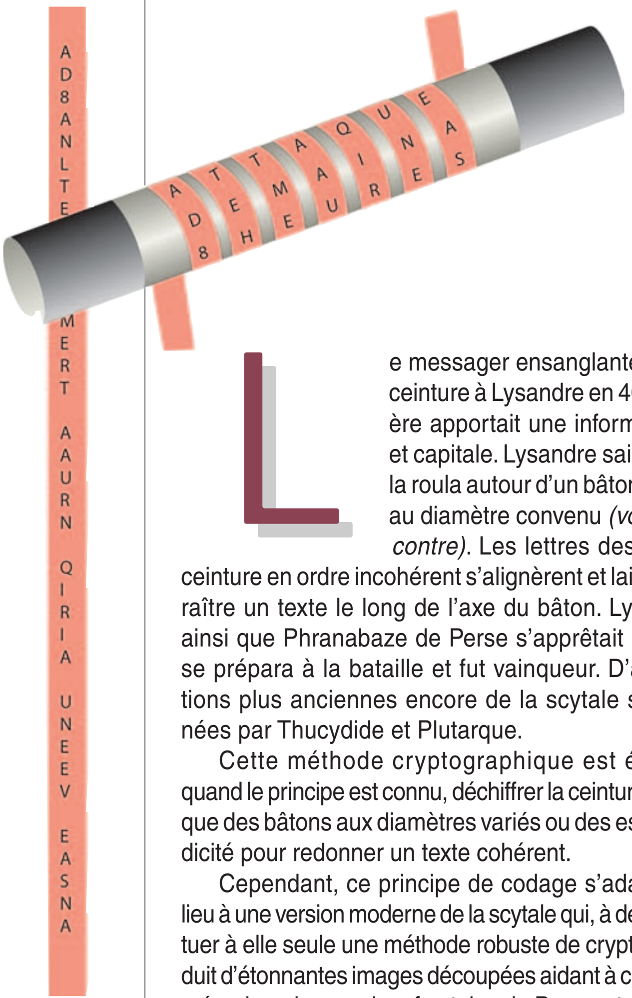


Une scytale informatique

En faisant glisser les points d'une image le long d'une courbe fractale, on obtient de jolies figures... et une méthode cryptographique amusante.

Le messager ensanglanté qui tendit sa ceinture à Lysandre en 404 avant notre ère apportait une information secrète et capitale. Lysandre saisit la ceinture, la roula autour d'un bâton – la scytale – au diamètre convenu *(voir la figure ci-contre)*. Les lettres dessinées sur la ceinture en ordre incohérent s'alignèrent et laissèrent apparaître un texte le long de l'axe du bâton. Lysandre apprit ainsi que Phranabaze de Perse s'apprêtait à l'attaquer. Il se prépara à la bataille et fut vainqueur. D'autres utilisations plus anciennes encore de la scytale sont mentionnées par Thucydide et Plutarque.

Cette méthode cryptographique est élémentaire : quand le principe est connu, déchiffrer la ceinture ne demande que des bâtons aux diamètres variés ou des essais de périodicité pour redonner un texte cohérent.

Cependant, ce principe de codage s'adapte et donne lieu à une version moderne de la scytale qui, à défaut de constituer à elle seule une méthode robuste de cryptographie, produit d'étonnantes images découpées aidant à comprendre les méandres des courbes fractales de Peano et de Hilbert.

Cette scytale numérique associe le principe des transformations bijectives d'images à celui des courbes fractales qui recouvrent une aire non nulle. Revenons sur ces deux idées.

L'inévitable retour

Une transformation bijective d'images est une opération qu'on effectue sur une image numérisée et disponible sous la forme d'un rectangle de $N \times M$ pixels. La transformation change la disposition des pixels dans le rectangle sans en supprimer aucun, et donc sans en dupliquer aucun : une transformation bijective d'images mélange les pixels comme le battage d'un paquet de cartes change l'ordre des cartes.

Lorsqu'on se fixe une transformation bijective d'images et qu'on l'applique plusieurs fois de suite, l'image est modifiée à chaque étape, ce qui donne une série de dessins différents, parfois intéressants. Cependant, et c'est inévitable, l'image initiale finit par réapparaître après un certain nombre d'opérations *(voir la figure 1)*. Cette propriété, l'inévitable retour, est la conséquence d'une situation mathématique bien connue : les transformations bijectives d'un ensemble fini constituent un groupe fini. Or si T est un élément d'un groupe fini, il existe un entier positif n tel que T.T. ... .T (n fois) = 1. Autrement dit, en composant n fois l'application T avec elle-même, on obtient l'application identité – la transformation qui ne change rien –, ce qui a pour conséquence le retour de l'image initiale. Avec un jeu de cartes, la propriété est également vraie : si vous mélangez un jeu de cartes exactement de la même façon plusieurs fois de suite après un certain nombre de mélanges identiques, vous reconstituerez l'ordre initial du paquet.

Dès que la taille de l'image est fixée et que la transformation bijective d'images est définie, le temps nécessaire au retour de l'image initiale est déterminé. Ce nombre n est parfois petit ; par exemple, il vaut 2 si le mélange consiste à opérer une symétrie par rapport au centre de l'image. Pour les transformations fractales que nous allons considérer, il est égal au nombre de pixels de l'image, mais n peut être beaucoup plus grand encore. Des méthodes mathématiques permettent de le connaître sans avoir à mener d'expérimentations prolongées.

