

TP4 ACV

L'objectif de ce TP est d'effectuer un codage inter-images afin de visualiser les mouvements présents dans la vidéo. Pour cela quelques hypothèses sont à prendre en compte : L'illumination de la scène est constante, les mouvements ne sont pas dû à l'utilisateur de la caméra. Qui plus est notre modèle de mouvement sera translationnel afin d'éviter des coût de calcul trop important.

Le principe de la prédiction temporelle du mouvement repose sur le matching de blocs entre l'image I à un instant $t+1$ et l'image à un instant t . Celui-ci se fera de la manière la plus simple possible, nous expérimentons les 3 normes L1, L2 et Linf puis nous sélectionnons celui avec la plus petite erreur.

Ainsi il nous faut des échantillons de bloc à comparer à celui présent dans l'image I à l'instant $t+1$. Pour cela nous définissons un paramètre L définissant un nombre de blocs. Nous effectuerons donc des comparaisons exhaustif dans une fenêtre de taille L x la taille du bloc.

Etant donné que nous souhaitons analyser une compensation de mouvement, notre image d'erreur représentera la différence entre les pixels des 2 blocs ayant match. Une fois la carte d'erreur obtenu, nous en profitons par la même occasion pour enregistrer la distance.

Question 3

Les temps de calcul dans les résultats qui suivent sont relativement grand car nous n'utilisons pas les fonctions Rectangles permettant de paralléliser certains calculs, qui plus est, nous utilisons .at pour accéder à nos éléments ce qui n'améliore pas nos temps de calcul. De même pour la fonction pow.

Tableau récapitulatif des résultats entre l'image 7 et 9 pour une erreur Quadratique

Taille B / Taille L	8 / 2 16*16 pixels	8 / 5 40 *40 pixels	8 / 10 80 * 80 pixels	16 / 5 80*80 pixels	16 / 10 160*160 pixels	16 / 15 240 * 240 pixels
PSNR	infini	infini	infini	infini	infini	infini
Entropie	3.31	2.674	2.483	2.994	2.95	2.93
Norme du plus grand vecteur	11.31	28.28	56.57	55.17	106.104	161.23
Temps d'exécution calcul du mouvement	3.13 s 0.2 s (sans pow)	15.65 s (avec pow) 0.99s (sans pow)	62.78 s 4.34 s (sans pow)	62.66 s (avec pow) 4.44 s (sans pow)	237.8 s (avec pow) 16 s (sans pow)	479.29 s (avec pow) 29.5 s (sans pow)

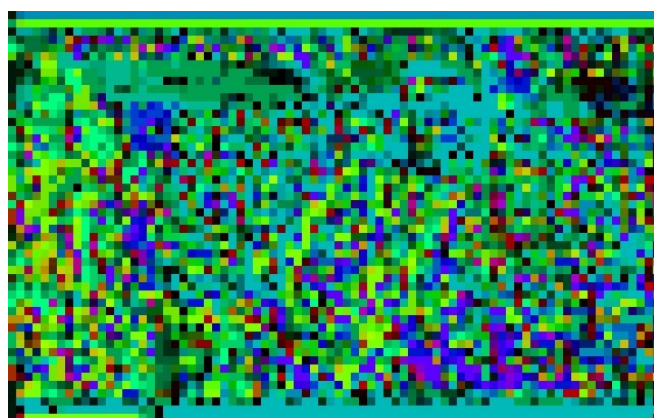
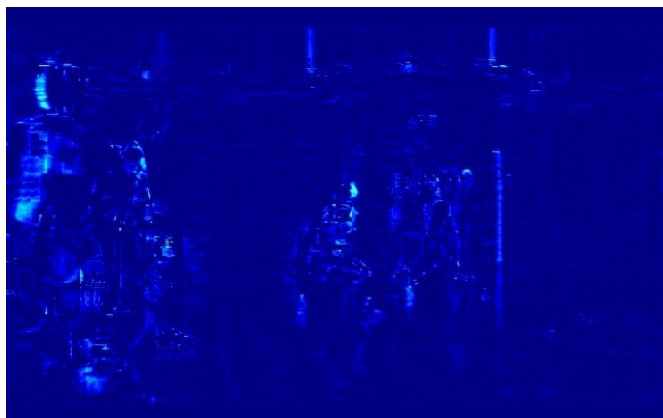
Le PSNR est infini, ceci est tout à fait normal car notre calcul est entièrement réversible grâce à la sauvegarde des cartes de mouvements et d'erreurs. Qui plus est, l'étape de quantification ne faisant pas partie du codage, nos valeurs sont parfaitement conservées dans leur état d'origine.

