

N-642

CODAGE D'IMAGES SPATIO-TEMPORELLES

S. BEUCHER

FONTAINEBLEAU

FEVRIER 1980

CODAGE D'IMAGES SPATIO-TEMPORELLES

L'objet de cette note est d'apporter une ébauche de solution au problème de l'acquisition de l'information lors du suivi de corps mobiles dans un champ. Le procédé décrit n'a pas la prétention d'être le plus efficace (il n'a d'ailleurs pas été testé), mais il est intéressant en ce sens qu'il fournit à la fois des renseignements sur la trajectoire et sur la chronologie des déplacements des objets en mouvement.

I/ OBJETS EN MOUVEMENT. QUELQUES HYPOTHESES

Les objets étudiés seront supposés de forme géométrique simple (grossièrement convexes) et non déformables au cours du temps. Le nombre d'objets évoluant dans le champ à un instant donné n'est pas critique, pourvu qu'il ne soit pas trop élevé. Le champ d'analyse est une partie D de R^2 . Les objets sont supposés se mouvoir dans l'espace à deux dimensions uniquement. On interdit notamment, par cette hypothèse, le chevauchement d'un objet par un autre (figure 1)

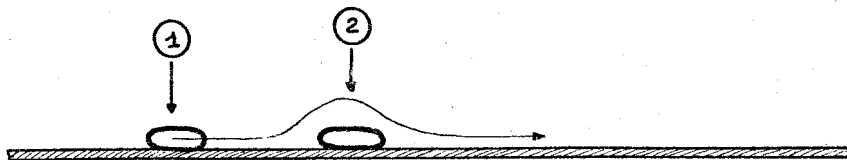


Figure 1 : Trajectoire non autorisée

Certains objets peuvent être immobiles. La trajectoire des objets mobiles sera supposée relativement simple. Nous préciserons plus tard ce qu'on entend par simple.

II/ ACQUISITION DE L'INFORMATION : DES PROBLEMES PETITS ET GROS.

L'acquisition s'effectue classiquement de la façon suivante : On dispose d'un capteur (par exemple une caméra de télévision) et on enregistre à intervalles de temps réguliers une image. L'échantillonnage est supposé instantané (cela signifie en pratique que la vitesse d'évolution des objets est suffisamment lente pour que le capteur puisse les "voir" lorsqu'ils

traversent le champ). On dispose donc après n tops d'échantillonnage de n images, dont la succession restitue, comme un film, les trajectoires et la chronologie des mouvements. Malheureusement, cette méthode implique que l'on dispose de n mémoires de stockage d'accès relativement rapide. Le nombre maximum d'images stockables est donc de quatre pour l'AT4 et de huit pour le Leitz-TAS. Si ce nombre est déjà assez important pour le TAS, une fois l'opération de stockage réalisée, il ne reste plus aucune mémoire de travail pour l'analyse des images. Inversement, pour AT4, si on peut temporairement libérer des mémoires par stockage sur "floppy" disques, le nombre d'images enregistrées semble quelque peu insuffisant. On peut alors effectuer l'union de toutes les images. Cette union n'occupe qu'une seule mémoire de l'analyseur. Malheureusement, on a perdu la chronologie des mouvements. Cette chronologie est fondamentale pour analyser les trajectoires. Pour s'en persuader, considérons les quelques exemples suivants, où pour simplifier, la trace de la trajectoire est supposée continue :

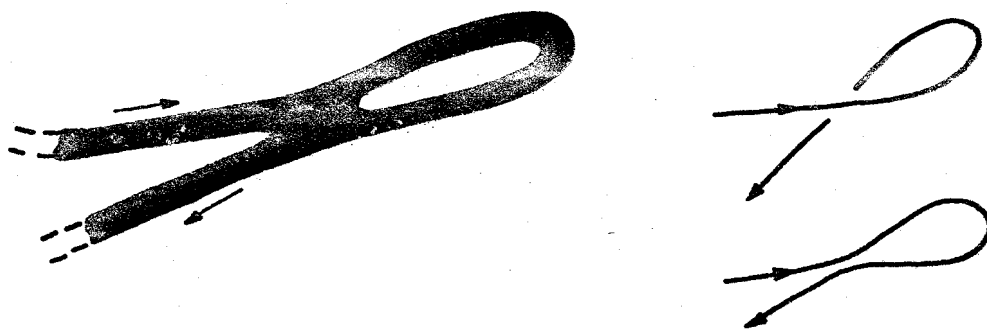


Figure 2 : Cas d'un seul mobile

La figure 2 représente la trace d'un objet unique. La trajectoire est ambiguë : Y a-t-il eu croisement ou non ? Il est alors impossible de le dire. Ce cas n'est pas trop gênant, car il n'y a qu'une seule particule. Beaucoup plus critique est le cas d'une image ou plusieurs objets évoluent

dans le champ (figure 3). Dans ce cas, il n'est pas possible d'affecter les divers arcs simples constituant la trajectoire aux mobiles qui leur correspondent.

