

Rapport de bureau d'études

Traitement et analyse d'image sous Raspberry pi

Bouyjou Gauthier

27 Mai 2018

Tuteurs : Alain Crouzil, Jean-Baptiste Puel

Etablissement : Université Paul-Sabatier, 31000 Toulouse

Formation : L3 informatique

Présentation du sujet

Présentation

Le sujet de ce bureau d'étude consistait à se familiariser avec le développement sous Raspberry Pi (micro carte informatique), que ce soit au niveau installation du système d'exploitation Raspbian qui est un Debian développer exclusivement pour marcher sur des machines ARM comme le Raspberry Pi et permet d'utiliser les GPIO.

Puis d'installer des librairies complémentaires d'analyse d'image tel que Opencv.

Travail à réaliser

- Installation du raspberry pi avec un environnement dédié au traitement d'image, par exemple l'installation d'Opencv depuis les sources qui peut prendre un certain temps à compiler, la connexion des caméras (modules à activer).
- Réalisation d'applications d'analyse d'image sur le raspberry pi.
- Création de guide d'apprentissage (wiki) permettant de présenter les réalisations.

Cadre de travail

Le bureau d'étude se déroulant en salle 1R1-014 (sous sol du bâtiment des mathématiques), salle de tp/td avec environ 20 poste de travail, et un tableau.

Équipe de 7 étudiants donc effectif raisonable.

On a pour ce BE à notre disposition environ 7 raspberry pi (3 rpi3, 4 rpi2), de nombreuses cartes SD car seules ces cartes peuvent servir de mémoires de masse pour le rpi.

Nous avons à notre disposition 4 caméras qui peuvent se brancher directement sur le rpi, dont 2 caméras nocturnes (sans filtre IR).

Déroulement du stage/BE

Installation du réseau (première semaine)

En premier lieu étant dans une salle de tp on a travaillé avec les machines de la salle, téléchargé l'ISO de Raspbian et formaté les cartes SD pour faire marcher les rpi.

En connectant les rpi aux réseaux de la salle de tp on a remarqué que leurs IP n'étaient pas attribuées automatiquement (DHCP), après l'obtention d'une nouvelle IP toute neuve attribuée à la main, on n'avait aucun accès internet, et oui le parefeu au-dessus acceptait seulement les machines (adresses MAC) qu'il connaît déjà.

Du coup on a décidé avec l'équipe que se serait plus simple de créer notre propre réseau à nous.

Le choix à été de faire un hotspot wifi avec un des raspberry déjà à notre disposition, ceci permettrait d'utiliser SSH au sein de ce sous-réseau, car en effet le réseau WIFI EDUORAM ou UPS n'accepte pas les échanges SSH entre machines.

Le problème persiste quand même, on n'a toujours pas accès à Internet dû au pare-feu, la solution a été de prendre une adresse MAC d'une des machines de la salle de tp déjà présente et de modifier celle de notre hotspot afin d'avoir un réseau complet, sachant que le service DHCP est proposé par notre hotspot.

Après tout cela nous étions prêts à travailler efficacement car le matin nous n'avions qu'à brancher les rpi hotspot en premier et tout se connectait automatiquement via le protocole WPA_SUPPLICANT après quelques configurations.

Toutefois l'installation du réseau a pris une bonne semaine, et quelques maintenances les semaines suivantes.

Apprentissage de la librairie opencv (sous python personnellement) (deuxième semaine)

<https://opencv-python-tutroals.readthedocs.io/en/latest/index.html>

J'ai réalisé ce tutoriel entièrement afin d'être un minimum à l'aise avant de commencer à faire des choses un peu inédites, ou de combiner plusieurs fonctionnalités différentes à la fois.

Travail réalisé

Scanner papier

Utilisation de la camera du raspberry pour prendre une photo de haute qualité (5 Megapixels, 2592x1944), afin de pouvoir récupérer idéalement chaque caractère.

Utilisation d'une librairie additionnelle à opencv, Tesseract OCR (Optical Character Recognition) pour permettre de récupérer du texte.

Cette librairie est très efficace pour reconnaître des caractères dans des fichiers PDF, car le texte est bien aligné, et la police facilement reconnue.

Le problème se pose maintenant, on doit appliquer des opérations de traitement d'image afin de pouvoir faire une ocr sur une image propre et alignée.

Le travail a consisté premièrement à récupérer la zone d'écriture sur une photo, par exemple le blanc d'une feuille A4 blanche, pour cela j'ai utilisé l'histogramme vu en cours de traitement d'image, en trouvant la couleur la plus présente sur la photo on pourrait se dire que ca serait la page blanche.

Après avoir trouvé les quatres coins de la feuille, une rotation et une déformation est nécessaire pour aligner la feuille.

Puis viennent des techniques de seuillage telles que la binarisation avec Otsu pour se retrouver avec une feuille noir et blanche de meilleur qualité pré-ocr.

Enfin l'utilisation de Tesseract sur une image beautifiée.

50 % des mots ont été reconnus de cette façon, il faut dire que la prise de la photo n'était pas idéale, sans flash, sans lumière blanche.

