

Auto et intercorrélation, recherche de ressemblance dans les signaux : application à l'identification d'images floutées

J.-M Friedt, 29 avril 2011

1 Introduction

La disponibilité d'une masse croissante de données *via* internet permet à chacun d'appréhender des problèmes qui restaient confinés jusque récemment à un cercle restreint d'utilisateurs suffisamment chanceux pour accéder à ces données. Ainsi, seule la compétence de traitement limite aujourd'hui la capacité de toute personne à extraire un maximum d'informations de ces données. Par exemple, une équipe anglaise a ainsi identifié au moyen d'un accès gratuit aux images satellites de résolution médiocres (*preview* [1]) la localisation de toutes les colonies de manchots en Antarctique, tâche impossible à réaliser (ou financer) par un mode d'observation dédié à cet objectif [2]. Une des approches les plus intuitives est la recherche de l'occurrence (ou absence) d'une information dans cette masse de données. Pour appliquer plus précisément ce concept aux images, il s'agit de rechercher l'occurrence d'un motif. Nous proposons donc ici une présentation d'un outil classique de recherche de ressemblances entre motifs : la corrélation. Concrètement, après une brève introduction théorique, nous appliquerons ce concept aux images (rechercher l'occurrence d'un motif) et étudions dans des séquences temporelles d'images de déplacement de motifs (champ de déplacements), qui serviront de mise en place des outils en vue du cœur du sujet, la recherche de zones floutées dans les images. Enfin, nous concluons sur un cas particulier où l'utilisateur choisit le signal sonde qui va servir à détecter une cible, et montrons comment un choix judicieux des propriétés statistiques permet de maximiser les chances d'identifier l'écho et sa localisation.

L'ensemble des expérimentations de traitement statistique du signal se fera au moyen de GNU/Octave (www.gnu.org/software/octave) [3], un port opensource de l'outil propriétaire Matlab de Mathworks totalement fonctionnel depuis l'implémentation de la majorité des *toolbox*, et notamment *signal processing toolbox* dans le cas qui nous intéressera ici. L'objectif de cette étude, au-delà d'acquérir quelques connaissances utiles pour extraire toute l'information de la masse de données accessibles sur le web, est d'être conscient des risques associés à ces données, notamment par une analyse systématique de la statistique du signal. Par exemple, il est bien connu qu'une analyse du flux de données fournit une information en cryptographie sans nécessairement permettre l'identification de l'information elle-même : il est peu judicieux de ne crypter par PGP que les messages à vocation de secret, messages qui ressortiront trivialement lors d'une analyse de flux [4]. Une extrapolation audacieuse qui conclura notre étude est que de la même façon, il est peu judicieux de manuellement modifier les propriétés statistiques (flouter) les zones considérées comme sensibles sur les images aériennes, la signature associée à cette opération étant identifiable.

2 L'intercorrélation : quelques notions de base

Le concept que nous proposons de mettre en œuvre au cours de cet exposé est l'intercorrélation entre deux signaux. Ce concept vise à extraire le "taux de ressemblance" entre deux signaux unidimensionnel (signaux temporels par exemple) ou bidimensionnels (images). La définition mathématique de ce concept, que nous nous efforcerons d'intuiter par la suite, est que l'intercorrélation entre deux signaux, supposés unidimensionnels dans un premier temps, de moyenne nulle, $s(t)$ et $s'(t)$ est

$$xcorr(s, s')(t) = \sum_{i=-N}^N s_i s'_{i-t}$$

Le lecteur qui ne désire pas entrer en détail dans la compréhension de cette formule – et par là esquiver toute chance de comprendre le sens du concept – pourra se contenter d'expérimenter avec les fonctions `xcorr()` et `xcorr2()` (pour la version 2D applicable en imagerie) de GNU/Octave. La philosophie de l'intercorrélation est de rechercher le retard (ou translation t) qui permet de maximiser la ressemblance entre s et s' . En effet, ce n'est que pour un retard t qui maximise la ressemblance que la somme s'effectue de façon cohérente avec des produits s_i et s'_{i-t} tous positifs (de l'ordre de s_i^2). Dans le cas contraire, le produit des valeurs de s_i se fait avec un signal s' de valeur moyenne nulle qui a toutes les chances de

fournir un résultat proche de 0. Ainsi, nous allons rechercher t qui maximise la somme des éléments du produit $s(i) \times s'(i - t)$ lorsque le retard t permet à s de ressembler au mieux à s' décalé de t .

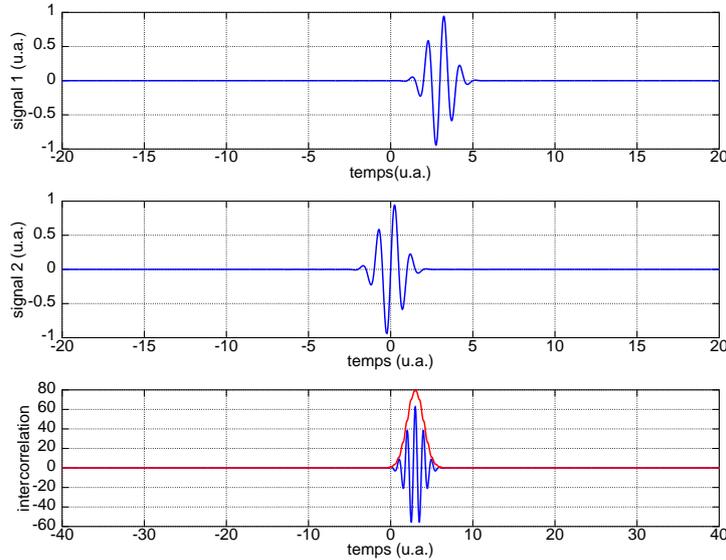


FIG. 1 – Deux signaux qui pourraient s’apparenter à une impulsion de RADAR, au milieu le signal émis, et en haut le signal récupéré après un certain temps de vol et réflexion sur une cible statique. En bas, l’intercorrélation entre le signal émis et le signal reçu n’est non-nul que lorsque le signal reçu ressemble au signal émis. Le maximum d’intercorrélation, par exemple identifié ici comme maximum de l’enveloppe de l’intercorrélation (valeur absolue puis filtre passe-bas) indique le retard (ou temps de vol) entre impulsion émise et reçue.

On notera la forte ressemblance entre la définition de la corrélation $xcorr(s, s')(t)$ et la convolution définie par

$$conv(s, s')(t) = \sum_{i=-N}^N s_i s'_{t-i}$$

Dans ce cas, le produit dans l’intégrale se fait en balayant s selon un axe du temps i croissant, tandis que le second signal s' est décalé d’un retard t mais balaie le temps à l’envers ($-i$). Cette opération reflète l’application du filtre défini par s' au signal s . Cette digression est nécessaire pour entrer dans les détails pratiques de la mise en œuvre de l’intercorrélation : par soucis d’efficacité, il nous faut impérativement passer par le domaine spectral (transformée de Fourier). En effet, une relation fondamentale [5] pour la mise en place d’un calcul d’intercorrélation est le fait que **la transformée de Fourier (TF) de la convolution de deux signaux est le produit des transformées de Fourier de chaque signal**

$$TF(conv(s, s')) = TF(s') \times TF(s)$$

Cette affirmation, apparemment anodine, permet de passer d’un nombre de multiplications de l’ordre de N^2 (avec N le nombre d’éléments du vecteur à analyser) vers $N \ln N$ grâce à l’algorithme de transformée rapide de Fourier (*Fast Fourier Transform* – FFT) [6]. Une subtilité subsiste : nous nous proposons de travailler sur la corrélation, alors que nous énonçons un théorème sur la convolution. On aura remarqué, par rapport à la définition de la corrélation, l’inversion du signe sur le terme de la variable d’intégration : dans un cas la variable glisse le long de l’axe du temps ($s'(i - t)$), dans l’autre cas elle glisse dans le sens opposé ($s'(t - i)$). Le passage de la convolution à la corrélation va donc soit s’obtenir par inversion de l’axe du temps dans le domaine temporel (rotation de 180° du motif s') [7], soit dans le domaine spectral en prenant le complexe conjugué de la transformée de Fourier. On pourra se convaincre que cette dernière

opération inverse la direction du temps en considérant que le complexe conjugué de $\exp(i \times \omega t)$ est $\exp(-i \times \omega t)$ (avec t le temps, ω une pulsation et $i^2 = -1$).

