39-Cartographie spatiale

Dans ce didacticiel, nous examinerons les moyens de remapper la disposition spatiale des cellules dans une matrice Jitter en utilisant une deuxième matrice comme carte spatiale. En cours de route, nous étudierons les moyens de générer des cartes complètes par l'interpolation de petits ensembles de données, ainsi que les moyens d'utiliser des expressions mathématiques pour remplir une matrice Jitter avec des valeurs.

Les matrices Jitter peuvent être remappées spatialement à l'aide de tables de référence d'une complexité arbitraire. Dans le *didacticiel 12*, nous avons appris que l'objet *jit.charmap* mappe les valeurs des cellules (ou les informations de couleur) d'une matrice Jitter sur la base d'une table de référence fournie par une seconde matrice Jitter. De la même manière, l'objet *jit.repos* prend une matrice Jitter contenant des valeurs spatiales qu'il utilise ensuite comme table de référence pour réorganiser les positions spatiales des cellules dans une autre matrice.

• Ouvrez le patch du didacticiel.

Le patch du didacticiel montre la sortie de matrice d'un objet *jit.movie* traitée par un nouvel objet, appelé *jit.repos*. L'entrée droite de *jit.repos* reçoit les matrices de la sortie d'un objet *jit.xfade* alimenté par deux matrices nommées, chacune ayant reçu un nom (***stretch*** et ***cartopol***) afin qu'elles partagent des données avec des objets *jit.matrix* ailleurs dans le patch. À droite de la chaîne de traitement principale se trouvent deux zones distinctes où nous construisons ces deux matrices, qui serviront de cartes spatiales pour ce didacticiel.

• Cliquez sur la boîte de message qui lit ***read colorswatch.pict*** pour charger une image dans l'objet *jit.movie*. Cliquez sur le *toggle* intitulé Display pour démarrer l'objet *metro*.

Remarquez qu'à ce stade, nous ne voyons qu'un cadre blanc uni apparaître dans l'objet *jit.pwindow* en bas à gauche du patch. Cela est dû au fait que nous n'avons pas chargé de carte spatiale dans notre objet *jit.repos*.

• Ouvrez le sub-patch ***generate_stretch***.

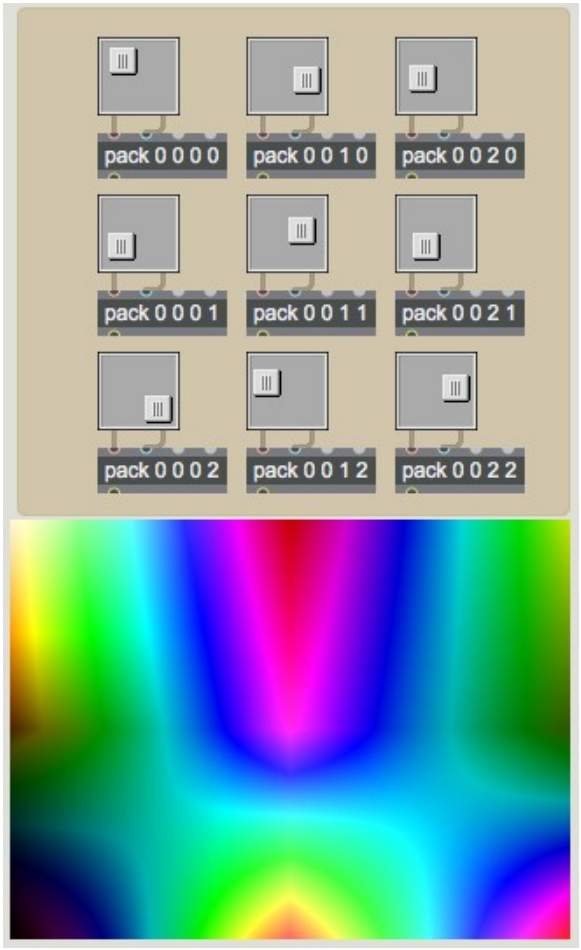
• Cliquez sur la première boîte de l'objet *preset* dans la zone du patch intitulée ***Generate Stretch Matrix***. Vous verrez les objets *pictslider* s'enclencher à différentes positions et un dégradé progressant du magenta au vert apparaît dans la fenêtre *jit.pwindow* au bas de cette section.