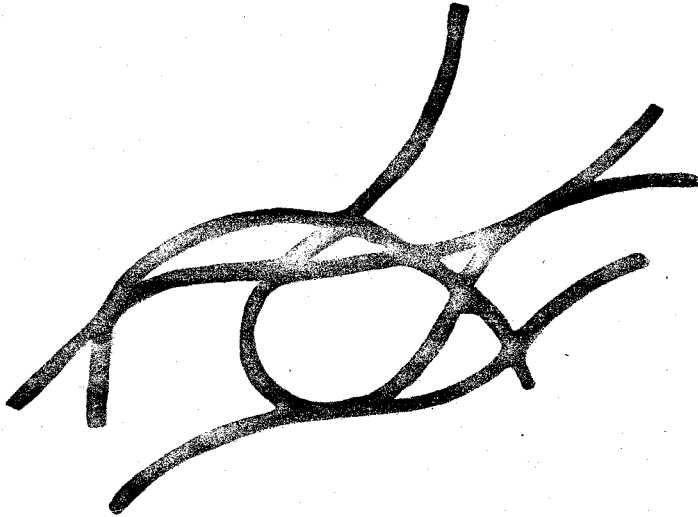


Figure 3 : Evolution de plusieurs mobiles.

Dans l'analyse d'un mouvement en général, l'objet considéré est un ensemble de l'espace $\mathbb{R}^2 \times t$. Faire l'union revient donc à projeter cet objet sur le plan \mathbb{R}^2 . La coordonnée temps t est donc, par cette opération irrémédiablement perdue. Il s'ensuit que l'équation horaire du mouvement n'est plus accessible. L'union des images conduit donc à une impasse.

Il faut donc conserver l'ensemble des images, et ceci malgré un nombre restreint de mémoires, d'où l'intérêt d'un codage.

III/ CODAGE DES IMAGES

Le codage s'effectue entre deux échantillonnages. Ce codage doit être suffisamment rapide pour ne pas trop faire baisser la fréquence d'échantillonnage. Comme l'ensemble de toutes les images ne constituent pas une partition du plan, on s'y ramène de la façon suivante :

Soit M_i l'image au temps t_i , et $\bigcup_{k=1}^{i-1} M_k$, l'union de toutes les images précédentes. Le codage va affecter la valeur i aux points de l'ensemble $M_i - \bigcup_{k=1}^{i-1} M_k$.

Le résultat final constitue bien, comme on peut s'en rendre compte sur la figure 4 une partition de l'espace.

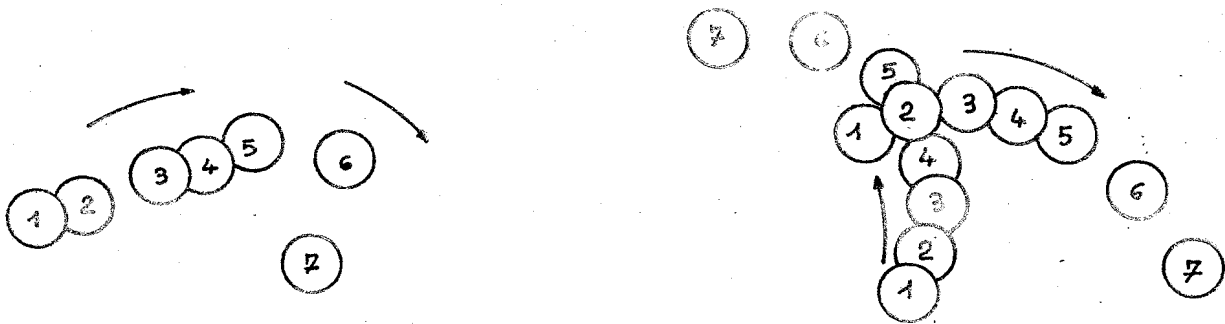


Figure 4 : Exemples de codage.

Par ce procédé, on peut donc stocker n images successives sur $\log_2(n+1)$ mémoires, puisque ce nombre correspond au nombre de bits nécessaires au codage. Ainsi trois mémoires permettent le stockage de 7 images successives.

Ce type de codage ne conserve pas toute l'information. En effet, partout où un mobile croise une trajectoire déjà enregistrée, sa position n'est pas conservée (figure 5)

