



Cahier des Charges Fonctionnel – Projet 41

Traitement de données pour le géo-positionnement du
TGV IRIS 320

Version 1.1

Date : 26/03/24

Professeur Encadrant : M.PAJUSCO



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Baptiste HUBERT

Eric KHELLA

Antoine GATARD

1. INTRODUCTION ET PRÉSENTATION GÉNÉRALE DU BESOIN

Le département Micro-onde d'IMT Atlantique participe à un projet pour établir des modèles de canaux de transmission pour le déploiement du futur standard de télécommunication ferroviaire, le FRMCS (Futur Railway Mobile Communication System). Dans ce but, le TGV IRIS 320 a permis d'effectuer certaines mesures de propagation, ainsi le géo-positionnement de ces mesures devient primordial pour mener à bien ce projet.

Pour géopositionner de la manière la plus précise ce TGV IRIS 320, nous disposons de plusieurs fichiers vidéo (caméras à l'avant et à l'arrière du train, ainsi qu'une caméra 360° au-dessus du train) et un fichier XML. Notre travail consiste donc à utiliser ces fichiers et les assembler pour déduire la position du TGV, la vitesse du TGV, l'instant et la frame associée.

Le projet, sous la supervision de M. PAJUSCO, vise donc à synchroniser et à analyser ces heures de vidéos sous différents angles, en les coordonnant dans une perspective spatio-temporelle. Grâce aux données fournies par les différents fichiers (vitesse, point kilométrique, heure précise à la seconde), le défi consiste à affiner la localisation du train à une échelle extrêmement précise (précis à la milliseconde, on ne peut se permettre d'être moins précis car le projet est utilisé dans le cadre de mesures de propagation dans les canaux de transmission). L'ambition est de parvenir à déterminer la position exacte du TGV à l'aide d'une interpolation avancée des données issues des vidéos et des capteurs, en utilisant les technologies de programmation Python et XML.

2. Périmètre du Système

2.1 Acteurs

2.1.1 Utilisateurs

Le principal utilisateur de ce système sont les personnes en charge du projet FRMCS.

2.1.2 Clients

Les clients principaux de ce système sont d'une part le département Micro-onde, et donc indirectement les responsables du projet FRMCS, qui bénéficieront des résultats obtenus par l'analyse des données fournies. Les attentes et exigences de ce projet d'un nouveau cadre 5G va permettre aux opérateurs ferroviaires de moderniser leurs services.

2.1.3 Composants/Systèmes Externes

Les composants et systèmes externes comprennent des fichiers vidéo agissant en tant que sources d'entrée. Ces fichiers contiennent des informations essentielles sur la vitesse du TGV, la borne kilométrique correspondante, et une diode qui s'allume au début de chaque nouvelle seconde avec une précision de l'ordre de la milliseconde. Le fichier XML, également externes au système, permet de répertorier les différentes trames vidéo, les caméras utilisées, et les heures de début à la seconde près. Dès lors, il servira afin de mutualiser les différentes analyses. L'interaction avec ces composants externes est cruciale pour l'efficacité globale du programme.

2.2 Entrées/Sorties

2.2.1 Entrées

Les données d'entrée du système sont constituées de fichiers vidéo au format avi et mp4. Ces fichiers intègrent des informations cruciales telles que la vitesse du TGV, la borne kilométrique correspondante, et une diode permettant de repérer précisément le début d'une seconde. Le fichier XML complète ces données en mutualisant les informations des différentes vidéos afin de pouvoir retracer leur chronologie.

2.2.2 Sorties

Les sorties du système consistent en un fichier CSV structuré contenant les informations extraites (numéro de frame, heure, vitesse instantanée, borne kilométrique). En outre, le programme offre la possibilité d'exporter les résultats dans un format utilisable, que ce soit sous forme de liste ou de dictionnaire, facilitant ainsi une utilisation programmable des données analysées. Enfin, il doit offrir la possibilité d'accéder aux différentes prises vidéos en entrant une heure à la milliseconde près.

2.2.3 Clarification sur ce qui n'est pas inclus dans le système

Le périmètre du système est délimité de manière à traiter spécifiquement les fichiers vidéo et XML mentionnés précédemment. Ainsi, le programme n'inclura pas le traitement d'autres types de fichiers vidéo ou de données non spécifiées dans les consignes. De plus, les aspects liés à la visualisation des résultats, tels que la représentation graphique des données, ne sont pas pris en compte dans le cadre de ce système.

