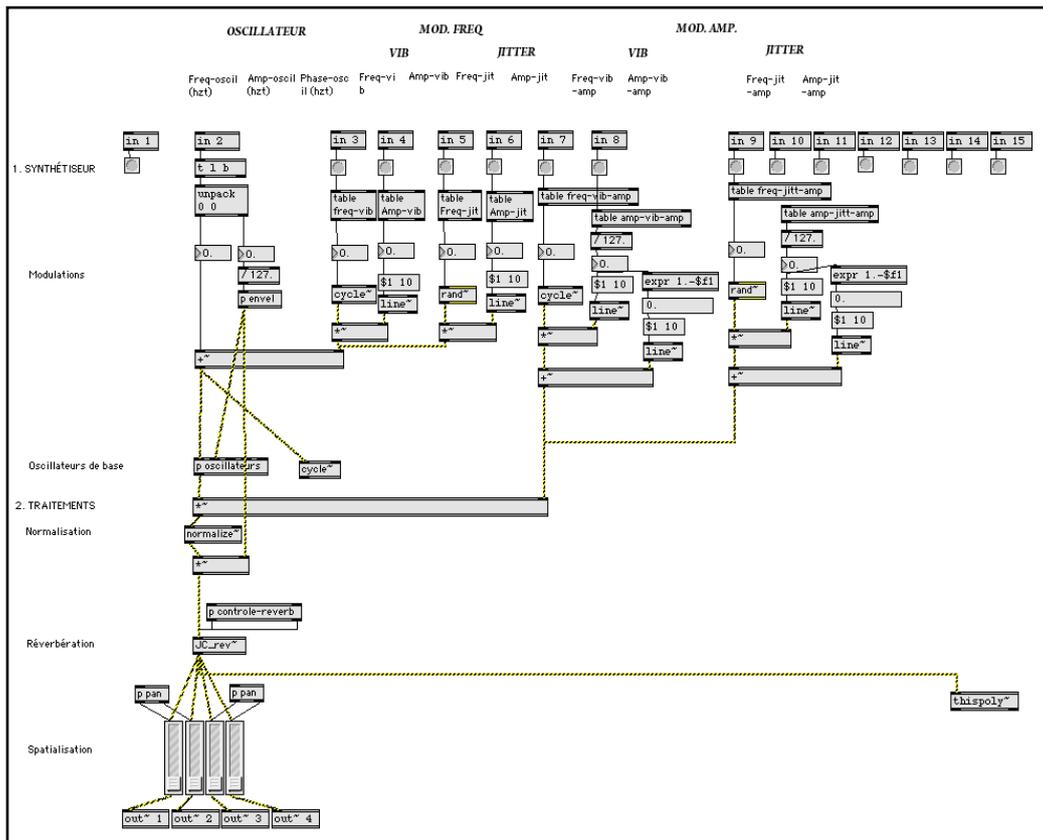


Figure 2 : Synthétiseur sonore hybride temps réel sur la base du modèle additif

## 2. Le synthétiseur image

Dans l'attente de l'implémentation d'un synthétiseur additif, le choix s'est porté sur la visualisation filmée et transformée du public en des tableaux virtuels abstraits. L'image du visiteur se trouve reflétée dans l'œuvre, les gestes déformant les formes du tableau devenu interactif.

Cinq programmes participaient à cette modification de l'image : un fractionnement et déplacement en cellule basés sur la luminance<sup>8</sup>, un effet de modification du taux d'échantillonnage sur l'image<sup>9</sup>, une combinaison de matrices<sup>10</sup> et une spatialisation associée à un mixage automatisé de ces différents traitements<sup>11</sup>.

<sup>8</sup> *Jit.plume* prend deux matrices en entrée, analyse chaque cellule de la seconde matrice et, se fondant sur la valeur moyenne de ces dimensions, déplace la cellule correspondante dans la première matrice.

<sup>9</sup> *Jit.ameba* (nom du programme) sature de l'image par une augmentation ou une diminution du taux d'échantillonnage spatial.

<sup>10</sup> *Jit.eclipse* combine deux matrices à quatre dimensions (canal alpha, rouge, vert, bleu) en entrées selon des modes différents d'interaction.

<sup>11</sup> A partir de deux matrices, *Jit.mxform2d* produit une seule et même matrice. Un second objet *Jit.xfade* mixe ces deux matrices.

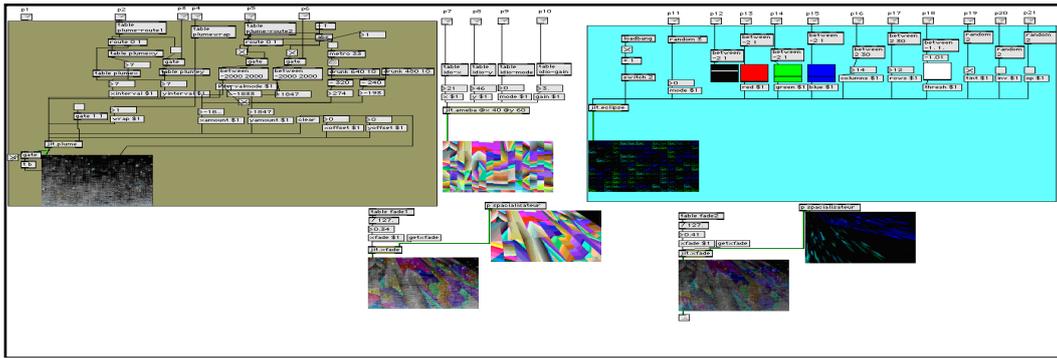


Figure 3 : Programme du synthétiseur image réalisé dans Max /MSP /JITTER

1. Fractionnement et déplacement de l'image en cellule basé sur la luminance  
Cet objet<sup>12</sup> prend deux matrices en entrée, analyse chaque cellule de la seconde matrice et, se basant sur la valeur moyenne de ces dimensions, déplace la cellule correspondante dans la première matrice.
2. Effet d'une modification du taux d'échantillonnage sur une image<sup>13</sup>  
Cet objet<sup>14</sup> joue sur la saturation de l'image (un mode définit comment la saturation va être manipulée) par une augmentation ou une diminution du taux d'échantillonnage spatial.
3. Combinaison de matrices  
Cet objet<sup>15</sup> prend deux matrices à quatre dimensions (canal alpha, rouge, vert, bleu) en entrées et les combine selon un mode de combinaison différent.
4. Spatialisation et mixage  
Un premier objet<sup>16</sup> permet la combinaison de deux transformations différentes afin de créer une seule et même matrice. Un second objet<sup>17</sup> mixe ces deux matrices, à savoir quelle transformation sera plus visible que l'autre (un pourcentage d'apparition sur les deux matrices est donné).

