

Acquisition d'images par un réseau de caméras asynchrones



Etudiants :

Fabien GAIFFE
Nicolas HUGUET
Raphaël PREVOST

Marthe CUNY
Marie-Solène TEURNIER

Enseignant-responsable du projet :

Abdelaziz BENSRAIR

Date de remise du rapport : **16/06/2014**

Référence du projet : **STPI/P6/2014 – 001**

Intitulé du projet : **Acquisition d'images par un réseau de caméras asynchrones**

Type de projet : **Bibliographique et expérimental**

Objectifs du projet :

* Découvrir, via de nombreuses présentations, la vie d'un laboratoire d'informatique et de traitement de l'information et des systèmes (LITIS), les différentes activités et recherches qu'on y mène, ainsi que leurs applications et usages pratiques dans le quotidien

* Nous pencher sur le cas concret d'un réseau de caméras asynchrones et de la notion de stéréovision :

- découverte de son mode de fonctionnement
- prise en main expérimentale d'un réseau asynchrone sur banc stéréoscopique, de la calibration à l'acquisition d'une banque d'images
- comprendre le traitement et l'interprétation informatique des informations et images captées
- pressentir les usages d'un tel dispositif et ses intérêts

Mots-clefs du projet :

- **Caméras asynchrones**
- **Stéréovision**
- **Calibration**
- **Acquisition**
- **Véhicule autonome**

Table des matières

1.Introduction.....	5
1.1.Contexte	5
1.2.Lien avec le LITIS.....	5
1.3.Notre projet.....	6
2.Méthodologie / Organisation du travail.....	7
3.Travail réalisé et résultats.....	8
3.1.Résumés des présentations.....	8
3.2.Matériel utilisé.....	10
3.2.1.Objectifs.....	10
3.2.2.Capteurs.....	10
3.2.3.Moyens de connexion.....	11
3.3.Manipulations.....	11
3.3.1.Acquisition d'images par une seule caméra.....	12
3.3.2.Acquisition d'images par un réseau de caméras synchrones.....	14
3.3.3.Acquisition d'images par un réseau de caméras asynchrones.....	16
3.4.Calibration.....	18
3.4.1.Paramètres intrinsèques d'une caméra.....	18
3.4.2.Paramètres extrinsèques d'une caméra.....	19
3.4.3.Le fonctionnement théorique en quelques mots.....	20
3.4.4.La calibration, en pratique.....	21
4.Conclusions et perspectives.....	26
4.1.Conclusions sur le travail réalisé.....	26
4.2.Conclusions sur l'apport personnel de cet E.C. projet.....	26
4.3.Perspectives pour la poursuite de ce projet.....	28
5.Bibliographie.....	29
6.Annexes.....	30

1. INTRODUCTION

1.1. Contexte

Dans le domaine de l'automobile, un secteur en pleine expansion est celui de l'intelligence artificielle qui y trouve de nombreuses applications : il s'agit d'assister la conduite, ou de communiquer avec d'autres voitures, et peut-être à terme d'exécuter des trajets sans apports humains. Il y a de fortes demandes puisque cela permet de gagner beaucoup de temps pour l'utilisateur, pour l'armée d'effectuer des trajets dans des zones à risque ou d'éviter les accidents ou les bouchons pour l'utilisateur moyen.

Le géant américain Google a développé la Google Car sur 8 voitures qui ont déjà parcouru des centaines de milliers de kilomètres en Californie sans provoquer d'accidents. Leur système de pilotage automatique utilise un lidar ou radar laser, très cher, une caméra, des radars, un récepteur GPS et des capteurs sur les roues motrices. Le véhicule doit être conduit de manière ordinaire une fois sur le trajet qu'il empruntera ensuite de manière autonome, afin qu'une équipe vérifie que le véhicule a enregistré tous les signaux importants le long du trajet. Le véhicule n'a plus ensuite qu'à gérer les modifications de signalisation pendant son trajet autonome.

Une des limitations du système est son incapacité à agir suivant les gestes d'un agent de police faisant la circulation

Des compétitions sont organisées, à l'image du DARPA (Defense Advanced Research Projects Agency) Grand Challenge, une compétition mettant en jeu un prix de 2 millions de dollars, et défiant les compétiteurs de concevoir des véhicules autonomes capables de parcourir plus de 200 kilomètres en moins de 10 h, dans le désert, ou de remplir des missions d'approvisionnement en s'intégrant en sécurité au trafic routier.

1.2. Lien avec le LITIS

En partenariat avec l'INSA de Rouen, le LITIS (Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes) est l'unité de recherche dans le domaine des sciences et technologies de l'information de Haute-Normandie. Il regroupe 88 permanents dont 18 enseignants chercheurs de l'INSA. Ses domaines d'application vont de l'accès à l'information au traitement de l'information biomédicale en passant par l'intelligence ambiante et mobilité,

Le laboratoire développe des démarches pour mieux comprendre et maîtriser la nature de l'information et de son utilisation dans le contexte. Les recherches portent à la fois sur des aspects théoriques, algorithmiques et sur la mise en œuvre

de systèmes dépendants du contexte, du capteur à la base de données.

Dans le domaine qui nous intéresse, les recherches de l'équipe STI (Systèmes de Transports Intelligents) concernent l'apport des sciences et technologies de l'information pour l'aide à la conduite de véhicules par l'obtention et le traitement des données.

Son objectif est le développement de systèmes embarqués capables de fournir en temps réel des informations utiles au conducteur, dans un environnement dégradé avec une faible infrastructure pour optimiser et sécuriser les déplacements.

Le laboratoire aborde ce problème à travers trois axes : la perception de l'environnement grâce à des systèmes de vision intelligents, puis le guidage dynamique du véhicule, et l'exploitation de ces résultats dans un problème routier.

1.3. Notre projet

Le système d'IA de la voiture a besoin pour fonctionner d'acquérir des informations exploitables, sur sa localisation, la distance avec les objets qui l'entourent, sa vitesse, la nature de son environnement, sa destination, les plans et le trafic avoisinant.

Notre projet porte sur la détection d'obstacles et de l'environnement par des capteurs optiques (caméras), dans le but final d'avoir une banque d'images propre et utilisable d'une voiture se déplaçant en milieu urbain comme rural ou sur les grands axes, pour avoir des situations diverses. Ces banques de données sont utilisées pour de nombreuses études comme test pour leurs algorithmes.

Notre problème est donc : Comment acquérir une banque d'images utile destinées à un usage dans l'IA appliquée au domaine automobile par un réseau de caméras intéressant pour les industriels?

Les véhicules auront besoin d'un guidage par caméra, qui leur permettra de voir leur environnement, mais une seule caméra n'aurait pas assez de champ de vision pour guider une voiture, et ne permettrait pas d'avoir une idée de la distance à laquelle se trouvent les objets entourant le véhicule. Pour cela, il faut utiliser au moins deux caméras, qui ont le même objet dans le champ de vision. Ensuite, grâce à la stéréovision, des algorithmes seront capables de donner de la profondeur au milieu vu par le véhicule.

Cette méthode sera détaillée plus bas, mais nous rajoute la contrainte d'avoir un réseau de caméras pour obtenir une banque d'images utiles.

De plus, nous avons un critère supplémentaire qui est que le réseau de caméra sera asynchrone : il sera plus compliqué à gérer mais permettra de se passer des fils entre chaque caméra, gênants dans un véhicule.

