

Flou de profondeur de champs

Hadjer DJERROUMI & Jean-Manuel ERIALC

Rapport du projet de traitement d'image M1

UNIVERSITÉ GUSTAVE EIFFEL

25 mai 2020

Convention et notation

Dans ce qui suit, *depth* désignera la valeur de profondeur du pixel sélectionné par l'utilisateur. La convention est que plus cette valeur est grande plus le pixel est proche de la caméra.

Selection de l'objet *focus*

La reconnaissance d'un objet quelconque sur une image est un problème de *machine learning*. Nous nous sommes contenté des valeurs de profondeur associées à chaque pixel pour déterminer si le pixel fait partie de l'objet ou pas. Pour cela, nous avons fixé une valeur par défaut *DEFAULT_DEPTH_DIFFERENCE* égale à 10. Le principe est, étant donnée la profondeur *depth* du pixel sur lequel l'utilisateur a cliqué, le programme cherche dans le voisinage de ce pixel, les voisins de profondeur *d* dont la distance $|depth - d|$ est au plus égale à *DEFAULT_DEPTH_DIFFERENCE*.

Le résultat est une zone de l'image qui désigne l'objet *focus* et que l'utilisateur pourra voir en appuyant sur la touche *o*. Le point fort de cette méthode est que les autres objets qui ont la même profondeur que l'objet *focus* seront floutés.

L'idée est qu'en général, les pixels appartenant à un même objet, ont des profondeurs proches. Ceci ne donne pas de bon résultats si l'objet considéré possède des parties *proches* et d'autres *loins*.

Un autre point à préciser est qu'il faut *ajuster* la valeur *DEFAULT_DEPTH_DIFFERENCE* selon la taille des objets de l'image, en effet, si les objets sont petits (*resp. grands*), il faut le diminuer (*resp. l'augmenter*) en appuyant sur les touches *m* ou *-* (*resp. p* ou *+*).

Choix du flou : le flou gaussien

Le flou utilisé dans ce projet est le *flou gaussien*. Ce choix est justifié par la possibilité d'agir sur le paramètre réel σ pour nuancer le flou ce qui permet une forte précision. Dans le cas du filtre moyenner, la seule possibilité de nuancer le flou est d'agrandir ou de diminuer

la taille de la matrice, or vu que la taille est un entier impair, nous sommes obligés de passer de 3 à 5 à 7 à 11 et nous pourrions finir par de grandes matrices qui peuvent avoir une taille proche de celle de l'image et donc attribuer la moyenne de toute l'image à ce pixel, ce qui n'est pas un bon flou. Pour éviter cela, il faut travailler sur des plages de valeurs de profondeur et donc perdre en précision.

D'autre part, le flou gaussien est une moyenne pondérée du voisinage dont le poids le plus fort est attribué au pixel lui-même et plus le voisin est loin, plus son poids est faible, ceci résout le problème de matrice de grandes tailles et préserve la définition. En effet, avec le moyennage le flou est puissant même avec des noyaux de petite taille par contre le flou gaussien nous donne la possibilité de nuancer le flou avec une grande précision (la définition de l'image diminue sensiblement).

Flou gaussien sélectif

Un autre argument puissant qui justifie notre choix est la technique du *flou gaussien sélectif* dont l'une des applications consiste à flouter l'arrière-plan d'une image. Il s'agit d'ignorer les voisins de valeur v dont la distance avec la valeur du pixel à flouter est strictement supérieure à une valeur δ fixée. Il sert à conserver le contraste de l'image.

Nous l'avons adapté à notre projet en remplaçant les valeurs des pixels par les valeurs de profondeur correspondantes, et en remplaçant la distance par *profondeur du voisin - profondeur du pixel* car nous voulons ignorer les voisins qui sont plus proches du pixel.

Nous avons remarqué une dégradation dans le rendu en fixant le δ à 0 (les niveaux de flou sont très visibles à l'œil), et à partir de $\delta = 1$ le rendu est lisse et sans bruit. Nous l'avons donc fixé à 1 car nous cherchons la plus petite valeur pour ignorer un maximum de voisins plus proches.