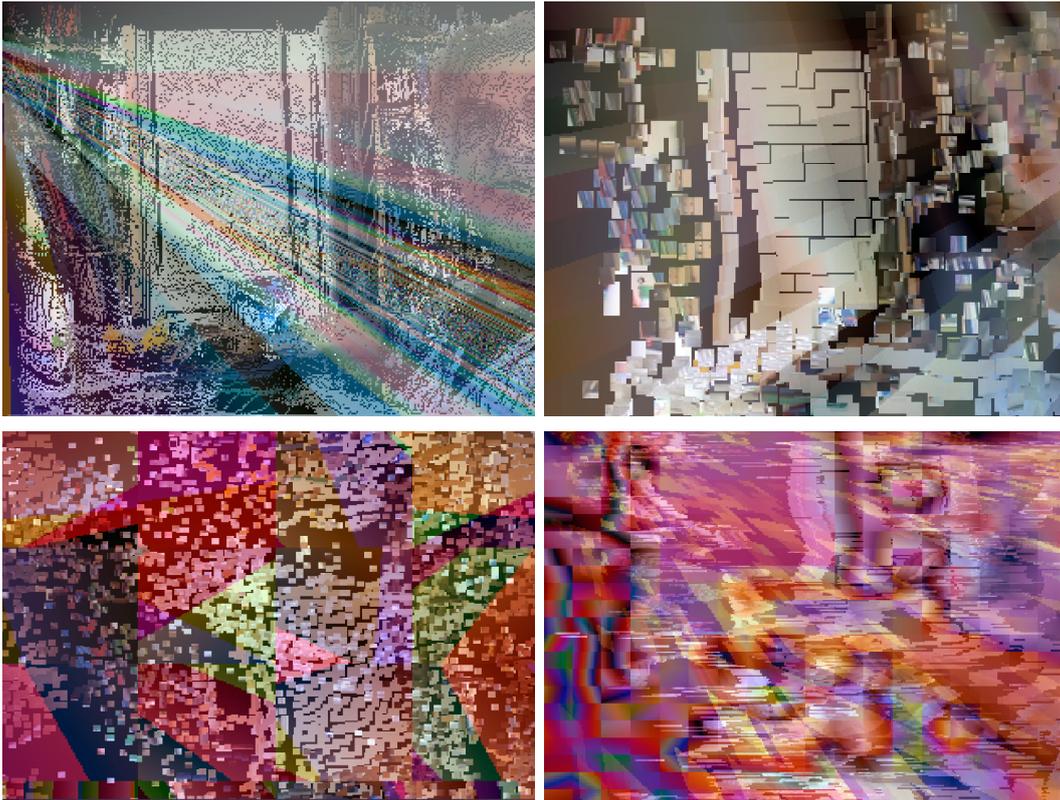


Figure 4 : Quatre exemples de résultat du programme image : le public est filmé et son image est déformée en tableaux virtuels

<sup>12</sup> Jit.plume.

<sup>13</sup> « upsampling » ou « downsampling ».

<sup>14</sup> Jit.ameba.

<sup>15</sup> Jit.eclipse.

<sup>16</sup> Jit.mxform2d.

<sup>17</sup> Jit.xfade.

## Le contrôle de la synthèse

Décrivons la programmation des processus de contrôle de ces deux synthétiseurs.

Pour répondre à nos problématiques esthétiques initiales, nous ne cherchons pas à développer une « combinatoire de l'écriture », principe qui ferait appel « de manière classique » à l'émergence de cellules thématiques et ses transformations, structuration très souvent usitée de la mémoire dans l'histoire de l'art et en particulier de la musique savante occidentale.

A ce stade de la réflexion, une absence de mémoire prévisible du matériau n'est pas problématique bien au contraire : rappelons que nous cherchons « simplement » à percevoir un matériau diversifié, sans direction début ni fin, une matière en mouvement toujours différente.

### Principe général lié au contrôle de la synthèse

La problématique est d'inventer un premier processus de contrôle et de création du son et de l'image qui soit suffisamment généraliste et puissant pour être appliqué à tous les paramètres de la synthèse quel que soit le type de modèle utilisé.

Pour contrôler l'ensemble des paramètres de la synthèse poly-sensorielle (dans ce cas les deux synthétiseurs image et son), le principe est de générer une succession diversifiée de clicks dans le temps.

Puis à partir d'un processus de synchronisation /désynchronisation de ces clicks initiaux, nous créons une partition polyrythmique qui fera vivre et contrôlera l'ensemble du système.

### Premier programme de contrôle : le cœur du système - invention et le contrôle du temps, encapsulation « p durées »



Figure 5 : Système de contrôle du temps dans Max-MSP-JITTER

Le premier programme « p durées » produit une succession de durées, à partir d'un contrôle probabiliste d'échelles de durée qui oscillera en définitive entre la perception d'une régularité (temps pulsé) et de durées irrégulières (temps lisse). Puis ces clicks sont distribués pour synchroniser ou désynchroniser la synthèse du son et de l'image.

### Présentation générale du programme « p durées »

Voici résumé les instructions de ce premier programme :

1. Tirage probabiliste de durées à partir d'une échelle de durée choisie (modifiable graphiquement en temps réel).
2. Sélection entre :
  - a. Conservation de la durée tirée
  - b. Subdivision de la durée
    - i. Subdivision irrégulière
    - ii. Subdivision régulière avec un coefficient de distorsion
3. Répartition probabiliste des clicks entre trois situations : son /son-image /image (respectivement 0 ou 2 ou 1)

Les dix paramètres de ce premier programme montrent une omniprésence de processus probabilistes pour tenter de répondre aux choix esthétiques de l'œuvre.

Tirage pseudo-aléatoire de durée selon des probabilités d'apparition (sorte de métronome pseudo-aléatoire)

P1 : échelle de durée

P2 : probabilité d'appariation

Subdivision éventuelle des durées selon deux modèles : division irrégulière /division régulière avec distorsion

P3 : durée subdivisée ou non

p4 : durée subdivisée irrégulièrement ou régulièrement (0 ou 1) (table « reg-irreg »)

p5 : nombre maximum de subdivisions des durées aléatoires

p6 : Probabilité(s) de tirage du nombre(s) de subdivisions irrégulières (table « pulse-irreg »)

p7 : nombre maximum de subdivisions des durées aléatoires initiales

p8 : Probabilité(s) de tirage du nombre(s) de subdivision régulières (table « pulse-reg »)

p9 : taux de distorsion de la durée subdivisée en %

Répartition probabiliste des bangs selon trois situations : 1 - son /2 image et son /3 image

p10 : Probabilité(s) de tirage « son-image » : répartition des clicks entre 3 situations : son image son/image (respectivement 0 ou 2 ou 1)

## L'algorithme

Voici la description de l'algorithme principal<sup>18</sup> puis des deux sous-programmes de subdivisions.

```
Entier x, tmp, z, y, tmp2, x1, p ;
DEBUT
x = table(durs-initiales) ; // tirage de la durée de référence
metro(x) ; // on lance le métronome
tmp = table(durdiv) ; // tirage de la subdivision ou non
si (tmp == 0) // on ne subdivise pas la durée
    z = table(son-image) ; // tirage du type de paramètres
// modifiés
// on va subdiviser la durée
si (tmp == 1)
    faire {
        y = table(reg - irreg) ; // tirage du type de subdivision
        si (y == 0) // subdivision irrégulière
            faire {
                tmp2 = table(pulse - irreg) + 1 ; // tirage du nombre de
                // subdivisions
                pulse - irreg(x, tmp2) ; // fonction qui subdivise
                z = table(son - image) ; // tirage du type de
                // paramètres modifiés
            }
        Si (y == 1) // subdivision régulière
            faire {
                tmp2 = table(pulse - reg) + 1 ; // tirage du nombre
                // maximal de subdivisions
                x1 = x / tmp2 ; // durée de la subdivision
                p = table(distor) ; // tirage du taux de
                // distorsion appliqué à la
                // durée de la subdivision
                pulse - reg() ; // fonction qui subdivise
                z = table(son - image) ; // tirage du type de
                // paramètres modifiés
            }
    }
si (z == 0) // paramètres du son modifiés
    choix - voies - son() ; // programme II.5
si (z == 1) // paramètres du son et de l'image
    // modifiés
    faire {
        choix - voies - son() ; // programme II.5
        choix - voies - image() ; // programme II.5
    }
si (z == 2) // paramètres de l'image modifiés
    choix - voies - images() ; // programme II.5
FIN
```

Ci-dessous la présentation de l'algorithme dans MAX.