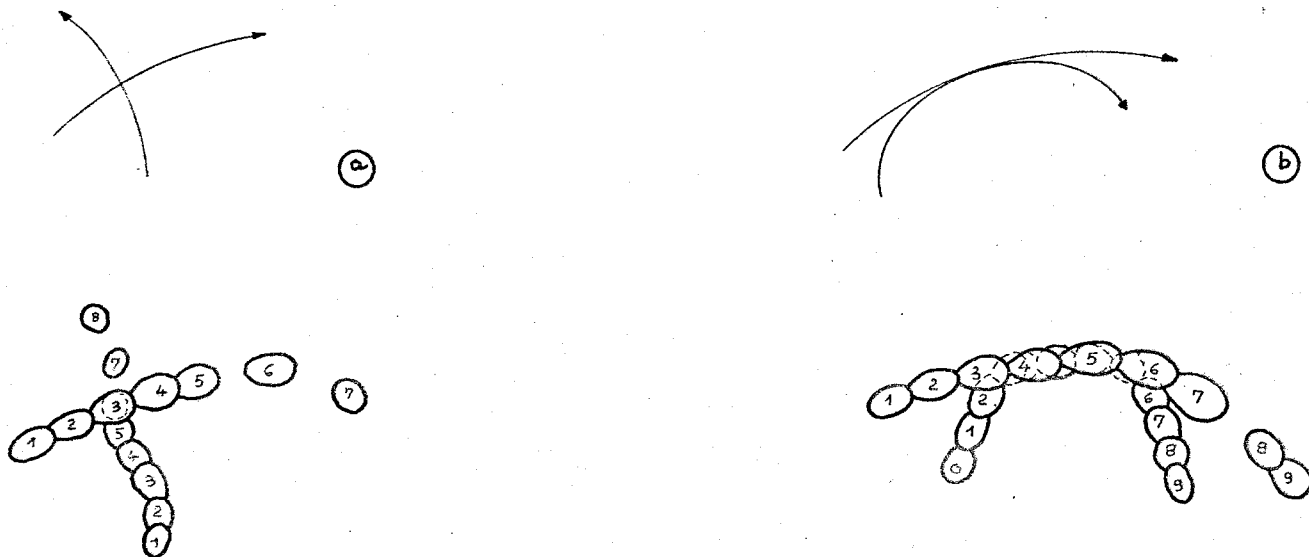


Figure 5 : Trajectoires autorisées
a) Trajectoire permise ; b) Trajectoire interdite

On est donc conduit à ne prendre en compte que les trajectoires dont l'intersection deux à deux est une union finie de points. De la même façon une trajectoire se recoupant sur une longueur non nulle est interdite.

IV/ ALGORITHMS DE CODAGE ET DE DECODAGE

1) Programmes mis en oeuvre sur AT4

Sur AT4, 4 mémoires sont disponibles. On peut donc théoriquement stocker 15 images successives. Les contraintes du codage sont les suivantes :

- l'échantillonnage doit s'effectuer à intervalles de temps constants.

- le nombre d'opérations entre deux échantillonnages doit être le plus faible possible. Si n_i est ce nombre entre les instants t_i et t_{i+1} , le temps qui s'écoulera entre deux prises d'images sera au minimum égal à :

$$\Delta t = \text{Max}_{i=1}^{14} (n_i) \times 20 \text{ ms}$$

- L'algorithme de codage doit tenir compte de la particularité de la mémoire BR, qui ne peut pas être à la fois la source et la destination d'une transformation.

La figure 6 donne le diagramme synoptique d'un codage remplissant les précédentes conditions. Le code utilisé n'est pas le code binaire pur. Mais on s'y ramène simplement en permutant et en inversant le contenu des différentes mémoires. Ce codage s'effectue en quelque sorte en "remontant le temps", puisque les images les plus anciennes ont les valeurs les plus élevées (dans la correspondance binaire).

Le nombre maximal d'opérations par échantillonnage est égal à 2. On a donc

$\Delta t = 40$ ms, soit une fréquence d'échantillonnage maximale de 25 images/s. Cette fréquence correspond à la fréquence de l'oeil. Ce codage permet donc d'analyser tout mouvement perceptible par l'oeil.

A la fin du codage, les 4 plans de bits sont stockés sur "floppy" disques, pour utilisation ultérieure.

La figure 7 donne l'algorithme de décodage. Cet algorithme de décodage comprend une procédure de test pour corriger les erreurs éventuelles de mesure du nombre de particules. En effet, une particule peut être coupée, si elle croise une trajectoire déjà utilisée (figure 8).

Transformation 1	Transformation 2	Image	Codage			
			BX	BY	BZ	BR
SV → BX, BY, BZ		1	0	0	0	0
SV ∪ BX → BX, BY	$\overline{BY} \rightarrow BY$	2	0	0	1	0
SV ∪ BX → BX	$(BX \cap BY) \cup BZ \rightarrow BZ$	3	0	1	0	0
SV ∪ BX → \overline{BX}	$\overline{BY} \rightarrow BY$	4	0	1	1	0
$(SV \cap BX) \cup BY \rightarrow BY$	$(BX \cap BY) \cup BZ \rightarrow BZ$	5	1	0	0	0
$(SV \cap BX) \cup BY \rightarrow \overline{BY}$		6	1	0	1	0
$(SV \cap BX \cap BY) \cup BZ \rightarrow BZ$	$\overline{BY} \rightarrow BY$	7	1	1	0	0
$(\overline{SV \cup BZ \cup BY}) \cap BX \rightarrow BR$	$\overline{BX} \rightarrow BX$	8	1	1	1	0
$(SV \cap BR) \cup BX \rightarrow BX$	$(BX \cap BR) \cup BZ \rightarrow BZ$	9	0	0	0	1
$(SV \cap BR) \cup BX \rightarrow BX$	$(BX \cap BR) \cup BY \rightarrow \overline{BY}$	10	0	0	1	1
$(SV \cap BR) \cup BX \rightarrow BX$	$(BX \cap BY \cap BR) \cup BZ \rightarrow BZ$	11	0	1	0	1
$(SV \cap BR) \cup BX \rightarrow \overline{BX}$	$\overline{BY} \rightarrow BY$	12	0	1	1	1
$(SV \cap BR \cap BX) \cup BY \rightarrow BY$	$(BX \cap BY \cap BR) \cup BZ \rightarrow BZ$	13	1	0	0	1
$(SV \cap BX \cap BR) \cup BY \rightarrow \overline{BY}$		14	1	0	1	1
$(SV \cap BY \cap BX \cap BR) \cup BZ \rightarrow \overline{BZ}$		15	1	1	0	1
Codage du fond			1	1	1	1

Figure 6 : Algorithme de codage

SV est le seuil de l'image i (particules en blanc)

