

19-Enregistrer un film

Ce didacticiel explique comment enregistrer une matrice unique ou une séquence de matrices sur disque sous forme de film. Nous montrerons l'utilisation de l'objet *jit.record* pour l'enregistrement en temps réel et en temps différé. En cours de route, nous vous montrerons également comment régler certains paramètres de la vidéo de sortie.

Comme la plupart des objets Jitter, l'objet *jit.record* fonctionne selon un modèle événementiel plutôt temporel : les images envoyées à un objet *jit.record* activé pour l'enregistrement sont ajoutées au film en cours en tant que "prochaine image" en fonction des caractéristiques temporelles du film de sortie, sans tenir compte du temps relatif entre les images qui arrivent. Cette méthode permet d'obtenir des films très fluides avec une synchronisation cohérente entre les images, mais elle nécessite une certaine préparation avant l'enregistrement.

L'objet *jit.record* offre également une méthode d'enregistrement basée sur le temps (temps réel) en plus du mode par défaut décrit dans ce tutoriel. Bien que le mode temps réel puisse sembler plus simple à utiliser, la sortie enregistrée ne sera pas aussi fluide, car les images de la vidéo entrante seront généralement supprimées assez régulièrement pendant l'enregistrement. Vous pouvez définir le mode temporel pour l'objet *jit.record* en utilisant le message **realtime**, mais il est très différent du fonctionnement en temps réel décrit dans ce didacticiel.

Votre kilométrage peut varier

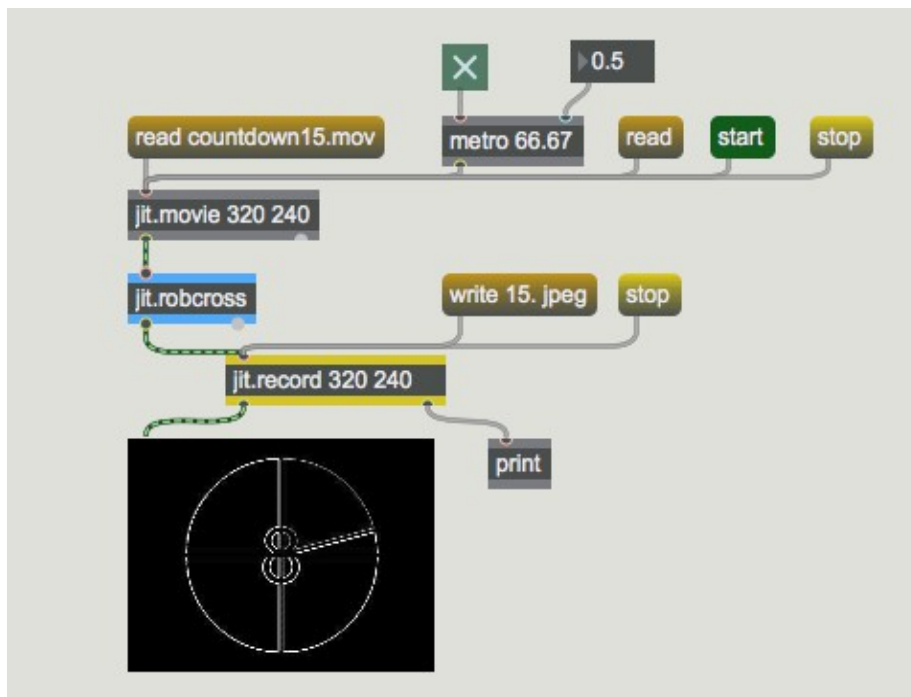
Chaque matrice qui arrive à un objet *jit.record* doit être compressée avant de faire partie de votre film de sortie. La vitesse de cette compression dépend de plusieurs facteurs: la vitesse de votre processeur et de votre disque dur, et, le plus important, le type de compression utilisé (appelé **codec**, abréviation de compression/décompression). Les exemples de ce tutoriel utiliseront le codec Photo-JPEG. Il devrait donner des résultats assez cohérents d'une machine à l'autre, sans nécessiter une forte utilisation du processeur et des accès au disque. Sur des systèmes plus anciens ou plus lents, il est possible que vous remarquiez une certaine imprécision dans la synchronisation avec la première série d'exemples de ce didacticiel. Si c'est le cas, ne vous inquiétez pas; la deuxième partie de ce didacticiel vous montrera comment obtenir d'excellents résultats, même sur les systèmes le plus modestes, en travaillant en mode non temps réel.