Certaines transformations amusantes comme *la transformation du boulanger* ou *Le Photomaton* ont déjà été décrites dans cette rubrique il y a quelques années. Une image soumise au *Photomaton* est présentée sur la figure 2, où l'on voit le miracle de la réapparition soudaine d'Amélie Poulain. Ce que nous voulons observer et comprendre aujourd'hui, ce sont les effets de la transformation fractale associée à la courbe de Hilbert.

Courbes et aires

Dans un article publié en 1890, *Sur une courbe qui remplit toute une aire plane,* Giuseppe Peano introduit une étonnante courbe définie par une fonction continue qui, à chaque nombre réel t entre 0 et 1, associe un point M (t) du plan. Cette courbe méritait une publication, car elle possède la propriété inattendue de passer par tous les points du carré $[0, 1] \times [0, 1]$ sans en oublier aucun, ce que l'on croyait impossible avant sa découverte : comment une courbe sans épaisseur pourrait-elle remplir une surface ? La courbe de Peano a une longueur infinie ; de plus, quand t varie, le point M (t) se déplace sur la courbe avec une vitesse infinie à chaque instant : entre deux instants distincts t et t' aussi proches qu'on veut l'un de l'autre, le point M (t) parcourt une distance infinie.

D'autres courbes du même type furent proposées plus tard et portent toutes le nom générique de « courbes de Peano ». Elles confirmèrent que l'absurdité géométrique n'était qu'apparente, et que nous devions nous habituer à prendre au sérieux ces troublants tracés infinis que le monde mathématique autorise, sans bien sûr qu'ils puissent exister dans le monde réel.

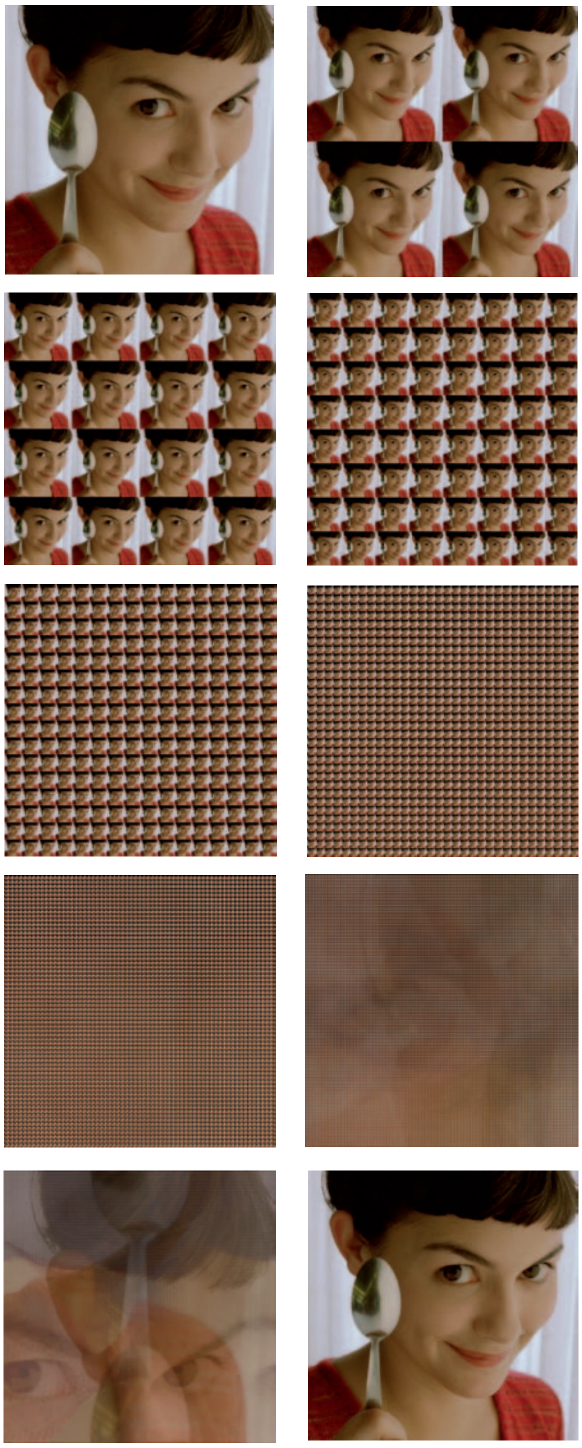
Une façon de se faire une idée du parcours précis de ces courbes serait de les dessiner. C'est impossible, ou plutôt c'est sans intérêt, car seul un carré tout noir représente une courbe de Peano. En revanche, on peut dessiner une suite de courbes C_0, C_1, C_2, ..., C_N, ... qui convergent vers la courbe étudiée quand N augmente. La vision de ces étapes intermédiaires illustre ce qui se passe.