Pour ce projet, il va falloir comprendre les technologies des caméras et comparer leurs différentes caractéristiques. Il faudra qu'elles ne soient pas trop chères et que les algorithmes et diverses études fonctionnent avec les caméras qui seront réellement sur la voiture, c'est-à-dire de qualité moyenne et peut être abimées, en milieu dégradé.

Cette méthode avec des caméras asynchrones pour se repérer et se diriger est donc sensée être suffisamment précise et sûre pour être utilisable par un véhicule, et peu onéreuse.

2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

Un cadre de travail nous était imposé : des séances d'une heure et demie chaque semaine durant l'intégralité du second semestre, au sein du laboratoire (bâtiment Bougainville). Les séances commençaient généralement par une présentation en lien avec le projet, par les professeurs, ou des intervenants extérieurs. Ensuite nous travaillions avec le professeur, Mme Rawia Mhiri, sur le côté expérimental et bibliographique de notre projet, par exemple en veillant au bon fonctionnement des caméras et en réalisant quelques acquisitions. En son absence, ne disposant pas des accès administrateurs nécessaires à l'accès aux données informatiques, nous nous penchions plus sur la rédaction de notre rapport, la quête de sources fiables et de divers travaux ou thèses déjà réalisés sur des thèmes proches.

Pour rédiger, il a fallu établir des méthodes de travail claires et rigoureuses, avec par exemple la définition d'une sorte d'organigramme interne au groupe (nomination d'un chef d'équipe -Fabien Gaiffe- et de responsables de thématiques -chaque membre restant se voyant attribuer la responsabilité d'un grand axe de notre projet-). La répartition des tâches de travail a elle aussi été essentielle. De fait, celle-ci s'est fait équitablement, et suivant les préférences de chacun, en veillant à ce que tout le monde y trouve son compte. A chaque fin de séance, l'un des 5 membres était désigné (en alternant chaque semaine, de manière à ce que tout le monde ait été concerné) pour rédiger un récapitulatif complet d'une page environ sur les tâches réalisées au cours de celle-ci, et les enseignements tirés.

Un « Groupe » Facebook s'est révélé être un bon outil de travail et de communication au sein du groupe. En effet, il nous a assuré une certaine coordination de nos travaux individuels de par une organisation facilitée.

Enfin, nous avons opté pour la rédaction de notre projet via un Google Doc ouvert à l'ensemble de l'équipe et modifiable par tous dans le but de pouvoir constater l'avancement de chacun et ainsi éviter tout problème de redondance.

3. TRAVAIL RÉALISÉ ET RÉSULTATS

3.1. Résumés des présentations

Aux cours de nos séances nous avons assistés à des présentations de thésards du département spmii qui soit nous ont permis de mieux comprendre les aspects techniques de notre sujet ou soit nous ont présenté différentes applications des réseaux de caméra.

La première présentation a eu lieu le 4 février et nous a été animé par Mme Rawia Mhiri. Cette présentation avait pour but de nous apprendre certaines bases sur les caméras et leurs applications générales.

Les caméras sont composées de deux éléments : un capteur et un objectif. Plusieurs types de capteur existent, il existe des capteurs qui traitent le visible et d'autre qui traitent les ondes infrarouges. Chaque type de capteur a une gamme d'applications différentes.

Un système de caméra synchrone peut permettre de reconstituer une image en 3D par exemple grâce à un système de deux caméras synchrones. Pour retrouver la 3D, deux méthodes sont possibles :

- Par triangulation: les 2 images sont traitées pour trouver des « points importants » sur les deux images. Ensuite on fait correspondre ces points et grâce à un programme informatique on peut retrouver les coordonnées 3D du point. On parle de méthode éparses car on n'a pas la correspondance de tous les pixels de l'image mais seulement les plus importants.
- Par stereomatching: pour commencer, les images sont rectifiées. Pour cela on prend un point objet commun aux deux images. Puis on mesure la distance entre les coordonnées des deux points. Cela permet de trouver la profondeur du point objet et de reconstituer l'image en 3D. On parle de méthode dense, tous les pixels sont traités. Cette méthode ne marche que si les deux caméras sont synchrones et si on connaît leur position l'une par rapport à l'autre.

Le but de la thèse de Mme Rawia Mhiri est de parvenir au même résultat mais avec un réseau de caméras asynchrones. L'objectif est de pouvoir mettre ce dispositif dans les voitures pour les rendre plus intelligentes à moindre coût et sans changer le mode de fabrication.

La deuxième présentation a eu lieu les 11 février par Mme Fan Wang. La présentation avait pour thème la calibration d'image par ordinateur. Ce sujet sera développé dans un paragraphe plus tard dans le dossier.

La troisième présentation s'est déroulée le 25 mars. M.Pierre Yver nous a présenté son métier et ses recherches liées à la 3D. M.Yver et ses collègues ont créé un logiciel d'aide à la production de vidéo en 3D. Ce logiciel a pour but de simplifier et faciliter l'utilisation de plusieurs caméras pour faire une acquisition en 3D. Lors d'une prise de vue en stéréovision, plusieurs problèmes apparaissent tel que le roulis des caméras, les problèmes de zoom ou de différence de hauteur entre les caméras et rendent le traitement de l'image compliqué. Le logiciel permet de résoudre ces problèmes automatiquement sans aide humaine. Le logiciel propose notamment une gestion automatique des profondeurs et la correction chromatique (il uniformise les couleurs sur les deux images). Ce logiciel intéresse le monde du cinéma, dont James Cameron qui utilise ce logiciel pour ses films.

Une quatrième présentation a été faite par M.Fabien Bonardi le 8 avril. Le sujet de sa thèse est la cartographie dans les systèmes embarqués. Son but est de créer un véhicule autonome avec des caméras peu chères et sans test avant la mise en circulation. C'est-à-dire faire en sorte que le véhicule puisse être autonome même sur une route inconnue, sans avoir besoin de faire des tests sur cette route pour apprendre à la voiture ce qu'elle doit faire. Une voiture autonome pourrait être plus intelligente et elle pourrait permettre une meilleure gestion de l'énergie. Automatiser un véhicule pourrait aussi diminuer le nombre d'accident sur la route car les robots et ordinateurs peuvent réagir plus vite que l'homme en situation dangereuse.

Enfin nous avons eu une cinquième présentation par le docteur Amnir Hadachi. Il a réalisé une thèse à l'INSA de Rouen sur le trafic urbain routier dans les ruelles de Rouen. Il évolue aujourd'hui à l'université de Tartu en Estonie. Au cours de la présentation il nous a parlé de ses travaux. Il travaille notamment sur la faculté d'un système embarqué dans une voiture à prédire son trajet, ainsi que sur la reconstruction 3D d'une scène à partir de photos de téléphone.

Ces présentations nous ont permis d'avoir un peu de culture générale sur les caméras et leurs applications.

3.2. Matériel utilisé

Pour choisir son matériel il faut prendre en compte le coût et la qualité. On ne peut pas faire nos expériences avec des caméras haut de gamme car cela dissuaderai des entreprises qui voudraient commercialiser le système de caméras asynchrones. En revanche, un minimum de qualité est nécessaire pour pouvoir utiliser l'information contenue dans les images. Nous allons voir quels types de caméras numériques existent et pourquoi la camera Basler aca1600-20gc a été choisie. Elle a un capteur de type CCD. Elle peut délivrer jusqu'à 20 images par seconde avec une résolution de 2 millions de pixels. Elle détecte les couleurs et chaque pixel est stocké sur 12 bits.