Remarque: les recommandations relatives au codec et à la dimension dans ce didacticiel sont obsolètes. Le codec **h264** est un bon codec pour équilibrer la taille de fichier, la qualité d'image et les besoins en processeur. Pour un enregistrement sans perte, ou pour enregistrer un canal alpha, utilisez **prores4444** sur Mac et **animation** sur Windows.

Sur l'horloge

- Ouvrez le sub-patch **Basic_recording**.

Le premier exemple montre une configuration d'enregistrement très basique. Notez que l'objet *jit.record* prend deux arguments (**320** et **240**) qui spécifient la largeur et la hauteur de tous les films que nous enregistrons avec cet objet.



Un simple patch d'enregistrement

- Cliquez sur la boîte de *message* intitulée **read countdown15.mov** pour charger un film dans l'objet *jit.movie*, puis démarrez l'objet *metro* en cliquant sur le *toggle* relié à son entrée. Si l'attribut *preview* est réglé sur 1, vous verrez l'image du film apparaître dans l'objet *jit.pwindow*, puisque l'objet *jit.record* passera ensuite toutes les matrices qu'il reçoit par la sortie gauche. (L'objet *jit.robcross* de ce patch est un simple détecteur de bords que nous utilisons pour traiter les matrices qui le traversent).
- Cliquez sur la boîte de *message* qui dit **write 15. jpeg** pour commencer à écrire le film (nous examinerons le contenu de la boîte de *message* plus en détail dans un instant). Une boîte de dialogue de fichier apparaîtra, vous demandant un nom de fichier. Lorsque vous entrez un nom valide et cliquez sur le bouton *Save*, l'enregistrement commence avec le nom de fichier que vous avez choisi (à titre d'exemple, supposons que vous ayez nommé votre film *myfilename.mov*). Cliquez sur la boîte de *message* **stop** pour arrêter l'enregistrement.

L'objet *jit.record* envoie un message par sa sortie droite après une opération d'écriture pour confirmer que le film a été écrit avec succès. Puisque nous avons connecté cette sortie à un objet *print* dans notre patch, nous pouvons voir les résultats en consultant la console Max. Si tout se passe comme prévu, vous devriez voir **print: write myfilename.mov 1** dans la console Max. En cas de problème, le nombre après le nom du fichier ne sera pas un **1**.

- Cliquez sur la boîte de *message* **read** dans le patch principal et lisez le film que vous venez d'enregistrer. Comme cela se présente-t-il ?

