Architecture et conception du logiciel

Projet de conception

1.0

Samuel Harvey

Jérôme Blackburn Saucier

Aicha Kaba

19-12-2022

Historique des révisions

Date	Version	Description	Auteur
19-02-2022	1	Finalisation	Jérôme

Définitions

Terme	Définition

Abréviations/acronymes

Abré./Acro.	Définition

Table des matières

1. Introduction

1.1. Objectifs

Le but du présent document est de fournir une description de la conception et de l'architecture logicielle en se basant sur les exigences fournies dans le document des spécifications du logiciel. Afin d'illustrer ces interactions, ce document contient la conception détaillée des éléments du logiciel ainsi que différentes vues représentant l'architecture logicielle.

1.2. Portée

[Indiquez si le produit est un système adjacent, un sous-système ou est la mise à jour d'un produit existant. Présentez les systèmes et les logiciels avec lesquels le produit interagit. Aussi, présentez ce dont il ne fait pas parti]

2. Parties prenantes de la conception et leurs préoccupations

Jérôme Blackburn Saucier

- Les délais ont rapidement été dépassés.
- Difficulté d'établir les filtres stables lorsque la caméra bouge
- Problème de lenteur sur un vieil ordinateur
- Prévision de temps non respecté

Aicha Kaba

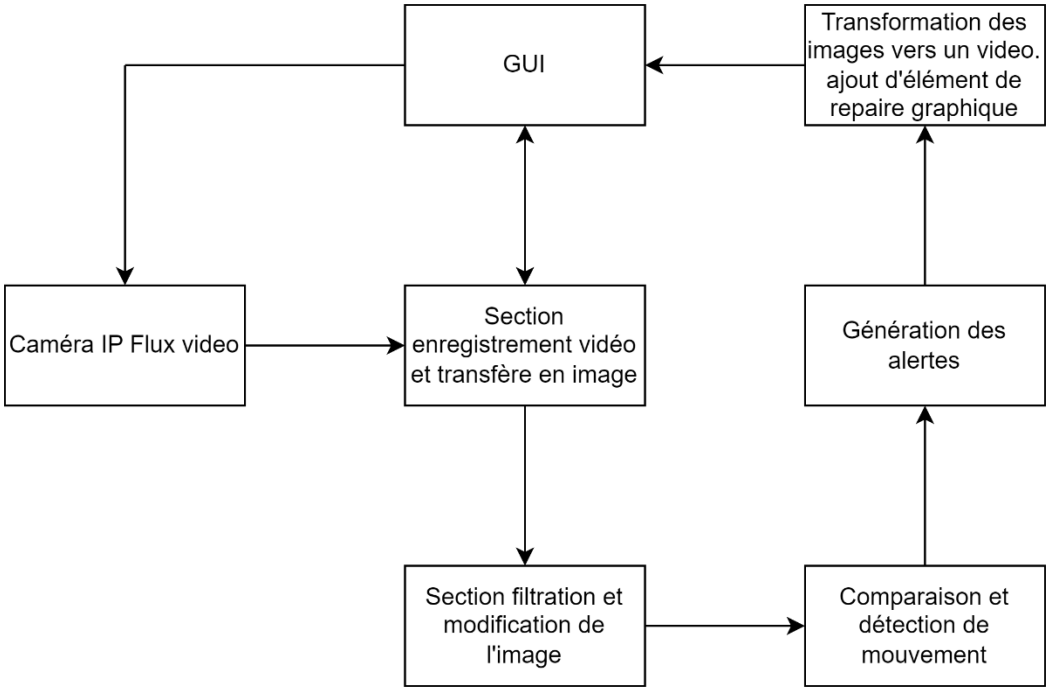
- Assurez la sécurité
- Répondre aux exigences

Samuel Harvey

- Prévision de temps non respecté
- Difficultés à interpréter quand déclencher les carrés de détection
- Peur d'oublier une ou des fonctionnalités du programme tel que décrit par le client

3. Architecture logicielle

3.1. Vue d'ensemble de l'architecture logicielle



3.1.1. Contraintes de conception qui s'applique à cette vue

L'ensemble des images sont traitées et envoyées d'une classe à une autre. La coordination des éléments devient importante puisqu'il n'y a pas de classe maitresse. L'utilisation de thread et de queue nous a permis de simplifier des éléments. Il est important de coordonner le tout sans quoi il y a décalage entre les images.

3.1.2. Exigences et préoccupations de conception

L'utilisation de l'interface graphique pour afficher la vidéo modifier on a eu de problème au niveau de la synchronisation des threads et des images. Il y avait du décalage. De plus on a rencontré des difficultés lors de l'utilisation des filtres. Un des principaux problèmes venait du fait que la caméra IP fournis avait une vibration constante. Cette vibration provoquait de fausses lectures de mouvement.

Une autre préoccupation vient du langage de programmation. Puisque nous en sommes à nos débuts sur python on doit souvent chercher et trouver des alternatives pour résoudre les petits casse-têtes

3.1.3. Description des éléments de la vue et Raisonnement.

Le GUI est offre d'entrer une URL de vidéo. En plus il affiche le vidéo modifié qui situe le mouvement détecté.

La section d'enregistrement permet de lire le vidéo provenant de l'adresse IP et d'ensuite la convertir en une série d'images.