$\overline{BX} = (BX, C0)$

Mémoire	plan de bit
BX	P ₃
BY	P ₂
BZ	P ₁
BR	P ₀

} poids fort

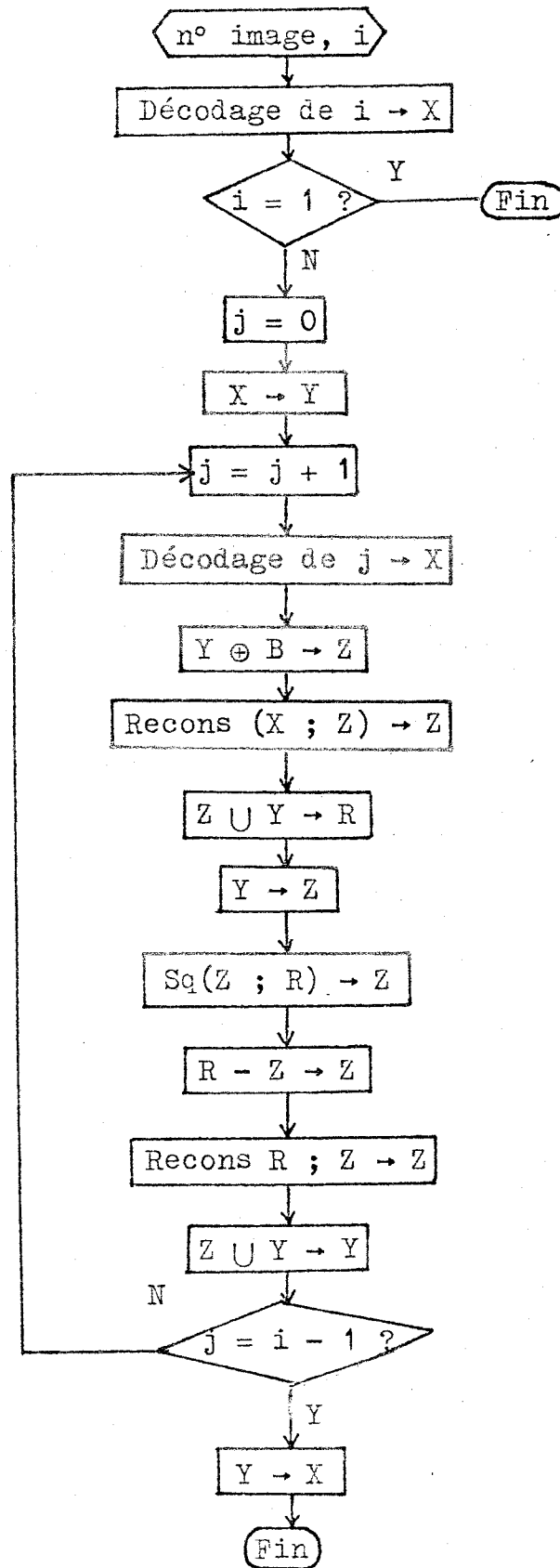


Figure 7 : Ordinogramme de Décodage (voir listing complet en Annexe)

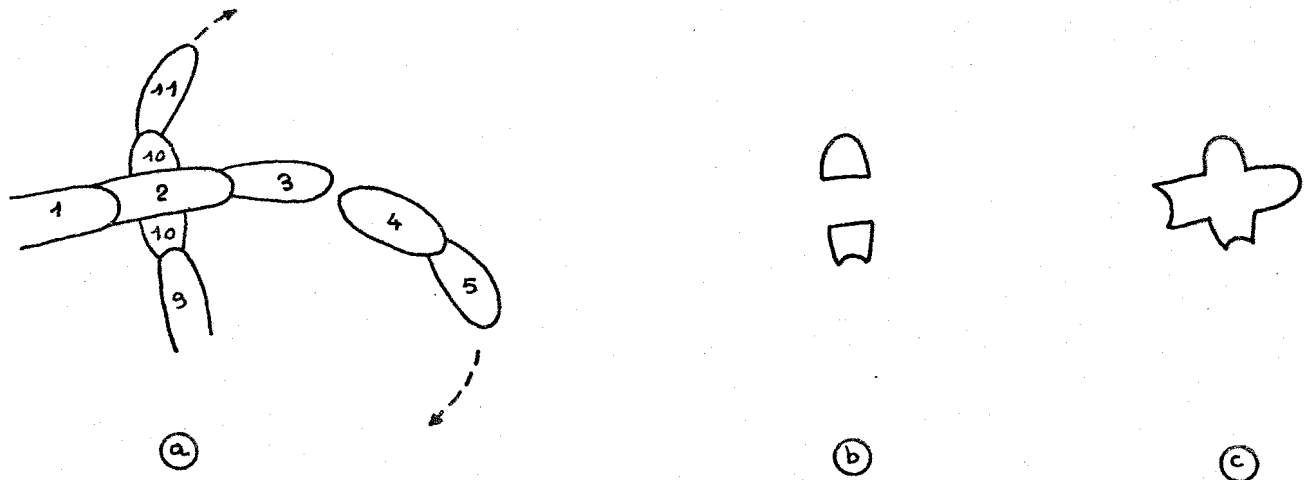


Figure 8 : Erreurs de comptage

- a) Codage ; b) décodage du niveau 10 sans correction
c) avec correction.

Cette procédure ne conserve pas la forme de la particule. Ce n'est pas gênant, car ce qui est important est sa position moyenne.