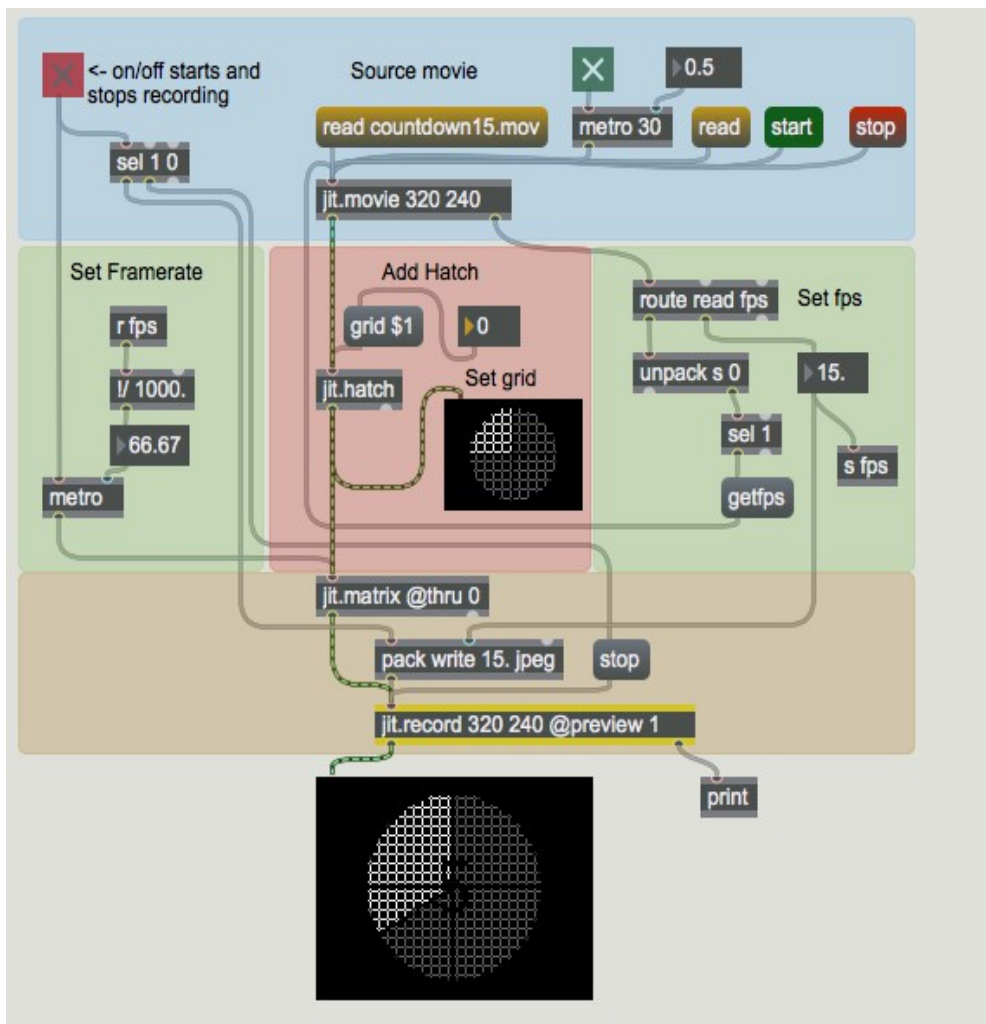
Examinons un instant le message **write** adressé à l'objet *jit.record*. Le message **write** place *jit.record* dans un mode d'activation d'enregistrement. L'enregistrement réel ne commence pas avant que l'objet ne reçoive une matrice dans son entrée. Le message **write** peut prendre plusieurs arguments optionnels pour spécifier un nom de fichier, la fréquence d'images, le type de codec, la qualité du codec et l'échelle de temps pour lire le film. Dans cet exemple, nous n'utilisons seulement deux arguments (**15.** et **jpeg**) pour spécifier la fréquence d'images (**15.** indique 15 images par seconde) et le type de codec (**jpeg** indique Photo-JPEG) du film de sortie. Pour une description complète de tous les arguments du message **write**, voir l'entrée de la référence d'objet pour l'objet *jit.record*.

Notez que l'objet *metro* est configuré pour envoyer un message **bang** toutes les 66,67 millisecondes. Ceci est fait pour correspondre à la fréquence d'images du film d'entrée. Si vous modifiez la fréquence de l'objet *metro* et enregistrez d'autres vidéos, vous constaterez qu'une fréquence de *metro* plus rapide rend la vidéo plus longue et plus lente, tandis qu'une fréquence de *metro* plus lente donne une vidéo plus courte et plus rapide. Cela se produit en raison du modèle événementiel utilisé par *jit.record*.

