Rapport :

Détection et élimination d'occlusions

IMA 206 - Télécom ParisTech

Encadrant : D. Yohann Tendero

Romain Deffayet
Saousan Kaddami

Sommaire :

I. Introduction

1. Définition du problème

Cette étude a pour objectif de résoudre un problème que les photographes peuvent souvent rencontrer. En effet, dans certaines situations, prendre une photo claire et sans obstacles s'avère énigmatique. C'est le cas, à titre d'exemple, quand le photographe a devant lui des passants qui l'empêchent de prendre la bonne image en passant devant: dès qu'un passant perturbe la prise d'image, il y'en a un autre qui fait de même en passant devant la scène ou le monument qu'il souhaite capturer. Ces dits obstacles peuvent également être des objets comme les voitures, les grilles ou les barrières. Pour prendre une photo correcte dans telles situations, il faut donc beaucoup de temps et de persévérance. Cette étude vise à proposer une méthode rapide et efficace pour résoudre ce problème en combinant plusieurs images. L'idée est d'utiliser plusieurs images avec obstacles pour en déduire une sans.

Plusieurs travaux ont essayé de s'intéresser à cette problématique. Quelques auteurs se concentrent sur des types d'obstacles bien spécifiques comme les réflexions, les grilles ou les barrières. D'autres utilisent une suite très dense d'images pour en dériver un flux optique à partir duquel une carte de profondeur est déduite et la surface la plus éloignée est gardée. En parallèle, d'autres auteurs utilisent de spécifiques configurations denses d'un capteur qui permettent une décision statistique. Si aucune information de la scène de base n'est observée, une ultime stratégie consiste à procéder à un inpainting sur les zones qui manquent avec le contenu le plus probable après avoir utilisé un masque de détection. Toutefois, les stratégies d'inpainting sont généralement vouées l'échec car elles introduisent des erreurs et des artefacts en arrière plan. De surcroît, dans beaucoup de situations, la vitesse de trame n'est pas assez élevée pour permettre un flux optique consistant pour les calculs et les hypothèses sur les tailles de masques ne s'avèrent pas vérifiées.

Pour toutes ces raisons, nous convenons qu'un algorithme simple, rapide et efficace serait recherché. La solution proposée dans cette étude repose sur les mouvements du photographe combinés aux mouvements de masques pour assurer que la suite d'images révèle tout l'arrière plan une ou plusieurs fois. En alignant géométriquement et photométriquement les images, on forme un paquet d'images. Par conséquent, pour chaque pixel, nous obtenons un paquet de valeurs et nous décidons de l'arrière-plan. Comme sous-entendu, la méthode proposée n'assume aucune taille, couleur, mouvement ou texture spécifiques pour les occlusions. En surcroît, l'algorithme proposé est simple, rapide et peut être comparé avantageusement à une approche plus sophistiquée qu'est celle du robuste PCA.

2. Hypothèses réalisées

Dans cette étude, nous imposons des hypothèses sur l'objet d'intérêt, dit arrière-plan, que l'on souhaite extraire des images.
Nous supposons donc que cet arrière plan issu de la suite d'images est :

1- **Quasi Lambertien** : sa luminance est *uniforme angulairement*, c'est-à-dire identique dans toutes les directions. Ainsi, cet algorithme n'est pas conçu pour marcher correctement quand on observe des arrières-plans avec des surfaces réfléchissantes ou spéculaires comme les miroirs.

2- Peut être **ramené à une image de référence**. Cette hypothèse est réalisée si la scène est **plane** ou si la caméra fait **une rotation autour de son centre optique**.

NB:

- Remarquons qu'on n'a fait aucune hypothèse sur le contenu de l'arrière-plan, sa distribution de couleurs, sa continuité ou sa texture. Par ailleurs, nous le supposons quasi Lambertien pour qu'il n'y ait pas de différences de couleurs significatives quand on le regarde de différentes positions.
- Nous supposons en plus que les conditions de luminosité sont presque constantes durant l'acquisition.