Appliquons ces concepts quelque peu abstraits en l'état au cas particulier de la recherche de l'occurrence d'une lettre dans un texte. Si la corrélation est un estimateur capable d'identifier le taux de ressemblance entre deux motifs, alors en prenant un texte (Fig. 2, en haut à gauche) dont on extrait une lettre (Fig. 2, en haut à droite), nous devons être capables de retrouver la position de toutes les occurrences de ces lettres dans le texte. Au-delà des points déjà mentionnés sur le retournement de l'axe des temps de s' , le dernier point fondamental à ne pas oublier est que s et s' doivent être de moyenne nulle : on prendra soin de toujours retrancher la valeur moyenne de s et s' avant tout calcul (fonction `mean()` dans GNU/Octave). Afin d'illustrer les deux modes de calcul cités auparavant, nous pouvons dans un premier temps appliquer la convolution en deux dimensions (fonction `conv2()` de GNU/Octave) :

```
dtex_z=imread('texte.ppm'); dtex_z=dtex_z-mean(mean(dtex_z));
da_z=imread('a.ppm');      da_z=da_z-mean(mean(da_z));
res=abs(conv2(rot90(da_z',2),dtex_z'));
s=(res>0.70*max(max(res)));      % imagesc(s);
imagesc(dtex_z+2*s(18:636,18:249)) % 18 car la lettre fait 35x35
```

Ce programme charge le texte (supposé contenu dans l'image `texte.ppm`) et le motif (la lettre "a" dans `a.ppm`). Puis il effectue la convolution 2D sur une rotation de 180 degrés (fonction `rot90()`) appliquée sur un des motifs d'une part, et l'autre motif d'autre part. Finalement, nous seuillons à 70% du maximum de l'intercorrélacion pour bien faire ressortir l'identification des positions des maxima (Fig. 2, en bas à gauche : les deux points blancs sur les lettres 'a' en haut à gauche de l'image et en bas à droite sont l'identification des positions de maxima).

Nous pouvons nous convaincre de la validité de l'approche en passant par le domaine spectral

```
dtex_z=imread('texte.ppm'); dtex_z=dtex_z-mean(mean(dtex_z));
da_z=imread('a.ppm');      da_z=da_z-mean(mean(da_z));
tfa_z=(fft2(da_z,232,619));
tftex_z=(fft2(dtex_z));
p=conj(tfa_z).*tftex_z;
image=abs(iff2(p));
s=(image>0.70*max(max(image)));
imagesc(dtex_z+2*s);colormap gray % doit etre decale' de 18,18
```

Nous effectuons la transformée de Fourier (fonction `fft2()` de GNU/Octave pour l'application sur une image bidimensionnelle) du texte (par défaut, GNU/Octave génère une matrice de mêmes dimensions que le motif original, ici 619×232 pixels), et la transformée de Fourier du motif en précisant la taille de la matrice résultante. Une fois ces deux matrices obtenues (dans le domaine spectral), le produit terme à terme (opposé au produit matriciel du symbole $*$) s'obtient par `p=conj(tfa_z).*tftex_z`, avec `conj()` le complexe conjugué (donc inversion de l'axe temporel pour la variable qui lui est associée). Finalement, nous retournons dans le domaine spatial par transformée de Fourier inverse (`iff2()`) et affichage comme précédemment. Cette fois (Fig. 2, en bas à droite) les maxima sont légèrement décalés par rapport aux lettres 'a', pour cause de convention sur la position de l'origine de la transformée de Fourier.

Notons pour le lecteur désireux d'expérimenter avec des images couleurs au lieu d'images noir et blanc qu'il faut absolument réduire l'espace des couleurs (RGB) à un scalaire (tons de gris), puisque les opérations de traitement d'images de GNU/Octave ne s'appliquent que sur des matrices. On pourra soit choisir de ne conserver qu'un des canaux de couleurs, soit de convertir l'image couleur lue dans un fichier en tons de gris par `rgb2gray()`.

3 L'intercorrélacion : recherche de ressemblances dans des séquences temporelles

Au-delà des concepts nécessaires à une mise en œuvre efficace de l'intercorrélacion, le point le plus important pour exploiter cet estimateur est de concevoir qu'il s'agit d'une recherche de maximum de ressemblance du motif s' dans la séquence s .

Après l'application simple de recherche de toutes les occurrences d'un motif pioché dans une image, le premier exemple concret qui va nous intéresser concerne le calcul du champ de déplacement d'un

a b c d e f g h i j k
 hello world je suis un
 test de detection de a

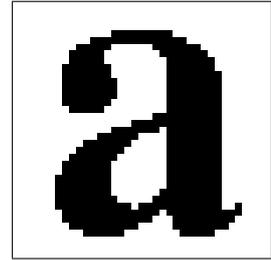


FIG. 2 – Recherche dans un texte (en haut à gauche) de motif d’une des lettres dans ce texte (en haut à droite). Les deux approches mentionnées dans le texte – passage par la convolution avec rotation de 180° d’un des motifs (en bas à gauche) ou par le domaine de Fourier (en bas à droite) donne des résultats identiques, avec un point blanc proche des symboles recherchés (les deux lettres ‘a’ recherchées sont en haut à gauche du texte et en bas à droite).

objet observé par une séquence d’images. Considérons un flux vidéo de cadence connue – dans notre cas des séquences de photographies haute résolution de glacier acquises plusieurs fois par jour – que nous désirons exploiter pour calculer le champ de déplacement [8]. Ce champ de déplacement, une fois divisé par l’intervalle de temps entre deux photographies, fournit le champ de vitesse et donc la dynamique de l’objet considéré.

Le principe de l’intercorrélacion consiste à rechercher, dans des segments de deux images d’une même objet, les vecteurs de translation qui permettent de retrouver le maximum de similarité entre les deux photos. Dans le formalisme proposé dans le chapitre précédent, il s’agit de trouver le retard t pour lequel l’intercorrélacion $\mathbf{xcorr}(t)$ est maximale. Si nous supposons que le champ de déplacement n’est pas homogène, *i.e.* que l’objet s’est déformé au cours de son déplacement, une unique recherche sur l’ensemble des pixels de l’image de départ et de fin n’a pas de sens. Il nous faut travailler sur des vignettes contenant des sous ensembles de l’image : chaque vignette doit être aussi petite que possible pour maximiser la résolution spatiale du résultat, tout en étant suffisamment grande pour garantir la présence d’une fraction suffisante d’un motif entre les images de début et de fin pour permettre à l’algorithme de reconnaître un déplacement. Si tous les motifs sont sortis de la fenêtre analysée dans l’intervalle de temps séparant les deux photos, il sera impossible de trouver t qui maximise l’intercorrélacion $\mathbf{xcorr}(t)$. Le résultat de l’implémentation de ces concepts

```
M=imread('2009-07-24_bionnassay-timelapse_LM 028.jpg');
a=.3*M(:, :, 1)+.5*M(:, :, 2)+.11*M(:, :, 3); % rgb2gray(a);
M=imread('2009-07-27_BI0nassay_timelapse_Lmoreau 031.jpg');
b=.3*M(:, :, 1)+.5*M(:, :, 2)+.11*M(:, :, 3); % rgb2gray(b);
a=a(1:2:end, 1:2:end); % reduit la taille de a (image enorme)
b=b(1:2:end, 1:2:end); % reduit la taille de b (image enorme)
[l,m]=size(a);
```

```

t=32;                % travail sur fenetres de 32x32 pixels
mm=0;
for x=t+1:t/4:m-t-1 % l-t-1
    mm=mm+1
    nn=0;
    for y=t+1:t/4:l-t-1 % m-t-1
        nn=nn+1;
        xa=double(a(y-t:y+t,x-t:x+t));
        xb=double(b(y-t:y+t,x-t:x+t)); % xcorr2 fait ce qu'il faut !
        xx=xcorr2(xa-mean(mean(xa)),xb-mean(mean(xb)));
        s=size(xx);
        m=max(xx(:));
        p=find(xx(:)==m);
        [cx(mm,nn),cy(mm,nn)]=ind2sub(s,p);
        v(mm,nn)=m;
    end
end
cx=cx-2*t;
cy=cy-2*t;

```

est l'application de l'intercorrélation sur des vignettes de 32×32 pixels entre deux images séparées d'environ une semaine d'images du glacier de Bionassay dans le massif du Mont Blanc, afin de fournir le résultat de la Fig. 3. Plus le paramètre de taille des images est grand (ici 32), plus le temps de calcul est long, et plus le champ de déplacement lisse des phénomènes potentiellement locaux. Il est donc judicieux de rechercher la plus petite fenêtre qui contienne les motifs pertinent sur toute l'image. On notera par ailleurs l'utilisation de la fonction `xcorr2()`, qui est simplement l'appel à la fonction `conv2()` suivant les concepts que nous avons explicités plus haut, et une mise en forme (normalisation) appropriée des données.