2) Les phénomènes parasites lors de l'échantillonnage d'images en mouvement.

Dans tout ce qui précède, l'acquisition d'image a été supposé instantané. Ce n'est bien sûr pas le cas en pratique. Le temps d'acquisition est égal au temps de balayage de l'image par le spot de la caméra. Il s'ensuit un phénomène de trainée sur l'image plus ou moins important selon la vitesse et le sens de déplacement de l'objet dans le champ de la caméra. Il s'ensuit une déformation des objets, d'autant plus importante que leur vitesse est proche de la vitesse de déplacement du spot de la caméra.

Le bruit sur les images seuillées ne semble pas être trop gênant. En effet, le codage a tendance à en supprimer une partie.

On trouvera en Annexe les programmes de codage et de décodage, ainsi que quelques exemples d'utilisation.

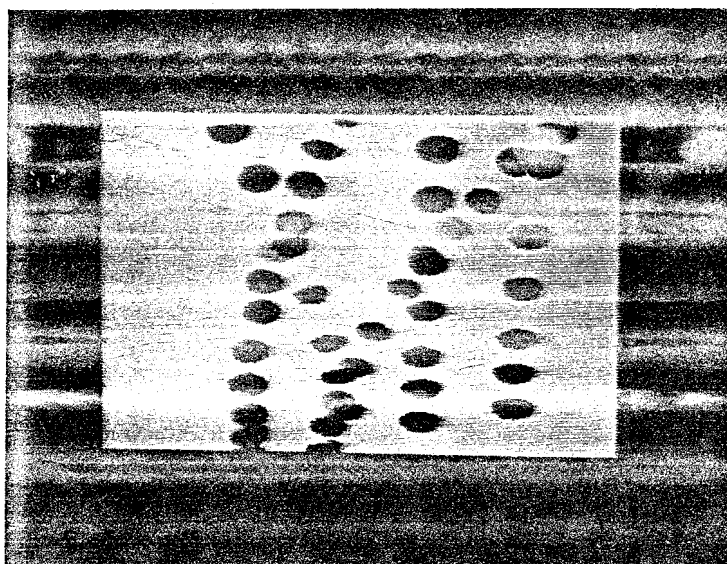
BIBLIOGRAPHIE

- H. DIGABEL : Codage, Décodage et cartographie d'images à plusieurs niveaux de gris. N-533 (Septembre 1977)
- C. LANTUEJOUL, S. BEUCHER : On the use of the geodesic metric in image analysis. N-638 (Décembre 1979).

Annexe :

On trouvera ci-après, les listings des programmes de codage et de décodage, avec un exemple d'illustration.

Le décodage est relativement long. On peut diminuer le temps de traitement, si les images ne sont pas trop compliquées, en effectuant un décodage simple. Il suffit pour cela de supprimer les lignes 150 à 295 et 360 à 465 du programme.



Exemple d'image codée.

```
0001 *
0002 * CODAGE IMAGES SPATIO-TEMPORELLES -- JANVIER 1980 -
0003 *
0004 1 JUMP 30
0005 * GENERATEUR DE DELAI
0006 10 IF (I.EQ.0)30,15
0007 15 FOR J=1,I,1
0008 20 NOL (BX)(BX)
0009 25 NEXT J
0010 30 EXIT
0011 * INITIALISATIONS
0012 50 STOP CHAMP, SEUIL(PARTICULES EN BLANC)
0013 55 NOL (SV, SC)(BX, BY, BZ, BR)
0014 60 VISU (CA, BX, BY, BZ, BR)
0015 65 TYPE DELAI EN MILLISECONDES , I
0016 70 I=1/20
0017 * DEBUT CODAGE
0018 75 STOP
0019 * IMAGE 1
0020 80 NOL (SV)(BX, BY, BZ)
0021 85 NOL (BX)(BX) DUMMY
0022 90 CALL 10
0023 * IMAGE 2
0024 95 NOL (SV, BX, UN)(BX, BY)
0025 100 NOL (BY, CO)(BY)
0026 105 CALL 10
0027 * IMAGE 3
0028 110 NOL (SV, BX, UN)(BX)
0029 115 COV 0, (BX, BZ, UN; BY, BZ, UN)(BZ)
0030 120 CALL 10
0031 * IMAGE 4
0032 125 NOL (SV, BX, UN)(BX, CO)
0033 130 NOL (BY, CO)(BY)
0034 135 CALL 10
0035 * IMAGE 5
0036 140 COV 0, (SV, BY, UN; BX, BY, UN)(BY)
0037 145 COV 0, (BX, BZ, UN; BY, BZ, UN)(BZ)
0038 150 CALL 10
0039 * IMAGE 6
0040 155 COV 0, (SV, BY, UN; BX, BY, UN)(BY, CO)
0041 160 NOL (BX)(BX) DUMMY
0042 165 CALL 10
0043 * IMAGE 7
0044 170 COV 0, (SV, BX, BY, CO; BZ, CO)(BZ, CO)
0045 175 NOL (BY, CO)(BY)
0046 180 CALL 10
0047 * IMAGE 8
0048 185 COV 0, (SV, BY, BZ, UN, CO; BX)(BR)
0049 190 NOL (BX, CO)(BX)
0050 195 CALL 10
0051 * IMAGE 9
0052 200 COV 0, (SV, BX, UN; BX, BR, UN)(BX)
0053 205 COV 0, (BX, BZ, UN; BZ, BR, UN)(BZ)
0054 210 CALL 10
0055 * IMAGE 10
0056 215 COV 0, (SV, BX, UN, BX, BR, UN)(BX)
0057 220 COV 0, (BX, BY, UN; BY, BR, UN)(BY, CO)
0058 225 CALL 10
0059 * IMAGE 11
0060 230 COV 0, (SV, BX, UN; BX, BR, UN)(BX)
0061 235 COV 0, (BX, BY, BR, CO; BZ, CO)(BZ, CO)
0062 240 CALL 10
```