3.2.1. Objectifs

Un objectif est un système optique convergent constitués de lentilles. Il est caractérisé par sa distance focale. Les rayons lumineux constitués de photons passent à travers l'objectif puis vont frapper le capteur. Pour notre application, il nous faut un objectif peu coûteux, résistant et qui donne au capteur une image fidèle à la réalité. L'objectif Fuji df5ha-1b a été choisi. Il a une distance focale de 6mm. L'angle de vue est de $56^{\circ} \times 43^{\circ}$. Il est conçu pour limiter la distorsion et est léger, robuste et peu coûteux ce qui respecte bien les contraintes liées aux applications de notre expérience.

3.2.2. Capteurs

Les caméras numériques se distinguent principalement par leur capteur. Leur rôle est de transformer la lumière reçue en information sous forme d'un courant électrique. Ils sont constitués de cellules photosensibles qui mesurent l'intensité de la lumière et sa couleur. Chaque point du capteur, qui compose une partie d'un pixel, enregistre l'intensité lumineuse pour produire une image. Le capteur est composé de deux éléments superposés : les cellules photosensibles et le dispositif à transfert de charges. Chaque cellule possède un filtre rouge, vert ou bleu. La proportion des filtres dans le capteur est la suivante $\frac{1}{4}$ de filtres bleus, $\frac{1}{4}$ de rouges et $\frac{1}{2}$ de verts. Cette proportion correspond à la sensibilité de la vision de l'homme. La résolution d'un capteur est définie par le nombre de pixels.

Il y a deux principaux types de capteur : les CCD et les CMOS.

Les CCD

Les capteurs CCD (Charge-Coupled Device) sont composés de cellules photosensibles qui envoient la charge sur un collecteur qui transfère l'ensemble des charges vers le convertisseur. Celui-ci a pour rôle de transformer le signal électrique en code numérique. Ils sont plus gourmands en énergie mais donnent des images de meilleure qualité.

Les CMOS

Les capteurs CMOS (Complementary Metal Oxyde Semi-conductor) fonctionnent de la même manière à quelques détails près. Ils se composent d'un champ de photodiodes sensibles à une des couleurs primaires (rouge, vert, bleu). Au lieu que la charge soit transférée vers un collecteur puis au convertisseur, elle est transférée au convertisseur directement.

Les capteurs CMOS consomment moins d'électricité, ont une meilleure vitesse de lecture et coûtent moins chers. En revanche, ils ont tendance à faire apparaître plus de bruit dans les images.

3.2.3. Moyens de connexion

On utilise un câble Gigabit Ethernet pour connecter notre ordinateur au hub. Ce câble permet de transmettre un gigabit d'information par seconde. La couleur d'un pixel est stocké avec 12bits donc si on a 5 cameras qui envoient 8 images par seconde composées de 1624*1234 pixels. Cela nous fait 961,927,680 bits envoyés par seconde à l'ordinateur ce qui est inférieur à un gigabit donc le câble Ethernet convient. On choisit combien d'images on prend par seconde. Ici, 8 est le nombre maximum d'images que l'on peut prendre par seconde sans dépasser la capacité du câble Gigabit Ethernet.

3.3. Manipulations

Mettre en place un réseau de caméras est une chose compliquée en soi. C'est pourquoi on essaiera d'abord de comprendre, puis de faire fonctionner une acquisition d'une seule caméra.

Une fois l'acquisition réussie, on mettra alors en place un réseau de caméras *synchrones*.

Enfin, lorsque les caméras fonctionneront correctement ensemble (en effet, de nouveaux problèmes peuvent apparaître en faisant fonctionner plusieurs caméras en même temps) et seront capables d'effectuer des acquisitions de caméras *synchrones*, on pourra alors rendre le réseau de caméras *asynchrones*.

3.3.1. Acquisition d'images par une seule caméra

On cherche donc à lancer d'abord une acquisition d'images pour une seule caméra. Pour cela, on aura besoin :

- d'un ordinateur,
- d'une caméra,
- d'un moyen de relier la caméra à l'ordinateur,
- de pilotes (drivers)
- d'une application.

Ici, le logiciel d'acquisition que l'on a par défaut sur notre ordinateur, IC Capture, ne fonctionne pas, car il ne reconnaît pas les drivers et car la caméra a besoin d'une application qui lui est spécifique.

On installe donc un logiciel (*Pylon*) et les drivers correspondant à la caméra *Basler* que l'on possède. Celle-ci doit être reliée à l'ordinateur grâce à un câble Ethernet.

Il faut, de plus, alimenter électriquement la caméra, afin qu'elle puisse fonctionner. On la connecte donc à un hub, branché sur le secteur, qui servira d'alimentation.

La caméra est maintenant en état de fonctionner, et est connectée à la carte réseau de l'ordinateur par câble Gigabit Ethernet.

Cependant, bien que la caméra soit détectée par le logiciel d'acquisition Pylon, on ne peut pas encore s'en servir.

En ouvrant *Pylon IP Configurator*, on constate que la caméra possède une adresse IP propre. Or cette adresse doit correspondre à celle de la carte réseau pour pouvoir fonctionner.

Pour cela, on doit aller dans les propriétés de la carte réseau, dans le protocole internet version 4 (TCP/IPv4), pour forcer une nouvelle adresse IP sur la carte réseau, afin qu'elle corresponde à celle de la caméra. On peut noter que, plus tard, chaque caméra du réseau devra avoir une adresse IP propre différente de celle de notre carte réseau (qui est, ici, 169.254.255.0).

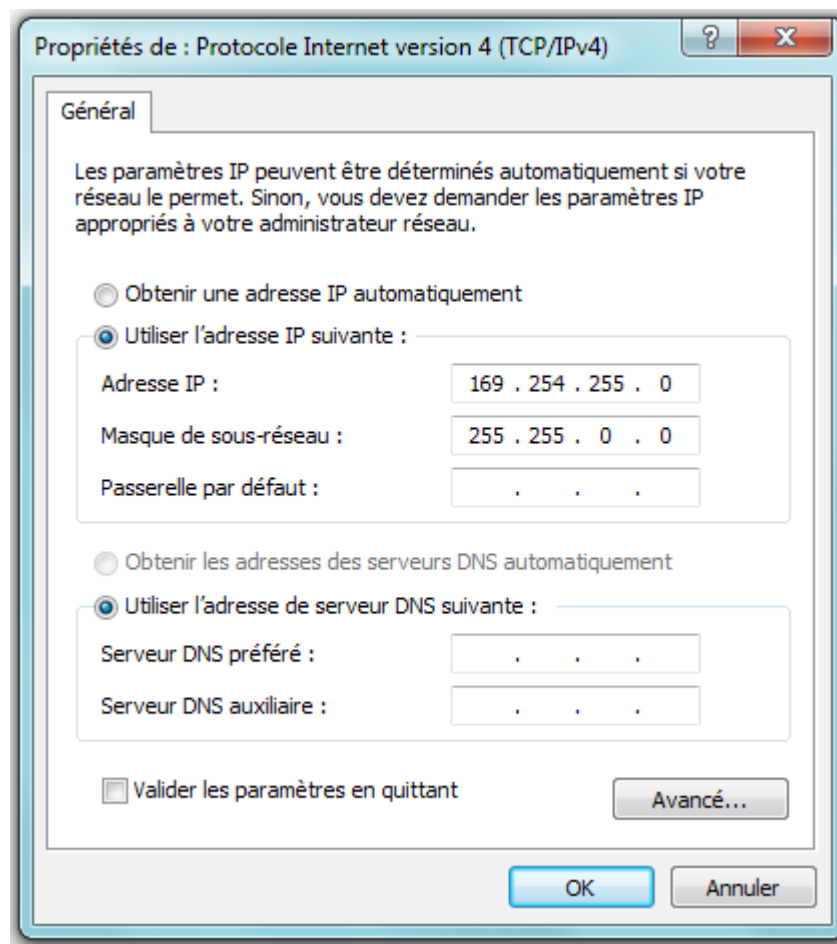


Illustration 1

Par exemple, on attribue ici à notre caméra l'adresse : 169.254.255.5 (avec le masque de sous-réseau: 255.255.255.0) grâce à *Pylon IP Configurator*.