Nous voyons maintenant une image normale apparaître dans la fenêtre *jit.p* au bas de notre patch.

Le vaste monde de jit.repos

• Ouvrez le sub-patch ***generate_stretch***.

• Essayez de manipuler les objets *pictslider* qui définissent des valeurs dans la ***Stretch Matrix***. Remarquez que l'effet est d'étirer et de tordre notre image. En manipulant les objets *pictslider* un par un, nous pouvons voir qu'ils déforment différentes zones de l'écran dans une grille 3x3. En cliquant sur la première boîte de l'objet *preset,* l'image revient à la normale.

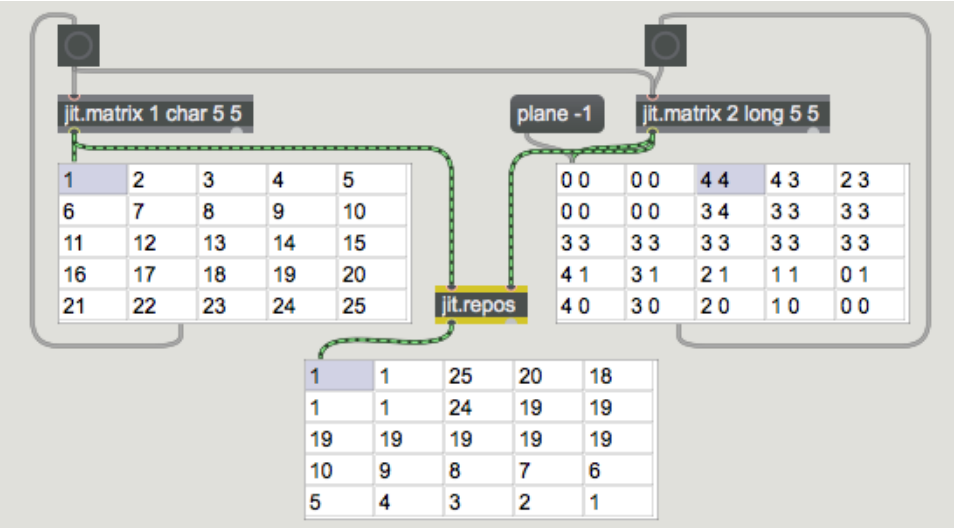


Salle des miroirs.

Les objets *pictslider* de notre patch définissent des cellules individuelles dans une matrice de Jitter de type ***long***, 3x3 à 2 plans,. Cette matrice est ensuite sur-échantillonnée avec interpolation vers une matrice 320x240 nommée ***stretch***. Cette matrice contient les instructions par lesquelles *jit.repos* mappe les cellules de notre matrice entrante. L'interpolation ici est nécessaire pour que nous puissions déduire une carte entière de 320 x 240 à partir de seulement neuf points de coordonnées spécifiés.

La carte spatiale fournie à l'objet *jit.repos* dans son entrée droite contient une description cellule par cellule de quelles cellules d'entrée sont mappées à quelles cellules de sortie. Le premier plan (plan 0) de cette matrice définit les coordonnées x pour le mappage; le deuxième plan (plan 1) représente les coordonnées y. Par exemple, si la cellule du coin supérieur gauche contient les valeurs **50 75**, alors la cellule aux coordonnées **50 75** de la matrice entrant dans l'entrée gauche de l'objet *jit.repos* sera placée dans le coin supérieur gauche de la matrice de sortie.

L'exemple suivant nous donnera une meilleure idée de ce que *jit.repos* fait avec ses instructions:



Une carte spatiale simple.

Dans cet exemple, nous avons une simple matrice 5x5 à plan unique remplie de nombres croissants qui est remappée par une carte spatiale. Remarquez la corrélation entre les valeurs de la carte spatiale et l'endroit où les cellules correspondantes de la matrice de sortie tirent leurs valeurs originales dans la matrice d'entrée. Par exemple, la cellule en bas à gauche de la matrice de la carte spatiale indique **4 0**. La cellule à la coordonnée {4, 0} dans la matrice entrante est définie sur la valeur **5**. Comme vous pouvez le constater, la cellule en bas à gauche de la matrice de sortie contient également cette valeur.