On remarque que la norme est presque égale au maximum dans quasiment tous les cas. Cela signifie que pour chaque application il arrive à trouver un bloc optimal proche des limites de la fenêtre définie.

On se rend compte que pour des blocs de petites tailles l'entropie est plus faible. Cela s'explique par le fait que les blocs de petites tailles permettent de capturer plus aisément les contours, et de garder une certaine homogénéité dans les mouvements. Il sera plus facile de capturer les contours. A l'inverse les grands blocs quant à eux auront de meilleurs résultats pour la mise en correspondance mais leur temps de calcul est non négligeable. Bien entendu pour une fenêtre de 16 pixels cela ne s'applique pas puisque la fenêtre de recherche est trop petite et comme le montre la norme maximale, le meilleur bloc n'est pas forcément le plus proche.

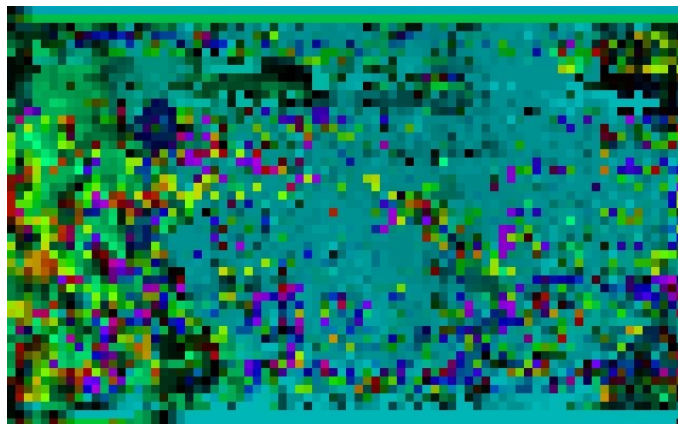
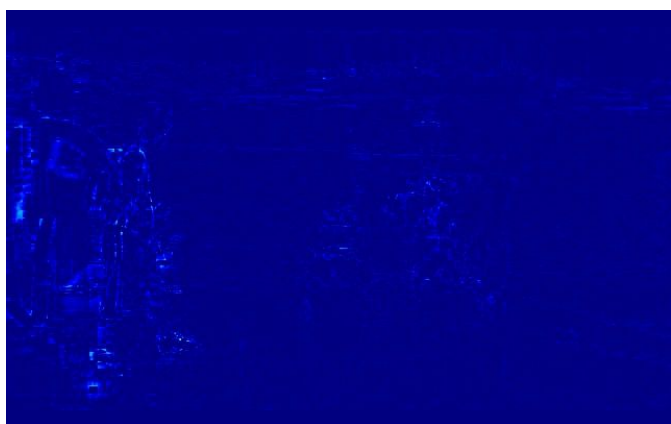
Dans les 2 cas l'augmentation de la taille de la fenêtre permet de réduire l'entropie.

Voici quelques exemples de cartes obtenues:



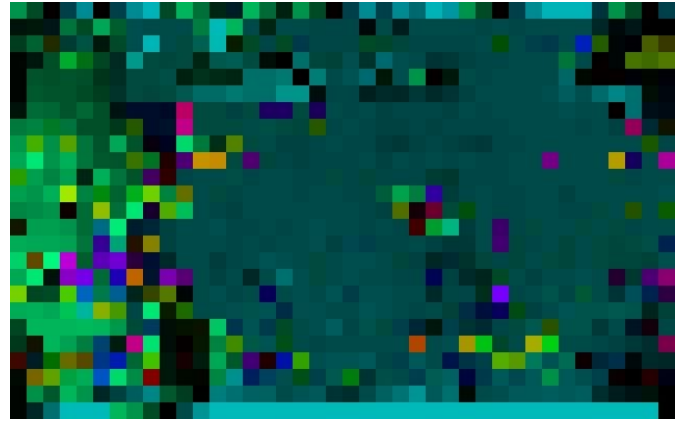
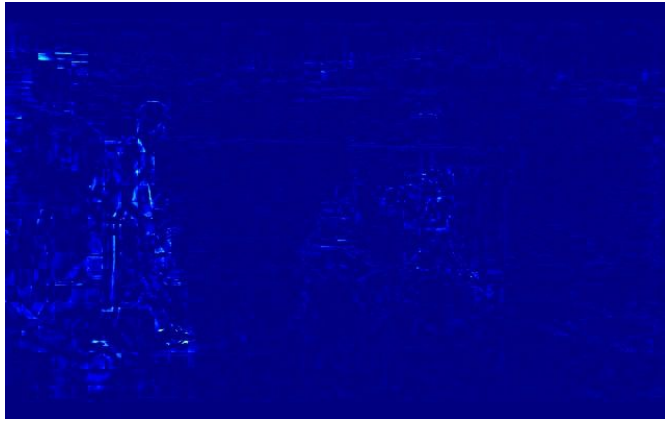
Carte d'erreur et flux 8/2
Erreur quadratique

Comme nous l'avons dit précédemment les résultats ne sont pas très représentatif du problème si nous prenons une taille de fenêtre trop petite. Cela se voit parfaitement sur la carte de flux. Celle-ci est construite à partir de la carte de mouvement. Malgré le fait que des valeurs soient bien calculées, nous ne pouvons rien en tirer car nous ne parvenons pas à en tirer un champ de mouvement constant dans l'espace.

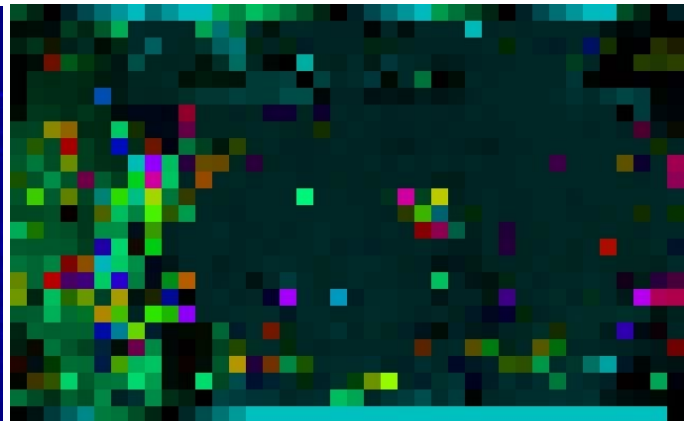
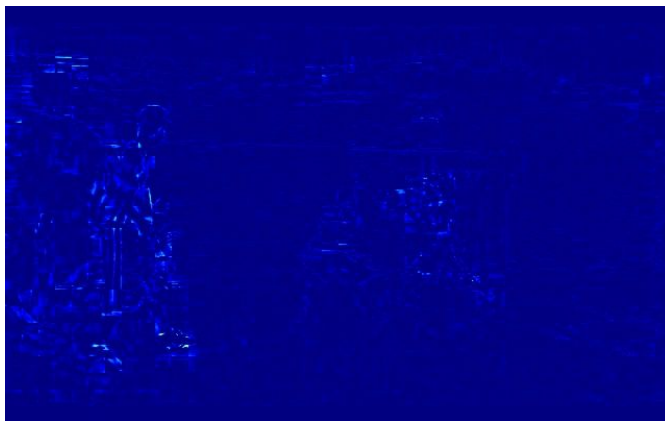


Carte d'erreur et flux 8/5
Erreur quadratique

Le résultat ici est nettement meilleur. Nous parvenons à voir parfaitement les contours des joueurs ce qui nous indique que le mouvement est important à cet endroit. Cela s'aperçoit notamment sur la carte de mouvement puisque l'on aperçoit bien deux zones relativement uniformes au centre. Le côté gauche quant à lui est mouvementé par les joueurs dans une direction opposé dû à la couleur rouge-jaune diamétralement opposé sur le cercle chromatique.