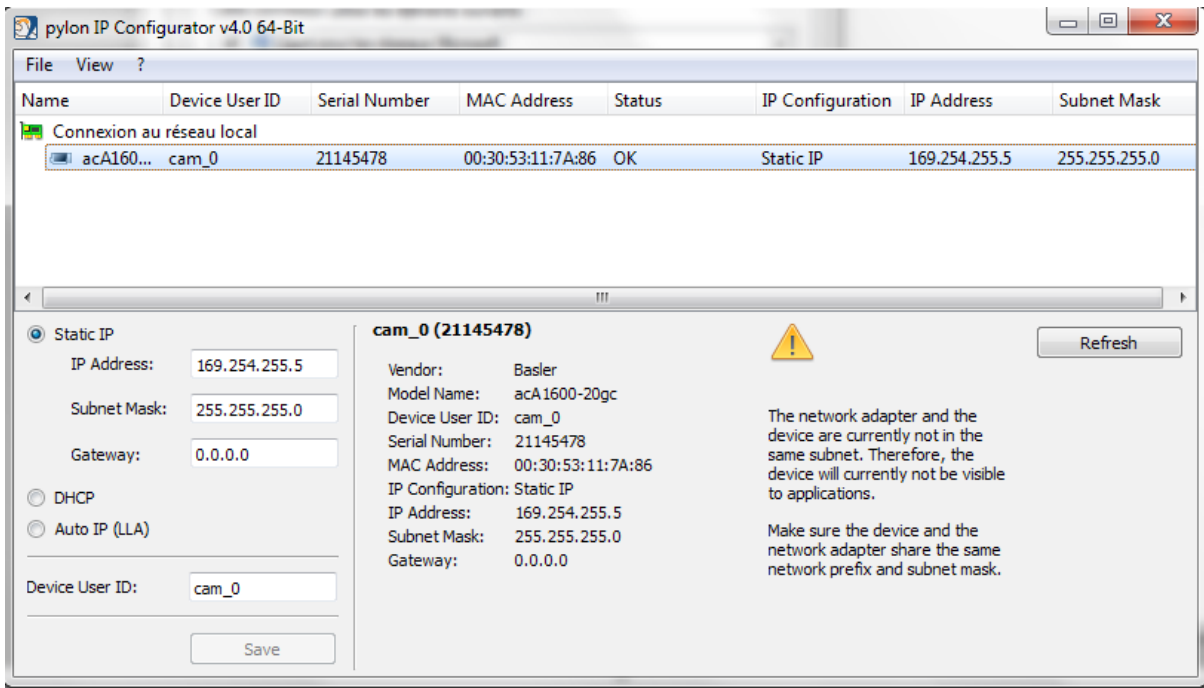


Illustration 2

Une fois cette opération effectuée, le logiciel *Pylon Viewer* nous permet d'obtenir à l'écran les images captées par la caméra.

Dans les réglages, on peut, entre autres, jouer sur le flou de la caméra (quasiment inutile dans notre cas, autant avoir la perception maximale une fois bien réglé pour ne plus y toucher) ou la luminosité.

3.3.2. Acquisition d'images par un réseau de caméras synchrones

On a donc réussi à faire fonctionner une seule caméra sur notre ordinateur.

On peut alors passer à l'étape suivante : mettre en place un réseau de caméras *synchrones*.

Pour cela, on branche les 5 caméras (toujours par un câble Ethernet) à un hub qu'on relie par un câble Ethernet à notre ordinateur. On vérifie bien dans *Pylon IP*

Configurator que la carte réseau a bien une adresse IP qui correspondra aux adresses IP des caméras, et que chaque adresse IP est unique.

On obtient au final un réseau de ce genre :

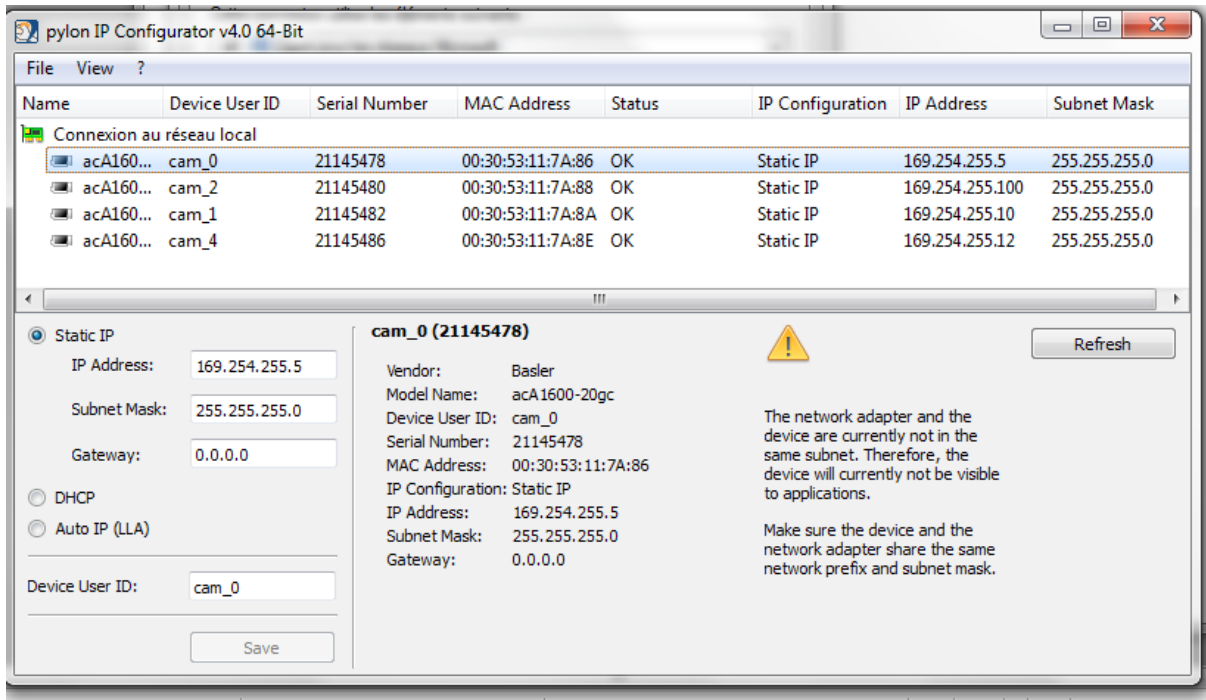


Illustration 3

Avec le hub qui possède l'adresse :169.254.255.0 (celle de la carte réseau).

