

[Eyes Motion]

Le présent document, décrit les besoins, exigences, et contraintes du projet. Il sert donc de feuille arienne qu'il faudra suivre et respecter lors de la réalisation du projet. Il résume les tâches réalisables par le système à concevoir, ainsi que ses limites. Le document est composé de plusieurs sections et parties devant intéresser d'une part l'équipe marketing et administrative et d'autre part l'équipe en charge de la conception interne du produit. Nous indiquons donc qu'il existe dans ce document des parties plus techniques ne pouvant pas intéresser tout le public en charge de ce projet. En effet, le présent document sert de planning pour la réalisation de chaque étape du projet. Décrit les différentes façons d'utiliser le produit. et de sa conception.

Ce document possède une partie très technique sur la spécification du produit, décrivant les technologies et techniques utilisés. Voilà pourquoi certaines parties sont susceptibles d'échapper à un public non averti.

@ Auteurs

Ndjanda Mbiada Jacques Charles

@ Responsables

Jean-Baptiste Yunès
Zacchioli Stefano
Yann Régis Gianas

introduction

Dans le but de l'UE Projet long de l'université Paris Diderot Paris 7, nous sommes amené à réaliser un projet de notre choix.

Sont mis à disposition des machines sous différents plateformes (NT, Linux, Unix) ainsi que des robots Nao, Lego® Mindstorms et des micro-ordinateurs Raspberry Pi. Il nous est demandé de créer un projet en rapport ou non avec ces robots et micro-machines. De se rapprocher un maximum de l'architecture d'entreprise quant à la teneur et réalisation du projet.

Le projet de créer un ordinateur robuste, mobile, multi fonction, futuriste et pas chère nous ayant séduit ; nous nous sommes donc lancé sur le projet [Eyes Motion] dont la présentation et description est faite ci-dessous.

[Eyes Motion] présentation

[Eyes Motion] le nom attribué à ce projet et produit fait allusion à son fonctionnement du point de vu externe et interne. ce nom décrit aussi bien le cas d'utilisation du produit que son fonctionnement. [Eyes Motion] est un logiciel de capture vidéo et d'analyse pour extraire des informations des captures de la dite vidéo. Nous nous concentrerons dans un première temps à l'extraction des informations concernant les silhouettes humaines et de leur analyse.

C'est à dire reconnaissance de silhouette humaine et leur traçage par la suite. Ce produit permet donc de garder un oeil actif sur une source vidéo afin d'en analyser le contenu.

Orientation principale

Nous souhaitons nous axer vers une utilisation précise du produit finale. En effet malgré l'ouverture que nous devons permettre au système finale, nous axerons notre travail sur l'utilisation du produit en tant que système de vidéo surveillance et de détecteur de mouvement (l'idéal serait d'avancer jusqu'à la reconnaissance de mouvement ou d'entité). Nous souhaitons pouvoir positionner l'objet devant une porte ou fenêtre et permettre la surveillance ainsi que la détection de toute personnes s'approchant de l'habitat.

Le but étant de pouvoir comptabiliser par exemple le nombre de passant en distinguant les femmes des hommes, les enfants des vieux. Les objets tel que les automobiles, les animaux et d'autres type d'analyse ultérieurement. Dans un premier temps reconnaître un objet mobile et comptabiliser semble correcte. La reconnaissance exacte du type d'objet mobile est un objectif secondaire.

Schéma d'utilisation

Cas d'utilisation

Nous présentons ci-dessous 3 usages possible du système. La colonne au titre vert correspond au point sur lequel nous nous concentrerons dans un premier temps.

les différentes configurations

Serveur domestique	Ordinateur personnel	Assistant personnel
<ul style="list-style-type: none"> • Gestion de l'alimentation électrique • Toujours connecté sans interruption possible. • Installation du software adéquate (serveur web, mail, PABX, ftp, etc.) 	<ul style="list-style-type: none"> • Interface graphique nécessaire • Support de la souris, clavier et écran, nécessaire. Ainsi qu'une webcam. • Reconnaissance de réseau domestique, tv, imprimante etc. 	<ul style="list-style-type: none"> • Nécessite une batterie embarqué • Capteur vidéo/audio pour commande gestuelle et/ou vocale. Pour analyse de l'environnement.