<sup>18</sup> Description des algorithmes par Tony Héron, Rapport de stage ENSEIRB : « Développement de programmes de contrôle temps réel du son et de l'image dans le cadre du projet CA – Creative Algorithm –, stage encadré par Stéphane de Gérardo.



Ci-dessous la présentation de l'algorithme dans MAX.

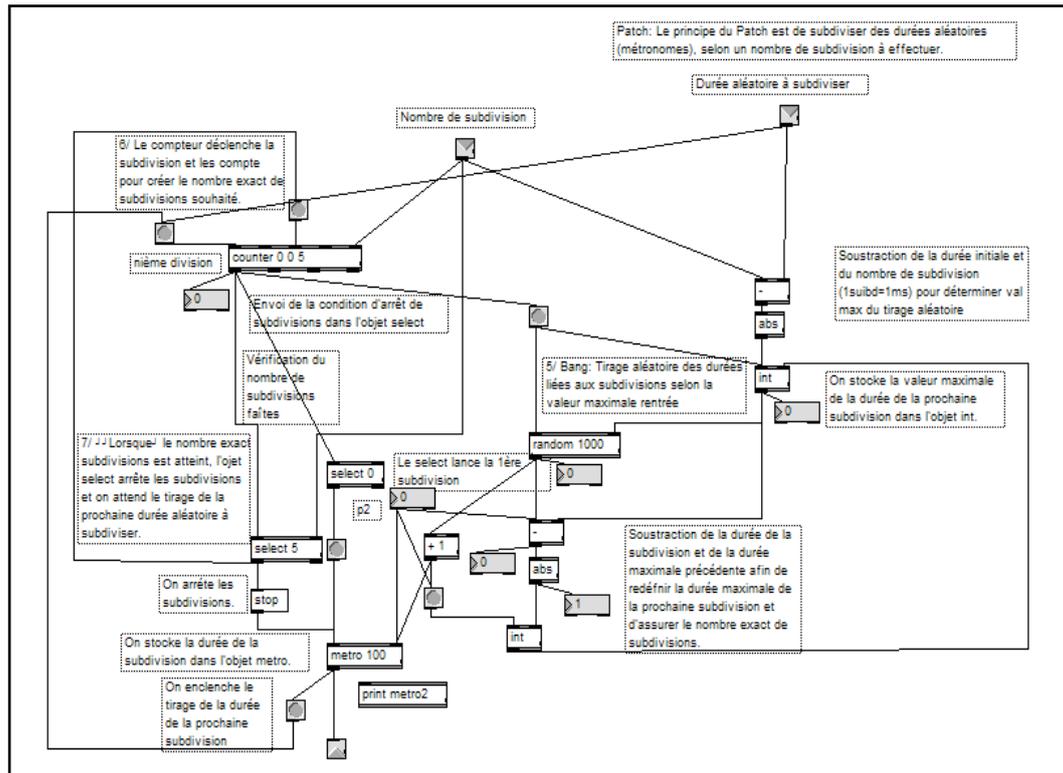


Figure 7 : Sous Programme de subdivision irrégulière des durées de référence

## 2. La subdivision régulière

On divise la durée de référence par une constante tirée aléatoirement dans une table (si on divise par 7, on aura au maximum 7 subdivisions). On effectue ensuite une distorsion sur la sous – durée engendrée. Cet algorithme prend donc en paramètres un taux maximal de distorsion, un nombre maximal de subdivision et la durée aléatoire à subdiviser. Un sous programme de distorsion a été créé. Le principe consiste à appliquer un taux de distorsion sur la sous – durée et ensuite de faire un tirage entre un minimum et un maximum créée<sup>19</sup>.

```

Bang pulse-reg (entier taux, entier n, entier d)
Flottant x ;
Entier y, z, taux ;
DEBUT
x = d / n ;
y = rand-distor1(taux, x) ;
z = d - y ;

tant que (z - y > 0)
    faire {
        metro(y) ;
        y = rand-distor1(taux, n, d) ;
        z = z - y ;
    }

Return bang ;
FIN
  
```

*// sous – durée créée  
// application d'une distorsion  
// on stocke le temps qui devrait  
// rester  
// Si il reste du temps on continue  
// de subdiviser, sinon on arrête  
// on lance le métronome  
// on calcule la prochaine sous  
// durée  
// préparation du test de la boucle*

Cet algorithme fait appel à « rand-distor1 » :

```

Entier rand-distor1(entier taux, flottant x) ;

Flottant x_min, x_max ;
DEBUT
x_min = x - (x / 100) * (taux / 2) ;
x_max = x + (x / 100) * (taux / 2) ;
y = E[between(x_min, x_max)]

return y ;
FIN
  
```

*// minimum possible de la sous – durée  
// maximum possible de la sous – durée  
// fonction qui fait un triage aléatoire  
// entre x\_min et x\_max*

<sup>19</sup> Un taux de 10% sur la valeur 30 ms engendre un tirage aléatoire entre 27 ms et 33 ms.

Ci-dessous la présentation de l'algorithme dans MAX.

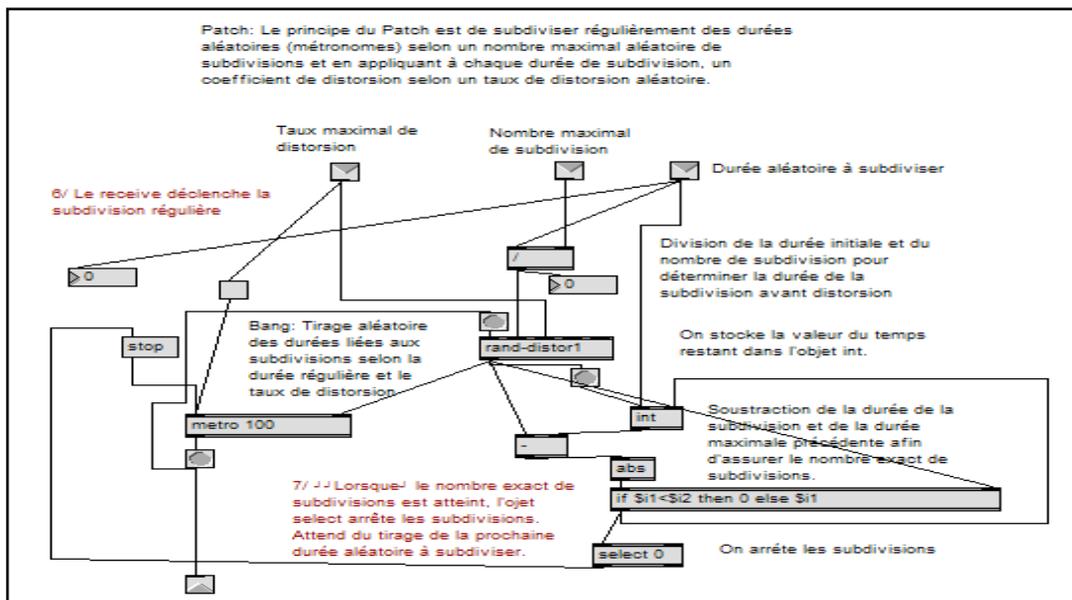


Figure 8 : Sous-programme de subdivision régulière des durées de référence

Ces bangs sont ensuite envoyés vers un programme qui va gérer la synchronisation /désynchronisation des paramètres de contrôle.