On ouvre alors le logiciel d'acquisition d'images (*Pylon Viewer*). On remarque que chaque caméra est bien détectée par le programme. On peut visualiser, comme pour une seule caméra, l'acquisition de différentes caméras.

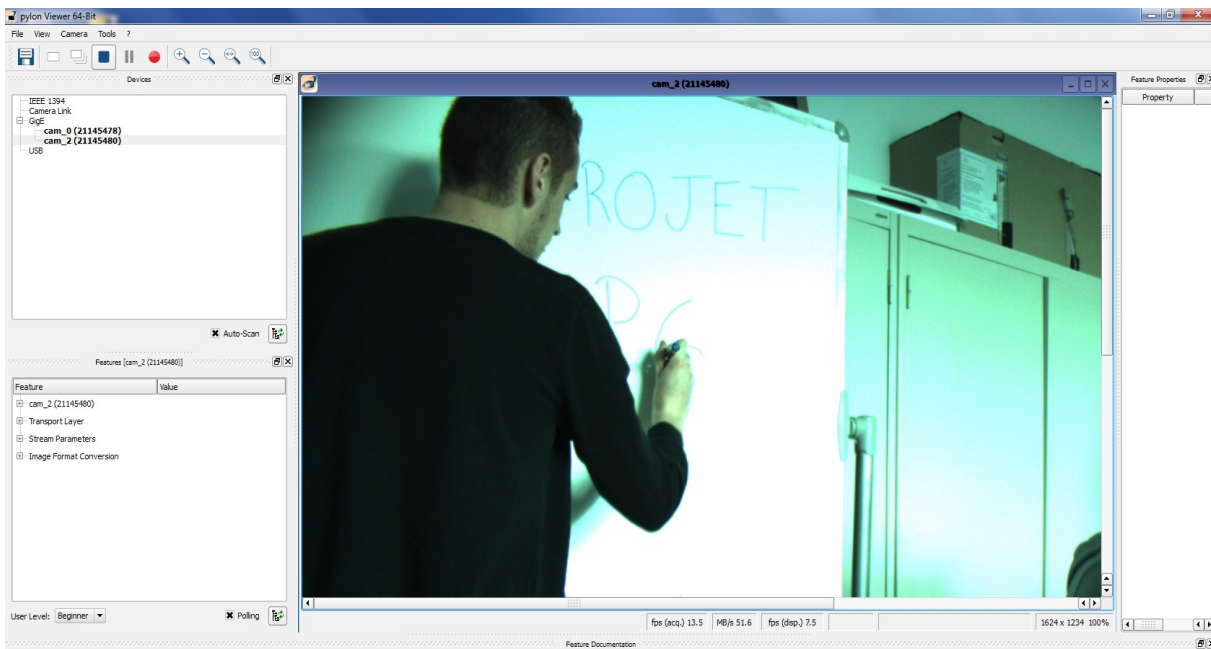


Illustration 4

Cependant, on se rend vite compte, en essayant de visualiser les images de chaque caméra, qu'une fois trois acquisition d'images lancées, l'ordinateur ne détecte plus le restant des caméras. Après avoir cherché quelques temps dans le logiciel, nous sommes arrivés à la conclusion que le problème vient, en réalité, de l'ordinateur lui même.

En effet, la carte réseau de notre ordinateur n'est pas assez puissante pour faire fonctionner plus de trois caméras en même temps, l'acquisition d'images simultanée ne fonctionne donc que pour un réseau ne dépassant pas trois caméras.

3.3.3. *Acquisition d'images par un réseau de caméras asynchrones*

Bien que le logiciel *Pylon* soit utile pour utiliser les caméras avec des fonction basiques, nous souhaitons pouvoir créer un réseau de caméras tout en ayant le contrôle de chaque option, afin d'avoir plus de libertés dans les réglages et les traitements de l'information (comme le recalibrage par exemple).

Pour cela, nous allons donc avoir besoin d'un programme fait directement pour nos besoins. L'intérêt de ce code, entre autres, est la possibilité d'automatiser les acquisitions d'images, afin de pouvoir constituer une bibliothèque d'images qui pourra être réutilisée dans d'autres cas plus tard.

La réalisation de ce programme dépassant de loin nos capacités et nécessitant d'être écrit en C, on utilisera un programme déjà existant, écrit par un doctorant sur le sujet.

Nous essayons d'abord de comprendre un programme en C plus simple, permettant l'acquisition d'une simple image, avant de commencer à utiliser le programme qui nous intéresse.

Ce programme d'acquisition d'image simple est un exemple mis à disposition par *Basler* : *SimpleGrab*. Nous essaierons de comprendre son fonctionnement global au mieux possible pour pouvoir peut être, si nécessaire, en modifier des petites parties pour notre utilisation. (voir le programme en annexe)

Si l'on résume de manière un peu simplifiée ce que fait le code, on obtient les étapes suivantes lorsqu'on lance l'acquisition d'image simple :

On initialise les SDK de pylon, puis on cherche à détecter les caméras présentes sur le réseau.

Le programme sélectionne la caméra et renvoie un message à l'écran pour informer l'utilisateur.

Ensuite, il réserve dans la mémoire un emplacement pour y mettre l'image qu'il va acquérir.

Puis, l'algorithme essaie de tester l'acquisition en prenant une image qu'il ne garde pas (juste pour tester la taille de l'emplacement mémoire).

Lorsque le programme est sûr d'avoir un emplacement réservé pour son acquisition, il procède à un enregistrement d'une image dans le flux de mémoire.

Ensuite, il vérifie si l'image a bien été prise.

Pour enregistrer l'image de la mémoire (RAM) sur le disque, il faut passer par une bibliothèque de traitement d'images (comme Opencv par exemple). On enregistre alors l'image au format Mat en précisant son extension.

Enfin, le programme libère la mémoire et libère sa caméra (met fin à tous les processus).

3.4. Calibration

Le traitement d'images repose sur l'usage des informations visuelles issues des capteurs. Les algorithmes avancés nécessitent les paramètres intrinsèques et extrinsèques des caméras. Lors d'une acquisition d'images, que ce soit avec une ou plusieurs caméras, il faut récupérer ces paramètres pour les utiliser pour les traitements. Pour ce faire, une calibration géométrique permet de faire correspondre un point dans l'espace avec le point associé dans l'image afin de trouver la relation entre les coordonnées spatiales d'un point de l'espace avec le point associé dans l'image prise par la caméra,

En pratique, la calibration sert à définir les paramètres intrinsèques et extrinsèques du capteur.

3.4.1. Paramètres intrinsèques d'une caméra

Les paramètres intrinsèques définissent le capteur en lui-même, ses caractéristiques propres. Ces paramètres sont répertoriés dans une matrice 3x3 sur le modèle suivant :

$$K = \begin{pmatrix} f_x & s_{uv} & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

avec f_x, f_y : focale de la caméra suivant respectivement \vec{x} et \vec{y}

u_0, v_0 : coordonnées du point principal de l'image

$s_{uv} \approx 0$ ce terme caractérise le pixel, si le pixel n'est pas rectangulaire $s_{uv} \neq 0$, mais aujourd'hui grâce aux progrès des caméras ce terme peut être considéré comme nul

Le point principal de l'image est souvent un point qui se trouve à quelques pixels près au milieu de l'image.