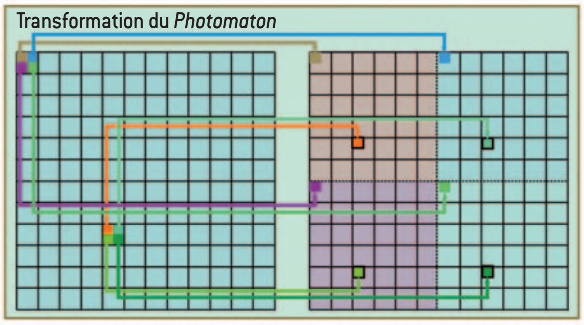
La figure 2 représente les premières étapes de la construction de la courbe de Hilbert qui est une courbe de Peano particulière. L'intérêt des étapes intermédiaires de la courbe de Hilbert est que chaque courbe C_i dessine un parcours fini qui s'étale assez uniformément dans le carré : le trajet ne laisse inoccupée aucune zone du carré ; précisément, la courbe de Hilbert C_N passe à une distance de moins de $1/2^N$ de chaque point du carré [0, 1] x [0, 1].

Les courbes intermédiaires ne passent jamais deux fois par un même point : ce sont des méandres soigneusement agencés pour se promener un peu partout dans un carré, sans jamais revenir à un endroit déjà visité. Suivre une telle courbe permet de passer de proche en proche par tous les pixels d'une image carrée de taille $2^n \times 2^n$ sans en oublier aucun et sans passer deux fois par le même.

La transformation de Hilbert

La courbe de Hilbert, comme toute courbe de Peano, est une fractale, car le motif utilisé se retrouve à diverses échelles dans le dessin : certaines parties sont semblables au tout. La courbe de Hilbert suit un parcours qui est composé de quatre parties exactement identiques et, de plus, semblables,





Le déplacement des points dans les transformations du *Photomaton*

1. Au *Photomaton* (qui joue un rôle central dans *Le fabuleux destin d'Amélie Poulain*), nous avons associé une transformation bijective d'images : l'image est quadrillée avec un nombre pair de lignes et un nombre égal de colonnes, ce qui permet de regrouper les pixels par paquets de 4 formant des carrés. Les quatre pixels d'un carré de 4 sont éclatés et chacun est utilisé pour reconstituer un pixel dans quatre images de taille 1/2. Ces quatre images de taille 1/2 sont disposées en carré : les quatre images obtenues ne sont pas exactement identiques (contrairement au vrai *Photomaton*), mais l'œil ne voit pas les petites différences. Cependant, aucun pixel n'a été perdu, ni dupliqué : les pixels ont été répartis différemment dans le rectangle. En recommençant l'opération d'éclatement des paquets de 4 pixels, on obtient 16 images d'Amélie Poulain, puis 64, etc. Cependant, avec cette image de taille 512 x 512, à la neuvième application de la transformation du *Photomaton*, on revient à l'image initiale : comme toute transformation bijective d'images, une série de transformations *Photomaton* finit par donner l'application identique. L'étude arithmétique explique pourquoi, avec une image de format 512 x 512, neuf étapes sont nécessaires pour ce retour. Les figures ci contre donnent le résultat des différentes étapes et la figure du bas explicite la transformation.

à échelle 1/2, à la courbe entière. Elle est donc composée de 16 parties exactement identiques entre elles et identiques, à échelle 1/4, à la courbe entière, etc.

La transformation bijective d'images « de Hilbert » est maintenant facile à définir :

a. On se fixe une image carrée dont le côté a un nombre de pixels qui est une puissance de 2. Pour $2^9 = 512$, il y a 512×512 pixels dans l'image.

b. On numérote tous les pixels de cette image en suivant le parcours de la courbe intermédiaire de Hilbert. Dans notre exemple, c'est la courbe C_9.

c. On convient que la transformation associée fait passer le pixel 1 à la place du pixel 2, le pixel 2 à la place du pixel 3,..., le pixel n à la place du pixel $n + 1$, ..., le dernier pixel revenant prendre la place du pixel 1.

Chaque pixel glisse de un pixel le long de la courbe et, au cours des transformations, chaque pixel suit donc le trajet de la courbe. Le temps de retour est le nombre de pixels de l'image.

Comme pour la scytale, dont elle est une version plate et alambiquée à l'extrême, le glissement masque l'image. Lorsque le glissement est faible, l'image est rendue floue, mais dès qu'il est important, l'image devient incompréhensible, à l'exception de quelques étapes particulières où l'image réapparaît découpée et mélangée *(voir la figure 3)*.

Pour comprendre la transformation de Hilbert, il faut imaginer un grand circuit de course automobile très tortueux où circulent des voitures placées tous les dix mètres le long du circuit. Les voitures avancent toutes à la même vitesse. Si, vues de haut, les couleurs des voitures dessinaient une image à l'instant du départ, celle-ci se brouille rapidement, comme lorsque la ceinture codée de la scytale n'est pas parfaitement enroulée autour du bon bâton.

Les propriétés de la transformation de Hilbert

Jouer avec la transformation de Hilbert est fascinant et permet de réaliser d'intéressants dessins géométriques, dont quelques-uns sont présentés sur la figure 3.

Avec l'aide de M. Braure, M. Dref et N. Kondratek, trois étudiants de l'IUT A de l'Université des sciences et technologies de Lille, nous avons réalisé le programme *Transform* (http://www.lifl.fr/~mathieu/transform/). Il vous permet d'effectuer des manipulations d'images avec toutes les transformations mentionnées dans cet article et aussi de mettre en œuvre les opérations de codage et de stéganographie décrites plus loin.