Erreur quadratique et flux 16/5



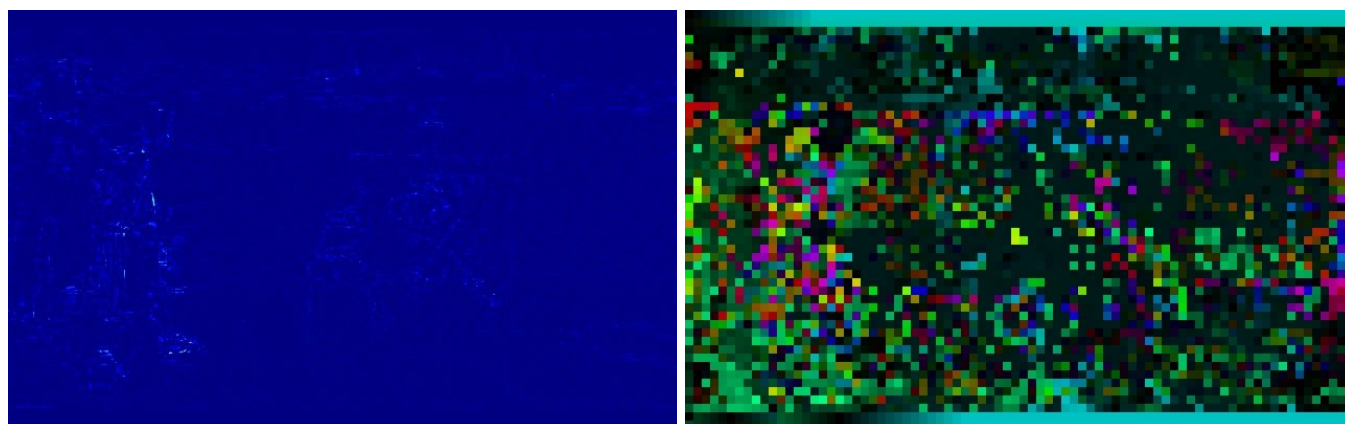
Carte d'erreur et flux 16/10
Erreur quadratique

Le flux est plus perceptible sur des blocs de taille 16, néanmoins l'inconvénient du block matching ici est que notre carte représentant le flux du mouvement se fait bloc par bloc et non pixel par pixel. On arrive à bien distinguer dans ces cas la différence de direction entre le joueur de gauche et le fond du terrain.

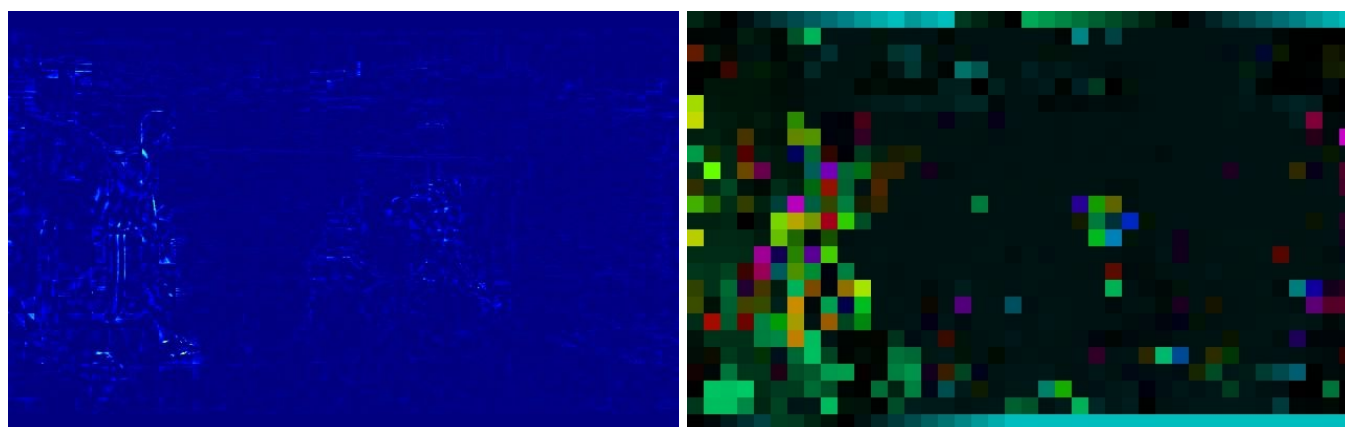
Tableau récapitulatif des résultats entre l'image 7 et 9 pour une erreur SVA