3.4.2. Paramètres extrinsèques d'une caméra

Les paramètres extrinsèques définissent la position du capteur dans l'espace. Quand on travaille avec plusieurs caméras, ces paramètres prennent en compte la rotation (orientation de la caméra) et la translation (distance) entre les différentes caméras. Grâce à une mire, on peut trouver ces paramètres et déterminer la matrice de pose (que nous appellerons ici T). La matrice T définit la transformation entre le repère des différentes caméra : la rotation et la translation.

Exemple pour un réseau de deux caméras :

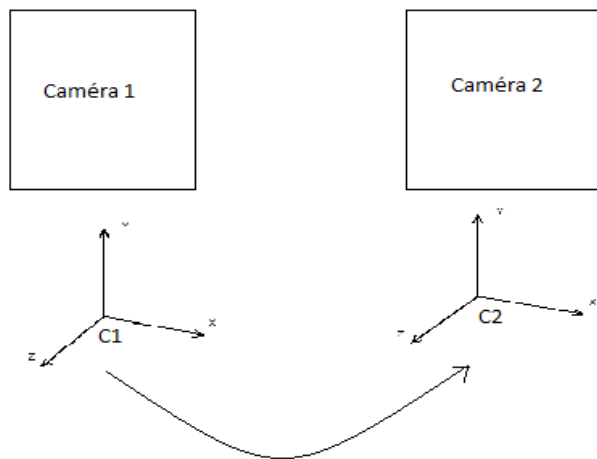
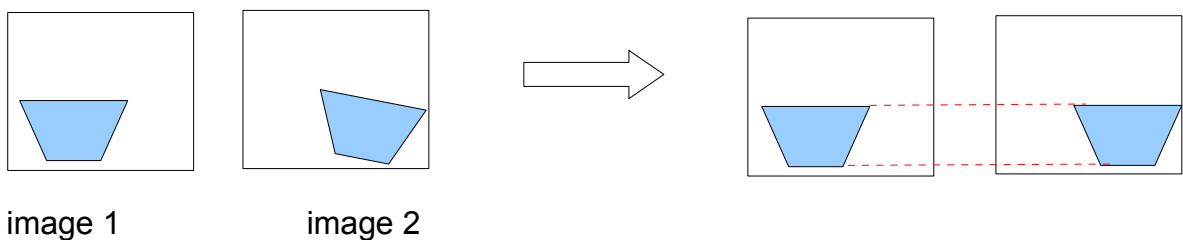


Illustration 5

On passe du repère C_1 au repère C_2 grâce à la matrice : $T = {}^{C_1}[R|t]_{C_2} = \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} & \lambda t_x \\ R_{yx} & R_{yy} & R_{yz} & \lambda t_y \\ R_{zx} & R_{zy} & R_{zz} & \lambda t_z \end{bmatrix}$ (2)

avec λ : distance entre les deux caméras (en mètres)

Avec cette matrice transformation, on peut modifier les images prises par les différentes caméras pour les faire correspondre les unes par rapport aux autres. Par exemple pour la rotation :



On prend une image de chaque caméra. On peut voir que l'objet dans l'image 2 est penché et décalé par rapport à l'objet dans l'image 1.

En récupérant les paramètres de calibrations (les paramètres intrinsèques k , les coefficients de distorsion et les paramètres extrinsèques), nous pourrions aligner chaque pixel de l'objet de l'image 2 par rapport à l'image 1. Cette étape s'appelle la rectification .

Grâce à ce processus, on obtient plus de précision pour les traitements par algorithme suivant (surtout pour utiliser les propriétés de la géométrie épipolaire) .

3.4.3. Le fonctionnement théorique en quelques mots...

Lors de l'acquisition, plusieurs transformations pour passer de la 3D à la 2D puis au format du capteur sont nécessaires.

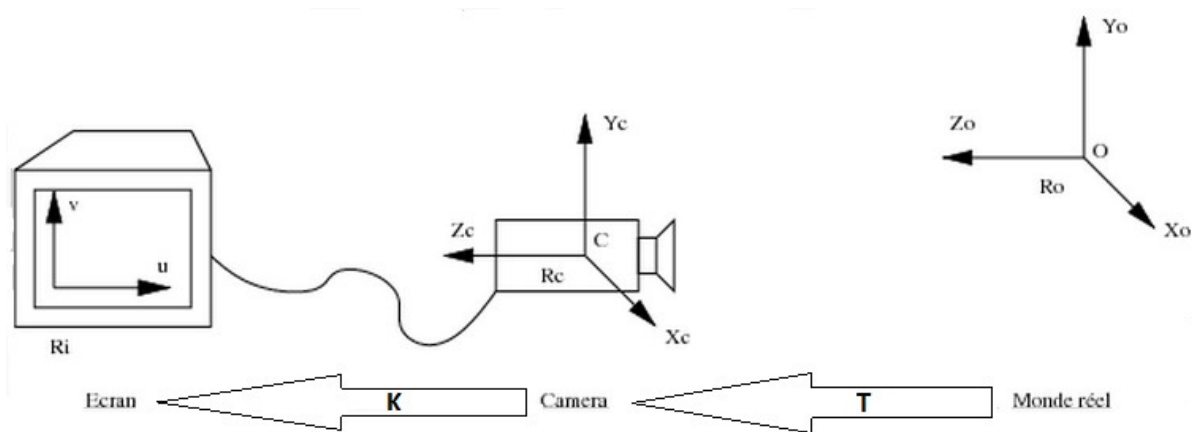


Illustration 6

La transformation T permet de passer d'une représentation 3D à une description en 2D :
$$\begin{pmatrix} x \\ y \end{pmatrix} = T \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Les coefficients de distorsion, grâce à la calibration, permettent de régler les problèmes de distorsion.

Les lignes courbées dans l'image sont corrigées pour retrouver les vraies coordonnées du point distordu :

$$x_{\text{corrigé}} = x_{\text{image distordue}} (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3)$$

$$y_{\text{corrigé}} = y_{\text{image distordue}} (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (4)$$

avec k_1, k_2, k_3 : paramètre de distorsion radiale

$$x_{\text{corrigé}} = x + (2p_1 xy + 2p_2 (r^2 + 2x^2)) \quad (5)$$

$$y_{\text{corrigé}} = y + (p_1 (r^2 + 2y^2) + 2p_2 xy) \quad (6)$$

avec (p_1, p_2) : paramètre de distorsion tangentielle

$(k_1, k_2, p_1, p_2, k_3)$: sont les coefficients de distorsion

Au final, pour passer du point 3D à la projection dans l'image, la transformation peut s'écrire sous cette forme :

$${}^i X_{2D} = K \cdot {}^c [R|t]_m \cdot {}^m X_{3D} \quad (7)$$

avec i : image

m : monde

c : caméra

R : matrice rotation

t : vecteur translation du capteur

De plus on peut aussi appeler $P = K \cdot {}^c [R|t]_m$: matrice de projection.

3.4.4. La calibration, en pratique

En pratique pour calibrer une ou plusieurs caméras, il faut utiliser une mire.