De même, vous pouvez voir que les cellules de la rangée centrale de la matrice de sortie ont toutes la même valeur (**19**). Cela est dû au fait que les cellules correspondantes de la carte spatiale ont toutes la même valeur (3 3). La cellule {**3, 3**} dans la matrice d'entrée contient la valeur **19**; cette valeur est ensuite placée sur toute la ligne de la matrice de sortie.

Dans notre patch de tutoriel, notre carte spatiale doit pouvoir contenir des valeurs dans une plage qui représente la taille complète (ou ***dim***) de la matrice à traiter. À cette fin, les plages de *pictslider* sont définies sur 320 x 240. Pour gérer des nombres de cette taille, nous devons utiliser une matrice de type **long** pour notre carte spatiale. Si elle était de type **char**, nous serions limités à des valeurs comprises entre 0 et 255, ce qui limiterait les cellules que nous pouvons choisir dans notre carte spatiale. Notez qu'afin de vusaliser correctement ces matrices **long** dans les objets *jit.pwindow*, nous utilisons l'objet *jit.clip* pour contraindre les valeurs de la matrice avant de visualiser la même plage que les matrices **char**.

> **Note technique**: le comportement par défaut de *jit.repos* est d'interpréter la carte spatiale comme une matrice de coordonnées absolues, c'est-à-dire une spécification littérale de quelles cellules d'entrée correspondent à quelles cellules de sortie. L'attribut **mode** de *jit.repos*, lorsqu'il est défini à 1, place l'objet en mode relatif, dans lequel la carte spatiale contient des coordonnées relatives à leur position normale. Par exemple, si la cellule {15, 10} de la carte spatiale contient les valeurs **-3 7**, alors un objet *jit.repos* travaillant en mode relatif définira la valeur de la cellule {15, 10} dans la matrice de sortie au contenu de cellule {12, 17} dans la matrice d'entrée (c'est-à-dire {15 - 3, 10 + 7}).

• Jouez un peu plus avec les objets *pictslider* pour avoir une meilleur idée du fonctionnement de *jit.repos*. Remarquez que si vous placez une rangée (ou une colonne) entière d'objets *pictslider* approximativement à la même position, vous pouvez «étirer» une zone de l'image pour couvrir une bande horizontale ou verticale entière dans l'image de sortie. Si vous réglez les objets *pictslider* des colonnes de gauche sur des valeurs normalement contenues dans les colonnes de droite (et vice