Plus musclé

- Ouvrez le sub-patch **Advanced_recording**.

Le deuxième exemple de patch est un peu plus compliqué, mais il comporte deux ajouts importants (et plusieurs ajouts mineurs).



Même idée, patch différent

Premièrement, nous détectons automatiquement la fréquence d'images du film d'origine (à l'aide du message **getfps**) et nous l'utilisons pour définir à la fois la fréquence de l'objet *metro* qui pilote *jit.record* et la fréquence d'images du film de sortie. Examinons cette partie du patch en détail:

- . Lorsqu'un film est chargé avec succès dans *jit.movie*, l'objet envoie le message **read filename.mov 1** par sa sortie droite. (Si l'opération **read** échoue, le nombre ne sera pas **1**.) Nous utilisons l'objet *route* pour récupérer ce message.

. L'objet *unpack s 0* décompose les autres éléments du message. Nous ne sommes pas intéressés par le nom du fichier, seulement par le succès ou l'échec de l'opération de lecture ; donc nous n'avons rien attaché à la sortie gauche de *unpack*, et attaché un objet *select*, qui teste le nombre entier **1**, à la sortie de droite. Si notre opération **read** est réussie, l'objet *select* produira un **bang**.

. Le **bang** est utilisé pour envoyer le message **getfps** à *jit.movie*, qui répondra en envoyant le message **fps [fps]** par sa sortie droite. L'élément de message **[fps]** est un nombre à virgule flottante représentant la fréquence d'images du film. Cette valeur est envoyée par la sortie centrale de l'objet *route* et est utilisée pour définir la vitesse de l'objet *metro* et la fréquence d'images de la vidéo de sortie.

Ce patch comprend une deuxième amélioration. Plutôt que de verrouiller l'intégralité du patch sur la fréquence d'images du film d'origine, nous avons ajouté un objet *jit.matrix* avec **thru** désactivé pour collecter la sortie du patch, et nous utilisons un second *metro* pour lui envoyer des messages de type **bang** à la fréquence d'images correcte pour *jit.record*. Cela permet au reste de notre patch de fonctionner de manière indépendante, à une fréquence aussi rapide ou lente que nous le souhaitons, avec seulement la toute dernière partie du patch spécifiquement synchronisée pour le processus d'enregistrement.

Enfin, nous avons ajouté un filtre de hachures (l'objet **jit.hatch**) avec son propre contrôle d'édition (la boîte de *nombres* et la boîte de *messages grid \$ 1*). Vous pouvez l'utiliser pour modifier votre film pendant que vous l'enregistrez et observer les effets sur la lecture. Vous pouvez substituer n'importe quel type d'objet Jitter ou de patch Jitter que vous voulez à ce stade du patch d'enregistrement et modifier ce que vous enregistrez en temps réel.