La Fig. 3 mérite une description détaillée :

- bien que le champ de déplacement soit relativement homogène sur le glacier, nous distinguons bien les zones englacées (en mouvement) du premier plan constitué du socle rocheux (immobile),
- les montagnes en arrière plan ne bougent pas,
- le ciel présente des vecteurs de déplacement aberrants, que nous aurions pu éliminer en seillant sur le maximum d'intercorrélation (vignette en bas à droite dans la série de 4 figures de droite). En effet, le ciel ne présente pas de motif représentatif d'un déplacement (soit le ciel est vierge de nuages et il n'y a pas moyen d'identifier les déplacements, soit comme ici les motifs de nuages sur un intervalle de 1 semaine sont incohérents). Cette notion de qualité de l'intercorrélation est importante, car elle permet de quantifier la validité du champ de déplacement observé [9],
- la représentation des vecteurs de déplacement se fait soit sous forme de flèches (fonction `quiver()` de GNU/Octave) dont la direction donne l'orientation du mouvement, et la longueur la magnitude du déplacement, ou (vignettes de droite, en haut) nous pouvons fournir individuellement les contributions en abscisse et ordonnée (variables `cx` et `cy`) puis la norme du vecteur déplacement (vignette en bas à gauche) qui démontre à l'évidence le mouvement du glacier. Connaissant la géométrie de l'installation (ouverture angulaire de l'appareil photo, distance de l'appareil aux différents plans de la scène) il est possible de remonter à un déplacement quantitatif en fonction de chaque saison.

Un point fondamental de la méthode d'intercorrélation pour identifier le déplacement est la disponibilité de points de repère. Dans le cas des glaciers, seuls les blocs de roche, crevasses ou bandes de Forbes permettent l'application de l'intercorrélation. Le lecteur désireux d'expérimenter ce concept par lui-même sur la Mer de Glace est encouragé à récupérer les photographies numériques disponibles par exemple à <http://www.compagniedumontblanc.fr/webcam/CMM1MERDEGLACE.jpg>, tout en étant conscient que le format de compression avec pertes JPEG et la faible résolution des images ne permettent pas un traitement automatique sur des intervalles de temps courts. Une base de données sur plus de 2 ans de ces images, acquise sur le web 4 fois par jour, est disponible auprès de l'auteur sur demande.

Parmi les utilisateurs de l'intercorrélation, la communauté de la mécanique des fluides est constamment à la recherche de méthodes d'identification du champ de déplacement des fluides. Normalement, les structures visibles sont injectées sous forme de particules ou fluorophore afin de créer les marqueurs représentatifs du mouvement du fluide. Nous allons illustrer cette méthode de mesure sur une bulle de savon [10, 11], dont il est bien connu que les motifs colorés sont représentatifs de l'épaisseur de liquide inclus entre les deux couches de surfactant qui définissent la surface de la bulle. L'hypothèse est donc que l'épaisseur de la bulle est représentative des mouvements de liquides qui s'y déroulent [12] et que

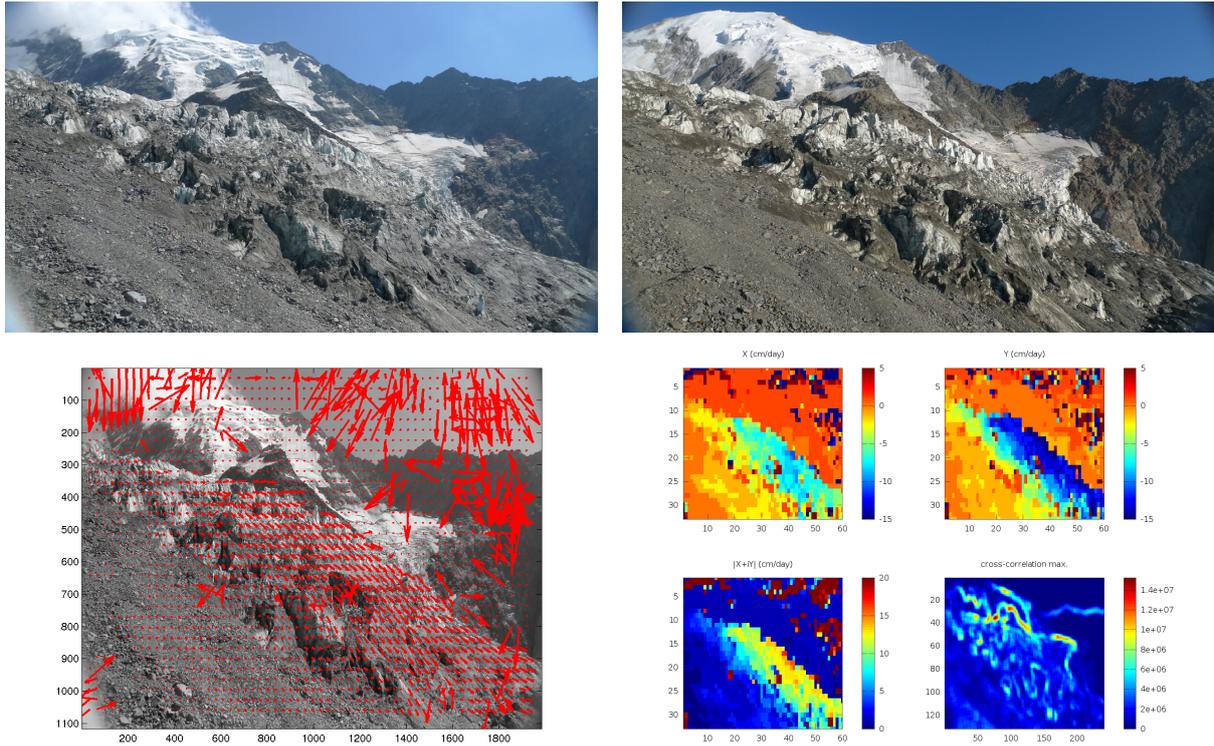


FIG. 3 – En haut : deux images séparées de une semaine du glacier de Bionassay. En bas, intercorrélation pour la détection de mouvement. La position du maximum d’intercorrélacion sur des zones de 32×32 pixels est analysée sur chaque zone de l’image à droite. En haut, les translations en X et Y, et en bas à gauche le module de la translation qui maximise l’intercorrélacion entre deux images successives. Un point fondamental de cette analyse est la qualité de l’intercorrélacion (en bas à droite) : une valeur trop faible indique l’absence de point de repères et donc une intercorrélacion non-significative. Gauche : les mêmes informations de translation en X et Y maximisant l’intercorrélacion, représentées sous forme de champ de déplacement (fonction `quiver()` de GNU/Octave). Les flèches aberrantes en haut à droite de cette image sont associées à une qualité de mesure faible et donc non-significative (absence de points de repère sur le ciel uniforme).

l’on peut y appliquer les règles de description eulérienne du fluide [13, p.99] dans laquelle un observateur fixe regarde passer des particules mobiles de fluide et en étudie la statistique en un point donné (Fig. 4).

Une fois le champ de vitesses calculé, la porte est ouverte à une exploitation quantitative des séquences d’images, notamment pour un calcul de l’évolution de la vorticité sur ces fluides [13, p.127] : pour un champ de vitesse bidimensionnel \vec{v} , le calcul du rotationnel $\vec{R} = \vec{\nabla} \wedge \vec{v}$ ne fournit que une composante normale au plan de l’écoulement $R_z = \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}$. Ce calcul est effectué, depuis la version 3.4 de GNU/Octave (qui n’est pas encore disponible en paquet Debian à la date de rédaction de ce document) par la fonction `curl(cx,cy)`, dont une illustration est proposée Fig. 5. Une extension de ce calcul est la validation expérimentale du théorème de Stokes [14, p.200], qui affirme que l’intégrale sur une surface fermée A de la vorticité doit être égale à l’intégrale de la composante du champ de vitesse le long du contour l définissant cette surface : $\iint_A R_z dA = \int_l \vec{v} \cdot d\vec{l}$.

Une autre application pratique de ces concepts est la proposition d’encodage dans un certain nombre de versions récentes de MPEG : au delà du codage des différences entre images JPEG successives (cas du MPEG1), l’objectif serait de coder les déplacements de zones d’images, sans recoder les parties statiques de l’image qui sont redondantes dans un flux vidéo ni recoder l’objet lui même (MPEG4). Nous avons cependant vu que la puissance de calcul pour appliquer ces méthodes est considérable, surtout pour une application en temps réel pour la transmission d’un flux vidéo.