3. Exemple

Nous remarquons dans ce cas que l'ensemble d'image représentant la boulangerie est perturbée d'un côté par les passants qui passent devant, et d'un autre par le motard (image 2). Ainsi, le but est de combiner toutes ces images différentes pour en tirer une où il n'y a aucun obstacle.

II. Méthode par la médiane

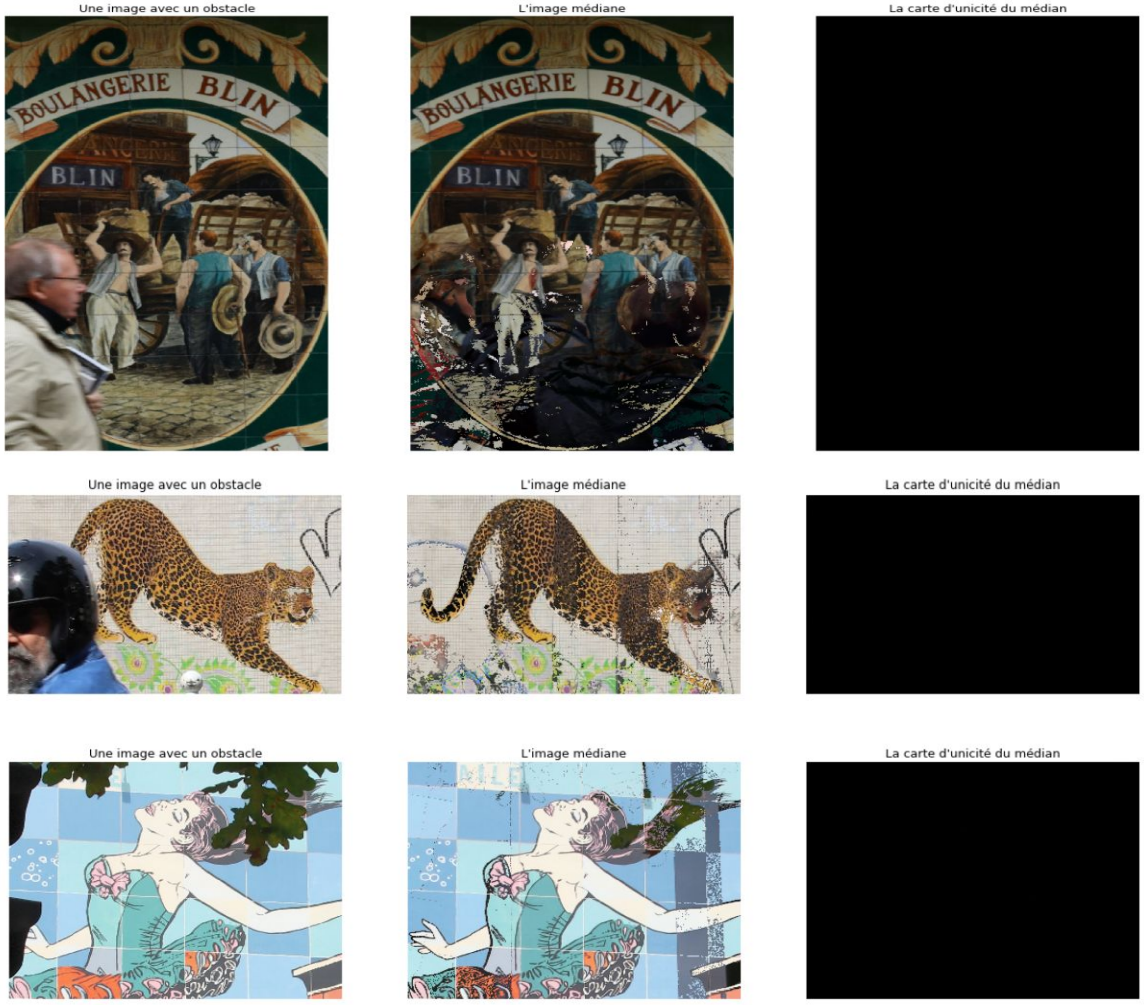
1. Médian vectoriel

Selon cette approche, on détermine pixel par pixel l'image à retenir en tant que fond. On dispose de N vecteurs RGB de longueur 3, et on sélectionne alors le vecteur médian. Pour cela, on définit sur le pixel x le médian comme le vecteur minimisant la somme des distances aux autres vecteurs :