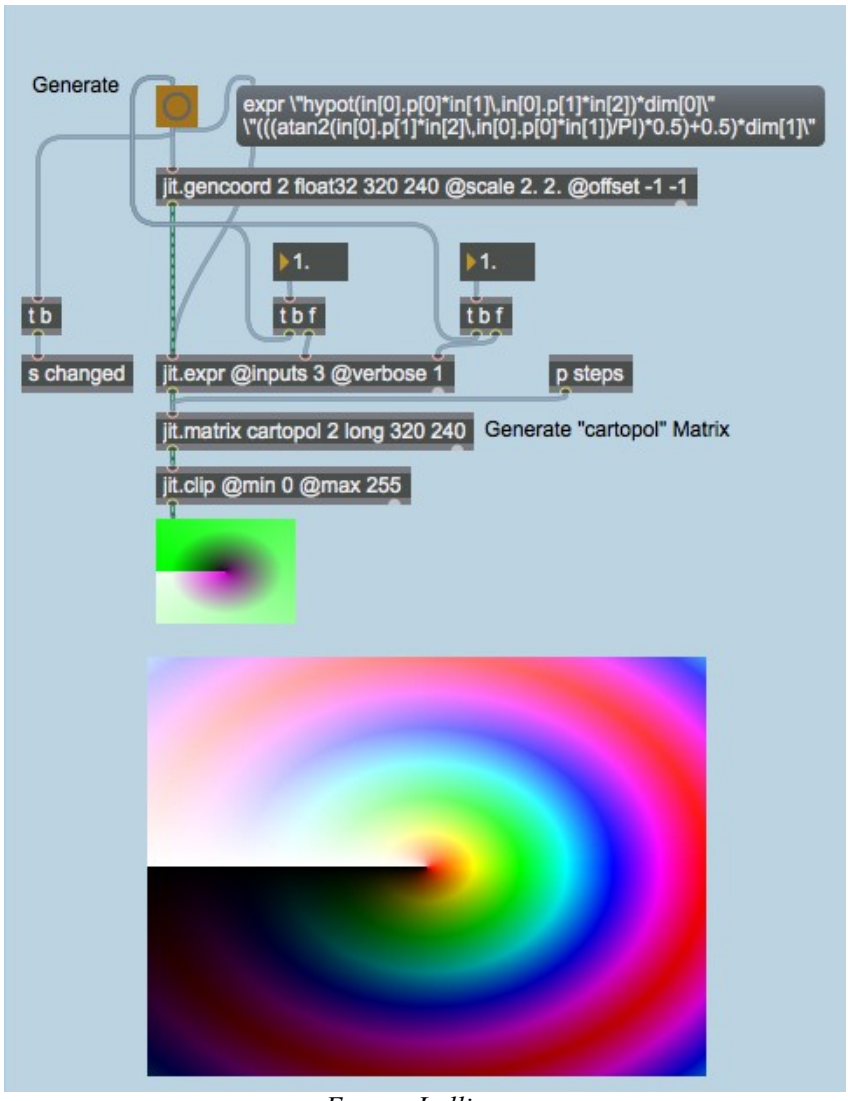
versa), vous pouvez retourner l'image sur son axe horizontal. En ne le faisant qu'à moitié, ou en positionnant les valeurs des curseurs quelque part entre leurs positions sur la carte «normale» (telle que définie par l'objet *preset*), vous pouvez obtenir le type de distorsion spatiale que l'on traouve couramment dans les miroirs d'attractions, où un pli dans le verre déforme la lumière réfléchie pour produire une exagération spatiale.

Expressions spatiales

Maintenant que nous avons expérimenté la conception de cartes spatiales à la main, pour ainsi dire, nous pouvons examiner certaines façons intéressantes de les générer de manière algorithmique.

• Ouvrez le sub-patch ***generate_cartopol***.

• Cliquez sur le *button* intitulé ***Generate***. Définissez la boîte de *nombre* intitulée Crossfade (attachée à l'objet *trigger* dans la partie principale du patch) sur **1.0**.

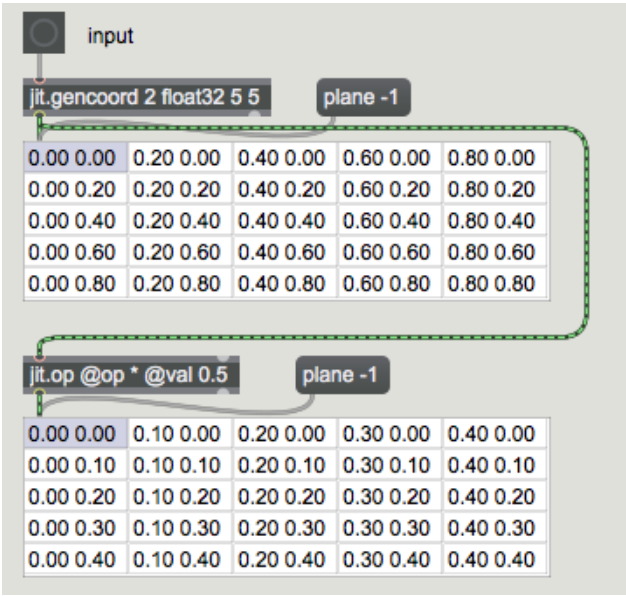


Espace Lollipop.