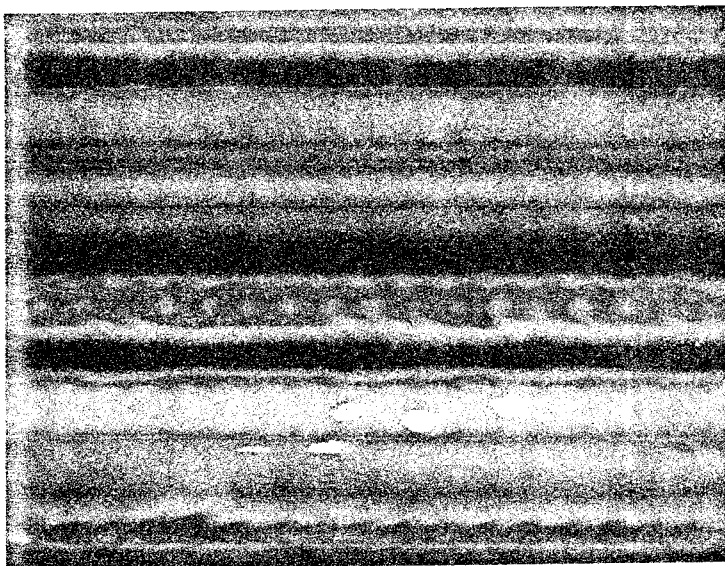
```
0063 *          IMAGE 12
0064 245 COV 0, (SV, BX, UN; BX, BR, UN) (BX, CO)
0065 250 NOL (BY, CO) (BY)
0066 255 CALL 10
0067 *          IMAGE 13
0068 260 COV 0, (SV, BX, BR, CO; BY, CO) (BY, CO)
0069 265 COV 0, (BX, BY, BR, CO; BZ, CO) (BZ, CO)
0070 270 CALL 10
0071 *          IMAGE 14
0072 275 COV 0, (SV, BX, BR, CO; BY, CO) (BY)
0073 280 NOL (BX) (BX)          DUMMY
0074 285 CALL 10
0075 *          IMAGE 15
0076 290 COV 0, (SV, BX, BY, BR, CO; BZ, CO) (BZ)
0077 295 STOP
0078 *          FIN CODAGE - STOCKAGE FLOPPY EVENTUEL
0079 300 TYPE 'STOCKAGE FLOPPY', K
0080 305 IF (K, EQ, 1) 310, 55
0081 310 TYPE 'NUMERO PLAN DE FAIBLE POIDS', L
0082 315 TYPE 'DRIVER NO', M
0083 320 IF (M, EQ, 0) 325, 335
0084 325 NDSK 0
0085 330 JUMP 345
0086 335 IF (M, EQ, 1) 340, 315
0087 340 NDSK 1
0088 345 SAVR *L
0089 350 NOL (BZ) (BR)
0090 355 L=L+1
0091 360 SAVR *L
0092 365 NOL (BY) (BR)
0093 370 L=L+1
0094 375 SAVR *L
0095 380 NOL (BX) (BR)
0096 385 L=L+1
0097 390 SAVR *L
0098 395 STOP
0099 /*
```

```
0001 *
0002 *          DECODAGE IMAGES SPATIO-TEMPORELLES -JANVIER 1980-
0003 *
0004 1 JUMP 300
0005 *          DECODAGE NIVEAU
0006 10 IF (I1.GT.8)25,15
0007 15 K1=2*(I1-1)
0008 20 JUMP 30
0009 25 K1=2*(I1-9)+1
0010 30 NOL (SV,SC,UN)(BX)
0011 35 N=4
0012 40 N=N-1
0013 45 LO=L+N
0014 50 N1=1
0015 52 IF (N.EQ.0)70,55
0016 55 FOR IO=1,N,1
0017 60 N1=2*N1
0018 65 NEXT IO
0019 70 LGDR *LO
0020 75 IF (K1.LT.N1)80,90
0021 80 COV O,(BX,BR,CO)(BX,TR)
0022 85 JUMP 100
0023 90 NOL (BX,BR)(BX)
0024 95 K1=K1-N1
0025 100 IF (N.EQ.0)105,40
0026 105 EXIT
0027 *          SOUELETTE PAR ZONE D'INFLUENCE
0028 150 K=0
0029 155 K1=K
0030 160 LDC 1,(BZ;O,11 00 )(EX,TR)
0031 165 COV O,(BZ,BX,UN;BR)(BZ,TR)
0032 170 LDC 1,(BZ;O, 11 00)(EX,TR)
0033 175 COV O,(BZ,BX,UN;BR)(BZ,TR)
0034 180 LDC 1,(BZ;O,0 11 0)(EX,TR)
0035 185 COV O,(BZ,BX,UN;BR)(BZ,TR)
0036 190 LDC 1,(BZ;O,00 11 )(EX,TR)
0037 195 COV O,(BZ,BX,UN;BR)(BZ,TR)
0038 200 LDC 1,(BZ;O, 00 11)(EX,TR)
0039 205 COV O,(BZ,BX,UN;BR)(BZ,TR)
0040 210 LDC 1,(BZ;O,1 00 1)(EX,TR)
0041 215 COV O,(BZ,BX,UN;BR)(BZ,TR)(AR,K)
0042 220 IF (K1.EQ.K)225,155
0043 *          EBARBULAGE
0044 225 K1=K
0045 230 LDC 1,(BZ; , 1111)(EX,TR)
0046 235 COV O,(BZ,BX,UN;BR)(BZ,TR)
0047 240 LDC 1,(BZ; , 1111 )(EX,TR)
0048 245 COV O,(BZ,BX,UN;BR)(BZ,TR)
0049 250 LDC 1,(BZ; ,1111 )(EX,TR)
0050 255 COV O,(BZ,BX,UN;BR)(BZ,TR)
0051 260 LDC 1,(BZ; ,111 1)(EX,TR)
0052 265 COV O,(BZ,BX,UN;BR)(BZ,TR)
0053 270 LDC 1,(BZ; ,11 11)(EX,TR)
0054 275 COV O,(BZ,BX,UN;BR)(BZ,TR)
0055 280 LDC 1,(BZ; ,1 111)(EX,TR)
0056 285 COV O,(BZ,BX,UN;BR)(BZ,TR)(AR,K)
0057 290 IF (K1.EQ.K)295,225
0058 295 EXIT
0059 *          PROGRAMME PRINCIPAL
0060 300 STOP CHAMP
0061 305 TYPE 'NO DRIVER ',M
0062 310 IF (M.EQ.0)315,325
```