Serveur domestique

Installer les logiciels de monitoring, Et services nécessaires pour en faire un serveur domestique. Configurer l'utilisation d'un espace e stockage distant (dans l'intranet ou extranet). L'usage en serveur multimédia domestique serait idéal. La raspberry pi n'étant pas une bête de course, la soumettre à des requêtes robustent et intempestive serait une mauvaise idée.

Ordinateur personnel

Installer un gestionnaire de fenêtre pour l'utilisation quotidien. Configurer l'utilisation d'un écran, imprimante, souris et clavier. Configurer un VNC serveur sur la raspberry pi et un VNC Client sur

une des machines existante et disposant des interfaces clavier - souris - écran. Ou pour l'utilisateur avancé, configurer l'accès ssh pour une commande en ligne de commande. Mais alors est ce encore une machine personnel ?

Assistant personnel

Nous souhaitons ici implémenter un système mobile capable d'utiliser les différents périphériques disponibles pour assister au mieux l'utilisateur. Ce système est construit autour du Raspberry Pi version 512 Mo de mémoire RAM et ayant 800MHz de puissance de calcul CPU. Aucune carte graphique n'est incorporé de base.

L'assistant personnel a besoin pour son fonctionnement minimal :

- Un capteur vidéo
- Un capteur audio
- Un transmetteur audio
- Un transmetteur vidéo

On notera ici que dans un premier temps, un des transmetteurs audio et vidéo devrait suffire. Et nous nous concentrerons sur l'acquisition d'un capteur video.

Pour ce faire, il est demandé d'implémenter dans le système :

1. Un module de reconnaissance de forme ;
2. Un module de tracking de forme ;
3. Un module de transmission video.

Etude fonctionnelle

Capture video

La capture des vidéos est prise en charge par la librairie OpenCV. Librairie largement répandu dans le domaine de la vision assisté par ordinateur. Nous avons développé le monitor EM pour permettre la monotorisation d'une ou plusieurs camera du programme.

Cette classe s'appel EMMonitor. Une fois Cette classe crée, elle se charge de pouvoir créer une ou plusieurs caméra auxquelles elle assigne un flux de capture et d'écriture (permettre la sauvegarde des vidéos dans un fichier). Une camera représenté par la classe Camera, ne sert que gestionnaire d'un flux de capture et d'enregistrement, en claire elle ne génère que les frames utiles. La classe en charge de leur analyse est CAMAnalyser. Un moniteur comporte une CAMAnalyser pour permettre de traiter les images des cameras.

Reconnaissance de forme.

La reconnaissance de forme est possible grâce au capteur vidéo intégré à [Eyes Motion]. Ce capteur permet de distinguer des formes par logiciel sur une distance de [10; 50] mètres. A noter que la reconnaissance des formes à moins de 10 mettre est possible, mais la configuration de base suppose des objets d'une certaines taille minimal et d'un certaines taille maximal. L'intégration du capteur au Raspberry Pi est une partie à prendre comme difficile dans le projet. Nous supposons que cette partie est faisable et donné, Il est indispensable d'être dans un premier temps capable de capturer et enregistrer la vidéo provenant du capteur vidéo pour pouvoir détecter les mouvements de ce flux.

L'algorithme utilisé fait appel à des opération matricielles de bases. Dont dispose la librairie utilisé (librairie exposé en Annexes).