La carte spatiale **cartopol** a été définie par une expression mathématique opérée sur la sortie de l'objet *jit.gencoord* par l'objet *jit.expr*. Les résultats de cette expression, effectuée sur des matrices Jitter ***float32***, sont ensuite copiés dans une matrice ***long*** qui sert de carte spatiale. Cette carte spatiale amène *jit.repos* a remappé spatialement notre matrice entrante de l'espace cartésien à l'espace polaire, de sorte que le bord le plus à gauche de notre matrice entrante, s'enroulant en

spirale jusqu'à l'anneau final, qui représente la bord le plus à droite de la matrice d'entrée. L'attribut **boundmode** de *jit.repos* indique à l'objet d'envelopper les coordonnées au-delà de cette plage vers l'intérieur, de sorte que le bord final de notre image de sortie commence à se replier à nouveau vers le côté gauche de l'image d'entrée.

L'objet *jit.gencoord* génère une simple carte cartésienne de valeurs à virgule flottante; lorsque vous multipliez les valeurs de la matrice par le **dim** de la matrice, vous pouvez facilement obtenir des coordonnées qui correspondent littéralement aux emplacements des cellules, comme le montre cette cette illustration:



Carte cartésienne normale.

Les attributs **scale** et **offset** de *jit.gencoord* nous permettent de manipuler la plage de valeurs émises par l'objet, ce qui nous évite d'utiliser plusieurs objets *jit.op* pour obtenir notre carte cartésienne dans la plage souhaitée. Pour que notre équation fonctionne, nous voulons que notre carte de départ soit dans l'intervalle {-1, 1} dans les deux dimensions (par conséquent, nous multiplions l'espace cartésien normal par 2 et le décalons par -1).

L'objet *jit.expr* prend la matrice entrante fournie par *jit.gencoord* et exécute une expression mathématique sur celle-ci, tout comme l'objet Max *vexpr* travaille sur des listes de nombres. L'attribut **expr** de l'objet *jit.expr* définit l'expression à utiliser. La notation ***in [i] .p [j]*** indique à *jit.expr* que nous voulons opérer sur les valeurs contenues dans le plan **j** de la matrice arrivant à l'entrée **i**. Le mot clé ***dim [i]*** renvoie la taille de la dimension **i** dans la matrice. Dans notre cas, la notation ***dim [0]*** correspondrait à 320; ***dim [1]*** se résoudrait à 240.

Notez qu'il existe deux expressions mathématiques définies dans la boîte de *message* fixant l'attribut **expr**. La première définit l'expression à utiliser pour le plan 0 (les coordonnées x de notre carte spatiale), le second définit l'expression pour le plan 1 (les coordonnées y).

Placée dans une notation plus lisible, l'expression *jit.expr* pour le plan 0:

*hypot (in [0] .p [0] * in [1] \, in [0] .p [1] * in [2]) * dim [0]*

donne quelque chose comme ceci, si x est la valeur dans le plan 0 de l'entrée, y est la valeur dans le plan 1 de l'entrée, et a et b sont les valeurs dans les objets de la boîte de *nombre* attachés aux deuxième et troisième entrées de l'objet *jit.expr*:

sqrt (ax2 + by2) * 320

De même, l'expression pour le plan 1:

*(((atan2 (in [0] .p [1] + in [2] \, in [0] .p [0] + in [1]) / PI) * 0,5) 0,5) * dim [1]*

Cela donne quelque chose comme ceci:

(((atan (by / ax) /) / 2) +0,5) * 240

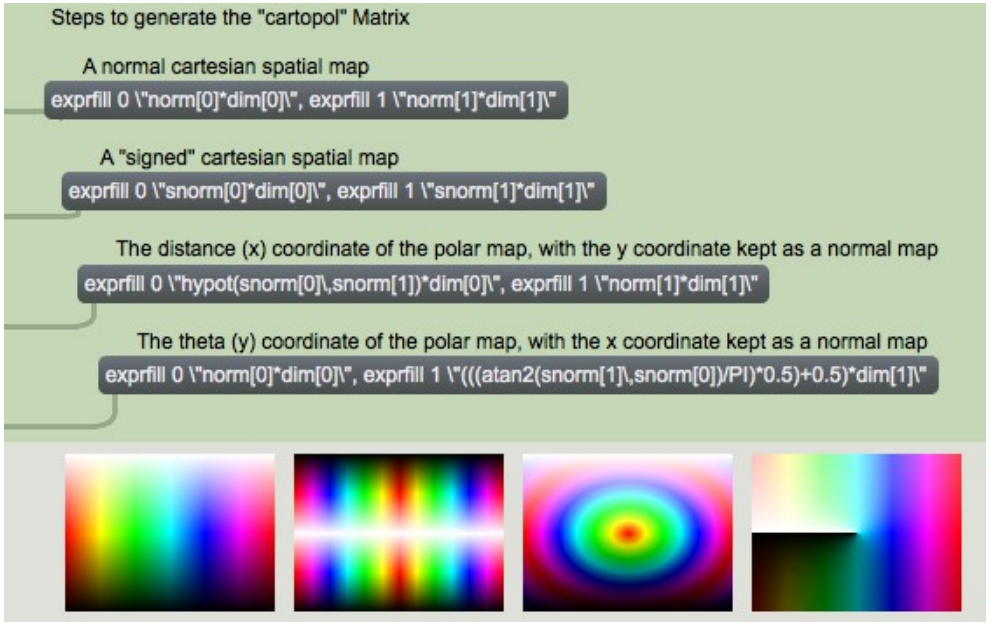
Par conséquent, nous pouvons voir que nous remplissons les valeurs x (plan 0) de notre carte spatiale avec des nombres correspondant à la longueur de l'hypoténuse de nos coordonnées (c'est-à-dire leur distance par rapport au centre de la carte), tandis que nous définissons les valeurs y (plan 1) valeurs pour représenter l'angle (Ø) de ces valeurs autour de l'origine.

• Dans la section inférieure droite du patch du didacticiel, modifiez les objets de boîte de *nombres* attachés à l'objet *jit.expr*. Notez qu'ils vous permettent d'étirer et de réduire la carte spatiale le long des axes x et y.

L'objet *jit.expr* peut prendre des valeurs numériques Max (**int** ou **float**) dans ses entrées au lieu de matrices. Ainsi, la notation ***in [1]*** et ***in [2]*** dans l'attribut **expr** fait référence aux nombres arrivant aux entrées du milieu et de droite de l'objet.

> **Note technique**: l'objet *jit.expr* analyse les expressions mathématiques contenues dans l'attribut **expr** comme des chaînes individuelles (une par plan). Par conséquent, chaque expression individuelle doit être contenue entre guillemets et les virgules doivent être espacées par une barre oblique inverse (\). Ceci afin d'empêcher Max d'atomiser l'expression en une liste séparée par des espaces ou une séquence de messages séparés par des virgules. Nous pouvons obtenir un aperçu de la façon dont *jit.expr* analyse ses expressions en activant l'attribut **verbose** de l'objet (comme nous le faisons dans ce didacticiel). Lorsqu'un attribut **expr** est évalué, l'objet imprime la hiérarchie d'évaluation de l'expression dans la console Max, afin que nous puissions voir exactement comment les différentes parties de l'expression sont segmentées et interprétées.

• Ouvrez l'objet patcher intitulé ***steps***. Cliquez sur les différentes expressions contenues dans les boîtes de *message*, chacune d'entre elles montrant une étape plus simple de la façon dont nous avons calculé la carte cartopol. La notation ***norm [i]*** et ***snorm [i]*** reproduit essentiellement ici la fonction d e *jit.gencoord*, générant une carte de valeurs cartésiennes (**0 à 1**) ou de valeurs cartésiennes signées (**-1 à 1**) sur la dimension spécifiée.