La transformation de Hilbert possède quelques propriétés remarquables qui en font l'intérêt. D'abord, contrairement au *Photomaton*, la déformation de l'image est progressive, ce qui n'est pas étonnant puisqu'elle est construite sur une courbe continue. Si vous l'appliquez une seule fois, la transformation trouble très légèrement l'image qui reste lisible malgré le léger flou. Si vous l'appliquez deux fois, l'image est déjà un peu plus brouillée, et plus vous l'appliquez, plus l'image devient difficile à identifier.

En revanche, si vous l'appliquez N fois en prenant pour N une puissance de 2 (ou un multiple d'une puissance de 2), vous obtenez une image composée de carrés bien nets, ceux de l'image initiale tournés, mélangés et retournés (c'est-à-dire résultant d'une symétrie par rapport à une droite). Plus la puissance de 2 est élevée, plus ces carrés sont de grande taille. Ainsi, avec une image de taille 512×512, si vous appliquez la transformation $256 \times 256 = 65\,536$ fois, le résultat est une image composée des quatre carrés bien nets correspondant aux carrés que donne le découpage de l'image de départ en quatre parties égales *(figure 3 VII)*. Le carré en haut à gauche est passé en bas à gauche, mais en plus, il a été tourné de 90 degrés et a été retourné. Le carré en bas à gauche est passé en bas à droite sans rotation, ni retournement. Le carré en bas à droite est passé en haut à droite en subissant une rotation de 90 degrés et un retournement. Le carré en haut à droite est passé en haut à gauche en tournant de 180 degrés. L'explication de cette danse complexe provient de ce que les points suivent la courbe de Hilbert : puisque la courbe de Hilbert est composée de quatre fois la même forme, lorsqu'un quart du parcours a été fait, tous les pixels se redisposent exactement comme au départ dans les quarts du carré initial, chacun ayant changé de carré. L'autosimilarité fractale de la courbe explique la réapparition mélangée du dessin.

Toujours avec C_9, en appliquant la transformation de Hilbert $128 \times 128 = 16\,384$ fois, on obtient encore une image composée de carrés bien nets et cette fois, il y en a 16. Comme précédemment, ils suivent la courbe qui s'est déplacée d'un seizième de sa longueur. Ils subissent des rotations et retournements selon une règle que l'on finit par comprendre : le fait que le dessin soit retourné ou non dépend du sens de parcours du motif répété qui compose la courbe de Hilbert.

Figures invariantes

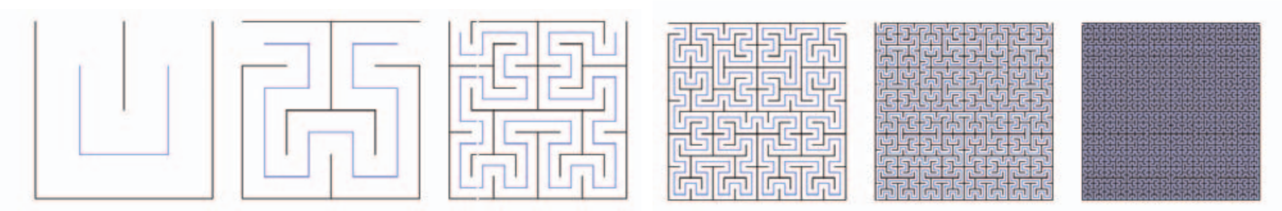
La question des invariants de la transformation de Hilbert peut être résolue complètement et sans trop de mal, justement parce que la transformation provient d'une courbe.

Un dessin D est invariant par une transformation T, si T $(D) = D$. On dit aussi que D est un point fixe de T. Pour simplifier les choses, nous ne considérerons que des images en noir et blanc : un pixel est noir ou blanc. Notons H la transformation de Hilbert.

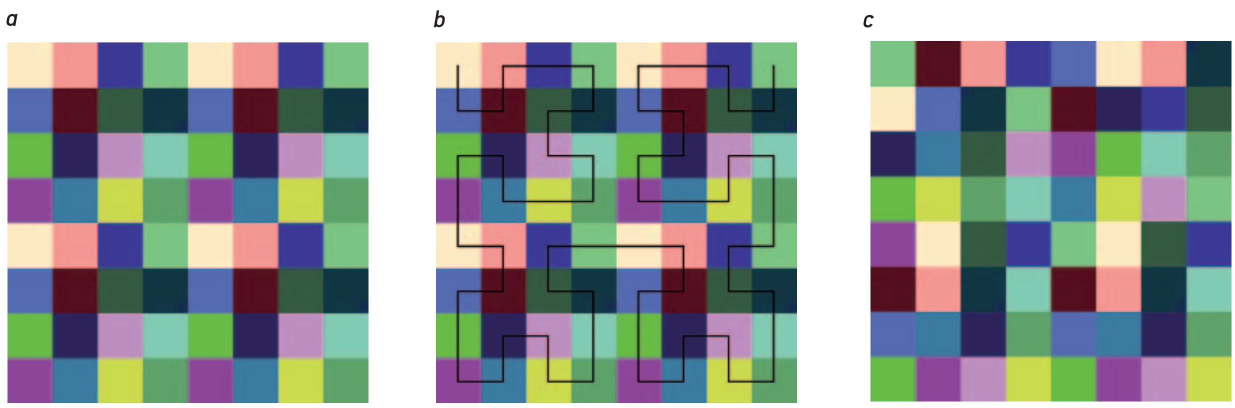
Pour qu'une image soit invariante par H, il faut que chaque pixel, quand il se déplace sur le chemin de la courbe de Hilbert, retombe sur un pixel de la même couleur. Il faut donc que le premier pixel soit identique au second, que le second soit identique au troisième, etc. : tous les pixels doivent donc être identiques ! Les images invariantes par H sont donc le carré tout blanc et le carré tout noir.

