

CSC_5RO17_TA (ROB317)

Vision 3d - 2024

Corrigé du Contrôle

1 Caméra RGB-d à lumière structurée

- **Quel est l'objectif de la double calibration d'une caméra RGB-d à lumière structurée, i.e. celle de la caméra d'une part, et celle du projecteur de la mire d'autre part ?**
 - L'objectif de la calibration pour la caméra est d'estimer la géométrie du plan focal (distance focale, centre de projection, éventuellement paramètres de distortion de la lentille), afin de connaître, pour chaque pixel, la direction du point 3d qui s'y projette.
 - L'objectif de la calibration pour le projecteur est d'associer à chaque motif élémentaire de la mire une direction, à partir de la projection de la mire sur un plan de référence placé à une distance fixe.
- **Définir le concept de profondeur de champ, et expliquer comment il s'applique à la fois à la caméra et au projecteur dans le cas d'une caméra RGB-d à lumière structurée.**
 - Pour la caméra, la profondeur de champ désigne la largeur de l'intervalle de distances pour lesquelles les points de la scène vont apparaître nets (i.e. vont former une tache de la taille de l'ordre du pixel). Elle est inversement proportionnelle à l'ouverture du diaphragme. En lumière structurée, la caméra doit maximiser sa profondeur de champ pour bien détecter les motifs quelque soit leur distance.
 - De même, le projecteur doit pouvoir projeter tous les motifs de la mire avec le même niveau de netteté quelque soit la distance de l'objet sur lequel ils se projettent. Il faut donc maximiser sa profondeur de champ, c'est-à-dire minimiser son ouverture.

2 Homographies et segmentation en plans

- **Qu'est-ce qu'une homographie et dans quels cas pratiques la rencontre-t-on en vision par ordinateur ?**
 - Une homographie est une transformation projective 2d qui préserve l'alignement de tout triplet de points.
 - On la rencontre en pratique dans deux cas particuliers de transformation liant une paire d'images : (1) une scène plane vue de deux points de vue distincts. (2) une scène quelconque vue de deux points de vue liés par une rotation pure (sans translation) autour du centre optique.
- **Considérer le cas d'un drone volant dans un environnement intérieur de bureaux (sol, plafond et murs plans). Si le drone est muni d'une seule caméra RGB, ima-**

gérer (dans les grandes lignes) un algorithme de segmentation des images en ses différents plans, fondée sur l'estimation d'homographies.

- Plusieurs propositions sont bien sûr possibles. Une solution assez simple serait de réaliser itérativement, à partir d'une paire d'images acquise par le drone en déplacement, une estimation d'homographie. Pour cela on peut utiliser un algorithme de type RANSAC pour choisir aléatoirement, parmi un ensemble de points d'intérêt calculés sur les deux images et appariés, des couples de points utilisés pour l'estimation (on peut biaiser RANSAC en ne considérant les points que dans un quadrant de l'image par exemple). Une fois l'homographie estimée, on exclut ses inliers de la liste des couples et on réitère le processus.

3 Contrainte épipolaire

- **Qu'est-ce que la contrainte épipolaire qui lie 2 images de la même scène perçue depuis deux points de vue différents ? Comment s'exprime-t-elle mathématiquement ?**
 - Il s'agit d'une contrainte géométrique liant deux points m_1 et m_2 provenant du même point 3d M et projetés sur deux plans focaux dont les centres optiques ne sont pas confondus. M , m_1 et m_2 forment alors un plan qui coupe chaque plan focal en une droite dite épipolaire qui contraint donc la recherche de points homologues en géométrie multi-vues.
 - Elle s'exprime mathématiquement grâce à la matrice fondamentale F dans le cadre de la géométrie projective, par la contrainte quadratique :

$$m_1^t F m_2 = 0.$$

Les droites épipolaires sont alors données par :

$$d_1 = F m_2 \text{ et } d_2^t = m_1^t F.$$

- **Lorsque les deux points de vue sont fournis par la même caméra qui s'est déplacée entre deux instants t_1 et t_2 , comment peut-on utiliser cette contrainte pour détecter les objets qui se sont déplacés par rapport à la scène entre t_1 et t_2 ?**
 - Pour un point M qui se serait déplacé indépendamment de la caméra, projeté en m_1 à t_1 et en m_2 à t_2 , la contrainte épipolaire n'est pas respectée. On peut donc utiliser comme indice de mobilité une mesure de déviation par rapport à la contrainte épipolaire, telle que la distance de m_2 à la ligne épipolaire :

$$\delta(m_2, d_2) = \frac{|d_2^t m_2|}{\sqrt{(d_2^1)^2 + (d_2^2)^2}}.$$

4 Apprentissage auto-supervisé de cartes de profondeur

On considère dans cette partie un réseau de neurones entraîné de façon auto-supervisée à prédire une carte de profondeur pour une caméra monoculaire en mouvement.

- **Expliquer comment on passe de la carte de profondeur à un nuage de points 3d. Quelles sont les informations nécessaires en plus de la carte de profondeur ?**
 - On peut passer de la carte de profondeur \mathbf{z} à un nuage de point 3d si l'on connaît les paramètres intrinsèques de la caméra - distance focale f et centre optique (c_x, c_y) - fournis par la matrice K , en rétroprojetant chaque pixel p comme suit :

$$X(p) = \mathbf{z}(p) K^{-1} p, \text{ avec } K^{-1} = \begin{pmatrix} \frac{1}{f} & 0 & -\frac{c_x}{f} \\ 0 & \frac{1}{f} & -\frac{c_y}{f} \\ 0 & 0 & 1 \end{pmatrix}$$

- On a vu en cours le principe des méthodes auto-supervisées fondées sur la rétro-projection par la caméra C_1 et re-projection sur la caméra C_2 pour le calcul de la loss photométrique. Dans le cas où le réseau prédit à la fois la carte de profondeur et les paramètres de déplacement de la caméra, proposer une autre méthode auto-supervisée, fondée sur la rétro-projection de nuages de points par les deux caméras C_1 et C_2 . Quelle fonction de loss proposez-vous ?
- Si $\{X_1(p)\}$ et $\{X_2(p)\}$ sont les nuages de points retroprojetés par les caméras C_1 et C_2 , et qu'on a une prédiction de la matrice de déplacement P , on peut comparer les 2 nuages $\{X_2(p)\}$ et $\{PX_1(p)\}$. Une loss possible est :

$$\sum_p \|X_2(p) - PX_1(p)\|^2.$$

- Considérons maintenant que le réseau est entraîné seulement à prédire une carte de profondeur, mais pas les paramètres de déplacement de la caméra. Peut-on combiner la méthode précédente avec l'algorithme ICP (*Iterated Closest Point*) ? Décrire brièvement dans ce cas la nouvelle méthode et la fonction de loss proposée.
- Dans le cas où les paramètres de déplacement P ne sont pas prédits par le réseau, on peut utiliser un algorithme de recalage de nuages de points tel que ICP, qui estime itérativement une transformation rigide $P^{(i)}$ à partir de l'appariement des plus proches voisins, et utiliser la valeur finale de la fonction de coût obtenue par l'algorithme (typiquement $\sum_p \|X_2(p) - P^{(\infty)}X_1(p)\|^2$) comme fonction de loss.