$$I_{med}(x) = \underset{\mathrm{I_k(x)},\mathrm{k}\in\{1,\ldots,\mathrm{N}\}}{\operatorname{argmin}} \sum_{i\in\{1,\ldots,N\}} \|I_k(x) - I_i(x)\|^2$$

2. Résultats

Lorsqu'on applique cet algorithme, on reconnaît le fond mais de nombreux artefacts polluent le résultat. En effet la méthode du médian suppose qu'au moins 50% des images représentent une partie du fond au pixel étudié afin de garantir un résultat satisfaisant. De plus, aucune forme de cohérence de l'image n'est ici retenue, ce qui favorise l'apparition d'artefacts de petite taille.
Par ailleurs, on peut observer la carte d'unicité du médian, qui montre ici que le vecteur minimisant les distance est toujours unique dans notre cas.



III. Méthode par cliques

1. Principe

Puisque la méthode de la médiane a des performances limitées, on souhaite proposer une nouvelle stratégie qui résolve ses limitations.
Nous ne supposons aucun modèle spécifique pour le signal, les masques ou la proportion de masques sur l'arrière-plan pour le paquet d'images. Pour ce faire, nous remarquons que si beaucoup d'images révèlent l'arrière-plan à un pixel donné, leurs valeurs seraient très proches les unes des autres. Par conséquent, pour chaque pixel x, nous cherchons un **sous-ensemble dense**, dit **clique** définie comme suit:

Definition 1. (Dense clique) *Let* $v_1, \ldots, v_n \in \mathbb{R}^3$ *and* $V := \{v_1, \ldots, v_n\}$. *A clique* $C \subset V$ *such that* card $V = m$ *is said dense if* $\forall\, v \in C$ *its* m*–*1 *nearest neighbors in* V *are in* $C \backslash \{v\}$.

A partir de cette définition, et du paquet d'images Φ(x), tel que :

$$\Phi(\mathbf{x}) = \{\mathbf{I}_i(\mathbf{x}),\ i \in \{1, \ldots, n\}\}$$

où $I_i(x)$ est un vecteur de $\mathbb{R}^3$ représentant la valeur de l'image i au pixel x, nous calculons les cliques denses. Ainsi, si des groupes d'images ont une même valeur pour un pixel donné, l'un de ces groupes peut vraisemblablement représenter l'arrière-plan.

L'algorithme utilisé pour obtenir les cliques denses pour un pixel donné et en considérant les m plus proches voisins est le suivant :

Data: Set $\Phi(\mathbf{x})$ (see (1)), positive integer m
Result: Meaningful cliques set $S(\mathbf{x})$.
Set $S = \varnothing$ and compute the $n \times m$ matrix made with indexes of nearest neighbors (NN) of $\mathbf{I}_i(\mathbf{x})$. Namely $\forall i \in \{1, \ldots, n\}, \text{col}(M, i) = (i, 1\text{st-NN} \ldots, m\text{–1th-NN})$.
for *i=1,…, n-m+1* **do**
  for *j=0,…,m-1* **do**
    if $col(M, i) \neq col(M, i + j)$ **then**
      Break
    end
    if *j==m-1* **then**
      $S := S \cup \text{col}(M, i)$
    end
  end
end
return S

Algorithm 1: Dense clique computation.