### Deuxième programme de contrôle : synchronisation /désynchronisation des paramètres de contrôle (encapsulation « choix-voies-image » / « choix-voies-son »)

Ce nouvel algorithme définit la distribution polyrythmique des clics.

#### Présentation générale

Le principe de synchronisation /désynchronisation est au centre du contrôle rythmique de la synthèse de l'œuvre, inspiré d'une technique compositionnelle permettant d'engendrer une polyrythmie (structure verticale) à partir d'une voie rythmique unique (structure horizontale). Les bangs générés au cours du temps sont distribués entre plusieurs « voies » symbolisant chacune un paramètre de contrôle. Cette synchronisation /désynchronisation a pour objectif de créer ou non des relations entre la synthèse du son et de l'image.

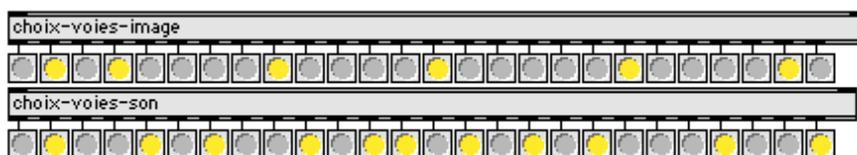


Figure 9 : Système de synchronisation /désynchronisation des voies

#### L'algorithme

A partir du nombre de paramètres mis en jeu, un tirage aléatoire (principe des tables de probabilité) est déclenché à chaque bang<sup>20</sup> sur le nombre de paramètres (ou voies) qui vont être

<sup>20</sup> Les bang sont ceux qui proviennent du programme précédent (son pour les paramètres du son, image pour les paramètres image et son /image pour tous les paramètres).

synchronisés en sortie. Une deuxième table de probabilité est présente afin d'éviter qu'une voie soit choisit plusieurs fois<sup>21</sup>.

L'algorithme détaillé pour les paramètres de l'image est le suivant<sup>22</sup> (il est similaire à l'algorithme pour les paramètres du son).

```

Tableau nombre-voies-image[25]; //représente le nombre de voies à
//synchroniser (de 1 à 25)
Tableau repeat[25]; //contient les probabilités de tirage
//affectées à chaque voie
Tableau ref[25]; //table temporaire assurant qu'une voie
//ne soit pas tirée deux fois (x voies
//tirées différentes)

Entier i, j, x ;

DEBUT
Pour i allant de 0 à taille(repeat) - 1
    ref[i] = repeat[i]; //on travaille sur la table ref
x = tirage(nombre-voies-image) + 1; //un bang sur la table envoie une valeur
//en sortie selon une probabilité de tirage
//ici on tire le nombre de voies qui vont
//être synchronisées

Pour j allant de 1 à x
Faire {
    y = tirage(ref) + 1; //on tire la voie qui va être synchronisée
    ref(y) = 0; //on met la probabilité de la valeur tirée à
//0 afin d'assurer qu'elle ne soit pas
//retirée
    afficher(y); //on affiche la valeur de la voie tirée
    return y; //on envoie le bang sur la voie associée
}
FIN
  
```

L'algorithme implémenté dans Max Msp Jitter est le suivant.

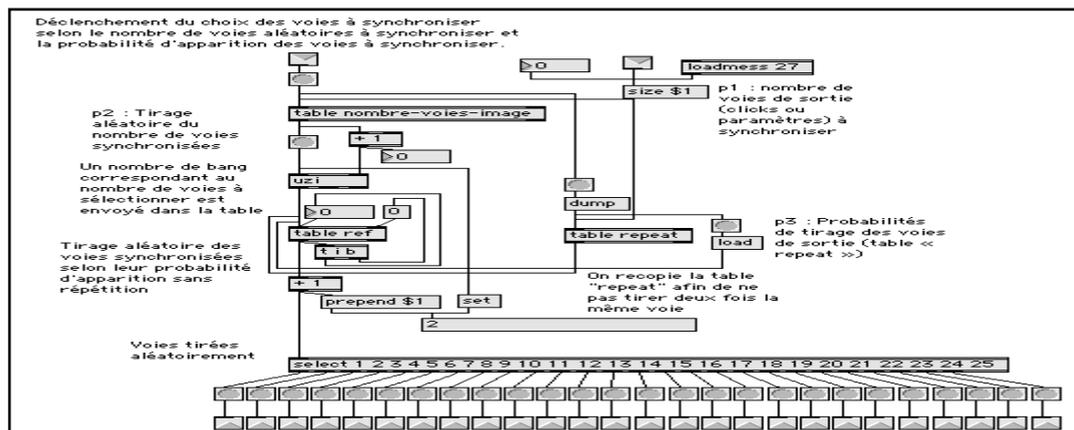


Figure 10 : Programme de synchronisation des paramètres de CA dans MAX-MSP-JITTER

Si le contrôle temporel de relations entre paramètres devient effectif grâce à ce programme, les phases rythmiques sont trop redondantes, les échelles de paramètres étant les mêmes, entraînant une monotonie visuelle et sonore.

### Troisième programme de contrôle : processus itératif d'auto-modification des tables (tirage probabiliste de grandeurs numériques)

Pour que l'œuvre se renouvelle dans le temps, il est donc nécessaire de trouver un processus faisant évoluer les données du programme.

### Présentation générale

La question était ici d'envisager une auto-modification des paramètres de sélection des données liées aux tables pour que l'œuvre puisse se renouveler dans la durée.

<sup>21</sup> A chaque fois qu'une voie est choisie, on met sa probabilité de tirage à 0, ainsi elle n'est plus choisie tant qu'un autre bang n'est pas généré.

<sup>22</sup> Le paramètre 25 dans l'algorithme correspond aux 25 voies (ou paramètres) en sortie de l'algorithme.

Ce programme va symboliser « l'intelligence » du système, suffisamment généraliste pour être complexifié puis appliqué sur lui-même<sup>23</sup> et au contrôle de tous les paramètres de l'écriture de l'œuvre comme la composition du timbre et son évolution, des fréquences, des durées, des dynamiques, de la spatialisation...