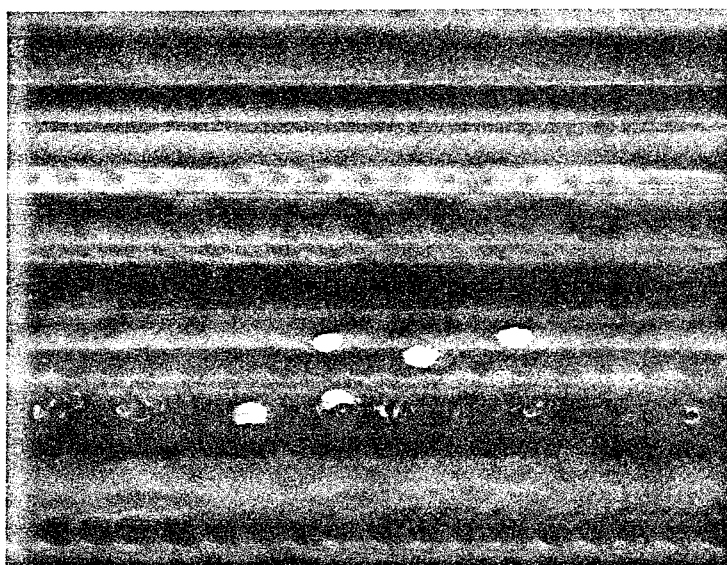
0063 315 NDSK 0
0064 320 JUMP 335
0065 325 IF (M. EQ. 1) 330, 305
0066 330 NDSK 1
0067 335 TYPE (NO PLAN FAIBLE POIDS), L
0068 340 VISU (BX, BY, BZ, BR)
0069 345 TYPE (NO IMAGE (COORD. TEMP.)), I
0070 350 I1=I
0071 355 CALL 10 DECODAGE
0072 360 IF (I. EQ. 1) 470, 365
0073 365 J1=I-1
0074 370 NOL (BX)(BY)
0075 375 FOR J=1, J1, 1
0076 380 I1=J
0077 385 CALL 10 DECODAGE
0078 390 DIL 1, (BY)(BZ, TR)
0079 395 K=0
0080 400 K1=K
0081 405 DIL 1, (BX, BZ)(BZ, TR)(AR, K)
0082 410 IF (K. EQ. K1) 415, 400
0083 415 COV 0, (BZ, BY, UN; BX, BY, UN)(BR, TR)
0084 420 NOL (BY)(BZ)
0085 425 CALL 150 SQUELETTE
0086 430 COV 0, (BR; BZ, CO)(BZ, TR)
0087 435 K=0
0088 440 K1=K
0089 445 DIL 1, (BZ, BR)(BZ, TR)(AR, K)
0090 450 IF (K. EQ. K1) 455, 440
0091 455 COV 0, (BR, BY, UN; BZ, BY, UN)(BY, TR)
0092 460 NEXT J
0093 465 NOL (BY)(BX)
0094 470 VISU (BX)
0095 475 STOP
0096 480 JUMP 340
0097 /*

Décodage de l'image :