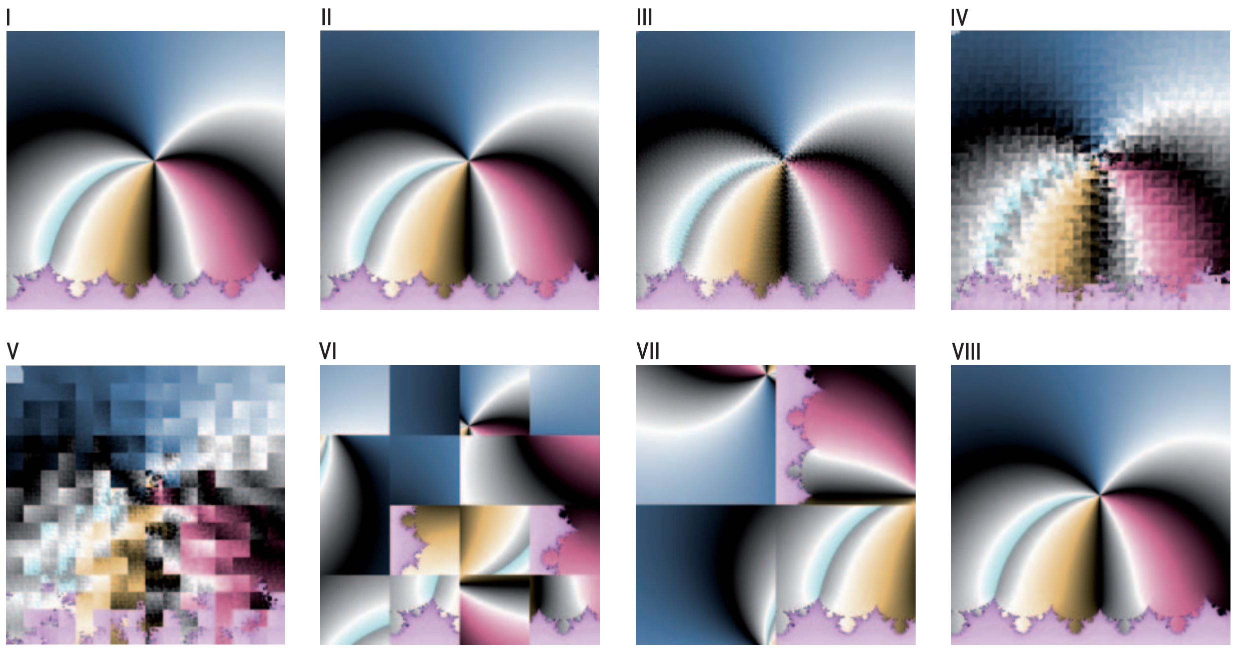
Pour H^2 (H appliqué deux fois consécutivement), il faut que chaque pixel, après s'être déplacé deux fois, retombe sur un pixel identique. L'image invariante est donc entièrement déterminée dès qu'on connaît les deux premiers pixels, et il y a donc quatre images invariantes selon que les deux premiers pixels sont blanc-blanc, blanc-noir, noir-blanc ou noir-noir.

Pour H^3, comme pour H, il n'y a que deux images invariantes : l'image toute noire ou l'image toute blanche. Cela est dû à ce que le nombre 3 est premier avec 2^n, ce qui fait qu'en sautant de 3 en 3, au bout de plusieurs tours de l'image, un point finit par passer par chaque point. De manière



2. La courbe de Hilbert est une courbe continue qui recouvre une aire non nulle du plan : elle résulte d'un processus itératif dont les étapes seront utilisées ici. Chaque étape donne une courbe finie parcourant les cases d'un carré découpé en carrés plus petits. La courbe limite est continue et passe par tous les points du carré. Les courbes définissent des méthodes pour énumérer, sans faire de saut, les différentes cases d'un carré découpé en $2^n \times 2^n$ petits carrés. Cette propriété est utilisée pour la définition de la transformation de Hilbert.





