ROB 317
Vision 3d
Examen 2023 - Corrigé

1 Caméras RGB-d actives

Quelles sont les contraintes qui peuvent rendre difficile l'emploi de caméras RGB-d actives (i.e. qui projettent de la lumière sur la scène) *en environnement extérieur* ? Dans quelles conditions extérieures les caméras temps de vol (ToF) vont-elles mieux fonctionner que les caméras à lumière structurée ? Dans quelles conditions sera-ce l'inverse ? Expliquer pourquoi dans tous les cas.

La difficulté principale est liée à la présence possible d'une forte luminosité extérieure qui peut masquer la lumière projetée par la caméra RGB-d active.

— En présence d'une forte lumière extérieure, les caméras à lumière structurée ne pourront pas fonctionner en général car comme elles doivent projeter une mire nette quelque soit la profondeur, le projecteur possède une faible ouverture pour maximiser sa profondeur de champ, ce qui limite également l'intensité de la mire projetée. Les caméras ToF n'ont pas ce problème car la lumière qu'elles émettent est uniforme spatialement.

— À l'inverse, en environnement sombre et en présence de grandes profondeurs (dans un tunnel par exemple), les caméras à lumière structurée pourront mieux fonctionner que les caméras ToF qui peuvent rencontrer des problèmes d'ambiguïté de phase pour les points les plus lointains.

2 Géométrie projective

Une transformation projective 2d ($\mathbb{P}^2 \longrightarrow \mathbb{P}^2$) préserve-t-elle toujours l'alignement de 3 points ? Si oui, le démontrer. Si non, montrer un contre-exemple. Mêmes questions pour le parallélisme de 2 droites.

— Une transformation projective H préserve toujours l'alignement de trois points $\{p_1, p_2, p_3\}$ car elle s'exprime comme une application linéaire de $\mathbb{P}^2$. Or si $p_2 \in [p_1 p_3]$, il existe $(\alpha, \beta); \alpha+\beta = 1$ tels que $p_2 = \alpha p_1 + \beta p_3$, et donc $Hp_2 = H(\alpha p_1 + \beta p_3) = \alpha H p_1 + \beta H p_3$.

— En géométrie projective, deux droites parallèles $l_1 = (a, b, c_1)$ et $l_2 = (a, b, c_2)$ se "coupent" toujours en un point à l'infini $m = (-b, a, 0)$. Mais si l'on applique une homographie H générale (troisième ligne non vide, *cf* transparent 14 du cours 3d-1) à ces droites, leur point de concours devient $Hl_1 \times Hl_2 = H(l_1 \times l_2) = Hm$, par linéarité du produit vectoriel. La troisième coordonnée de Hm vaut $-bH_{31} + aH_{32}$, qui est non nul en général. Le point de concours de Hl_1 et Hl_2 n'est plus à l'infini, et le parallélisme n'est donc pas préservé.

3 Atterrissage automatique

Considérons un drone muni d'une caméra ventrale avec stabilisateur (gimbal), assurant que la caméra pointe au nadir (axe optique toujours vertical, orienté vers le bas). Décrire un système possible d'atterrissage autonome pour le drone en utilisant comme piste une mire peinte sur une surface plane. Quel est le rôle du stabilisateur ici ? Quels paramètres de la caméra est-il nécessaire de connaître ? Quelles informations le drone doit-il embarquer pour assurer cette fonction ? Quelles sont les difficultés potentielles ?

L'idée est que l'image de la mire peinte sur la piste et perçue par le drone lui donne une information de sa position relative vis-à-vis de la piste. Le rôle du stabilisateur est ici de corriger la rotation de façon à maintenir une orientation constante vis-à-vis de la piste d'atterrissage. Cela permet de simplifier l'estimation de la pose du drone à la seule composante de translation. Pour cela, il est nécessaire que le drone embarque (1) une image de la mire représentant la piste à une distance connue (de préférence une faible distance à partir de laquelle le drone pourra déclencher une manœuvre d'atterrissage "en aveugle"), (2) un algorithme de détection de la mire robuste à la perspective, (3) un algorithme d'estimation d'homographie pour estimer les paramètres de l'homographie entre la mire perçue et celle de référence qu'il a embarquée. La difficulté principale vient du fait que les paramètres estimés de l'homographie sont normalisés par (a) la distance au sol, et (b) les paramètres intrinsèques de la caméra, qui ne sont pas forcément connus (un altimètre et une calibration préalable de la caméra permettent néanmoins de simplifier le problème). Il vaut donc mieux ne pas chercher à estimer un vecteur de translation en une seule fois, mais plutôt estimer simplement la direction d'approche (avec un vecteur de norme ε fixé), et fonctionner de proche en proche.