Pour simplifier, rappelons qu'à un instant  $t$ , le programme sélectionne un nombre ou plusieurs nombres parmi un ensemble de nombres déterminés. Plus précisément, l'algorithme exécute un tirage pseudo-aléatoire à partir d'un ensemble de valeurs définies ( $x$ ) selon des probabilités d'apparitions ( $y$ ). Mais pour éviter de stocker ou de déterminer une succession d'états probabilistes à l'image des chaînes de Markov, chaque tirage modifie au même moment les paramètres du tirage, en faisant évoluer, selon le contexte,  $X$  et  $Y$ <sup>24</sup>.

C'est comme si les conditions d'existence de l'œuvre étaient en perpétuelle évolution, le système pouvant évoluer avec le matériau de l'œuvre dans un jeu d'interaction temps réel, avec plus de cinquante tables de contrôle pour l'ensemble de l'œuvre.

Traduisant le concept d'auto-modification, le tirage pseudo-aléatoire est conditionné par dix autres paramètres à déterminer et présentés ici en trois catégories. Voici la liste des paramètres de ce programme qui permettent de contrôler l'évolution des échelles et des valeurs tirées.

1. Modifications des valeurs de  $y$  (probabilités non nulles) associées aux valeurs de  $x$ 
  - p1 : nombre de valeurs de  $x$  modifiées
  - p2 : coefficient de déformation de  $y$  (ajouté ou soustrait à la probabilité de la valeur tirée)
  - p3 : activé = probabilité modifiée jamais nulle
2. Ajout de valeurs de  $x$ 
  - p4 : nombre de valeurs  $x$  ajoutées
  - p5 : valeur de déformation de  $x$  (ajouté ou soustrait à la valeur tirée)
  - p6 : non activé = expansion de l'échelle des valeurs de  $x$  en fonction de p5
  - activé = ambitus du filtrage à déterminer (p7 et p8)
  - p7 : minimum de l'ambitus de filtrage des valeurs de  $x$
  - p8 : maximum de l'ambitus de filtrage des valeurs de  $x$
3. Mise à zéro de la probabilité d'apparition des valeurs de  $x$  (« éliminer » les valeurs de  $x$ )
  - p9 : nombre de valeurs éliminées (probabilité de la valeur tirée mise à 0)
  - Limite d'expansion supérieure des valeurs de  $x$
  - p10 : valeur maximale possible de  $x$  (grandeur de la table associée aux paramètres de la synthèse)

Comme à d'autres niveaux de composition de l'œuvre, des notions de continuité /discontinuité sont contrôlables. L'auto-modification des tables rend possible la succession d'échelles proches ou éloignées grâce à une définition des valeurs communes des échelles, un contrôle de l'écart de distorsion entre la valeur tirée et la nouvelle valeur de l'échelle...



Figure 11 : Système d'auto-modification des tables de probabilité stockant les valeurs du paramètre des hauteurs

## L'algorithme

L'algorithme est basé sur un système de contrôle du nombre de valeurs stockées dans la table. On peut modifier les probabilités non nulles, ajouter des valeurs (probabilité non nulle) ou encore éliminer des valeurs (mise à 0 de probabilités). L'algorithme peut changer l'échelle des valeurs et ainsi faire évoluer le système<sup>25</sup>.

### La modification des probabilités non nulles dans la table

Afin de modifier dans un premier temps des probabilités non nulles, il nous faut deux paramètres : le nombre de probabilités modifiées ainsi que le coefficient de déformation que l'on appliquera à cette probabilité d'apparition. Afin de s'assurer du nombre exact de probabilités non nulles modifiées, on utilisera une table de référence qui remettra à 0 temporairement les probabilités des valeurs déjà tirées. Un troisième paramètre définit si le choix d'une probabilité obtenant la valeur 0

<sup>23</sup> dimension fractale du processus que l'on peut « emboîter à l'infini » à l'intérieur de lui-même.

<sup>24</sup> échelles initiales et condition du tirage pseudo-aléatoire.

<sup>25</sup> La problématique d'une œuvre inédite, unique prend tout son sens.

après déformation reste à 0 ou à 1 (si une probabilité est à 0 alors la valeur correspondante ne pourra plus être tirée).

L'algorithme est le suivant.

```
Entier p1, p2, a, y, z, i, j ; // p1 et p2 sont des paramètres
// de contrôle
Booléen p3 ; // p3 est un paramètre de contrôle
DEBUT
Pour i allant de 1 à p1
    faire {
        pour j allant de 0 à taille(freq) - 1 // on travaille avec la table ref-freq
            ref-freq[i] = freq[i]; // on tire une valeur
        y = table(ref-freq); // on récupère sa probabilité
        z = ref-freq(y); // soit on augmente la probabilité
        a = random(2)+1; // soit on la diminue

        si a == 1 alors z += (100 - z) * p2 / 100;
        si a == 2 alors z -= z * p2 / 100; // on met la probabilité à 0 pour ne
        ref-freq(y) = 0; // pas altéré deux fois la même
        // probabilité

        si z == 0 et y == 0 alors break ; // quand aucune valeur est dans la
        // table, le tirage sort 0 par défaut
        // donc on évite cette situation

        sinon faire {
            si z == 0 et p3 == 1 // test sur le booléen p3
                ref-freq(y) = 0;
            sinon ref-freq(y) = z; // on change la probabilité
        }

        i++;
    }
FIN
```

### L'ajout de valeurs dans la table

Pour faire évoluer la table au cours du temps, il nous faut ajouter des valeurs dans la table notamment dans le but de changer l'échelle des valeurs. Ainsi, l'œuvre va évoluer au cours du temps et évitera la redondance des valeurs des paramètres. Les phases rythmiques seront donc changées. Le principe de l'algorithme est de tirer une valeur et de lui appliquer une déformation (p5), ainsi on ajoutera une valeur (une déformation sur les abscisses sera présente ce qui correspond bien à une addition d'une valeur).

Le paramètre de nombre de valeurs ajoutées (p4) sera maximal étant donné qu'il est possible qu'une valeur ajoutée soit déjà présente dans la table (dans ce cas là seule la probabilité sera changée). Quatre autres paramètres sont ajoutés : un booléen (p6) pour décider si on étend l'échelle maximale des valeurs, un minimum (p7) et un maximum (p8) si on veut limiter l'échelle des valeurs et enfin un paramètre (p10) qui définit l'échelle maximum des valeurs possibles si on étend l'échelle des valeurs sans limites (en effet, on doit éviter que le système s'emballé).

L'algorithme résultant, appliqué sur la table des fréquences, est le suivant.

```
Entier i, y, z, a, p4, p5, p6, p7, p8, p10 ;
DEBUT
Pour i allant de 1 à p4 // on ajoute au maximum
// p4 valeurs
    faire {
        y = table(freq); // tirage d'une valeur
        z = freq[y]; // on récupère sa probabilité
        a = random(2) + 1;
        si a = 1 alors y += p5 ; // on augmente l'abscisse
        si a = 2 alors y -= p5 ; // on diminue l'abscisse
        si ((y < p7 ou y > p8) et p6 == 1) // test pour l'expansion de l'échelle
            // des valeurs
            alors y = between (p7, p8); // on applique une valeur aléatoire
            // si on n'est pas dans l'ambitus
            // voulu
        si y > taille(freq) alors taille(freq) = y ; // on ajuste la taille de la table
        si y > p10 alors taille(freq) = p10 ; // de même
        freq[y] = z; // on ajoute la valeur
        i++;
    }
FIN
```

### L'élimination de valeurs

La possibilité d'enlever des valeurs est tout aussi nécessaire, condition pour que l'échelle des valeurs puisse évoluer. Les valeurs tirées (p9 différentes) voient leur probabilité remise à 0.