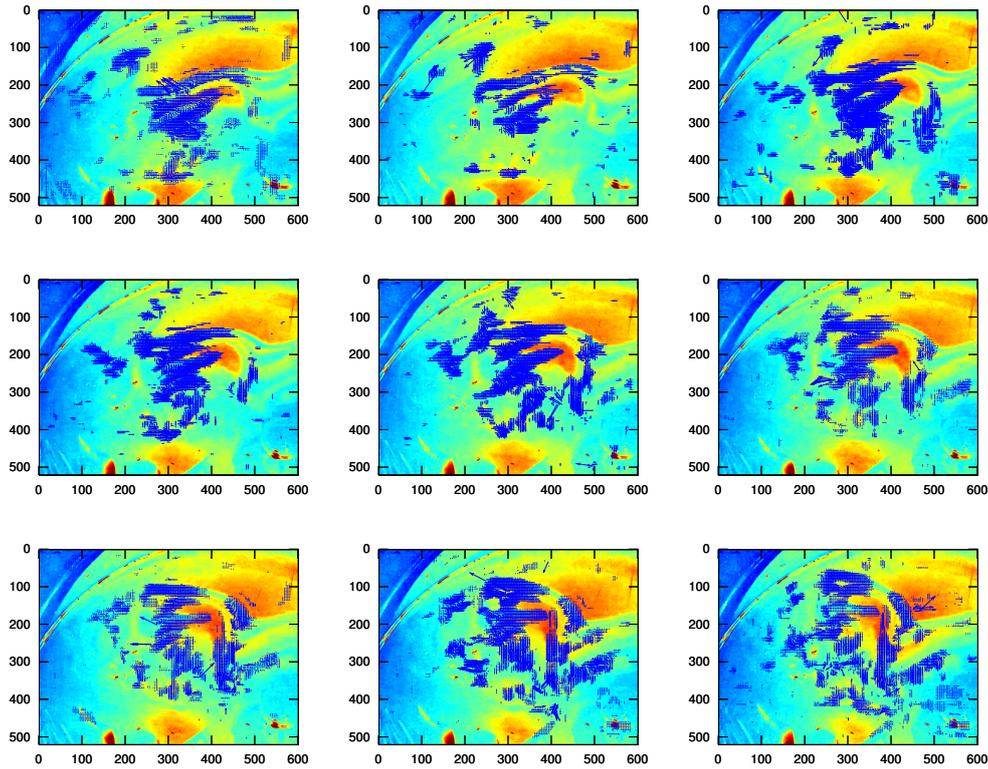


FIG. 4 – Champ de déplacement du fluide dans une bulle de savon suivie comme intercorrélation de zones de 16×16 pixels. L’hypothèse que la couleur des franges d’interférences optiques dans la membrane est représentative des mouvements de fluides permet de rechercher le déplacement de ces motifs. Une fois ces champs de vitesse calculés, nous pouvons y appliquer les méthodes de mesure classiques de mécanique des fluides (conservation du débit, évolution de la vortacité) pour valider expérimentalement les hypothèses derrière certains théorèmes. Pour cet exemple, nous n’avons conservé que la composante rouge des images, le passage en tons de gris fournissant un résultat moins contrasté et donc un calcul du champ de vitesses plus bruité.

4 L’autocorrélation

Un cas particulier qui va nous amener au cœur du sujet de cet exposé est le cas de l’intercorrélation pour des signaux s et s' identiques. On parle alors d’*autocorrélation*, dont nous allons voir la puissance dans l’exemple qui suit.

Une approche intuitive des performances de l’autocorrélation est proposée Fig. 6. Dans cet exemple, un phénomène physique supposé aléatoire subit deux retards avant d’être observé par l’opérateur. Notre objectif est d’identifier ces retards et donc les constantes de temps des réservoirs affectant la propagation du signal. La Fig. 6 (haut) propose un signal aléatoire décalé de 30 et 100 unités arbitraire généré par

```
x=rand(10000,1); x=x-mean(x); % signal aleatoire de moyenne nulle
xx=zeros(10100,1);
xx(1:10000)=x; % signal physique
xx(31:10030)=xx(31:10030)+x; % decalage de 30 unites de temps
xx(101:10100)=xx(101:10100)+x; % decalage de 100 unites
final=xx(101:10000); % observation
```

Ce signal apparaît, pour l’observateur non-averti, aléatoire. Cependant, un calcul d’autocorrélation tel que proposé sur la Fig. 6 (bas) démontre la présence de 3 maxima locaux de chaque côté du maximum correspondant au décalage nul (abscisse 900, le signal se ressemble à lui même en l’absence de translation).

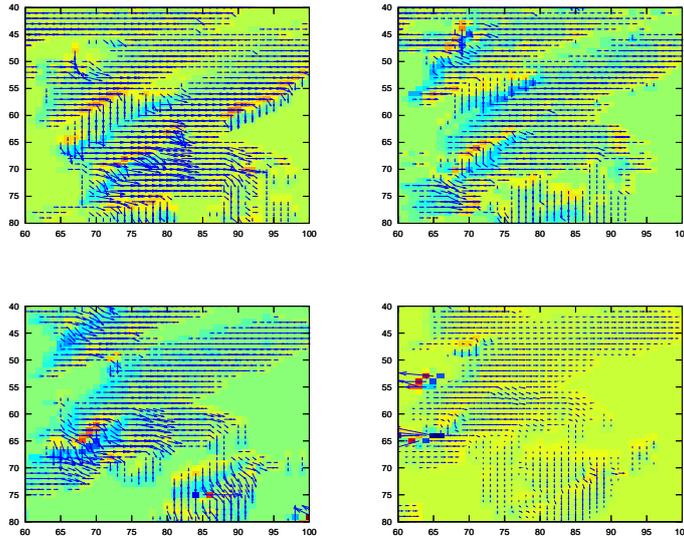


FIG. 5 – Comparaison du champ de vitesse \vec{v} (flèches) calculé à partir d’images successives d’une bulle de savon, et du champ de vorticité R_z obtenu par la fonction `curl` de GNU/Octave tel que décrit dans le texte (indexation par la couleur de fond). Nous constatons que le champ de vorticité est maximum dans les régions de changement de direction du mouvement du fluide, *i.e.* dans les régions où les éléments de fluides sont le plus “déformés” dans l’écoulement. Le calcul n’a été effectué que sur la zone centrale des 4 images du milieu de la Fig. 4.

Ces maxima sont situés en abscisse ± 30 et ± 100 (retard que nous avons choisi pour synthétiser ce signal), et un troisième maximum en 70 correspond à la ressemblance entre les deux signaux retardés de $\tau_1 = 30$ et $\tau_2 = 100$. Ainsi, la recherche d’une ressemblance au sein du signal permet de retrouver les retards qui se superposent au signal original. Cependant, au delà de l’aspect purement retard (présence ou absence de pic d’autocorrélation en dehors du retard nul), l’aspect de l’autocorrélation va nous indiquer si un retard est aussi associé à une “dilution” du signal (si par exemple le signal initial est non seulement retardé mais aussi emmagasiné avant restitution).

L’objectif de cette section est de démontrer l’inutilité – voir le danger – de vouloir changer les propriétés statistiques du signal dans l’objectif naïf d’en cacher le contenu. Tenter de soustraire des informations par le secret fournit au lecteur susceptible d’une recherche statistique sur la masse de données d’identifier une rupture des propriétés moyennes de la scène observée, et donc de focaliser son attention sur les quelques zones qui ont été suffisamment dignes d’intérêt pour mériter une manipulation manuelle. Nous allons en particulier démontrer la recherche des sites floutés sur Google Maps sur le territoire français [15, 16].

4.1 Obtention des données

Notre objectif est de traiter les images disponibles sur Google Maps en vue d’y rechercher les sites floutés. La licence d’exploitation de la base de données des cartes Google interdit de récupérer ces photos sans les afficher sur une carte ¹, donc nous allons rechercher sur le serveur de Google une photo du puzzle, l’afficher, la traiter puis l’effacer.

Afin d’automatiser ce processus, il nous faut connaître l’url pour rechercher une pièce du puzzle par une requête HTTP contenant les coordonnées de l’image à obtenir. En allant sur `maps.google.com` et en visualisant la région qui nous intéresse au zoom approprié, nous pouvons connaître l’url de chaque pièce composant une page web de la façon suivante :

