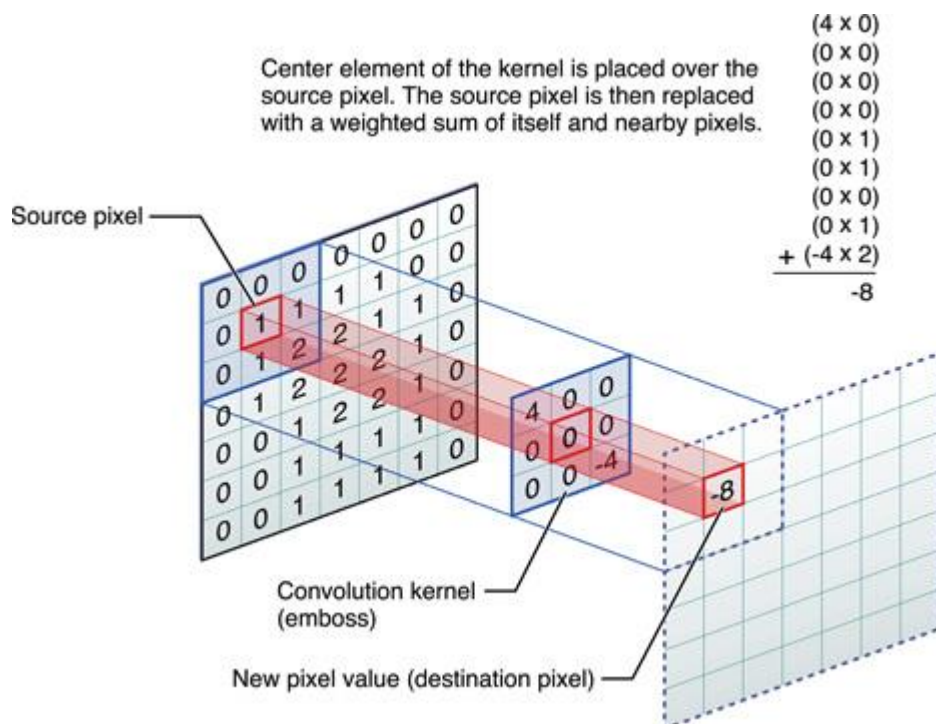


VIII. Filtres et convolution

1. Introduction

La convolution est courante en traitement d'images. Elle consiste en une opération de multiplication de deux matrices de tailles différentes (généralement une petite et une grande), mais de même dimensionnalité semblable (p.ex. 1D, 2D), produisant une nouvelle matrice (également de même dimensionnalité). La convolution est donc le traitement d'une matrice (p.ex. une image) par une autre petite matrice appelée matrice de convolution ou noyau (kernel).

Le filtre parcourt toute la matrice principale (p.ex. l'image) de manière incrémentale et génère une nouvelle matrice constituée des résultats de la multiplication. Notez qu'il y a une marge dans la matrice finale pour laquelle nous ne pouvons pas calculer de valeur. Dans le traitement d'image, ceci est utilisé par exemple pour effectuer un flou gaussien, ou détourner les éléments d'une photo (exemples sur <http://stephanieluu.com/image-convolution/>).



Exemple de convolution 2D (source [1])

2. Objectifs du TP

L'objectif du TP est de savoir appliquer des filtres de convolution sur des matrices, dans notre cas des modèles numériques de terrain (MNT). Les filtres sont appliqués dans différents domaines comme l'hydrologie, la météorologie, l'analyse d'images, le traitement du signal, etc.

Le filtrage d'une matrice permet d'en recalculer les valeurs à partir des valeurs des voisins. La matrice de convolution permet de définir les règles de voisinage (comment on combine les voisins pour recalculer la valeur du pixel central). On peut la désigner comme opération locale.

Exemple (filtrage par moyenne locale):

MNT	Nouveau MNT (après 1 ^{ère} itération)																														
<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>125</td><td>135</td><td>140</td><td>155</td><td>150</td></tr> <tr><td>130</td><td>127</td><td>124</td><td>126</td><td>135</td></tr> <tr><td>128</td><td>132</td><td>133</td><td>135</td><td>128</td></tr> </table>	125	135	140	155	150	130	127	124	126	135	128	132	133	135	128	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td>125</td><td>135</td><td>140</td><td>155</td><td>150</td></tr> <tr><td>130</td><td>130.4</td><td>124</td><td>126</td><td>135</td></tr> <tr><td>128</td><td>132</td><td>133</td><td>135</td><td>128</td></tr> </table>	125	135	140	155	150	130	130.4	124	126	135	128	132	133	135	128
125	135	140	155	150																											
130	127	124	126	135																											
128	132	133	135	128																											
125	135	140	155	150																											
130	130.4	124	126	135																											
128	132	133	135	128																											

La valeur du pixel rouge a été recalculée en faisant simplement la moyenne des voisins et lui-même. La matrice de convolution est dans ce cas de taille 3x3 (9 éléments) avec 1/9 dans chaque case. On peut changer la taille et les valeurs de cette matrice pour effectuer différents types d'opérations.

Le déplacement de cette matrice sur le MNT (la matrice qu'on va recalculer) s'appelle convolution.

3. Fonctions de convolution

`C = conv2(A,B)` calcule une convolution en 2D des matrices A et B, où A est la matrice principale et B est le noyau. La taille de la matrice résultante C est un peu plus grande que A [2].

`C = imfilter(A,B,options)` de la toolbox « Image Processing » fonctionne sur le même principe, mais par défaut la taille de C correspond à celle de A [2]. Il est également possible de donner des options supplémentaires pour définir la manière de traiter les bords : 'symmetric', 'replicate', 'circular' (voir l'aide et tester pour observer les effets).

4. Références

[1] iOS Developer Library (<https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html>)

[2] Aide Matlab

5. Auteur(s)

Loris Foresti (2009)

Pascal Horton (2015)