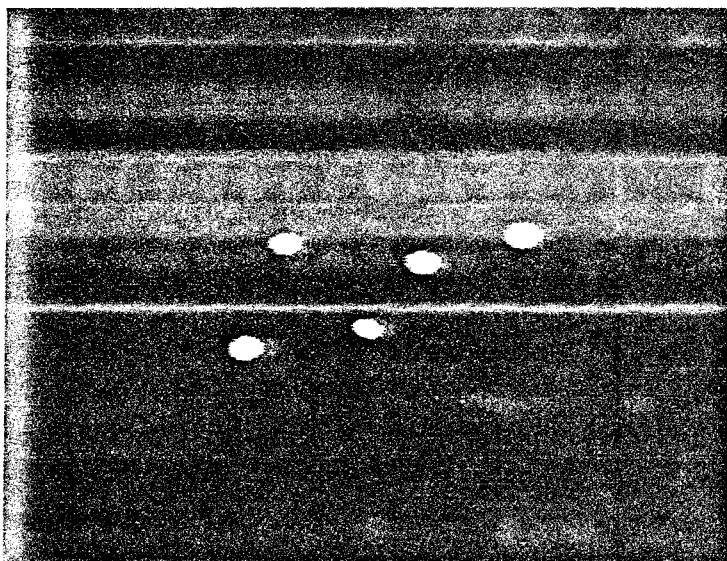
position des particules aux instants 1, 3, 5, 7, 9 et 11



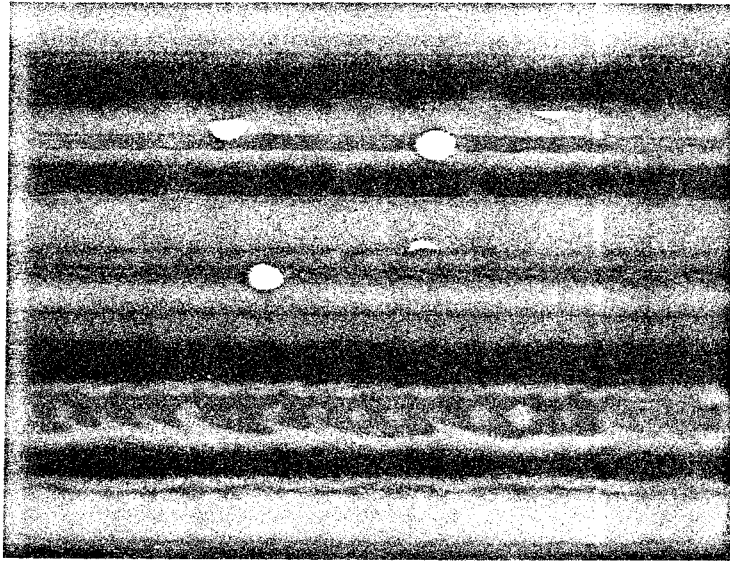
$t = 1$



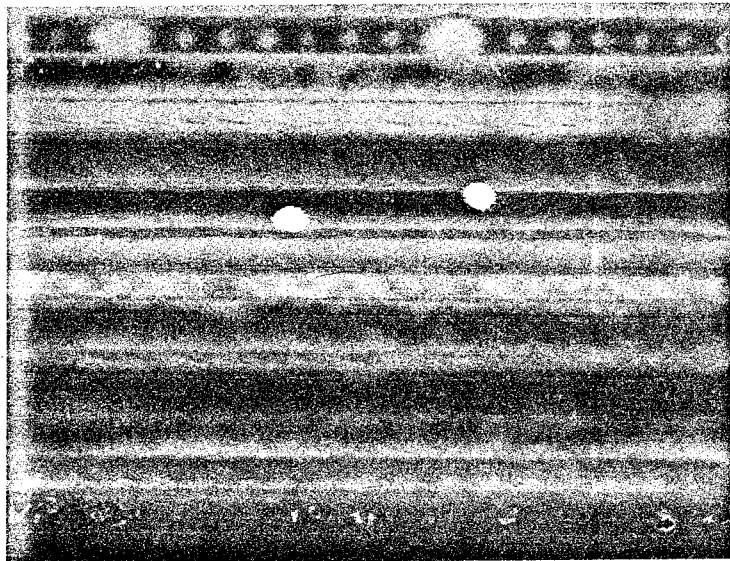
$t = 3$



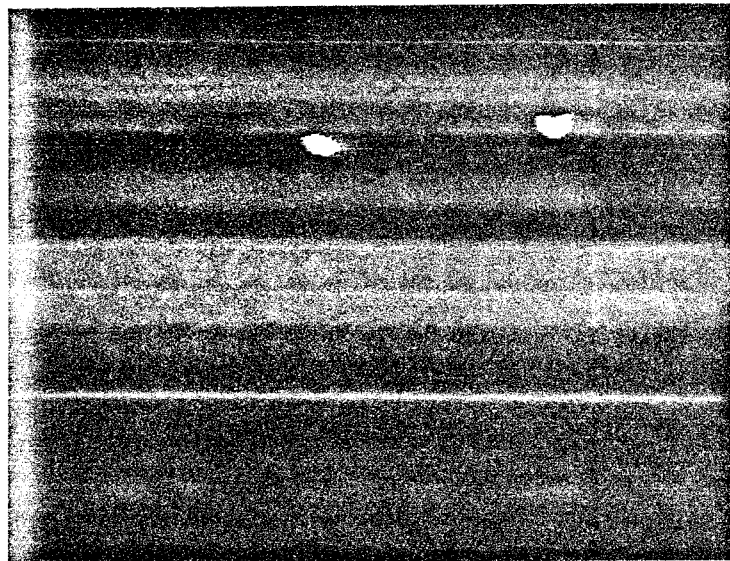
$t = 5$



t = 7



t = 9



t = 11