Par ailleurs, pour choisir entre ces cliques, on introduit la notion de clique significative (meaningful clique), définie par :

Definition 2. (Meaningful clique) *We posit the same setup as in definition 1. We say that a dense clique C is meaningful if every other dense clique $\widetilde{C}$ satisfies* $\operatorname{card} \widetilde{C} \leqslant \operatorname{card} C$ *and* $\operatorname{var} C \leqslant \sigma_T^2$, *where σ_T^2 is a given threshold.*

En pratique, l'algorithme implémenté pour déduire les cliques significatives des cliques denses est le suivant :

Data: Set $\Phi(\mathbf{x})$ (see (1)), treshold σ_T^2.
Result: Meaningful clique $C(\mathbf{x})$.
Set $n := \operatorname{card} \Phi(\mathbf{x})$, $m := 2$, $s := 0$, $S_{\text{pre}} := S_{\text{cur}} := \varnothing$
do
  Set $S_{\text{pre}} := S_{\text{cur}}$, $S_{\text{cur}} :=$ Algorithm 1 $(\Phi(\mathbf{x}), m)$,
  $m := m + 1$ and $s := \operatorname{card} S_{\text{cur}}$
while $s \geqslant 2$
Compute $\sigma^2 := \begin{cases} +\infty \text{ if } S_{\text{cur}} = \varnothing \\ \sigma^2 := \operatorname{var} C, \text{ for } C \in S_{\text{cur}} \end{cases}$
if $\sigma^2 \leqslant \sigma_T^2$ **then**
  return $C \in S_{\text{cur}}$
else
  return $\arg\min_{C \in S_{\text{pre}}} \operatorname{var} C$
end
Algorithm 2: Meaningful clique computation.

2. Résultats

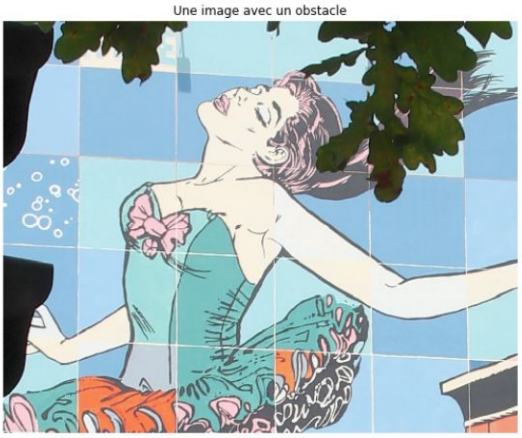
Les résultats obtenus par la méthode de clique pour les images de Boulangerie sont représentés ci-dessous. Nous remarquons la présence d'artefacts noirs dans l'image au milieu, tout en éliminant les obstacles.

Une image avec un obstacle

L'image obtenue

En revanche, les résultats obtenus pour le léopard sont quasiment satisfaisants. Nous ne remarquons aucun artefact ou tâche sur l'image. Cette méthode de clique a donc été efficace dans ce cas particulier.

La situation est par ailleurs différente pour l'image de la danseuse où on remarque des artefacts en haut à gauche.



Ainsi, nous avons conclu que cette méthode par les cliques marche plus ou moins bien selon les images considérées. On a donc pensé à d'autres stratégies pouvant potentiellement être généralement efficaces, comme la méthode des patchs.

3. Avantages et inconvénients

Par rapport à la méthode de la médiane, nous remarquons des résultats beaucoup plus satisfaisants quelque soit l'ensemble d'images considéré.
Pour le Léopard, le résultat obtenu est sans artefact, et très correct comparé à ce qu'on avait obtenu par la méthode de la médiane. De surcroît, quoiqu'on n'ait pas de résultat sans artefact pour la Danseuse, on remarque par ailleurs la grande amélioration par rapport à ce qu'on avait obtenu par la médiane (en l'occurrence pour les cheveux par exemple). Dans cette même optique, on remarque une amélioration pour le cas de la Boulangerie. En effet, par la méthode de la médiane, on avait quasiment perdu toute l'information en bas à droite, ce qui est différent par la méthode de clique; on se retrouve par ailleurs avec des artefacts en bas à droite qui font que le résultat n'est pas satisfaisant.