3. Le principe de la transformation fractale de Hilbert : on prend une courbe intermédiaire de Hilbert parcourant tous les pixels d'une image de taille 2^n x 2^n et l'on déplace chaque pixel en suivant la courbe : le pixel 1 va occuper la place du pixel 2, le pixel 2 va occuper la place du pixel 3, etc. Le dernier pixel revient prendre la place du pixel 1. Si on prend une figure de 8 × 8 pixels colorés *(a)*, que l'on applique la transformation de Hilbert associée à la courbe intermédiaire du carré 8 × 8 *(b)*, les pixels se déplacent et l'on obtient le dessin *(c)*. Prenons maintenant une image plus grande, de taille 512 × 512, et examinons ce qui se passe. L'application déplace tous les pixels d'un point à chaque fois qu'on l'utilise : elle reconstituera le dessin initial au bout de 512 × 512 étapes. Les figures I, II, III, IV, V, VI, VII, VIII représentent le résultat de l'application de la transformation de Hilbert du carré 512 × 512 aux étapes 0, 1, 10, 100, 1 000, 10 000, 16 384, 65 536, 262 144 (512 × 512). Le dessin II montre qu'une seule application brouille très peu l'image. Au bout de 10 étapes (dessin III) et encore plus, au bout de 100 étapes (dessin IV), le flou devient important. Lorsque la transformation a été appliquée 1 000 fois (dessin V) et 10 000 fois (dessin VI), l'image n'est plus identifiable. Notons cependant *(voir le texte de l'article)* que lorsque l'on applique la transformation *N* fois, avec pour *N* un multiple d'une puissance de 2, alors des carrés bien nets de l'image initiale sont visibles. Le dessin VI montre le résultat de l'application de la transformation de Hilbert 2^{14} = 16 384 fois (ce qui correspond à un déplacement d'un seizième le long de la courbe). Pour le dessin VII, correspondant à un déplacement d'un quart de la courbe, les quatre quarts de l'image initiale ont été déplacés et tournés. La figure VIII, identique à la figure initiale, est obtenue après 262 144 étapes.

générale, si l'exposant e est premier avec 2, il y a seulement deux figures invariantes par H^e.

Lorsque e est une puissance de 2, $e = 2^k$, le nombre de figures invariantes par H^e parmi les images ayant m points ($m = 2^n \times 2^n$, avec n supérieur ou égal à k) est 2^e. En effet, une image invariante est déterminée dès qu'on en connaît les e premiers points, car tout se passe exactement comme si on cherchait les différentes façons de colorier un collier circulaire de m perles noires ou blanches lorsqu'on le fait tourner de e perles et qu'on veut que cela ne change pas le motif formé par les perles.

Il y a ainsi quatre figures invariantes pour les carrés 2×2 et H^2, et 16 dessins invariants pour les carrés 4×4 et H^4 (*voir la figure 4*). Pour un plus grand nombre de pixels, il y a encore plus d'invariants.

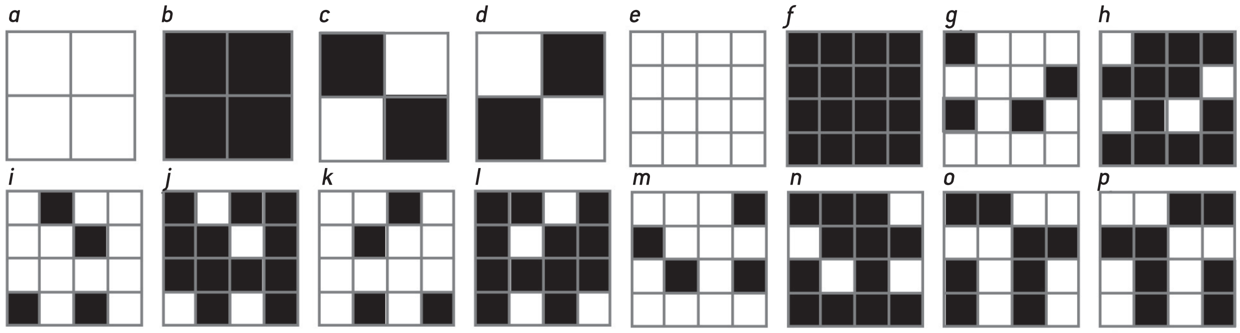
D'un point de vue cryptographique, la transformation de Hilbert n'est pas excellente, car elle maintient côte à côte deux pixels qui le sont sur le parcours de la courbe de Hilbert, ce qui limite le brouillage, même après un grand nombre d'applications de H.

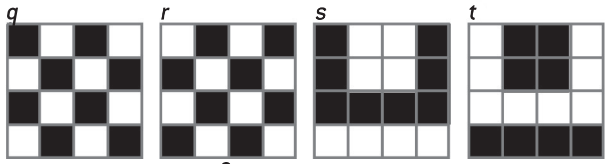
En revanche, en l'associant à d'autres transformations, on obtient un moyen de codage d'images assez efficace. Imaginons par exemple que nous souhaitions coder une image de taille 512 x 512 (on prend un carré dont le côté est une puissance de 2, de façon à ce que l'on puisse utiliser la transformation de Hilbert qui exige cela). On applique 3 141 fois la *Transformation de Hilbert*, puis 7 fois *Photomaton*. On obtient une image convenablement brouillée *(voir la figure 5c)*.