¹<http://code.google.com/apis/maps/documentation/geocoding/>

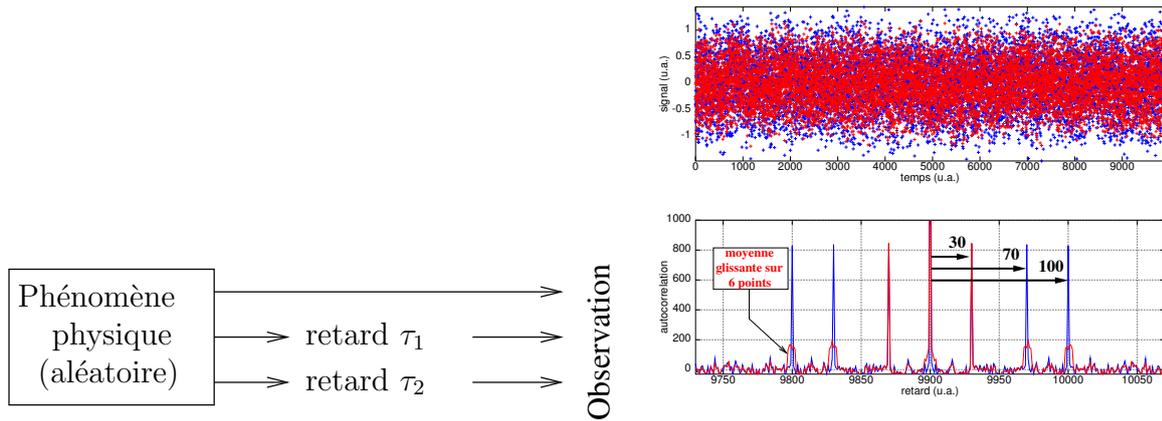


FIG. 6 – Gauche : principe de la mesure d’un signal physique retardé par deux délais τ_1 et τ_2 que nous cherchons à déterminer. Droite : en haut un signal aléatoire synthétisé avec deux retards $\tau_1 = 30$ et $\tau_2 = 100$ – apparemment lui même aléatoire par son aspect visuel (bleu) – et en bas l’auto-corrélation du signal qui présente 3 minima locaux, aux deux retards prévus et leur combinaison linéaire (bleu). Si le signal retardé de $\tau_2 = 100$ est en plus dilué (moyenne glissante sur 6 points, en rouge en haut – `m=ones(6,1)` ; `m=m/sum(m)` ; `xm=conv(m,x)` ; `xm=xm(3 :length(xm)-3)` ; et finalement `xx(101 :10100)=xx(101 :10100)+xm` ; à la place de la dernière affectation dans l’exemple du texte), les pics d’intercorrélacion associés sont élargis (courbe rouge).

- sous iceweasel : Tools → Page Info → Media. Nous obtenons alors une série d’urls de la forme `http://khm1.google.com/kh/v=81&x=66377&s=&y=45087&z=17&s=Ga` que nous tronquons en `http://khm1.google.com/kh/v=81&x=66377&s=&y=45087&z=17`. L’obtention des images s’obtient alors trivialement par une boucle sur `x` et `y` et une requête par `wget`. Chaque image contient 256×256 pixels et couvre une surface dépendant du facteur de grossissement (argument `z`).
- sous Google Chrome, nous exploitons parmi les menus accessibles au moyen de l’icône de la feuille cornée en haut à droite (*Control the current page*) → Developer → Developer Tools, qui propose d’afficher toutes les ressources, incluant les urls des images affichées.

Ayant identifié les chemins, nous fabriquons les urls au moyen d’un script shell qui incrémente les valeurs de `x` et `y`, obtenons les images par `wget` pour un traitement immédiat sans stockage local, limité à 2500 pièces par jour et par adresse IP pour interdire à tout contrevenant d’abuser du service fourni par Google.

4.2 Traitement des données

Prenons l’exemple d’une des pièces du puzzle obtenu au niveau de la base militaire aérienne de Luxeuil. Une analyse statistique d’une région floutée (en haut à gauche de l’image) et d’une région nette (en bas à droite) est proposée sur la Fig. 7. Pour chacune de ces régions, nous proposons l’affichage de l’autocorrélation (en bas à gauche pour la zone floutée, en bas à droite pour la zone nette, sur des images de 100×100 pixels dans les deux cas) définie comme l’intercorrélacion d’un signal avec lui-même.

Tout comme l’intercorrélacion, l’autocorrélacion recherche le taux de ressemblance entre une image, et cette même image déplacée d’un vecteur \mathbf{t} dans le plan. On peut se demander quel est l’intérêt de rechercher la ressemblance d’un signal avec lui même : le résultat d’une autocorrélacion en fonction du décalage est évidemment maximal pour un retard nul, puisque un signal se ressemble au mieux avec lui même. Cependant, les points en dehors du décalage nul présentent un intérêt que nous allons exploiter ici : plus une image est floue, plus les translatices de cette image sont proches de l’original. Au contraire, une image nette et contenant beaucoup de détails ne se ressemblera pas après translation de quelques pixels. Ce constat est visuellement illustré sur les deux graphiques en bas de la Fig. 7 : la zone floue présente une décroissance lente de l’autocorrélacion autour du vecteur de déplacement nul situé en $(100, 100)$, tandis que l’autocorrélacion de la région nette présente une distribution étroite qui chute rapidement dès que nous nous éloignons du vecteur de translation nulle. En prenant une coupe de ces distributions selon

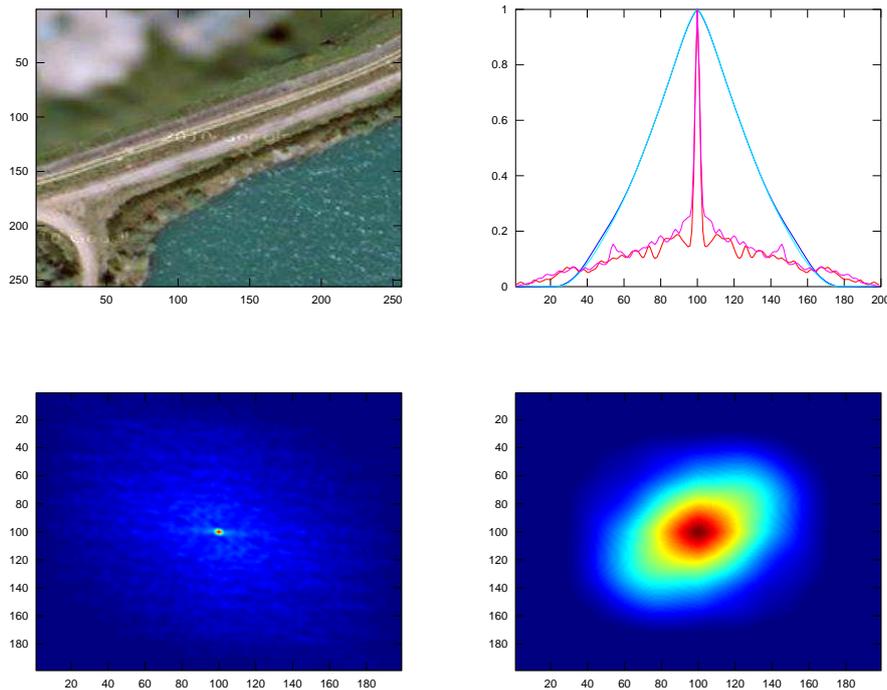


FIG. 7 – Exploitation de la fonction d'autocorrélation pour identifier une image floue par rapport à une image nette. En haut à gauche, l'image originale, avec une zone floutée en haut et une zone nette en bas. En bas : autocorrélation du coin de 100×100 pixels en haut à gauche (figure de droite) et du coin de 100×100 pixels en bas à droite (figure de gauche). En haut à droite : coupe des courbes d'autocorrélation en abscisse 101, position du maximum, pour la zone floue (bleu) et la zone nette (rouge).

l'ordonnée 100 (en haut à droite) et en normalisant par le maximum de l'autocorrélation, nous voyons apparaître un critère simple de sélection d'une zone floutée ou non. **La hauteur de l'autocorrélation à une position décalée du retard nul est d'autant plus importante que l'image est floue** (ici, le retard nul se trouve en abscisse 100). Pour chaque image, nous allons sélectionner des sous ensembles de quelques dizaines de pixels de côté (afin d'accélérer le calcul et pouvoir identifier des régions floutées au sein d'une image de 256×256 pixels obtenue sur le serveur de Google), et tracer la valeur de l'autocorrélation en des points situés à côté du maximum (translation nulle). Le choix de l'abscisse peut varier selon le rapport signal à bruit recherché, et nous avons constaté qu'il est peu judicieux d'aller trop loin sur les parties les plus faibles de la distribution. Nous nous sommes focalisés sur la mesure de la valeur de l'autocorrélation normalisée à des distances de 2 à 5% de la position du maximum (*i.e.* abscisse 95 ou 98 dans l'exemple de la Fig. 7 en haut à droite). En pratique, le script GNU/Octave rapatriant les images, affichant chaque pièce du puzzle avant d'en effacer la copie locale, puis traitement ressemble à :

```
clear all
more off
format long
page_output_immediately(1);
l=1;
m=1;
for x = 16572:16623-1
  for y= 11257:11306-1
    % fabrication de la requete http pour la piece du puzzle
    nom=['v=81&x=',num2str(x),'&s=&y=',num2str(y),'&z=15']
    url=['http://kml.google.com/kh/',nom]
    % appel au shell (commande wget) pour prendre l'image puis l'effacer
```

```

commande=['wget "',url,'"'];system(commande);
a=imread(nom);
commande=['rm "',nom,'"']; system(commande);

a=rgb2gray(a); % passage en tons de gris
imagesc(a); % affichage
% autocorrelation sur le centre de l'image
aa=a(131:150,131:150)-mean(mean(a(31:50,31:50)));
b=xcorr2(double(aa));
res3a(1,m)=b(20,20); % selectionne diverses zones de l'autocorr
res3b(1,m)=b(18,20); % pour analyse a posteriori
res3c(1,m)=b(15,30);
res3d(1,m)=b(20,18);
res3e(1,m)=b(20,15);
m=m+1;
end
m=1;l=l+1;
end

r=res3c./res3a;

rm=max(max(r)); % le plus flou est celui dont les lobes sont les plus larges
rm=rm*0.91; % le seuil doit etre ajuste' au cas pas cas pour le moment
k=find(r>rm);
r(k)=rm*2.2; % on met un point rouge pour les regions les plus floues
imagesc(r'); % affichage du resultat : valeur autocorrelation pour chaque
colorbar % imagette du puzzle

s=size(r);
[u,v]=ind2sub(s,k); % identification de quelle image a donne' le meilleur
u=u+16572-1 % score
v=v+11257-1