Taille B / Taille L	8 / 5 40 * 40 pixels	8 / 10 80 * 80 pixels	8 / 20 160*160 pixels	8 / 30 240*240	16 / 10 160*160 pixels	16 / 20 320*320 pixels
PSNR	infini	infini	infini	infini	infini	infini
Entropie	2.5762	2.41	2.325	2.273	2.9	2.86
Norme du plus grand vecteur	28.28	56.57	105.55	158	108.19	214.516
Temps d'exécution calcul du mouvement	0.74 s	2.6 s	9.15 s	19 s	8.36 s	26.7 s

Même constat que précédemment, le PSNR est infini ici. En revanche on s'aperçoit tout de même que l'entropie est légèrement meilleure pour un critère de SAV tout comme son temps d'exécution.



Carte d'erreur et flux 8/30
Erreur SVA



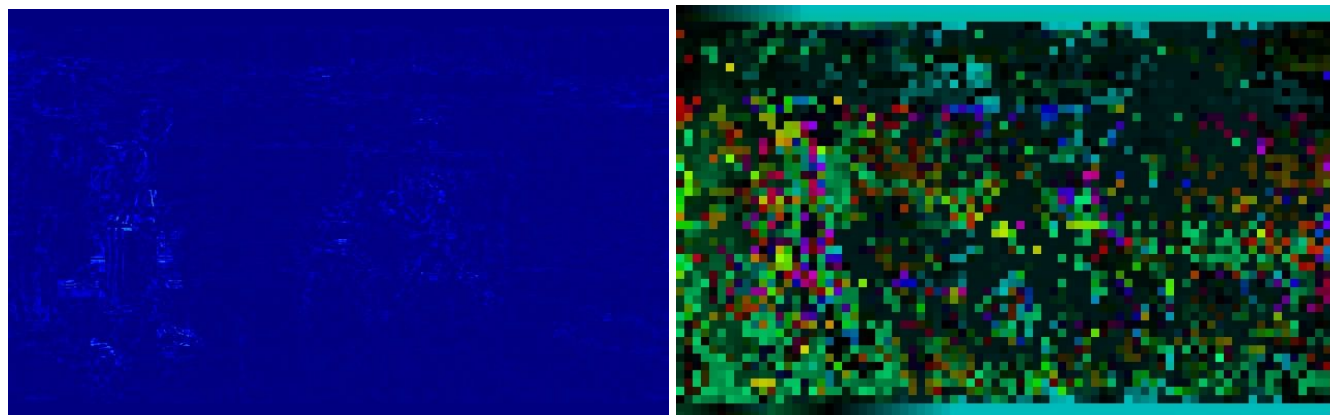
Carte d'erreur et flux 16/20
Erreur SVA

Nous apercevons quelques différences sur les cartes mais cela reste relativement minime. Les résultats présentés dans le tableau sont plus représentatifs des changements.

Tableau récapitulatif des résultats entre l'image 7 et 9 pour une erreur EM

Taille B / Taille L	8 / 5 40 *40 pixels	8 / 30 240*240 pixels	16 / 10 160*160 pixels	16 / 20 320*320 pixels
PSNR	infini	infini	infini	infini
Entropie	2.87	2.60	3.2	3.17
Norme du plus grand vecteur	28.28	164.1	112.43	219.276
Temps d'exécution calcul du mouvement.	1.09	27 s	11.24	36.6

Même constat que les 2 cas précédents, le PSNR est infini ici. Alors ici le critère n'est absolument pas adapté contrairement au reste des résultats. Certes les calculs sont plus rapides que l'erreur quadratique mais il n'empêche que l'on dispose d'une trop grosse différence entre les entropies.