Nous avons vu plus haut que CAMAnalyser permettait d'analyser les vidéos en provenance des caméras. Et c'est pourquoi la reconnaissance de formes lui est en partie assigné. En effet cette classe ne reconnait pas les formes à proprement dit mait les formes en mouvement. Si nous souhaitons vraiment la reconnaissance de forme (tel que la reconnaissance de visage ou d'une bouteille) nous déléguerions cette tâche à une classe utilisé par CAMAnalyser.

En réalité CAMAnalyser reconnait la zone où se trouve un objet en mouvement et passe au tracker ces éléments et c'est uniquement ce dernier qui les transforme en form de la classe Form.

Motion tracking

Après détection des forme l'étape important est leur traçage. Outre les afficher, l'on souhaite

tracer les objets ; notamment grâce à un identifiant et/ou un nom. La classe qui permet cela est TrackForms associé à la classe Form. TrackForms associé à chaque objet de zone une form avec cette zone et un identifiant.

Analyse de mouvement

Nous souhaitons ici présenter la technique utilisé pour la détection de mouvement dans une vidéo.

On rappelle que la difficulté est de pouvoir distinguer un objet en mouvement de l'arrière plan pseudo statique[1].

Première approche

Une capture vidéo est la prise de plusieurs photo dans une intervalle donné ; par exemple 25 images par seconde. Soit une image toute les $n = 60/25 = 2.4$ soixantième seconde.

L'approche la plus simple et conventionnelle pour détecter un mouvement serait de comparer deux à deux les dernières images capturées.

En effet en regardant la soustraction des deux images nous pourrions percevoir la différence entre ces deux frames.

Le problème qui se pose est la vitesse de mouvement de l'objet et de sa taille.

Sur 25 images, si nous décidons de prendre deux à deux les dernières captures pour comparer ; nous aurons très certainement du mal à détecter un mouvement. Sauf s'il est question d'un mouvement rapide ou d'un objet éloigné.

Une réponse possible à ce problème serait de réduire la vitesse de capture de la camera. ou autrement dit, de ne prendre une image que toute les n/k image par seconde avec $k > 1$. Si l'on souhaite avoir au moins une comparaison par seconde, on s'arrange alors à ce que $k < 13$ (au dela de 12 le nombre d'image pris ne contient plus une second).

Cette solution semble la plus à même de satisfaire rapidement une première version du produit. Afin de voir la possibilité de capture et de détection de mouvement d'[Eyes Motion].

[1] En effet l'environnement peut changer. Un objet de l'arrière plan peut se déplacer ou disparaître lors d'une longue capture. L'exemple d'une voiture garé en début de capture et s'en va à un temps t est la plus flagrante.

Seconde approche

Nous voulons à présent optimiser la technique de détection de mouvement d'un élément dans une capture vidéo. Pour ce faire nous introduisons une nouvelle technique toujours rapidement implémentable (nous verrons dans la partie qui suit, des techniques bien meilleures mais aussi bien plus difficile à comprendre et à implémenter).

Le détour d'élément dans une photo est une opération qui fait appel à des technique de filtrage successives.

Cette opération accompagné par des technique d'extraction par détermination de zone viable nous permet de vérifier une assertion donnée sur une image. (comme par exemple : Est ce là un objet en mouvement ?

Cette technique est pour le moment le plus simple et le plus efficace que nous puissions implémenter sans beaucoup de connaissance poussée en manipulation de matrices, la morphologie, géométries discrète, projective et épistolaire.

Reconnaissance d'objet

En utilisant une base de donnée de formes.

Architecture interne

Interface du Raspberry Pi

X11 GNOME - VNC avec tiny tight vnc server

En cas d'erreur avec Xlib et un manque d'extension XANDR, il est nécessaire d'installer *gtk2-engines-pixbuf* et ou à désactiver *xinerama*.

Logiciel du RaspberryPi

Une image

Les images sont générés par une application en ligne de commande qui permet de capturer la vidéo en provenance du module camera de la raspberry. Cette solution permet de palier aux éventuels problèmes de capture de openCV.

Le flux vidéo

Une forme

Un objet détecté

Un mouvement