L'algorithme est le suivant.

```
Entier i, y, p9 ;
```

```

DEBUT
Pour i allant de 1 à p9
faire {
    y = table(freq);
    freq[y] = 0;
    i++;
}
// on élimine exactement p9 valeurs
// tirage d'une valeur
// remise à 0 de sa probabilité
FIN
    
```

L'algorithme implémenté dans Max Msp Jitter est le suivant.

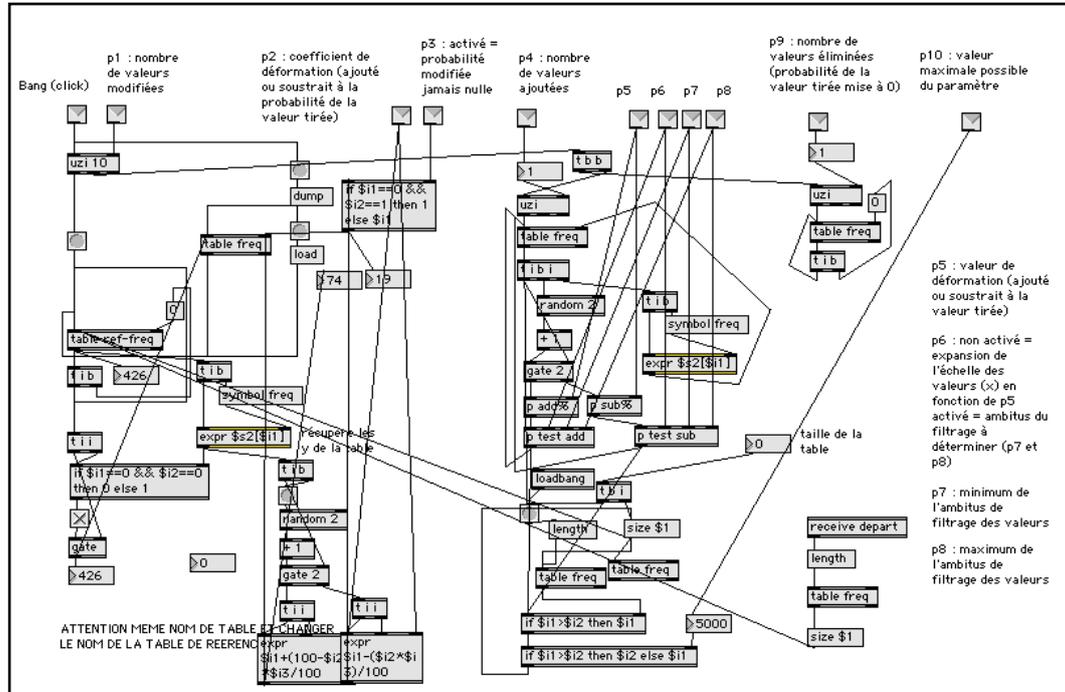


Figure 12 : Programme d'auto modification des tables dans MAX-MSP-JITTER

Nos premières études ont montré des risques d'augmentation de l'entropie du système amenant les données des tables vers un processus totalement aléatoire. Nous avons donc développé un processus de contrôle de l'auto modification des tables, afin de ne pas tendre vers l'aléatoire complet.

**Quatrième programme de contrôle : le régulateur**

Nous avons besoin de savoir si le système reste significativement probabiliste, à certains instants nous allons donc le vérifier et faire des modifications si nécessaire des tables.

**Présentation générale**

Lors des premiers tests effectués sur le principe d'auto-modification, il nous a apparu important d'insérer un système de « régulation », le système risquant d'évoluer trop aléatoirement.

On s'est alors acheminé vers un « contrôleur surveillant l'entropie du système à travers une mesure du nombre de paramètres utilisés dans des ambitus limites définis par l'utilisateur de CA »<sup>26</sup> dont le contrôle va nous permettre de modifier les tables de façon cohérente et intéressante en vue d'une œuvre.

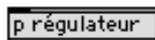


Figure 13 : Système de régulation des tables de probabilité

**L'algorithme**

<sup>26</sup> Stéphane de Gérando, *Trajectoire Oblique*, Bordeaux, icarEditions, seconde édition en 2008.

A un moment donné<sup>27</sup>, on analyse le nombre de valeurs présentes dans la table, et selon des paramètres de contrôle, l'algorithme additionne ou élimine des valeurs de façon continue ou discontinue, afin de créer un état proche du précédent, ou complètement chaotique. Plusieurs paramètres entrent en jeu : un nombre minimal de valeurs possibles dans la table (p1), un nombre maximal de valeurs possibles dans la table (p2), un entier (p3) définissant si on ajoute des points dans un ambitus défini (abscisse minimum (min) et maximum (max) présents dans la table) ou si on ajoute dans toute la table, et un paramètre (p4) qui définit l'ajout ou la soustraction de valeurs de façon continue ou discontinue<sup>28</sup>.

Afin de réaliser l'algorithme, il nous faut obtenir le nombre de valeurs de probabilité non nulle, la valeur (ou adresse) minimum et maximum de probabilité non nulle (fonction nb-vals-x-table).

```
Entier i, y, max, min, n = 0 ; //fonction nb-vals-x-table
DEBUT
Pour i allant de 0 à taille(freq) - 1 // on lit toute la table
    faire {
        y = freq[i] ; // on récupère les adresses
        si y ≠ 0 // test sur la probabilité
            alors {
                max = i ; // on veut l'adresse maximale où la
                // probabilité est non nulle
                n++ ; // on compte le nombre de valeurs
                // de probabilité non nulle
            }
            if front_montant(n) = 1 alors min = i ; // dès que l'on a la première
            // adresse de probabilité non nulle
            // on récupère l'adresse minimum
        i++ ;
        return n ; // on retourne le nombre de valeurs
        // de probabilité non nulle
    }
FIN
```

On peut donc décrire le programme général.

```
DEBUT
p3 = table(cmd2) + 1 ; // on appelle la fonction décrite
n = nb-vals-x-table() ; // ci-dessus
// ambitus continu
si p3 = 1 alors amb = max - min ; // ambitus discontinu
p4 = table(cmd3) + 1 ; // addition ou soustraction continue
si p4 = 1
    alors
    {
        si n - p2 > 0 // trop de points
        alors
        {
            Pour i allant de 1 à n - p2
            {
                y = table(freq) ; // on réduit le nombre de point au
                freq[y] = 0 ; // maximum p2
                i++ ;
            }
        }
        si n - p1 < 0 // pas assez de points
        alors
        {
            Pour j allant de 1 à abs(n - p1)
            {
                b = urn(amb) ;
                si p3 = 1 alors b += min ; // on augmente le nombre de point
                freq[b] = random(100) ; // au minimum p1
                j++ ;
            }
        }
    }
si p4 = 2 // addition ou soustraction
// discontinue
alors
{
    si n - p2 > 0 // trop de points
    alors
    {
        Pour i allant de 1 à n - p1
        {
            y = table(freq) ; // on réduit le nombre de point au
            freq[y] = 0 ; // minimum p1
            i++ ;
        }
    }
    si n - p1 < 0 // pas assez de points
    alors
    {
        Pour j allant de 1 à abs(n - p2)
```

<sup>27</sup> Un bang généré par un métronome, provenant d'une voie de référence ou encore des bangs aléatoires.