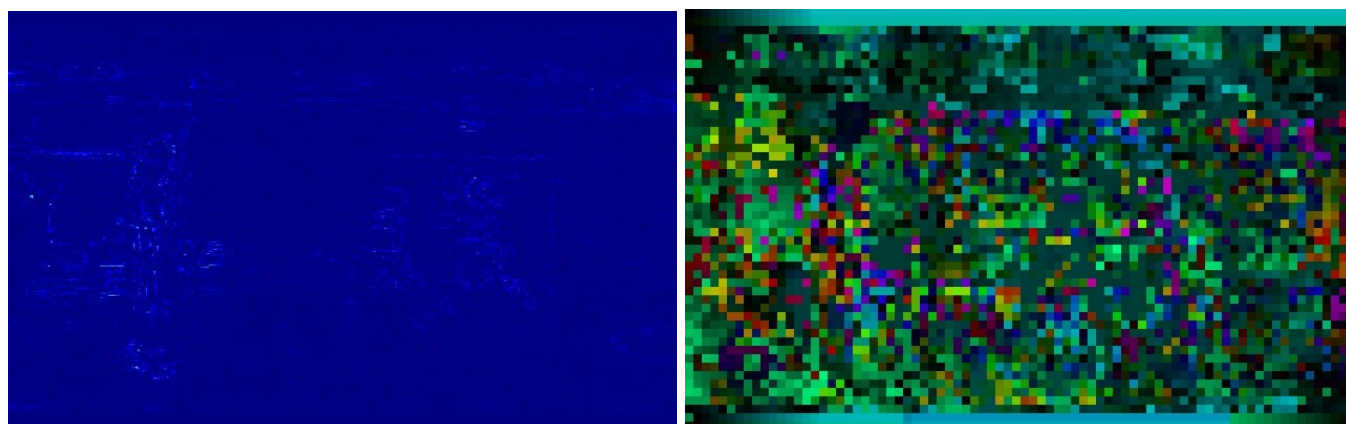


Carte d'erreur et flux 8/30
Erreur EM

Tableau récapitulatif des résultats entre l'image 9 et 13 pour une erreur SVA

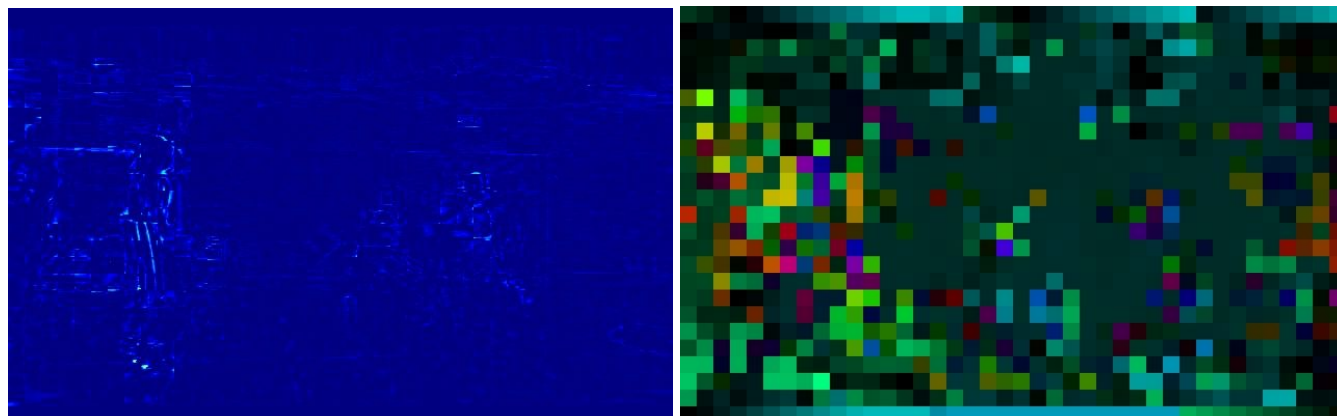
Taille B / Taille L	8 / 30 120*120 pixels	16 / 20 240 * 240 pixels
PSNR	infini	infini
Entropie	2.34	2.9
Norme du plus grand vecteur	166.88	220.658
Temps d'exécution calcul du mouvement.	19.17	27.81