Les images sont alors acheminées aux filtres qui les traites et les modifient. Le but étant ici d'avoir le moins possible de pixel a comparé et ainsi d'évité les fausses détections.

On envoie ensuite les images traitées vers un comparateur qui détecte les différences entre les images. Une fois détecté elles sont utilisées pour générer une nouvelle image qui contient les différences liées au mouvement.

Par la suite on génère une alerte si l'image de différence contient des pixels en état de mouvement.

L'image est alors ajoutée à l'image de base et est redirigée vers le GUI pour y être affichée.

4. Conception détaillée

4.1. Conception détaillée GUI

4.1.1. Vue structurelle

4.1.2. Vue comportementale

Il permet d'appliquer un choix de filtre, d'insérer une adresse et de lire la vidéo située à l'adresse.

4.1.3. Autres vues pertinentes

4.1.4. Raisonnement

Dans le but d'aider l'utilisateur, l'utilisation d'une interface graphique était la solution la plus simple et efficace pour parvenir a nos fins.

4.2. Conception détaillée du lecteur

4.2.2. Vue comportementale

Le lecteur est une classe qui sert à recevoir la vidéo. Une fois reçu elle est divisée en image elles sont alors envoyées à nos filtres pour qu'elle soit traitée. On utilise la librairie Open.cv.

4.2.4. Raisonnement

L'utilisation d'un moteur de lecteur et de conversion facile d'utilisation était un avantage.

4.3. Conception détaillée du filtreur

4.3.1. Vue structurelle

4.3.2. Vue comportementale

Les filtres traitent les images. On utilise principalement la librairie Pillow pour ce faire. On applique les modifications suivantes :

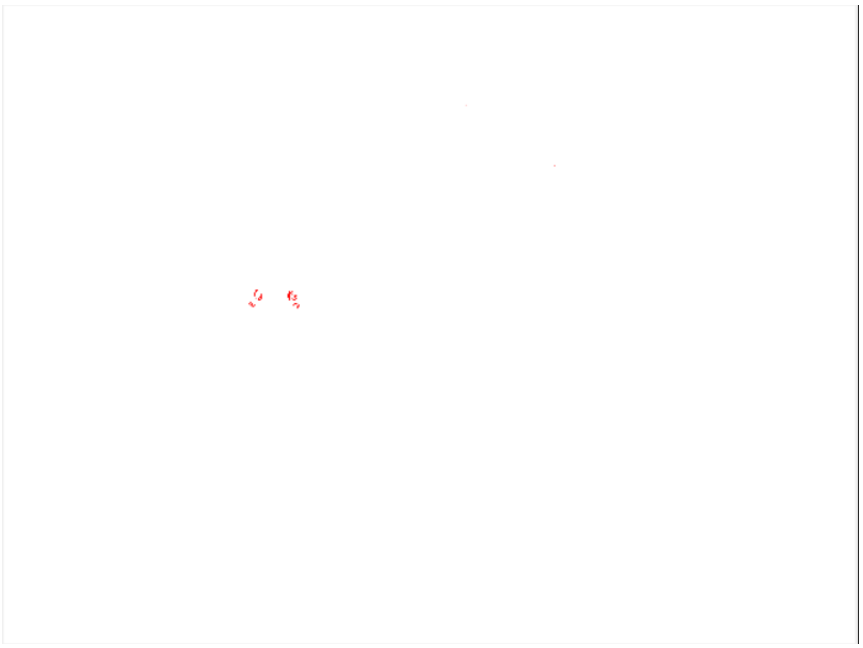
1. Convertir en format Pil
2. Convertir en format Greyscale
3. On applique un filtre qui permet d'augmenter la Sharpness
4. On applique un filtre qui fait un GaussianBlur
5. On applique un filtre qui change le median
6. On applique un filtre qui permet de trouver les Edges
7. On convertit les tons de gris en noir ou blanc
8. On rehausse la brightness.

4.3.3. Raisonnement

On y est allé en essais erreur ce fut une énorme perte de temps. La quantité et différence des pixels causait de sérieux problèmes pour les étapes suivantes.

4.4. Conception détaillé du comparateur

4.4.1. Vue structurelle



4.4.2. Vue comportementale

Notre comparateur utilise la librairie Pillow. Il utilise les fonctions suivantes :

1. On remet les images en RGB
2. On utilise « ImageChops » et on soustrait les images une à l'autre.
3. On les reconverti en Greyscale
4. On « recolorize » l'image et on retire le noir totalement en blanc et on met l'ensemble des tons en blanc. Seul le blanc original est transformé en rouge.
5. On rehausse la luminosité pour mieux voir les différences.

Une fois réalisé, si l'image contient du rouge c'est que l'on détecte du mouvement.

Par la suite on multiplie l'image de mouvement à l'image de base et on l'envoie dans notre interface graphique. Elle nous permet de voir lorsqu'il y a un déplacement.

Présentement notre solution n'est pas parfaite. Elle détecte beaucoup de déplacement. Elle devra être affiné.

4.4.4. Raisonnement

4.5. Conception détaillé du générateur d'alerte

4.5.1. Vue structurelle

Non complété

4.5.2. Vue comportementale

Non complété

4.5.3. Autres vues pertinentes

Non complété

4.5.4. Raisonnement

Manque de temps

Références

Spécifications des exigences, 20 décembre 2021, fourni sur le site du cours

Lien du repo : https://github.com/samm56784/Projet6GEI311.git