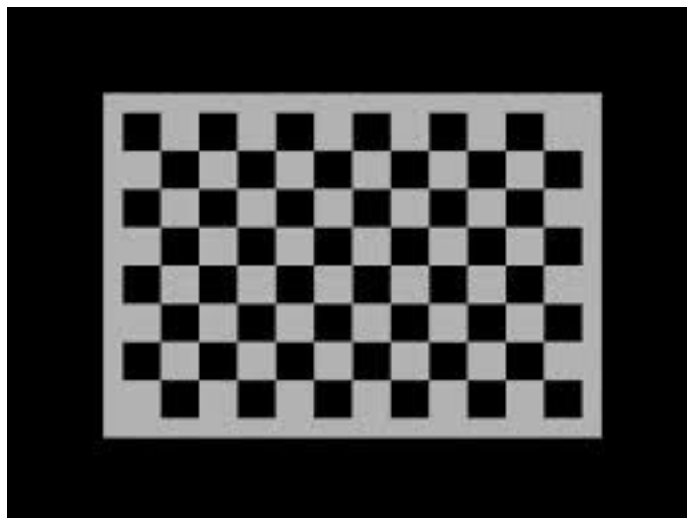


Illustration 7: Mire

Pour calibrer une seule caméra, on commence par définir la mire (l'objet) en donnant ses caractéristiques. Par exemple si c'est un damier il faut donner le nombre de coins des carrés en hauteur et en longueur. Il existe des algorithmes capables de retrouver ces points. Il faut pour cela faire plusieurs positions différentes. A chaque acquisition, on change la position, l'orientation d'angle, et la distance entre la mire et le capteur. En règle générale, dix acquisitions sont suffisantes. Les images obtenues permettront de définir les paramètres intrinsèques et extrinsèques de la caméra.

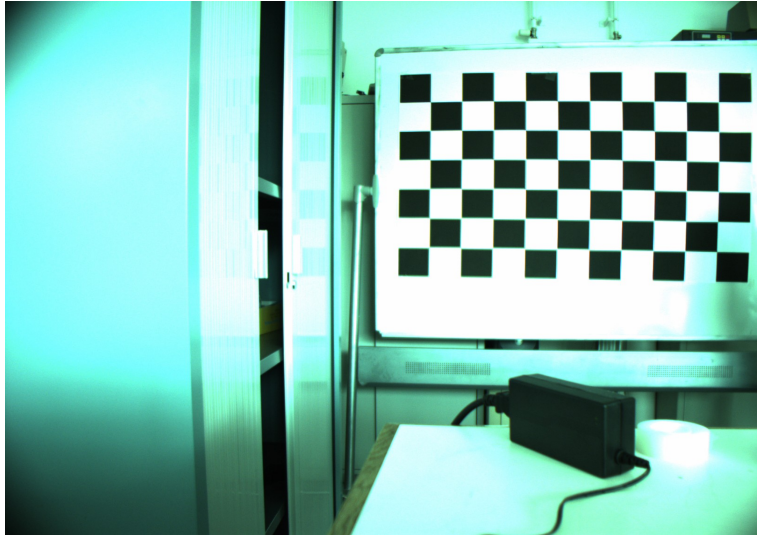
Remarque : En utilisant la mire il faut être vigilant au fait qu'elle doit être parfaitement plane pour ne pas fausser les calculs des paramètres et donc fausser la calibration. Le mieux est donc de coller la mire sur une surface rigide et complètement plane.

Dans le cas où le système comporte plusieurs caméras, il suffit de prendre une mire assez grande pour qu'elle soit vue par toutes les caméras et permette de définir les position relatives entre les caméras.

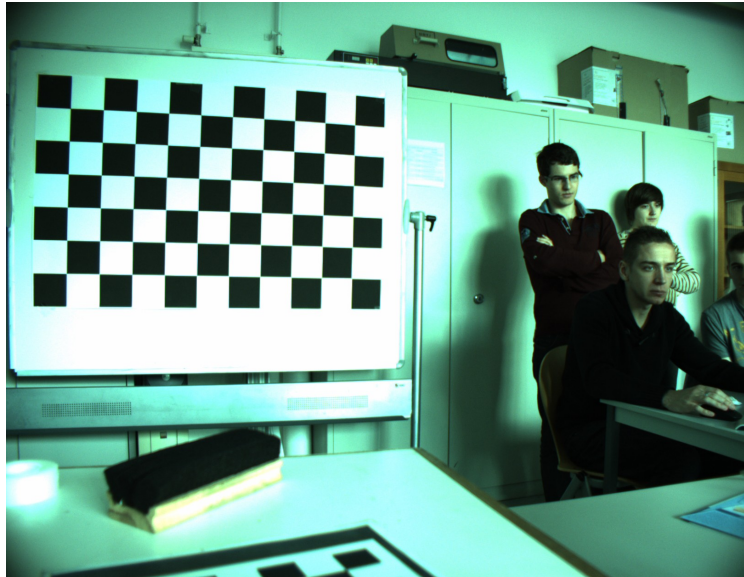
L'objet d'une de nos séances a été de calibrer deux caméras. Pour cela nous avons utilisé une mire assez grande pour être vue par les deux caméras. Nous l'avons fixé à un support mobile et plan pour avoir les meilleurs résultats possible. Nous avons ensuite fait les acquisitions.

Pour cela nous avons pris dix images pour chaque caméra. Nous positionnions la mire pour qu'elle soit vue des deux caméras en même temps puis nous faisons une acquisitions par caméra. Puis nous changions la mire de place et d'orientation et reprenions une image par caméra. En résumé pour chaque position de la mire, chaque « scène », nous avons une image prise par la caméra de droite et une image prise par la caméra de gauche.

Exemple d'images prises :



*Illustration 8
Image gauche*



*Illustration 9
Image droite*

Ensuite nous avons traité ces images grâce à un programme. Ce programme a pour but de faire une mise en correspondance des points entre l'image de gauche et l'image de droite, en lui donnant les points caractéristiques de la mire (nombre de carré en hauteur et nombre de carré en largeur).

Dans un premier temps le programme va analyser les images et retrouver les points caractéristiques de la mire (le programme « reconnaît » les coins des différents carré). Après cette analyse le programme enregistre de nouvelles images où les points reconnus sont identifiés.



Illustration 10

Ensuite le programme va prendre l'image gauche et l'image droite d'une même scène après reconnaissance des points caractéristiques. Les points caractéristiques sur les deux images étant les même le programme peut les mettre en correspondance, c'est à dire qu'il va créer une image où les points correspondants l'un à l'autre sont alignés.

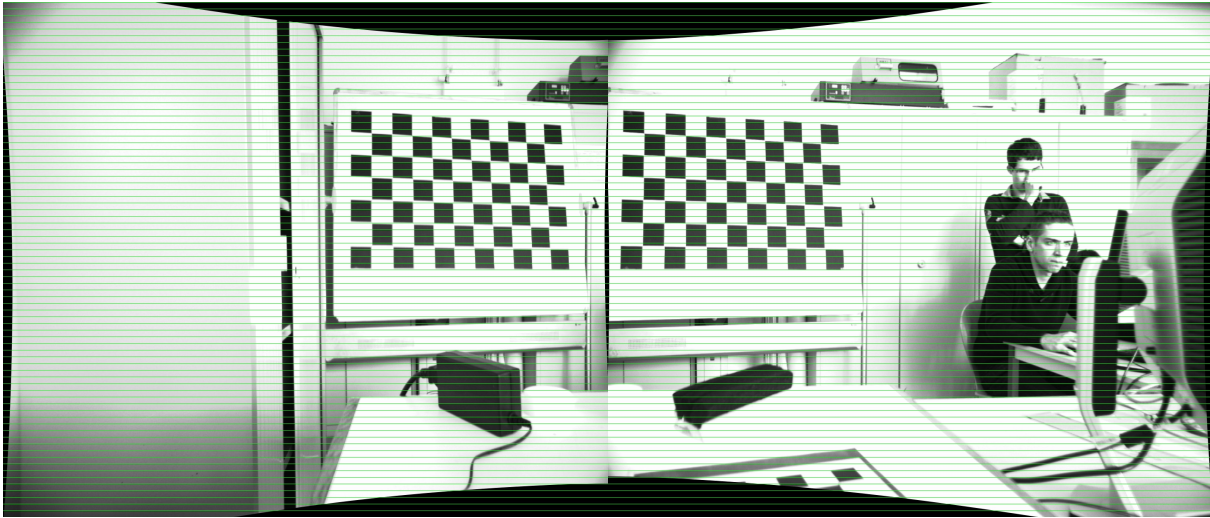


Illustration 11

Cette image finale permet par la suite d'avoir une reconstitution en trois dimensions de la scène.