Les résultats et les temps d'exécutions sont relativement similaires à ceux obtenus entre l'image 7 et 9. Cependant nous pouvons tout de même remarquer une légère augmentation dans le PSNR mais aussi dans la valeur du plus grand vecteur. Cela se justifie certainement par le fait que cette fois-ci nous travaillons avec des images plus espacées que l'exemple précédent. Le mouvement est donc plus fort, il est donc normal que le matching de bloc s'effectue à une distance plus élevée.



Carte d'erreur et flux 8/30
Erreur SVA

Au vue des résultats, le résultat semble en effet légèrement plus mouvementé si l'on regarde la carte du flux. Les zones colorées sur la carte d'erreurs ont par conséquent changées de place mais reste tout de même concentré sur les contours des joueurs.



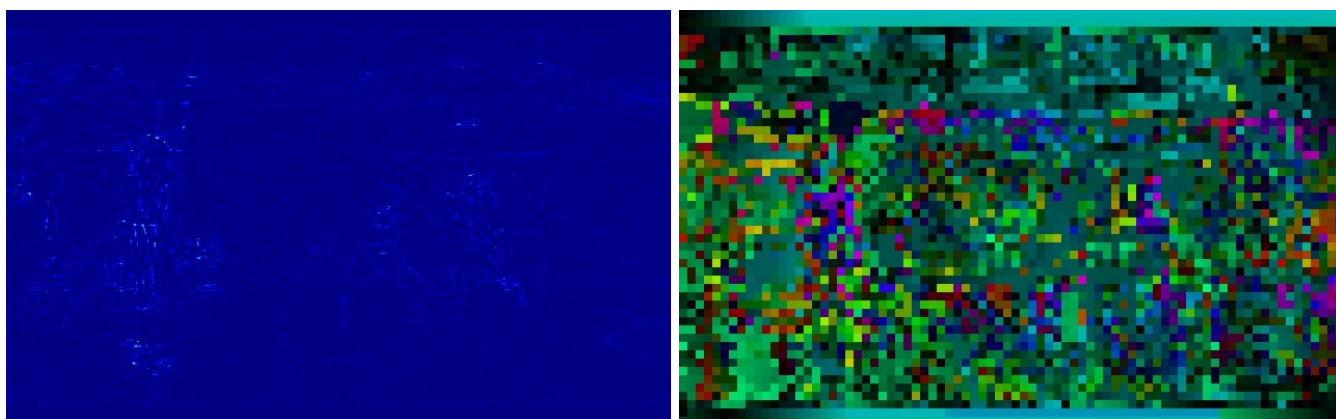
Carte d'erreur et flux 16/20
Erreur SVA

Même remarque que précédemment. Les changements sont plus perceptibles sur cette carte de flux pour un bloc 16. Les mouvements se sont accentués sur le côté gauche, prenant petit à petit de la place sur la zone uniforme du centre très différente de la première que nous avons aperçu.