4 Droites épipolaires

Qu'est-ce qu'une droite épipolaire pour une caméra en mouvement ? Est-elle toujours définie quelque soit le mouvement ? Comment la calculer ? Citer deux utilisations pratiques des droites épipolaires en vision par ordinateur.

Pour une caméra en mouvement, les droites épipolaires sont formées par l'intersection des deux plans focaux avec le plan formé des deux centres optiques et du point 3d qui se projette sur chaque plan focal. Elles ne sont définies que si les deux centres optiques sont distincts dans le mouvement, c'est-à-dire si la composante de translation est non nulle. On calcule les droites épipolaires en estimant la matrice fondamentale F associée à la pose de la caméra aux deux instants, ce qui est possible à partir d'un certain nombre de correspondances entre les deux images. Puis, pour chaque point m projeté dans le plan focal au temps t_1, la droite épipolaire associée dans le plan focal au temps t_2 est donnée par $l' = Fm$. On peut citer deux utilisations des droites épipolaires en vision par ordinateur :

- pour faciliter la recherche de m', le correspondant de m dans le deuxième plan focal, car on sait que, si le point n'a pas bougé entre t_1 et t_2, $m' \in Fm$.
- pour détecter les objets mobiles par rapport à la scène, en calculant les "vrais" correspondants m' de chaque point m (par un algorithme de flot optique par exemple), et en calculant la distance de chaque point à "sa" droite épipolaire, par $d(m', l') \propto |m'Fm|$.

5 Recalage de nuages de points

Rappeler brièvement le principe de la méthode ICP (*Iterated Closest Point*) pour recaler deux nuages de points 3d. Quelles sont les limitations de la méthode ? Montrer un exemple mettant en défaut l'algorithme. Quelles solutions existent pour accélérer l'algo-

rithme ? Dans le cas où on possède une information d'apparence sur chaque point (couleur, géométrie locale,...) comment cette information peut-elle améliorer l'algorithme ?

ICP est un algorithme de recalage de nuages de points 3d consistant à estimer itérativement une transformation rigide entre deux vues (i.e. correspondant à une translation et rotation du capteur par rapport à la scène supposée statique), à partir de la mise en correspondance des points entre chaque vue selon le critère du plus proche voisin dans l'espace 3d. Les limitations de cette méthode sont son coût calculatoire important et sa très grande dépendance à l'initialisation, qui exige que les deux nuages soient assez proches. Un cas typique mettant en défaut l'algorithme est celui du suivi d'un mur, où l'algorithme risque de très vite converger malgré l'ambiguïté du recalage. Concernant le coût calculatoire, deux solutions principales existent pour accélérer l'algorithme : (a) réduire le nombre de points utilisés pour estimer la transformation, et (b) utiliser un algorithme optimisé de recherche du plus proche voisin tel qu'un kd-tree. Enfin, si l'on dispose d'information sur la géométrie locale autour de chaque point, on peut améliorer l'algorithme en réalisant une mise en correspondance non plus sur la proximité spatiale (distance euclidienne en 3d), mais sur la ressemblance locale (distance dans l'espace des descripteurs).

6 Prédiction monoculaire de cartes de profondeur

Citer trois indices de profondeur qu'un réseau de neurones pourrait exploiter dans les images monoculaires. Pour chaque indice, imaginer un protocole d'apprentissage supervisé aussi simple que possible à partir d'images synthétiques. Dans chaque cas, proposer une fonction de loss permettant autant que possible de limiter les défauts du protocole proposé. Comment est-il possible d'intégrer les différents indices au sein du même apprentissage ?

Les réponses sont multiples ici : on peut parler des occultations, du flou, de la taille des objets, de la perspective, des ombres... Les protocoles d'entraînement et les fonctions de loss sont aussi très variés, par exemple :

- pour le cas des occultations, on pourrait entraîner le réseau avec des images synthétiques faites de formes ou d'objets 2d s'occultant partiellement et trouver une fonction de loss qui pénalise les zones pour lesquels l'ordre (partiel) de profondeur n'est pas respecté...
- pour la taille des objets, on pourrait de la même façon générer des images avec des objets de tailles différentes et proposer une fonction de loss qui force la prédiction de profondeur à être inversement proportionnelle à la taille des objets...
- pour le flou, on pourrait générer des images réalistes avec un moteur de synthèse physique permettant de simuler une caméra avec une faible profondeur de champ, et utiliser une fonction de loss qui minimise la différence avec une fonction de profondeur relative signée, où le zéro correspondrait à la distance du plan de netteté...

Pour intégrer les différents indices au sein du même apprentissage, on peut soit (1) générer une base d'entraînement de synthèse qui combinent les différents indices en utilisant une fonction de loss qui combinent celles associées aux différents indices, soit (2) appliquer séquentiellement les différents entraînement, dans l'esprit d'un curriculum learning.