Technique utilisée

Une fois l'objet *focus* sélectionné, on garde les valeurs initiales des pixels correspondant (aucun filtre n'est appliqué). Après, il s'agit de flouter à la fois l'avant-plan et l'arrière-plan en respectant la condition :

"le flou d'un objet proche se mélange avec les objets de l'arrière-plan mais la réciproque est fausse."

Nous avons défini l'avant-plan par l'ensemble des pixels qui sont plus proches de la caméra que l'objet *focus*. Plus précisément, l'ensemble des pixels de profondeur d pour lesquels $depth - d < 0$. De même, l'arrière-plan sera l'ensemble de pixels de profondeur d pour lesquels $depth - d \geq 0$.

Nous avons traduit la condition par :

"le flou d'un pixel de l'avant-plan prend en compte tous les voisins de son voisinage, et le flou d'un pixel de l'arrière-plan, ignore les voisins qui sont plus proches que lui."

Nous avons donc utilisé le *flou gaussien* pour les pixels de l'avant-plan, et le *flou gaussien sélectif* pour les pixels de l'arrière-plan. Le paramètre σ est égale à la distance entre *depth* et la profondeur du pixel à flouter + 1, car il doit être non nul (par définition de la gaussienne) et la distance peut être nulle. Et la taille des matrices gaussiennes est égale à $2 \times \left\lceil \frac{\sigma}{2.5} \right\rceil + 1$ (en divisant σ par 2, le flou était trop fort, et en le divisant par 3, le flou était trop faible, d'où le choix de 2.5).

Nous avons calculé une gaussienne pour chaque différence de profondeur en utilisant la *mémoïsation* pour éviter de recalculer une matrice déjà calculée.

Application

code

Le programme est composé de trois fichiers sources :

- `Commands.cpp` : contient le code gérant les callback lors d'un clic de souris ou l'appui d'une touche.
- `Context.cpp` : représente un context attaché aux callback contenant l'image d'entrée, l'image de profondeur des champs et l'image représentant l'objet calculé lors du dernier clic et tous les noyaux gaussiens déjà calculés.
- `focus.cpp` : programme principal.

Exécution

Nous avons travaillé uniquement sur des images *RGB-D*. Le programme se lance avec la ligne de commande :

`./bin/focus <my_image>`

où `<my_image>` est le chemin vers l'image *RGB* se trouvant dans le répertoire *input* dans lequel se trouve aussi l'image de profondeur (*depth map*) associée avec le nom `<my_image>_depth.png`.

Interaction utilisateur

Une fois le programme lancé, l'utilisateur verra son image affichée. En cliquant sur un objet de l'image, il verra l'image floutée affichée dans une nouvelle fenêtre. Les touches de clavier suivantes permettent de réaliser les commandes :

- `q` : quitte le programme.
- `s` : exporte l'image courante dans un fichier nommé "export_<clic_x> <clic_y>.jpg".
- `o` : affiche une fenêtre montrant l'objet correspondant au focus.
- `+` ou `p` : augmente la valeur de `DEFAULT_DEPTH_DIFFERENCE`.
- `-` ou `m` : diminue la valeur de `DEFAULT_DEPTH_DIFFERENCE`.

Exemples de rendu



FIGURE 1: le focus est le cône et $DEFAULT_DEPTH_DIFFERENCE = 10$



FIGURE 2: le focus est le livre *Networking* et $DEFAULT_DEPTH_DIFFERENCE = 6$

Difficultés rencontrées

La principale difficulté était le respect de la condition : *"le flou d'un objet proche se mélange avec les objets de l'arrière-plan mais la réciproque est fausse."* Nous l'avons interprétée de plusieurs façons différentes qui m'ont pas abouti, mais la découverte du flou gaussien sélectif était la piste qui nous a amené à un résultat acceptable : un *vrai flou*.

Une autre difficulté était le choix des paramètres : σ , matrices de tailles fixes ou variables, tailles en fonction de σ ou non. Ce n'est pas facile de trouver un choix universel.