Outils de cryptographie et stéganographie

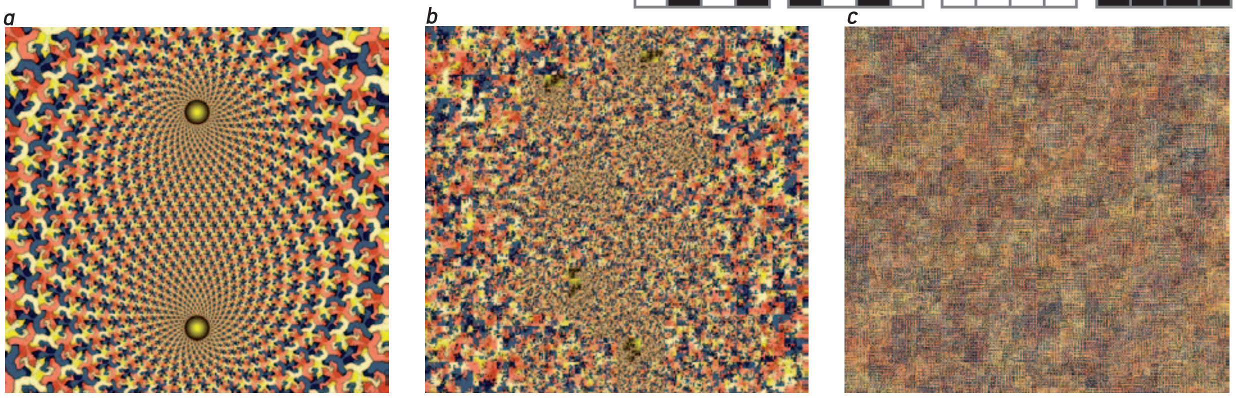
Si vous l'envoyez à un correspondant qui connaît votre clef de brouillage (en notation abrégée : HI 3141 ; PH 7), il saura reconstituer l'image. Il lui suffira de revenir en arrière et donc d'appliquer *Photomaton* – 7 fois, *Hilbert* – 3 141 fois (on reprend la liste à l'envers, et on change les signes).

Le logiciel *Transform* permet ce type de codage de manière directe : on charge l'image, on indique la clef retenue et le calcul de l'image transformée se fait tout seul. Il permet aussi le décodage rapide en indiquant uniquement la clef de codage (donc sans avoir à calculer soi-même la clef de décodage). En utilisant des transformations plus variées, l'espace des clefs devient considérable, et cela rend impossible un décodage par tâtonnements

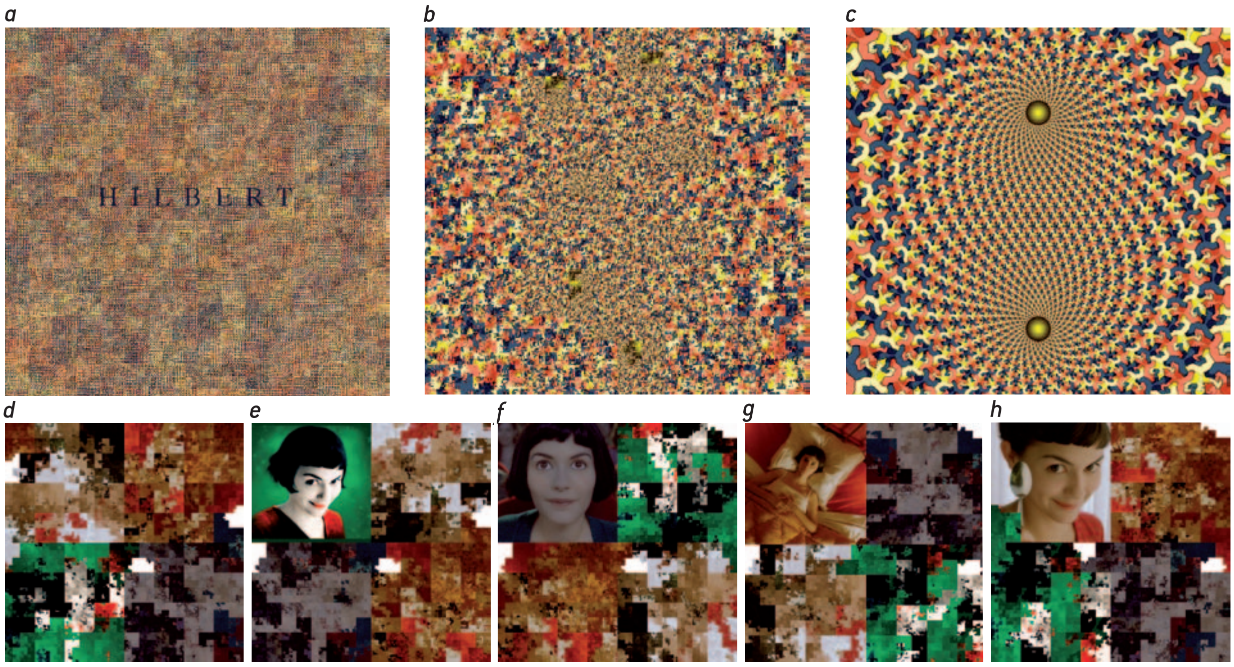




4. Invariants : il y a quatre invariants (figures qui ne sont pas modifiées par une transformation) pour le carré 2×2 et la transformation H^2, deux fois la transformation de Hilbert (*a, b, c, d*), et seize invariants (*e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t*) pour la transformation H^4, quatre fois la transformation de Hilbert et le carré 4×4.