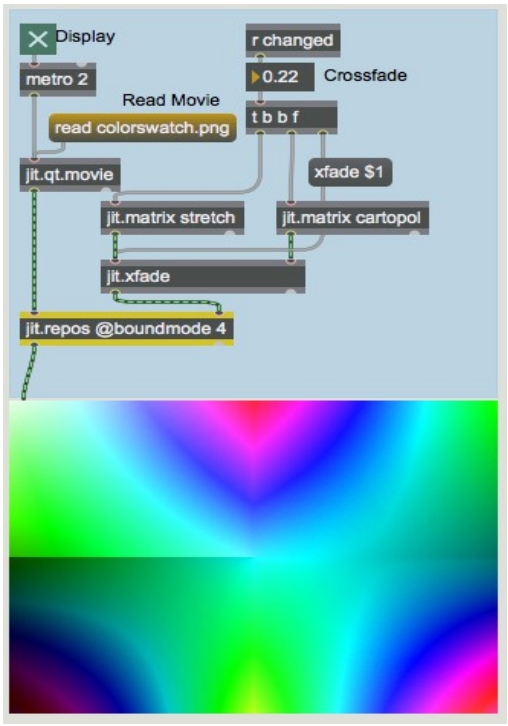


Différentes étapes de la cartographie cartésienne-polaire: (a) carte normale, (b) carte normale mise à l'échelle et décalée par rapport à une plage signée {-1, 1}, (c) hypoténuse uniquement, (d) thêta uniquement

Une chose importante à noter avec ce sub-patch est que les objets de boîte de *message* sont directement connectés à l'objet *jit.matrix* dans le patch principal. Le message ***exprfill*** à *jit.matrix* nous permet de remplir une matrice Jitter avec les valeurs d'une expression mathématique sans avoir à utiliser *jit.expr* comme un objet séparé. La syntaxe de la méthode ***exprfill*** est légèrement différente de celle de l'attribut **expr** de *jit.expr*: le premier argument de ***exprfill*** est le plan à remplir de valeurs, donc deux messages ***exprfill*** (séparés par des virgules) sont nécessaires pour générer la carte spatiale, en utilisant cette méthode.

Toujours des matrices

• Faites lentement passer la boîte de *nombres* attachée à l'objet *trigger* à gauche du patch de 1 à 0, et inversement.



Fondu enchaîné partiel des deux cartes spatiales utilisées dans ce didacticiel.

La boîte de *nombre* contrôle l'attribut xfade de l'objet jit.xfade directement attaché à l'objet jit.repos au cœur de notre chaîne de traitement. Ainsi, nous pouvons passer en douceur de notre carte spatiale définie manuellement (la matrice d'étirement) à celle définie algorithmiquement (la matrice cartopol). La possibilité d'utiliser des objets de traitement Jitter pour composer différentes cartes spatiales ajoute une autre couche de flexibilité potentielle. Comme les deux matrices contenant nos cartes ont la même typologie (type, dim et planecount), elles peuvent être facilement combinées pour créer une variété infinie de composés.

Résumé

L'objet *jit.repos* traite une matrice d'entrée basée sur une carte spatiale encodée comme une matrice envoyée à sa deuxième entrée. Le comportement par défaut est que *jit.repos* traite les matrices spatialement en utilisant des coordonnées absolues, de telle sorte que les valeurs des cellules dans la deuxième matrice correspondent aux coordonnées dans la matrice d'entrée dont les valeurs des cellules sont copiées dans la position correspondante dans la matrice de sortie.

Les cartes spatiales peuvent être créées en définissant explicitement les cellules dans une matrice et, si vous le souhaitez, en sur-échantillonnant avec interpolation à partir d'une matrice initiale plus petite. Les cartes peuvent également être créées en générant des matrices par des moyens algorithmiques, comme le traitement d'une carte cartésienne «normale» générée par un objet *jit.gencoord* par une expression mathématique évaluée par un objet *jit.expr*. L'objet *jit.expr* prend des chaînes d'expressions mathématiques, une par plan, et les analyse en utilisant des mots-clés pour représenter les valeurs des cellules entrantes ainsi que d'autres informations utiles, comme la taille de la matrice entrante. Si vous souhaitez remplir une matrice Jitter avec les résultats d'une expression mathématique directement, vous pouvez vous servir du message ***exprfill*** vers un objet *jit.matrix* pour obtenir les mêmes résultats sans avoir à utiliser un objet *jit.expr* distinct.

L'objet *jit.repos* peut être utilisé avec une grande variété de cartes spatiales pour différentes utilisations dans le traitement des images et des données. Le dossier «jitter-examples/video/spatial» du dossier exemples de Max contient un certain nombre de patches qui illustrent les possibilités de *jit.repos*.