<sup>28</sup> Si on n'a pas assez de valeurs, on peut décider d'ajouter le nombre de valeurs qui manquent (continu) ou ajouter le nombre de valeurs afin de s'approcher du maximum autorisé (discontinu). Il en est de même pour la soustraction).

```

    {
      b = urn(amb);
      si p3 = 1 alors b += min;
      freq[b] = random(100); // on augmente le nombre de points
                              // au maximum p2
      j++;
    }
  }
}
FIN

```

Le programme équivalent en langage Max Msp Jitter est illustré dans la figure ci-dessous.

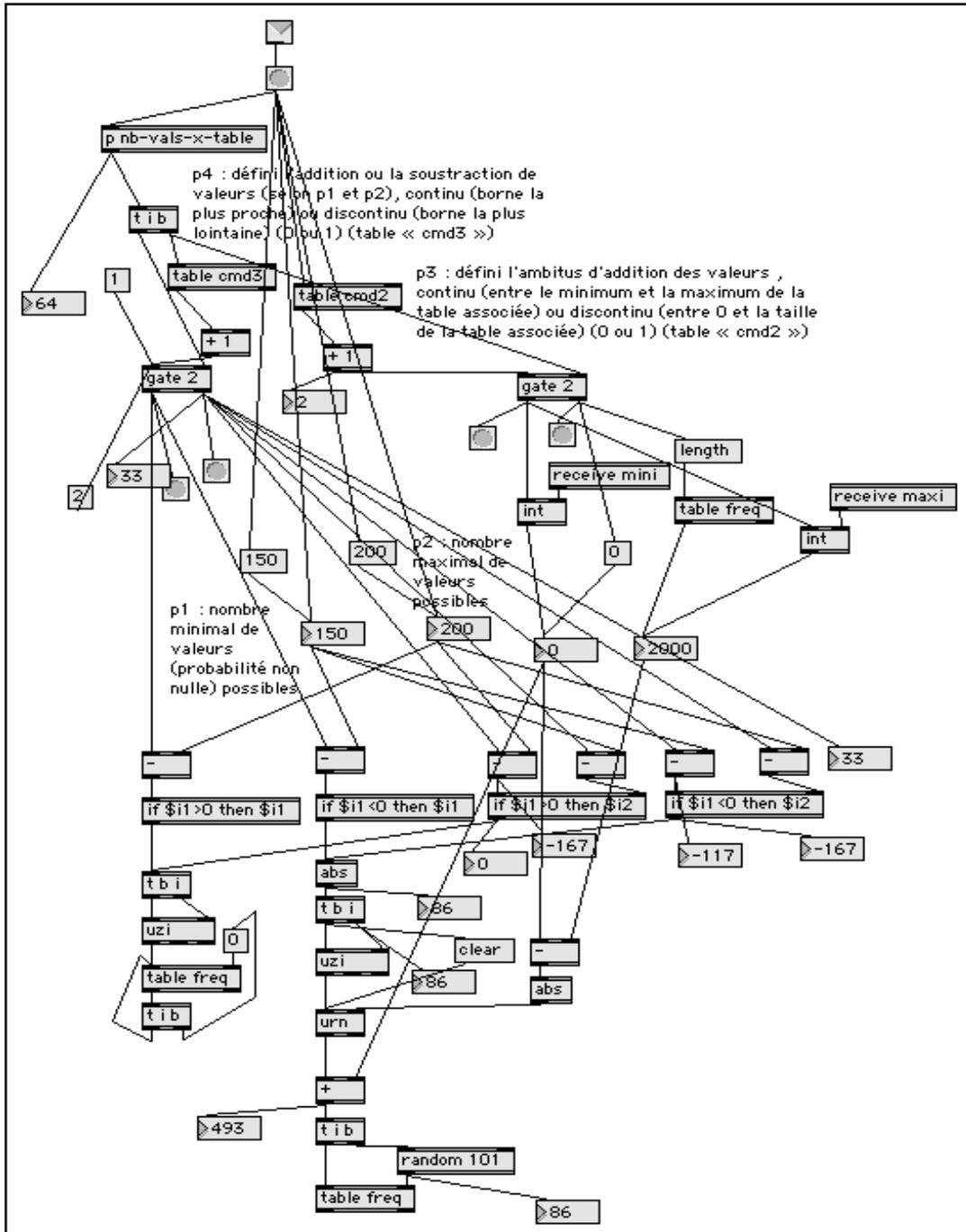


Figure 14 : Programme de régulation des tables dans MAX-MSP-JITTER

Un tel régulateur permet de contrôler l'entropie du système. Avec des paramètres adéquates de contrôle et des tests effectués, le programme possèdera un puissant outil d'évolution des tables.

## 3. PREMIER BILAN DE LA RECHERCHE

## Présentation synthétique de CA

Interface utilisateur	CA se présente en trois zones distinctes : une première zone dédiée au son et à l'élaboration du synthétiseur, une deuxième associée à la synthèse de l'image et une troisième réunissant les programmes de contrôle de l'image et du son.
Synthèse du son	La synthèse du son se fait grâce à un synthétiseur polyphonique hybride (spatialisation 4 sorties), sur la base d'un modèle additif associé à des modulations d'amplitudes, de fréquences et de phases et des programmes de traitement.
Synthèse de l'image	Les images proviennent de la visualisation du public transformées en des tableaux virtuels abstraits, par le biais de programmes temps réel. Il s'instaure alors un jeu d'interactions et de miroirs entre l'abstraction des images visualisées, leur possible modification par le public et la vision éphémère et discrète d'une altérité, sa propre image pouvant parfois « se refléter » dans le tableau.
Cœur du programme : la génération de « clicks »	Ce sous-programme <sup>29</sup> produit une succession de durées, clicks ou déclenchements <sup>30</sup> à partir d'un contrôle probabiliste d'échelles de durées <sup>31</sup> oscillant entre la perception d'une régularité (temps pulsé) et de durées irrégulières (temps lisse). La sortie correspond à une répartition probabiliste des clicks dédiée à la synthèse de l'image et du son selon trois situations : son, son-image, image. Trois diodes lumineuses symbolisent cette répartition. La génération des clicks représente à la fois l'unité et le cœur du système : c'est ce qui donne vie aux autres composants du programme. Notons que ce processus généraliste peut s'étendre au contrôle de la synthèse des cinq sens <sup>32</sup> .
Synchronisation du nombre de voies de contrôle	Ce nouveau sous programme synchronise /désynchronise les clicks initiaux selon un nombre de voies de sortie à définir qui correspondent aux paramètres de contrôle de la synthèse. Inspiré d'un principe compositionnel très répandu dans l'histoire de l'écriture musicale savante occidentale, principe permettant d'engendrer une polyrythmie (structure verticale) à partir d'une voie rythmique unique (structure horizontale).
Auto-modification (« tables itératives »)	Un des principes lié au contrôle de CA, est l'utilisation omniprésente des « tables » (plus de cinquante) dans lesquelles on définit une échelle de valeurs en x (durées, fréquences...) associée à des probabilités d'apparition en y. La question était de pouvoir non pas définir des états probabilistes prédéterminés et fixés définitivement ou une succession d'états contrôlés par des chaînes de Markov <sup>33</sup> comme cela a été souvent réalisé, mais bien d'envisager une auto modification des paramètres de sélection des données liées aux tables. C'est le caractère « sans début ni fin » de CA, œuvre ou installation qui tend vers une durée infinie qui nous a orienté vers cette modification temps réel des systèmes d'écriture de l'œuvre.
Continuité /discontinuité	Le contrôle probabiliste des données est très souvent envisagé sous la forme d'un contrôle des notions de continuité /discontinuité. Prenons l'exemple des fréquences. La succession des fréquences (horizontale et verticale) introduit soit une notion de changement de couleur harmonique ou mélodique (discontinuité), soit une continuité. Ces notions sont par exemple en rapport avec l'utilisation de notes communes dans la succession des échelles de hauteurs. En conséquence, l'auto modification des tables intègre la possibilité de favoriser la succession d'échelles de hauteurs continues et discontinues, ce traitement étant applicable à tous les autres paramètres de contrôle de la synthèse.
Régulateur	Le régulateur est une sorte de contrôleur « surveillant » l'entropie du système à travers une mesure du nombre de paramètres utilisés dans des ambitus limites définis par l'utilisateur.