En revanche, on remarque que cet algorithme marche plus ou moins bien selon les images considérées. On ne peut donc juger de son adéquation pour tous les cas de figures. De plus, la complexité temporelle est plus importante qu'avec la méthode de la médiane. Enfin, la proportion de pixels bien classés est nettement meilleure qu'en appliquant la méthode de la médiane, mais on considère l'image pixel par pixel, ce qui nuit à la cohérence du résultat final. Ces limites amènent donc à penser à d'autres stratégies par patchs, à titre d'exemple, pour avoir plus de cohérence globale et recourir aux problèmes posés par les cas de la Boulangerie et la Danseuse.

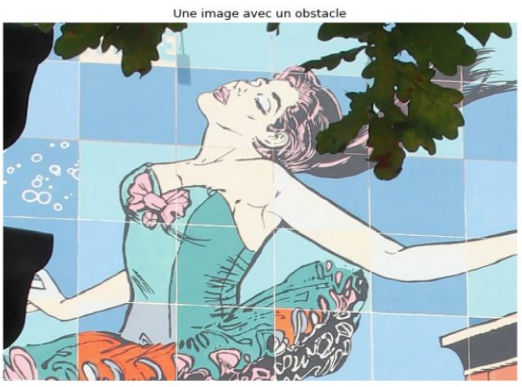
IV. Méthode par patchs

1. Principe

Dans cette partie, nous nous proposons de considérer non plus un voisinage simplement temporel pour chacun des pixels étudiés mais un voisinage spatio-temporel. Pour cela, on applique sur chacune des images un patch carré de taille $w \times w$. On peut alors appliquer la méthode par les cliques étudiée précédemment.
Il s'agit donc de trouver une clique significative parmi les $N \times w \times w$ vecteurs RGB sélectionnés. Néanmoins cela est très coûteux en temps de calcul et on ne peut pas obtenir des résultats en un temps raisonnable. Pour permettre ceci, nous avons utilisé une variante : en concaténant les $w \times w$ vecteurs RGB sélectionnés pour chacune des images, on se ramène à trouver une clique significative parmi N vecteurs de taille $w \times w \times 3$, ce qui diminue drastiquement le nombre de calculs de distances entre vecteurs à effectuer.

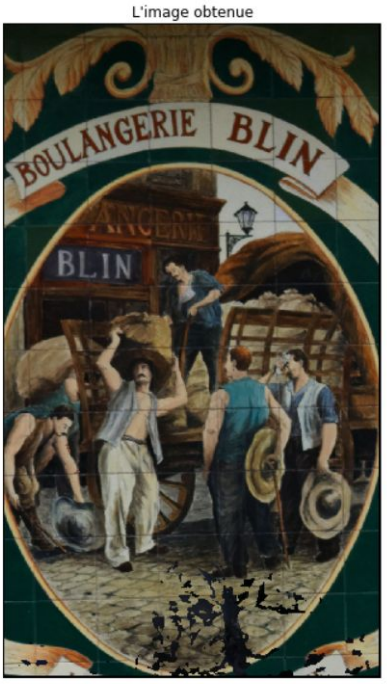
2. Résultats

- Les images représentants la danseuse ne présentaient que peu d'artefacts avec la méthode des cliques sans patchs. En appliquant simplement un patch de taille $w = 11$, on obtient le résultat escompté :