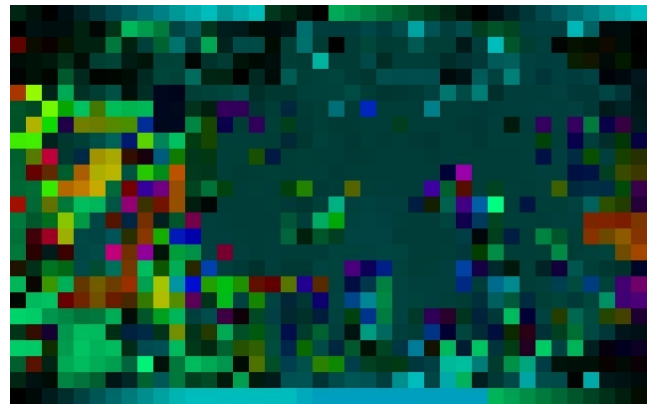
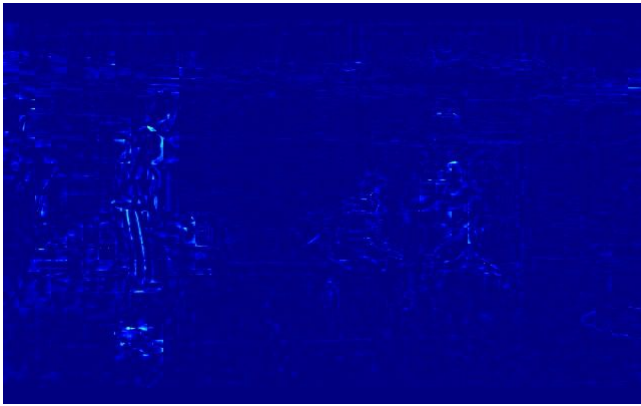
Tableau récapitulatif des résultats entre l'image 7 et 13 pour une erreur SVA

Taille B / Taille L	8 / 30 120*120 pixels	16 / 20 240 * 240 pixels
PSNR	infini	infini
Entropie	2.39	2.958
Norme du plus grand vecteur	168.291	215.258
Temps d'exécution calcul du mouvement.	19.97	28.35

Ceci est l'exemple permettant de justifier les propos précédent, excepté pour la norme qui dispose d'une valeur légèrement inférieure.



Carte d'erreur et flux 8/30
Erreur SVA



Carte d'erreur et flux 16/20
Erreur SVA

Question 4.3

Étant donné que le PSNR est infini partout, nous allons effectuer une analyse entropie/temps. Les meilleurs résultats sont obtenus avec un bloc de taille 8. Qui plus est, dû à sa petite taille, les temps de calcul n'en sont que meilleurs. En revanche un bloc de taille 16 nous permet d'avoir une meilleure visualisation de notre flux. À noter que le critère d'évaluation a lui aussi son importance comme nous l'avons vu. C'est la raison pour laquelle nous avons fait une majorité des tests avec le critère SVA.

Question 4.4

L'inconvénient est aisé à définir. L'algorithme mis en place représente juste une recherche exhaustive parmi tous les blocs possibles par conséquent sa complexité temporelle est un désavantage pour les grands espaces de recherche. Un léger inconvénient s'observe aussi sur la représentation du flux. En effet le mouvement n'est pas représentatif sur les bords du bloc. Cela est d'autant plus important si la taille du bloc est grande (Le mouvement n'est pas le même au sein du même bloc). Cependant l'algorithme n'en reste pas moins fiable, simple et minimise l'erreur de reconstruction dû à une potentielle quantification.

Question 5.5

Dans l'optique où l'on souhaite encoder la vidéo, l'étape de régularisation consiste à réduire l'entropie du champ de vecteur afin de diminuer la quantité d'information à transmettre lors de l'encodage et le décodage afin d'obtenir un compromis entre la qualité et quantité de bits utilisés pour l'encodage.

Dans d'autres cas la régularisation prend le sens de critère afin de sélectionner le meilleur bloc possible. Ainsi le meilleur bloc devra en plus de satisfaire l'EQM ou SVA..., satisfaire la contrainte imposée. Cela se traduit généralement par l'ajout d'une valeur dans le calcul. Dans notre cas, une régularisation qui me semble la plus adaptée concerne un critère venant réguler la distance d'un bloc. L'objectif est donc de minimiser la distance entre le bloc de l'image I et l'image I1. En effet pour 2 images très proches dans le temps il est tout à fait normal de trouver le bloc à une distance proche de sa position d'origine. Cela permet d'éviter des erreurs dû à la différence de l'intensité des pixels.

Conclusion Il existe plusieurs méthodes afin d'estimer le mouvement entre une série d'images. La solution adoptée dans ce TP est la plus facile à implémenter à condition de se mettre dans le cas d'hypothèses simples. Nous avons donc vu au cours de ce TP l'impact de certains paramètres sur le Block Matching et ainsi pu constater l'impact sur nos métriques de qualité de reconstruction.