```

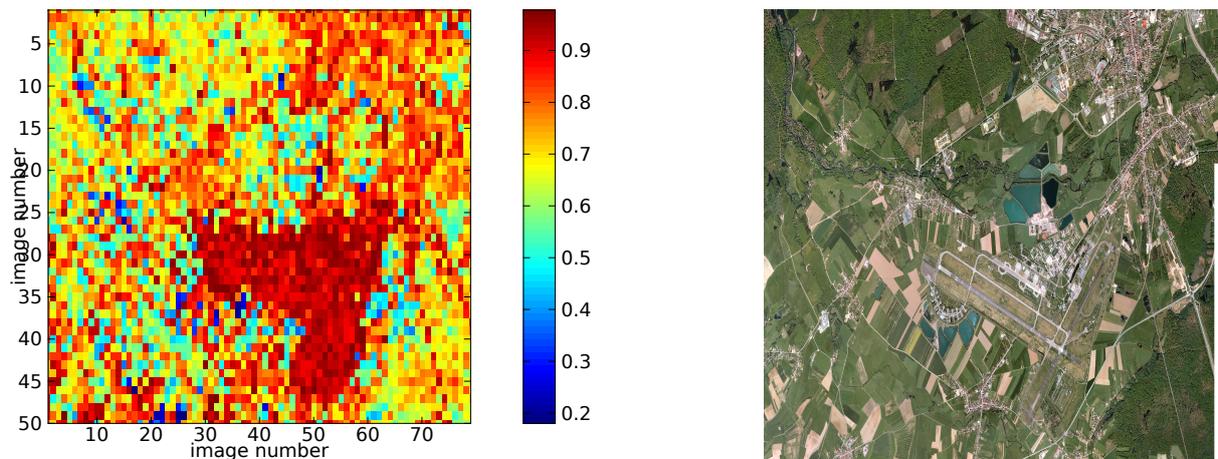


FIG. 8 – Exemple de l'application de la recherche par autocorrélation de zones floutées sur la base aérienne de Luxeuil. Un traitement automatique des images acquises sur Google Maps (droite) permet de mettre en évidence, sous forme de région rouge (valeur normalisée de l'intercorrélacion en abscisse 98 sur le graphique en haut à droite de la Fig. 7, entre les abscisses 30 à 60), les zones floutées (gauche).

Une application systématique de cet algorithme sur l'ensemble des images proposées par le serveur Google au grossissement $z=18$ fournit le résultat de la Fig. 8. L'aéroport flouté ressort en rouge sur un fond bruité mais de valeur nettement moindre de l'autocorrélation.

Le principe de fonctionnement de l'algorithme semble donc acquis, mais son exploitation au delà d'une base aérienne pour l'appliquer, avec un tel facteur grossissement, à une région plus large est impossible. En effet, couvrir une surface intéressante nécessiterait trop de requêtes à Google Maps pour permettre un résultat probant (ce qui ne veut pas dire qu'un organisme suffisamment motivé, avec suffisamment d'adresses IP à sa disposition, ne puisse pas mettre en œuvre une telle stratégie). Nous

estimons à $110 \text{ m} \times 110 \text{ m}$ la surface couverte par une image acquise sur le serveur Google au zoom 18. Ainsi, pour couvrir la majorité de la France par une analyse sur une surface de $800 \text{ km} \times 800 \text{ km}$, il faudrait effectuer 64 millions de requêtes, soit 70 ans au rythme de 2500 images/jour (en supposant que l'accès ne soit pas bloqué avant !). Nous allons donc considérer le travail sur un grossissement plus faible, au détriment des performances puisque l'effet de flouter l'image devient moins évident.

Le second exemple qui va nous intéresser concerne donc l'exploitation de cet algorithme à grossissement moindre, où la notion de flouté devient moins claire, induisant par conséquent plus de faux-positifs et un réglage plus fin de paramètres de seuillage en vue d'une identification automatique des régions floutées. Nous nous sommes proposés de travailler à zoom 15, soit sur une région où une même image de 256×256 pixels couvre une superficie 8×8 fois plus importante.

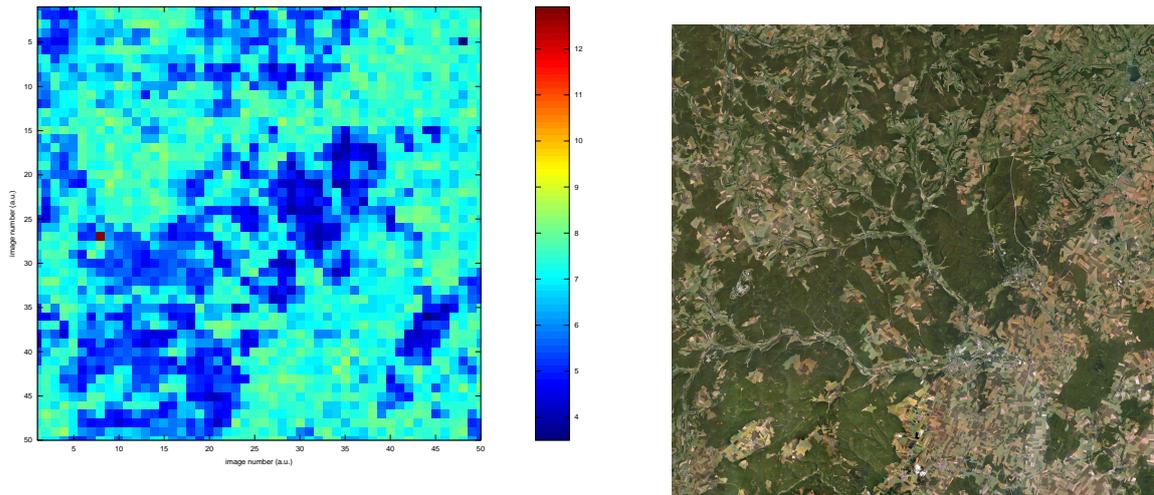


FIG. 9 – Exemple de l'application de la recherche par autocorrélation sur une zone plus large autour du site de Valduc du CEA : le centre du CEA flouté est effectivement identifié (point rouge à abscisse 7, ordonnée 27). Bien que le critère sélectionné ici (autocorrélation sur des images entières de 256×256 pixels combinée avec une autocorrélation locale sur des sous-ensembles de 20×20 pixels distribués le long des diagonales de la photo) permette une identification univoque, la majorité des tests induit un certain nombre de faux positifs qui nécessite un post-traitement manuel pour l'identification. Le rapport signal à bruit est significativement dégradé en réduisant le grossissement par rapport au cas de la Fig. 8.

Le problème d'une analyse sur une zone plus vaste (zoom 8 fois plus faible, en passant du grossissement 18 à 15) est le mauvais rapport signal à bruit associé à l'identification de zones faiblement contrastées (étendues d'eau, champs) comme des régions floutées. Nous pouvons néanmoins traiter rapidement des zones de $40 \text{ km} \times 40 \text{ km}$ (région autour du centre de Valduc du CEA, au nord-ouest de Dijon, Fig. 9) avec un taux de faux positifs de l'ordre de 0,5%. Le site flouté est bien identifié par un unique pixel représentatif d'un taux d'autocorrélation anormalement élevé, abscisse 7 et ordonnée 27. On notera sur cette même figure que au delà de la recherche des régions floutées, l'analyse par autocorrélation est un bon indicateur, au delà des analyses colorimétriques classiques, de la nature du terrain. Le pic d'autocorrélation s'avère fin dans les régions boisées présentant un fort contraste, tandis qu'il s'élargit dans les régions cultivées où le faible contraste des champs peut aller jusqu'à s'apparenter à une image floue. Un cas intermédiaire présentant un taux de faux-positifs important est autour de l'aéroport de Longvic à Dijon : bien que la zone floutée soit bien identifiée au centre de la région analysée (abscisses 23 et 24 de Fig. 10), plusieurs champs mal résolus sont identifiés comme floutés par le simple seuillage que nous appliquons.