5. La cryptographie et les clefs secrètes. Chiffrer une image se fait le plus souvent en chiffrant le fichier de l'image par une méthode cryptographique conçue pour chiffrer n'importe quel fichier informatique. On s'intéressera ici aux méthodes de chiffrement d'images d'interprétation géométrique simple, par les transformations bijectives d'images. Vous pourrez utiliser pour cela le programme *Transform*. On choisit quelques transformations bijectives d'images 512×512 et quelques nombres entiers. Par exemple : *Hilbert*, 3141, *Photomaton*, 7, la clef de chiffrage étant [HI 3141 ; PH 7]. Pour chiffrer l'image (*a*), on applique alors *Hilbert* 3141 fois (*b*), puis *Photomaton* 7 fois (dessin *c*). On obtient une image brouillée. Votre correspondant qui connaît votre clef de brouillage saura reconstituer l'image, car il lui suffira de revenir en arrière et donc d'appliquer *Photomaton* – 7 fois et l'on revient à (*b*), puis *Hilbert* – 3141 fois (on reprend la liste à l'envers et on change les signes) pour obtenir (*a*). Attention : pour pouvoir utiliser les transformations fractales, il faut partir d'une image carrée dont le côté, comme ici, est une puissance de 2. (L'image initiale est une image de Jos Leys.)



6. Pour la stéganographie (l'art de cacher des informations) l'idée consiste à prendre une image chiffrée (par exemple le dessin 5*c*), à lui superposer un message (ce qui donne le dessin 6*a*), puis à la déchiffrer, ce qui mélange les pixels utilisés par le message et les distribue un peu partout dans l'image initiale (dessin 6 *b* et 6 *c*). Cette modification de quelques pixels de l'image initiale passera inaperçue si l'image est assez grande et que le nombre de pixels modifiés par le message ajouté n'est pas trop important. Une telle image d'apparence anodine peut circuler ou être placée sur un site Internet. Celui qui connaît la clef utilisée peut facilement prendre connaissance de l'information cachée : il lui suffit d'appliquer les transformations bijectives d'images dont la clef indique la liste. Le dessin 6*d* cache quatre images différentes dont on ne peut prendre connaissance qu'à tour de rôle. Si vous lui appliquez la transformation de Hilbert – 67 000 fois, vous découvrez le dessin 6*e*. Si ensuite vous appliquez encore Hilbert – 67 000 fois, vous verrez cette fois le dessin 6*f*. Puis, en recommençant, vous tomberez sur les dessins 6*g* et enfin 6*h*. Nous vous laissons deviner comment a été fabriquée cette image.

sans connaissance de la clef, même pour celui qui connaît la méthode utilisée.

Une autre utilisation des transformations bijectives d'images est la stéganographie (prévue dans le logiciel *Transform*) : il s'agit de cacher un texte, ou un petit dessin, dans une image donnée en la modifiant d'une façon si discrète qu'on ne puisse même pas soupçonner que l'image modifiée contient un message caché. Avec la méthode de stéganographie que nous vous proposons, non seulement le codage dépend d'une clef qui, lorsqu'on l'ignore, rend indéchiffrable le message, mais en plus le message est si discret qu'un observateur non prévenu ne sait pas qu'il y a un message !

Voici le principe de fonctionnement de cette stéganographie par les transformations bijectives d'images.

a. Vous choisissez une image, si possible assez complexe, par exemple avec des feuillages ou des motifs géométriques nombreux. Pour notre exemple, nous avons utilisé une image créée par Jos Leys en s'inspirant d'un motif dû à Escher.

b. Vous appliquez un certain nombre de transformations (comme pour le codage vu précédemment, avec le même système de clef).

c. Vous écrivez alors un texte ou un petit dessin sur l'image brouillée *(voir la figure 6a)*. À ce point de la procédure, le texte est bien lisible, car le brouillage de l'image a créé une surface à peu près homogène sur laquelle ce que vous avez ajouté se détache clairement.

d. Vous décodez maintenant l'image créée *(figure 6a)* avec la clef inverse de la clef utilisée pour produire l'image brouillée. Cela a un double effet : d'une part, cela répartit les points qui correspondent au message un peu partout dans l'image initiale ; d'autre part, cela fait réapparaître l'image initiale dont seuls quelques pixels ont été modifiés. Les pixels modifiés sont à peu près uniformément répartis, ce qui les rend invisibles ou très peu visibles.

C'est l'image obtenue ainsi *(figure 6c)* que vous envoyez à votre correspondant. Pour prendre connaissance du message caché dans l'image à l'apparence anodine, celui-ci doit utiliser la clef de codage (dont vous aviez convenu avec lui). En appliquant les transformations bijectives indiquées par la clef, cela brouille l'image, mais cela reconstitue l'image que vous aviez produite *(figure 6a)*, et donc votre correspondant peut lire le message que vous avez écrit.

Indiquons que les méthodes de cryptographie et de stéganographie définies à partir des transformations bijectives d'images n'ont été présentées ici que pour leur intérêt récréatif. Elles doivent être considérées avec une certaine méfiance, car elles n'ont pas pour l'instant été soumises à des attaques sérieuses et il n'est pas certain qu'elles puissent résister longtemps à des spécialistes chevronnés de cryptologie...

Jean-Paul DELAHAYE et **Philippe MATHIEU** sont professeurs d'informatique à l'Université de Lille.

Jean-Paul DELAHAYE, Labyrinthes de longueur infinie, in *Complexités : aux limites des mathématiques et de l'informatique*, Belin/Pour la Science, 2006.

Jean-Paul DELAHAYE et Philippe MATHIEU, Images brouillées, images retrouvées, in *Jeux mathématiques et mathématiques des jeux*, Belin/Pour la Science, 1998.

Logiciel Transform : http://www.lifl.fr/ffmathieu/transform/