Hors-ligne

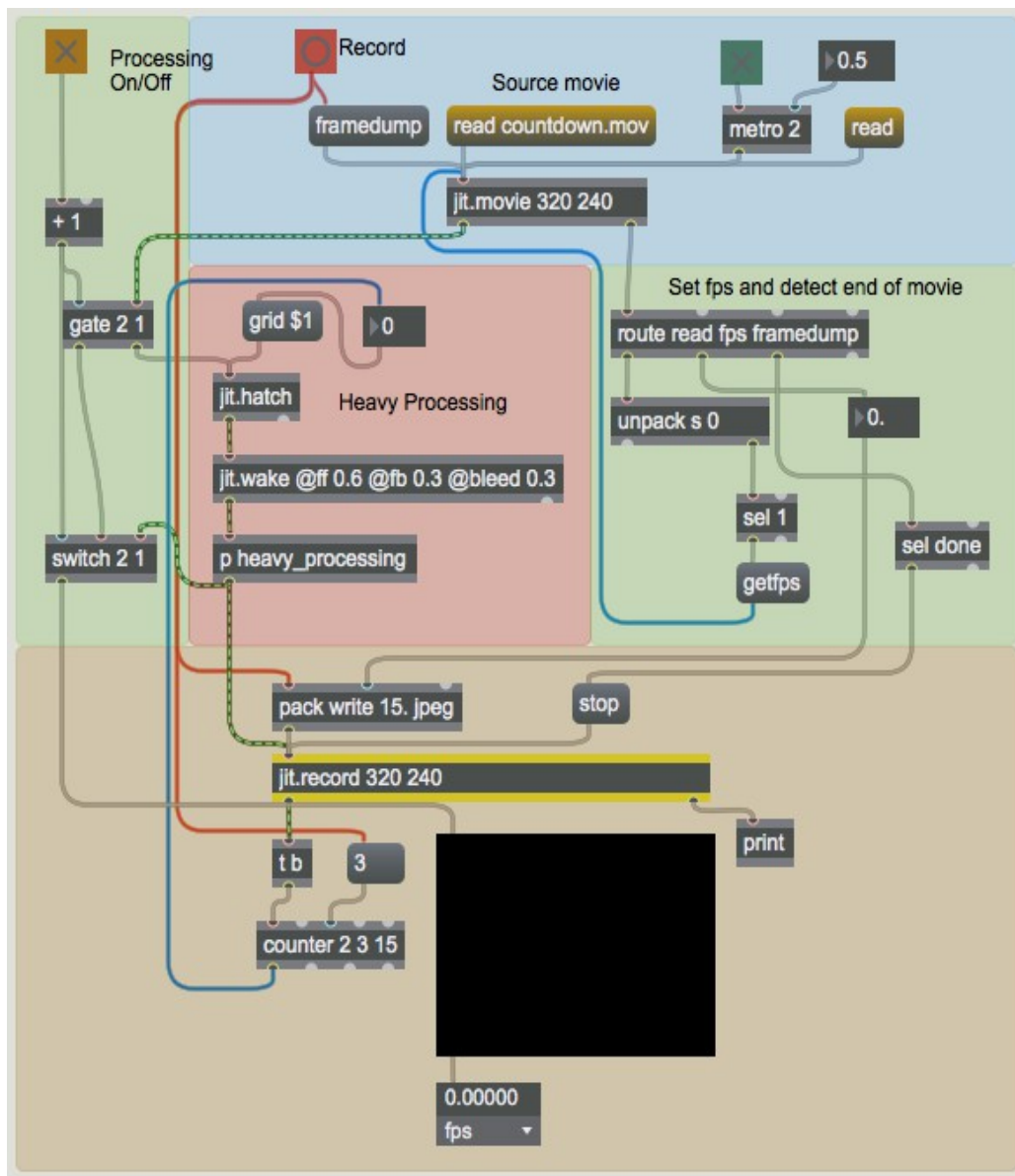
- Ouvrez le sub-patch **Off-line_recording**.

Les deux patches précédents ont un problème: sous des charges de processeur élevées, ces patches perdront des images. Dans de nombreux cas, ce n'est pas un problème, et les patches que nous avons examinés sont bien adaptés à l'enregistrement dans n'importe quel contexte, y compris les performances en direct.

Mais comment pouvons-nous enregistrer une séquence dans laquelle chaque image d'un film original a une image correspondante dans le film de sortie? Comment rendre des films dans Jitter de sorte que, même en cas d'une forte charge du processeur ou du disque, chaque image traitée soit capturée?

Il existe une solution. Le message **framedump** de l'objet *jit.movie* offre un mode de lecture non temps réel qui fonctionne parfaitement avec *jit.record*.

- Faites défiler la fenêtre du Patcher jusqu'en bas pour voir le troisième exemple.



Enregistrement en temps non réel avec **framedump**

A première vue, le troisième exemple de patch ne semble pas beaucoup plus compliqué que le précédent. Mais ne vous y trompez pas: ce patch sans prétention mettra même l'ordinateur le plus rapide à genoux.