<sup>29</sup> « p durées ».

<sup>30</sup> Mot équivalent.

<sup>31</sup> Chaque version de CA doit pouvoir facilement intégrer une complexification des algorithmiques de création d'échelles de hauteurs et des procédures de sélection des durées.

<sup>32</sup> Par exemple, la variation d'une synthèse olfactive pouvant très bien être conduit par ce procédé.

<sup>33</sup> Si la distribution de probabilité conditionnelle de  $X_{n+1}$  sur les états passés est une fonction de  $X_n$  seul, alors :

$$P(X_{n+1} = z | X_0, X_1, X_2, \dots, X_n) = P(X_{n+1} = z | X_n) \text{ où } x \text{ est un état quelconque du processus.}$$

## Commentaires critiques

L'imaginaire de l'œuvre et sa composition sont ici bouleversés : il n'est plus question de composer un objet définitivement déterminé, mais le compositeur invente les conditions d'existence d'une œuvre chaque fois différente, de sa micro à sa macro structure. On voit clairement l'impact des nouvelles technologies dans la façon de renouveler le concept d'écriture, en nous renvoyant au vieux rêve d'une « cyberœuvre » décidant, indépendamment de son créateur, de son existence et de son évolution.

Sur le plan technique, cette première phase de développement appelle plusieurs constatations. Les objectifs ont été atteints en dehors de la programmation d'un synthétiseur additif lié à l'image. Cela permet dès aujourd'hui de confirmer la puissance de potentiel d'un tel projet artistique : associer la réflexion historique et esthétique aux développements d'outils et leurs contrôles offre immédiatement des résultats singuliers. Cela confirme l'intérêt de la méthode, la relation art sciences ne peut se passer d'une réflexion historique et esthétique pour faire des choix et constamment guider la recherche.

La deuxième remarque concerne la volonté d'aboutir à un programme qui traite toutes les questions à la fois, de la création des synthétiseurs à leurs contrôles. Sur cette question, le bilan pourrait sembler mitigé. Si l'on s'en tient en l'état au résultat purement artistique, il est en soi insuffisant, ne pouvant confondre des aspects novateurs ou expérimentaux avec la réalité d'une œuvre. Mais d'un autre côté, le temps imparti ne nous a pas permis d'envisager une réelle phase de test et de réglage de l'algorithme afin de l'optimiser voir de l'améliorer dans cette première version. Plus encore, l'accroissement de la complexité de l'algorithme confronté au « contrôle du tout » a entrouvert des perspectives insoupçonnées. Cela pose la question de l'interaction entre différentes échelles de perception d'un même objet. Mais c'est aussi cette notion même de complexité qui est ici remise en cause.

## Perspectives

### 1. Phase expérimentale : optimisation des paramètres de contrôle de l'algorithme

L'algorithme représente déjà une structure complexe à utiliser. Son organisation arborescente nécessite une étape expérimentale permettant un réglage équilibré des paramètres à la fois lié à la synthèse du son et à la synthèse de l'image. Les seuils de déclenchement déterministes ou probabilistes doivent être étudiés et combinés avec précision pour que l'œuvre puisse se renouveler automatiquement dans la durée.

### 2. Phase expérimentale en relation avec l'idée d'une œuvre : optimisation de l'algorithme

A l'écoute de l'œuvre, la relecture du programme et le réglage des paramètres de contrôle de la synthèse aura des conséquences : optimisation, reprogrammation, simplification, complexification, ces étapes seront nécessaires.

### 3. Développer un synthétiseur additif dans le cadre de l'image

Les expériences acquises depuis 2007 dans le domaine de la synthèse de l'image ont enrichi les perspectives actuelles. A l'époque de la première phase de développement de CA, l'objectif était de réaliser des analogies image et son grâce à l'utilisation d'un même modèle de synthèse (additif). Par manque de temps, j'ai opté pour la modification temps réel d'images filmées. La création d'œuvres multimédias 3D via des logiciels d'images temps différé (comme « Le cercle de la sphère » en 2012) m'incite plus que jamais à approfondir des synchronisations au niveau « microscopique » de l'image et du son et de pouvoir être confronté au projet initial avec l'utilisation du même modèle de synthèse pour l'image et le son.

### 4. Développer des notions d'interactions temps réel et de poly-art

*Le labyrinthe du temps* - œuvre multimédia créée en 2011 au Festival de l'innovation à Paris – rappelle les difficultés techniques spécifiques d'une œuvre technologique temps différé. Cette nouvelle phase de développement de CA permettrait de réaliser une œuvre poly-artistique (image,

son, acteur, danseuse, instrumentiste, à préciser selon les moyens), introduisant des notions d'interaction temps réel sur scène.

##### 5. Engager des partenariats scientifiques et institutionnels internationaux

A partir de cette première vision d'ensemble de l'algorithme, il est aujourd'hui possible d'engager des recherches pluridisciplinaires très spécifiques pour améliorer, par étapes successives précisément ciblées en lien avec une partie de l'algorithme, l'efficacité du programme et la réalité de l'œuvre perçue. L'entreprise est colossale tant elle concerne des secteurs différents de recherche : informatique, mathématiques, physiques, électroniques, sciences cognitives, traitement du signal, acoustique, réalité virtuelle ou augmentée, philosophie... Car il s'agit bien dans ce projet d'inventer un métalangage qui pourra rendre cohérent l'enchaînement d'une odeur avec la lumière, un son dans un espace interactif confondant réalité et virtualité. L'invention de la forme, la création des échelles, la gestion du lien entre micro et macocosme, les contraintes perspectives (seuils différentiels, absolus, caches...), les systèmes de captation et d'interaction, la conceptualisation des espaces temps et des échelles différentes de perception, les logiques d'engendrement du matériau... autant de sujets qui doivent s'enrichir de compétences singulières, de la topologie à l'astrophysique, de la biologie à l'infographie ou la recherche en esthétique.