Cette conclusion est encore plus évidente autour du site de Cadarache du CEA (Fig. 11), où la région floutée recherchée n'est pas identifiée par un traitement automatique du fait de nombreuses zones de faible contraste – notamment lacs et rivière, ainsi qu'un paire de nuages que Google n'a pas pu éliminer par l'exploitation d'une image de meilleure qualité. Ces zones de faible contraste augmentent artificiellement le seuil de détection des régions floutées, qui devient insuffisant pour identifier le centre de Cadarache.

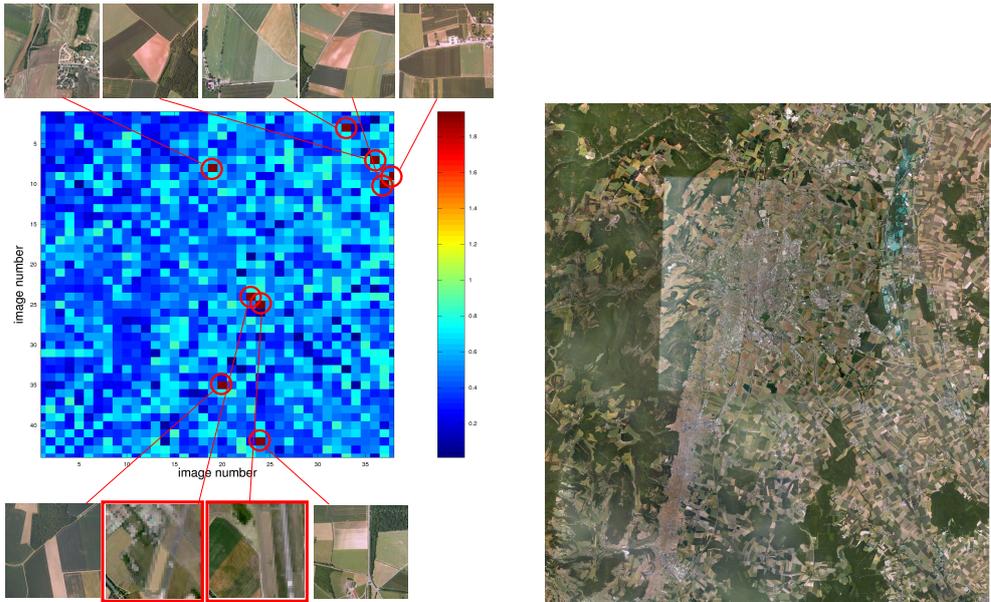


FIG. 10 – Gauche : recherche des zones floutées autour de Dijon : bien que l’aéroport de Longvic apparaisse bien en deux occurrences (abscisses 23 et 24, au centre de l’image de gauche, photographies aériennes encadrées en rouge dans le cadre), le taux de faux-positifs est important et nécessite un classement par un opérateur humain *a posteriori* de l’analyse automatique. Droite : vue d’ensemble de la région analysée : à ce grossissement, l’aéroport au sud-est de la ville de Dijon est à peine distinguable.

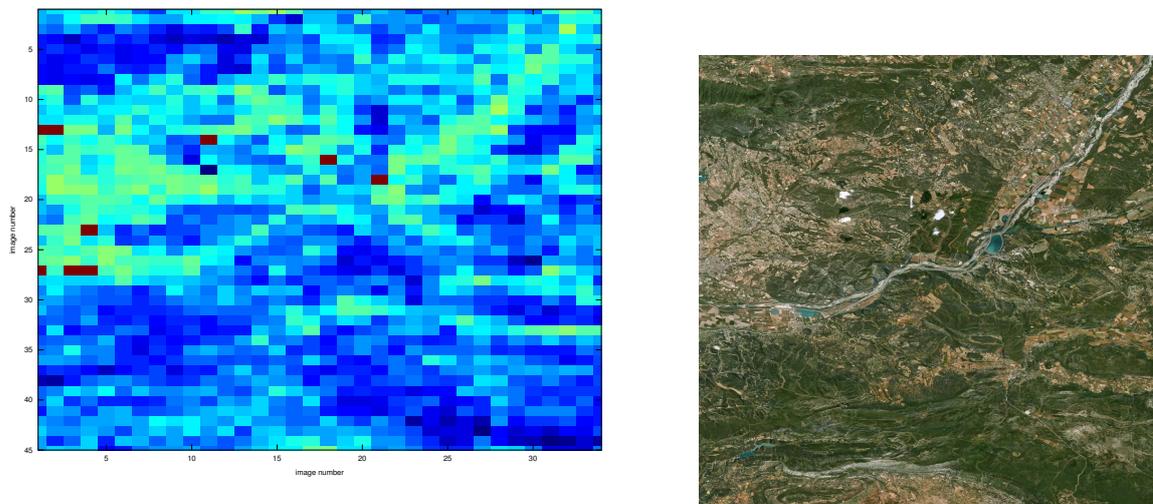


FIG. 11 – Exemple de l’application de la recherche par autocorrélation sur le site de Cadarache du CEA : la présence de nombreuses sources faiblement contrastées induit au taux de faux positifs excessif.

Il est surprenant de constater que l’application de cet algorithme sur un rayon d’environ 50 km autour de Paris ne permette pas d’identifier une seule zone floutée, et une vérification manuelle sur un certain nombre de sites que l’on pourrait considérer comme sensibles valide effectivement la netteté des images dans toutes ces régions.

Nous nous sommes jusqu’ici contents d’appliquer des outils de traitement du signal – auto et inter-corrélation – sur des données qui nous sont imposées, avec leur dynamique et leur qualité intrinsèque. Cependant, il existe de nombreux cas – dont les RADAR et le SONAR sont les plus connus – où l’utilisateur sonde son environnement avec un signal dont il choisit les propriétés statistiques. Pouvons-nous

alors améliorer les performances de la détection en choisissant un signal sonde optimisé en ce sens ?

5 La compression d'impulsion

L'intercorrélation est un estimateur classiquement utilisée pour retrouver le signal émis par un RADAR ou un SONAR dans le signal bruité enregistré par le système de réception (Fig. 1). Au-delà de la détection de l'écho, ces dispositifs s'intéressent à la localisation de la cible, qui pour la direction radiale s'obtient par une estimation du temps de vol.

Le problème de l'application d'une intercorrélation sur une impulsion délimitant une oscillation périodique est que rien ne ressemble plus à une période de sinusoïde qu'une autre période de sinusoïde (Fig. 12). Par ailleurs, en cas de décalage Doppler par une cible mobile, l'intercorrélation ne somme pas de façon cohérente le signal découlé au signal émis du fait du changement de fréquence lors de la réflexion de l'impulsion sur la cible mobile.

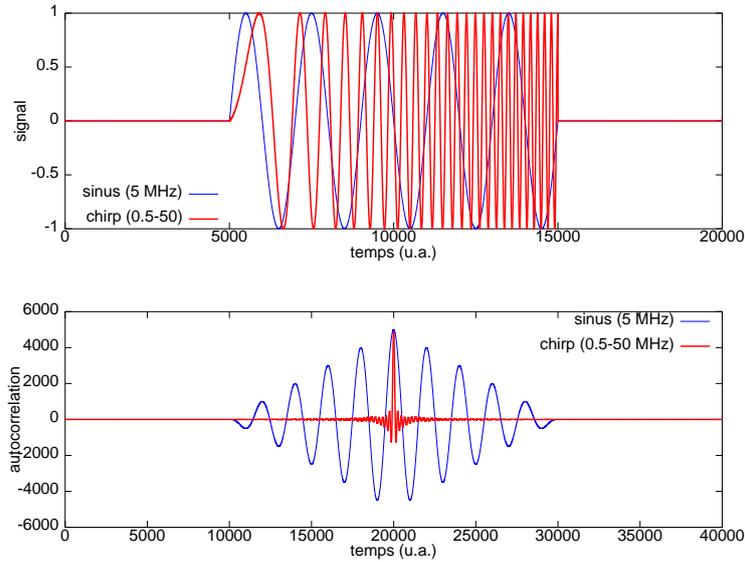


FIG. 12 – En haut : en rouge le chirp dans lequel la fréquence de la sinusoïde émise varie (linéairement) avec le temps. En traits fins, une sinusoïde de fréquence constante et de durée égale, s'apparentant au signal émis sur la Fig. 1. Grâce au balayage de fréquence, l'incertitude sur la ressemblance entre le début et la fin du signal est levée, l'autocorrélation n'est maximum que lorsque le signal du chirp se superpose à lui même (courbe rouge, en bas). Au contraire, la ressemblance de la fin de la sinusoïde (bleue en haut) avec le début de la sinusoïde étale l'intercorrélation et par conséquent ne permet pas une identification fine du temps de vol de l'impulsion.