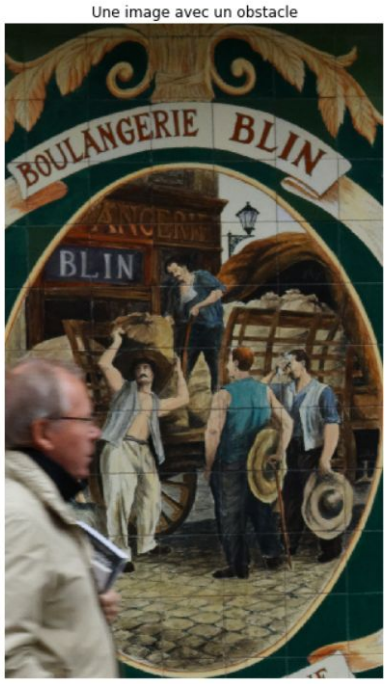


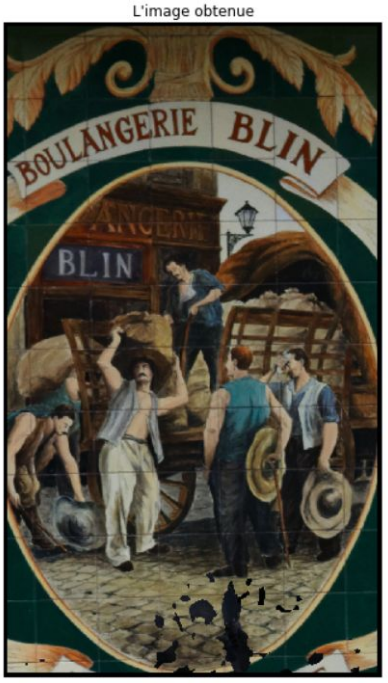
- Pour le set d'images représentant la boulangerie, où les artefacts étaient plus nombreux car deux cliques différentes s'affirment comme significatives, voici les résultats que l'on peut obtenir avec des patchs de taille respectives $w = 7, w = 17$ et $w = 31$:

$w = 7$:

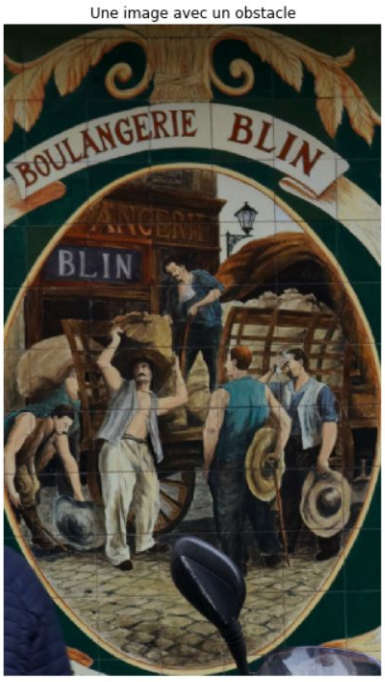


$w = 17$:





$w = 31$:



Les résultats obtenus montrent que l'augmentation de la dimension du voisinage spatial considéré entraîne la diminution du nombre d'artefacts, malgré une augmentation de la taille de ceux-ci.

3. Avantages et inconvénients

En utilisant des patchs sur les images, on réduit le nombre d'artefact présents sur l'image obtenue et on améliore ainsi la cohérence globale du résultat. Dans des cas peu ambigus (lorsqu'une clique se démarque très nettement des autres en tous points), cette méthode nous permet même d'obtenir le résultat parfait espéré. Néanmoins, le temps de calcul est significativement augmenté, particulièrement avec des gros patchs, rendant difficile l'application à une fonctionnalité disponible sur smartphone par exemple.
De plus, lorsque plusieurs obstacles ont des couleurs similaires, on voit apparaître une large zone ne correspondant pas au fond. Ainsi, cette méthode nécessite un certain nombre de prises de vues différentes afin de garantir que le fond soit correctement identifié en tous points.

4. Variante : seuil de distance

Une méthode permettant de limiter les mauvais choix effectués lors de la méthode des patchs est d'ajouter un critère de sélection des cliques denses, afin de favoriser

la clique correspondant au fond. Nous avons choisi d'imposer un seuil sur la somme des distances internes de la clique. Ainsi toute clique trop "variable" sera rejetée. Suivant les cas étudiés, cela peut nous permettre de ne garder que la clique intéressante. Néanmoins, cela requiert un recalage géométrique et colorimétrique très précis lors du pré-processing des images. De plus, il faut réaliser des tests sur chaque image afin de trouver le seuil adapté.

Seuil trop bas | Seuil correct | Seuil trop haut