Panorama

Réalisation de panorama avec prise de photo à l'aide d'un support pour raspberry pi composé de 2 servos moteurs permettant une vue 360 degrés.

Le but a été de voir comment un panorama peut se faire, en reconnaissant des points caractéristiques à deux images en créant des liens, c'est comme cela que l'on peut fusionner deux images.

J'ai donc proposé un code python permettant de lier deux images.

Puis vient l'utilisation des 2 servos moteurs grâce aux GPIO du raspberry pi, ces actionneurs fonctionnent par le principe de rapport cyclique PWM (Pulse Width Modulation) à une fréquence de rafraîchissement fixée au préalable.

L'utilisation d'alimentation externe car qui dit moteurs, dit forte consommation, ce qui pourrait endommager le rpi.

Pour finir on peut créer plusieurs types de panorama :

- Cylindrique
- Sphérique
- Stéréographique
- et plein d'autres

Transmission de donnée par led (en équipe avec DURAND Sylvain)

Transmission par led ou laser de message (octet) par une communication synchrone simplex entre 2 raspberry pi, l'émetteur grâce à l'utilisation des pins du GPIO fait clignoter une source de lumière sur une horloge qu'on considère synchronisée avec le récepteur car il s'agit de deux rpi.

L'émetteur au préalable demande une chaîne de caractères à émettre à l'opérateur, l'envoi se fait par le principe de décodage de chaque caractère (octet) en 8 bits, et envoie sur une fréquence fixée égale sur les 2 raspberry pi des 0 ou des 1 selon le code du caractère, et se fait ainsi pour tout le message, envoie le code 0x00 0b00000000 pour la fin du message (également la fin du string en C).

La réception requiert une initialisation pour permettre d'identifier le pixel qui correspond à celle de la source lumineuse dans l'image, ce qui implique que la caméra du récepteur doit être fixe.

La caméra du rpi permet de prendre 90 fps avec une résolution de 640x480 ce qui permet en théorie une transmission à une vitesse de 11.25 octets par seconde.

Malheureusement à cette fréquence la source lumineuse (partie électrique/composant diode) n'a pas le temps de s'éteindre, donc on ne passe pas pour une composante d'un pixel de 0 à 255 mais plutôt 200 à 255, ce qui génère des erreurs lors de la transmission déjà.

Suivi de personne comme pourrait le faire un vrai cameraman

Projet qui pourrait être utile pour enregistrer nos propres scènes de cinéma sans cameraman.

Utilisation de fonction de la librairie opencv permettant de reconnaître des parties du corps se basant sur l'apprentissage profond, ou deep learning.

La librairie propose déjà des fonctions pré entraînée pour reconnaître toutes parties du corps.

J'ai donc utilisé la reconnaissance faciale.

Cette solution permet de travailler sur une image mobile contrairement à une autre solution de detection de mouvement qui utilise le fait que le background ne bouge pas.

Utilisation des servos moteurs pour tracker le visage, nécessite quelques opérations géométriques, et un peu d'optique car selon le FOV (Field Of View) de la caméra et la résolution on se déplace d'un certain angle.

Création de tutoriel (wiki)

Chaque projet a été présenté sur un wiki, permettant à des personnes d'utiliser ce qu'on a pu apporter.

Edition d'un fichier pour le wiki avec un langage de balisage ressemblant à ce qu'on peut voir sur Github md (Markdown).

Création de screencast vidéo sur youtube.

Création de fichier spécifique pour les branchements électriques lors d'usage des GPIO avec Fritzing.

Bilan

Résultat

Les résultats sont plutôt satisfaisants dans l'ensemble, quelques points pourraient être améliorés niveau performance.

Difficultés

Quelques difficultés pour l'utilisation de python car novice, utilisation des listes indexées très perturbantes.

Optimisation de code cpp avec des méthodes inline par exemple, mais ce manque d'expérience en cpp va être corrigé l'année prochaine en Master 1.

Bilan personnel

Ce bureau d'étude m'a permis de me rendre compte qu'avec un micro pc tel que le raspberry pi, on peut créer une infinité de projets intéressants, que les librairies libres comme opencv peuvent permettre de grandes choses.

La réalisation de ce BE m'a conforter dans le choix de Master voulant faire (IGAI).

Les choses que l'on peut faire en traitement d'image sont très puissantes et très efficaces quand on peut voir le nombre d'opérations à la suite que l'on peut faire sur un flux vidéo temps réels avec un framerate assez raisonnable, on se demande quelles ont été les optimisations pour arriver à un tel niveau de performance.

En informatique graphique on utilisera sans doute la librairie opengl et je pense qu'avec des librairies comme celle-la (opencv + opengl) on peut arriver à créer des applications graphiques vraiment utiles.

Lien avec formation et poursuite d'études

Mon premier vœu étant le Master IGAI, ce choix de BE m'a confirmé ma conviction première.

L'utilisation de librairie, que ce soit dans le traitement d'image, ou en informatique graphique, me permettra de réaliser de grandes applications.

Comme je l'ai dit plus haut, lors de l'utilisation de opencv pour faire des operations de seuillage, binarisation, segmentation, on fait appel au cour de traitement d'image que l'on a suivi ce semestre.