- Cliquez sur la boîte de *message* **countdown.mov** pour charger un film dans l'objet *jit.movie*, démarrez l'objet *metro* en cliquant sur le *toggle* relié à son entrée et observez l'affichage de la sortie du taux d'images par seconde de l'objet *jit.fpsgui* situé au bas du patch. Si vous obtenez plus de 10 images par seconde, vous avez un ordinateur bien meilleur que le nôtre.

Bien que ce patch fonctionne lentement en temps réel, il permet de créer un film en temps non réel. En utilisant le message **framedump** de l'objet *jit.movie* et l'objet *jit.record*, le patch va capturer chaque image du film d'origine, la traiter et produire un film de sortie de longueur identique.

Voici comment cela fonctionne: le message **framedump** indique à l'objet *jit.movie* d'arrêter la lecture de son film, puis envoie chaque image, une par une. (N.B.: il n'est pas nécessaire d'envoyer un message **bang** ou **outputmatrix** à l'objet *jit.movie* pour que les images soient envoyées.) En raison de la façon dont le programmeur Max fonctionne, une nouvelle image ne sera pas envoyée avant que l'image précédente ait été entièrement traitée et enregistrée. Essayons-le.

- Cliquez sur la boîte *toggle* connectée à l'entrée de l'objet *metro* pour le désactiver. (L'envoi d'un **bang** à *jit.movie* pendant l'opération *framedump* provoquera l'envoi d'une image supplémentaire, ce qui n'est pas ce que nous voulons.) Nous avons connecté le *button* dans le coin supérieur gauche du patch à la fois à l'objet *pack* qui envoie le message **write**, et au message **framedump**. Appuyez sur le *button*, entrez un nom de fichier lorsque la boîte de dialogue de fichier apparaît et attendez que le film soit rendu. Une fois l'opération *framedump* terminée, l'objet *jit.movie* envoie le message effectué **framedump done** depuis sa sortie droite. Cet exemple de patch utilise ce message pour déclencher un message **stop** sur *jit.record*.

Le patch d'exemple tire également parti du fait que l'objet *jit.record* envoie une matrice par sa sortie gauche après avoir enregistré une image. Nous utilisons cette trame pour déclencher un message **bang** vers un objet *counter* qui pilote l'attribut **grid** du filtre *jit.hatch*. Cela crée une séquence répétable de changements de traitement qui se produisent au fur et à mesure que le film est capturé.

- Cliquez sur la boîte de message "**read**" pour ouvrir le film que vous venez de créer. Allumez le *metro* pour afficher le film. Cliquez sur le *toggle* connecté à l'objet *gate* pour voir votre nouveau film. Cela fait passer la sortie de l'objet *jit.movie* directement à la *jit.pwindow*, plutôt que via les objets de traitement.

Détails techniques: Ce tutoriel a été écrit dans les jours de l'antiquité, lorsque les ordinateurs (et Max) étaient moins efficaces. Il y a de fortes chances pour que votre ordinateur ne se contente plus de répondre aux exigences du sub-patch de traitement intensif sans se fatiguer. Le fait est que, lorsque vous atteignez ce niveau de traitement (et nous le faisons tous par la suite), le message *framedump* fera l'affaire.

Sommaire

L'objet *jit.record* fournit des moyens d'événements pilotés pour enregistrer des matrices Jitter sous forme de films. Vous pouvez enregistrer la sortie d'un patch Jitter en temps réel lorsque vous manipulez votre patch, ou hors temps réel, dans une opération de type rendu. Vous pouvez utiliser le message **framedump** de l'objet *jit.movie* pour traiter un fichier vidéo entier sans perdre d'images. Cette méthode vous permet d'enregistrer des vidéos de haute qualité, quelle que soit la charge imposée au processeur de votre ordinateur.