4. CONCLUSIONS ET PERSPECTIVES

4.1. Conclusions sur le travail réalisé

Finalement, ce projet nous a permis de réaliser un réseau de caméra synchrones et d'en comprendre les bases. Cependant, nous n'avons pas réellement vu ou manipulé de réseau asynchrone, dû au fait que ce soit le sujet de thèse de notre enseignante, donc pas encore concret ni facile à enseigner à des STPI 2. L'un des principaux apports que nous en avons retiré a été de se rendre compte de la difficulté de mise en application des réseaux de caméras. Concrètement, nous retiendrons la manière de calibrer des caméras en théorie et en pratique. Nous avons acquis les bases sur les différents types de caméras et la manière de faire une acquisition d'image, et sur le lien entre les caméras et l'ordinateur assez difficile à gérer, finalement, à l'heure du "plug and play".

Les présentations auxquelles nous avons assisté nous ont ouvert à un secteur entier regroupant l'intelligence artificielle, l'automobile et l'optique. Ces présentations ont mis en valeur les compétences et thèmes gravitant autour de ces axes, avec les véhicules autonomes, leurs moyens de perception optique et le cinéma 3D.

4.2. Conclusions sur l'apport personnel de cet E.C. projet

Apport de cet E.C pour Nicolas

Ce projet m'a permis de découvrir certaines des nombreuses applications de la reconstruction 3D comme la reconnaissance d'obstacles, l'aide à la conduite ou même les voitures autonomes. Il m'a aussi apporté une nouvelle expérience de travail en équipe.

Apport de cet E.C pour Marthe et Marie-Solène :

Ce projet nous a permis de découvrir le laboratoire du LITIS ainsi que les multiples applications des différents domaines que peuvent avoir les caméras et le traitement d'image (la domaine automobile, le cinéma, la modélisation...).

La partie pratique, à savoir, la mise en place d'un réseau de caméra, nous a fait constater les nombreux problèmes que l'on peut rencontrer lors de la création d'un réseau mais aussi nous a poussé à travailler notre esprit de déduction et à chercher plus loin. De plus, le travail sur les algorithmes de calibration et le code d'acquisition m'a confortée dans mon choix de partir en ASI. (Marthe)

Enfin, le travail de groupe a aussi été une expérience enrichissante dans le sens où l'on a pu rencontrer de nouvelles personnes, voulant partir dans des départements différents, mais aussi appris à nous organiser au sein d'un groupe et la mise en place d'un projet.

Apport de cet E.C pour Raphaël

L'un des grands intérêts de ce projet est le fait qu'il soit pluridisciplinaire : il croise les domaines de l'optique avec la stéréovision et les réseaux de caméras, de l'intelligence artificielle et de l'automobile. Nous avons pu avoir une bonne idée de ce qu'était le travail de chercheur, car nous avons suivis de nombreuses présentations et avons été au contact de personnes ayant des parcours très variés. De plus, j'avais déjà entendu parlé de la Google Car, par exemple, et le véhicule autonome est une idée qui m'avait fait rêver. Voir les applications de ce domaine, les recherches entreprises multiples et très diverses, débouchant souvent sur d'autres applications, et la complexité du sujet a été très instructif pour appréhender la manière dont un système aussi important qu'un véhicule autonome est mis au point.

Apport de cet E.C pour Fabien

Après avoir effectué un stage de découverte au Laboratoire d'Analyse et d'Architecture des Systèmes (CNRS) de Toulouse, ce projet s'inscrit dans une certaine suite logique de mes découvertes. En effet, après avoir pressenti en 3ème certains aspects de la robotique, celui-ci m'a permis d'aller plus en profondeur dans le monde de l'intelligence artificielle, de la perception et reconstitution d'un environnement en trois dimensions, mais aussi et surtout du déplacement autonome, de par l'exemple du véhicule.

Véritable expérience enrichissante sur le plan du travail en équipe et sur l'organisation, ce projet a aussi su bénéficier d'un intérêt certain dû à la découverte d'un laboratoire de recherche tel que le LITIS, son fonctionnement et ses activités.

Pour finir, les multiples présentations auxquelles nous avons eu la chance d'assister se sont avérées être très intéressantes bien qu'un peu en parallèle de notre projet, car elles nous ont permis de mettre du concret là où nos idées restaient parfois un peu floues...

4.3. Perspectives pour la poursuite de ce projet

Dans l'hypothèse où nous pourrions poursuivre ce projet, la suite logique serait de faire des acquisitions avec les caméras sur un véhicule, et de tenter d'obtenir une banque d'image propre et de situations variées.

Il serait ensuite intéressant de creuser sur le sujet du traitement informatique de l'image, essayer de comprendre et décrypter les algorithmes utilisés lors de ce projet, pour pouvoir ensuite les modifier suivant nos besoins et être capable de créer nos propres programmes. Cela nous permettrait de mettre en application la stéréovision et la perception de l'environnement que nous avons découvert lors de cet EC. Nous pensons que ce serait la partie la plus intéressante, puisqu'au fond c'est la partie utile de notre projet, la banque d'image n'étant là que pour tester les algorithmes.

Un objectif qui serait génial à réaliser serait de pouvoir faire avancer un véhicule autonome sur un parcours très simple. Cela nous permettrait de voir d'autres types de capteurs que les caméras, comme par exemple les capteurs de distance infrarouge, et de voir comment ils pourraient être complémentaires avec nos caméras.

5. BIBLIOGRAPHIE

* Liens utiles :

<http://www.photovore.fr/news-differences-capteur-ccd-cmos-mos.html>

<http://www.commentcamarche.net/contents/2116-le-capteur>

<http://www.baslerweb.com/products/ace.html?model=169>

<http://futic.vision.cs.chubu.ac.jp/04/pdf/PSV02.pdf>

<http://www.vision.cs.chubu.ac.jp/04/pdf/PSV04.pdf>

<http://www.darpagrandchallenge.com/>

http://www.ted.com/talks/sebastian_thrun_google_s_driverless_car

http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.145.7481&rep=rep1&type=pdf>

http://en.wikipedia.org/wiki/Gigabit_Ethernet

http://fr.wikipedia.org/wiki/Calibration_de_cam%C3%A9ra

* Lien du code (en annexe)

<https://docs.google.com/file/d/0B7cXhKI3XOcudVNCOWZoSHdHRTg/edit>

6. ANNEXES