Afin de palier à ces deux problèmes, et nous nous focaliserons sur le premier, il est d'usage de remplacer la sinusoïde formant la porteuse de l'impulsion par un signal dont la fréquence évolue dans le temps, pour générer un signal nommé le *chirp*. De cette façon, une période du signal ressemble moins à ses voisines, et l'intercorrélation saura déterminer au mieux la date de coïncidence des signaux émis retardés et du signal retour. Nous pouvons intuitivement dire que plus la variation de fréquence est rapide et dans une large plage, plus cette stratégie est efficace. Pour un balayage de la fréquence dans une bande de B Hz pendant une durée τ (secondes), nous définissons le facteur de compression comme le produit sans dimension $B \times \tau$.

Concrètement, un chirp a pour équation

$$\text{chirp}(t) = \cos(2 \times \pi(f_{\text{start}} + BW \times t/\tau) \times t) = \cos(2 \times \pi f_{\text{start}} \times t + 2 \times \pi BW \times t^2/\tau)$$

avec BW l'excurions de la fréquence de début f_{start} à la fréquence de fin du chirp pendant une durée τ . Il s'agit donc d'un balayage linéaire de la fréquence f_{start} (date 0) pour atteindre la fréquence $f_{\text{start}} + BW$

à la date τ . Si ce signal nous est retourné inchangé par la cible, alors l'intercorrélacion du signal reçu avec le signal émis est d'autant plus fine et importante que BW est grand (Fig. 13). La notion de compression d'impulsion ne se comprend donc que sur la chaîne de mesure complète, incluant le traitement du signal acquis, et pas juste sur le chirp lui-même.

Sous GNU/Octave, la démonstration de ces concepts se fait par

```
t=[0:0.1e-9:1e-6];           % duree de 1 us par pas de 100 ps
l=100;                         % taille de la fenetre pour filtre passe bas

s=chirp(t,100e6,t(end),105e6); % chirp de 100 a 105 MHz
r=conv(abs(xcorr(s,s)),ones(l,1)/l); % convolution + passe bas
plot(r/max(r),'r');
k=find((r/max(r))>0.5);(k(1)-(k(end))) % largeur a mi-hauteur
```

| f_{start} (MHz) | BW (MHz) | $\tau \times BW$ | largeur |
|-------------------|----------|------------------|---------|
| 200 | 0 | 0 (cyan) | 10031 |
| 100 | 0 | 0 (bleu) | 10017 |
| 100 | 5 | 5 (rouge) | 2471 |
| 100 | 20 | 20 (vert) | 575 |
| 100 | 100 | 100 (noir) | 125 |

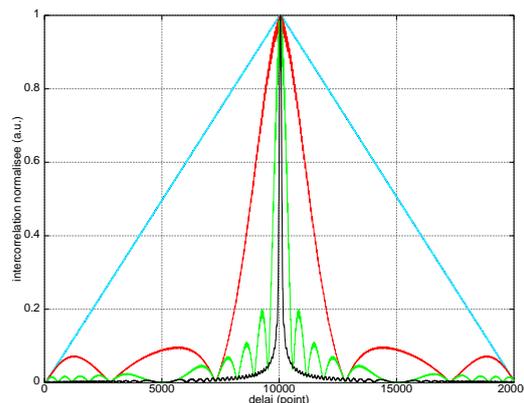


FIG. 13 – Gauche : autocorrélation de chirps de bande passante variable. Le tableau indique la largeur à mi-hauteur des diverses courbes d'intercorrélacion, démontrant l'efficacité de la compression d'impulsion en réduisant la largeur du pic d'un facteur bande passante \times durée du chirp (et donc à énergie constante, en augmentant d'autant la hauteur du pic).

6 Conclusion

Nous nous sommes efforcés de démontrer qu'une maîtrise de quelques outils simples de traitement du signal permet d'extraire un maximum d'informations disponibles, librement ou non, sur le web. Nous avons illustré ces idées dans un premier temps avec de la reconnaissance de position de motifs dans une image, avant de passer au calcul d'un champ de déplacement entre deux images d'un même site prises avec un intervalle de temps de l'ordre de la semaine. Après avoir exploité l'intercorrélacion, nous avons démontré la richesse d'un estimateur moins courant, l'autocorrélacion, pour la recherche de zones floutées sur des images aériennes. Enfin, nous avons montré comment un choix judicieux du signal sonde, lorsque l'utilisateur a la liberté de le synthétiser avec les propriétés statistiques requises, permet de considérablement améliorer le rapport signal à bruit et la localisation du signal émis dans un enregistrement bruité d'un récepteur.

Tous ces concepts méritent à être approfondis afin d'extraire le maximum d'informations de la masse de données disparates disponibles sur le web, et ainsi être maître de l'information plutôt que esclave d'un flot continu de données au rapport signal à bruit souvent bien faible. Seul expérimenter sur ses propres problèmes permet de s'appropriier les concepts présentés dans ce document : le traitement du signal ne nécessite aucun autre investissement que de la motivation et du temps.

Remerciements

Luc Moreau (www.moreauluc.com), glaciologue indépendant rattaché au laboratoire EDYTEM de Chambéry, a fourni les images du glacier de Bionassay pour leur traitement, ainsi qu'une motivation à

ce travail en installant des appareils photos automatisés dans divers sites du massif du Mont Blanc. Les discussions avec D. Laffly (Univ. Toulouse 2) ont été à l'origine du premier exemple sur la localisation de lettres dans un texte. C. Marlin (Univ. Paris 11) a proposé l'identification par autocorrélation des délais du signal retardé.

Références

- [1] l'accès aux previews des satellites dont les images sont commercialisées en pleine résolution par DigitalGlobe est disponible à <http://browse.digitalglobe.com/imagefinder/main.jsp>
- [2] P.T. Fretwell & P.N. Trathan, *Penguins from space : faecal stains reveal the location of emperor penguin colonies*, *Global Ecol. Biogeogr.* **18**, 543-552 (2009)
- [3] J.-M Friedt, *Affichage et traitement de données au moyen de logiciels libres*, *GNU/Linux Magazine France* **111** (Dec. 2008)
- [4] D. Kahn, *The codebreakers – the story of secret writing*, The MacMillan Company (1973)
- [5] F. Roddier, *Distributions and transformation de Fourier*, Edisciences International, Paris (1978) [chapitre 11]
- [6] un point de vue historique sur les performances d'un ordinateur nécessaires à ces calculs : Y. Granclaude & J.Y. Ranchin, *Corrélation et convolution par transformées entières*, actes du Septième colloque sur le traitement du signal et ses application (Nice, 1979), disponible à <http://documents.irevues.inist.fr/bitstream/handle/2042/10373/AR66.pdf>
- [7] une autre excellente illustration des concepts présentés dans le premier chapitre de ce document : <http://www.mathworks.com/help/toolbox/images/f21-17064.html>
- [8] J.-M Friedt, C. Ferrandez, G. Martin, L. Moreau, M. Griselin, E. Bernard, D. Laffly & C. Marlin, *Automated high resolution image acquisition in polar regions*, European Geosciences Union (2008), disponible à http://jmfriedt.free.fr/EGU_jmfriedt2008.pdf
- [9] J. Korona, É. Berthier, M. Bernard, F. Rémy & É. Thouvenot, *SPIRIT. SPOT 5 stereoscopic survey of Polar Ice : Reference Images and Topographies during the fourth International Polar Year (20072009)*, *ISPRS Journal of Photogrammetry and Remote Sensing* **64** 204-212 (2009)
- [10] Y. Couder, J.-M Chomaz & M. Rabaud, *On the hydrodynamics of soap films*, *Physica D* **37** (1989) 384-405, disponible à yakari.polytechnique.fr/people/jmarc/pdf/paru/PhyD1989Yves
- [11] J.-M Chomaz & B. Cathalau, *Soap films as two-dimensional classical fluids*, *Physical Review A* **41** (4), (1990) 2243-2245, disponible à <http://www.off-ladhyx.polytechnique.fr/people/jmarc/noticelp.html>
- [12] O. Greffier, Y. Amarouchene, & H. Kellay, *Thickness Fluctuations in Turbulent Soap Films*, *Phys. Rev. Letters* **88** (19), (2002) 194101, disponible à www.cpmoh.cnrs.fr/IMG/pdf/ThicknessPRL2002.pdf
- [13] E. Guyon, J.-P. Hulin & L. Petit, *Hydrodynamique physique*, Interéditions/Éditions du CNRS (1991)
- [14] R.W. Fox & A.T. McDonald, *Introduction to fluid mechanics*, John Wiley & Sons (1994)
- [15] <http://www.wired.com/dangerroom/2008/06/every-military/>
- [16] <http://www.gearthacks.com/dlcat19/Airports,-Air-Force-Bases,-Space-Centers.htm>
comparaison avec ikonos/previews