2.3 Justification de l'exhaustivité du périmètre

La délimitation du système est justifiée par sa capacité à répondre de manière exhaustive aux besoins du département micro-onde qui collabore avec le FRMCS. L'extraction précise des informations essentielles telles que la vitesse, la borne kilométrique, et les données temporelles de la diode garantit une analyse approfondie du déplacement du TGV. L'intégration du fichier XML enrichit cette analyse en fournissant des données complémentaires telles que les dates d'enregistrement des vidéos, la caméra utilisée ainsi que la durée des vidéos. Ainsi, le périmètre du système assure une couverture complète et précise des exigences exprimées par la SNCF, offrant une granularité temporelle optimale jusqu'à la milliseconde.

3. EXPRESSION FONCTIONNELLE DU BESOIN

Le langage de développement utilisé est Python, le code développé respecte la convention PEP8.

Fonctionnalités primaires :

Le programme de restitution des données doit générer un fichier CSV pour chaque vidéo, incluant le numéro de la frame, l'heure UTC+1 avec une précision jusqu'à la milliseconde, la position et la vitesse du train. Le temps de génération de ce fichier doit

être inférieur ou égal au temps de la vidéo. Cette limitation en temps de traitement s'explique par le fait que le client souhaite ne pas accumuler des vidéos à traiter, ainsi en respectant cette limite la vidéo enregistrée un jour n peut être traitée entièrement au jour n+1.

Pour récupérer les données caractéristiques du train (vitesse, position), plusieurs méthodes sont utilisées. Pour les fichiers vidéo, le programme doit extraire de chaque image les données de la vitesse et de la borne kilométrique ainsi que l'heure. Quant aux fichiers XML, le programme doit extraire la date de la vidéo, sa durée et la caméra utilisée. De plus, afin que le programme puisse récupérer de manière fiable les informations de chaque image, il faut que la police soit supérieure à 10 pt et que la résolution soit supérieure à 300 dpi (dots per inch) - d'après la documentation de la librairie Pytesseract



Image extraite d'un fichier vidéo de la caméra motrice arrière, en haut à gauche les kilomètres, en haut à droite l'heure, en bas à gauche la vitesse

```

326 1002" PK_fin="110718" Video="M1_231018_1329_216000_V1_110002.avi" Date="23/10/18 13:29" Duree="0:44" Raison="EndVoie" Camera="M1" />
327 PK_fin="10002" Video="M1_231018_1553_216000_V1_1.avi" Date="23/10/18 15:53" Duree="3:33" Raison="SplitDistance" Camera="M1" />
328 PK_fin="10002" Video="M2_231018_1553_216000_V1_6.avi" Date="23/10/18 15:53" Duree="3:33" Raison="SplitDistance" Camera="M2" />
329 107" PK_fin="20001" Video="M2_231018_1557_216000_V1_10007.avi" Date="23/10/18 15:57" Duree="2:59" Raison="SplitDistance" Camera="M2" />
330 102" PK_fin="20001" Video="M1_231018_1557_216000_V1_10002.avi" Date="23/10/18 15:57" Duree="2:59" Raison="SplitDistance" Camera="M1" />
331 101" PK_fin="30001" Video="M2_231018_1600_216000_V1_20001.avi" Date="23/10/18 16:00" Duree="2:27" Raison="SplitDistance" Camera="M2" />
332 106" PK_fin="30001" Video="M1_231018_1600_216000_V1_20006.avi" Date="23/10/18 16:00" Duree="2:27" Raison="SplitDistance" Camera="M1" />
333 106" PK_fin="40003" Video="M2_231018_1602_216000_V1_30006.avi" Date="23/10/18 16:02" Duree="2:2" Raison="SplitDistance" Camera="M2" />
334 106" PK_fin="40003" Video="M1_231018_1602_216000_V1_30006.avi" Date="23/10/18 16:02" Duree="2:2" Raison="SplitDistance" Camera="M1" />
335 108" PK_fin="50004" Video="M2_231018_1604_216000_V1_40008.avi" Date="23/10/18 16:04" Duree="2:0" Raison="SplitDistance" Camera="M2" />
336 108" PK_fin="50004" Video="M1_231018_1604_216000_V1_40008.avi" Date="23/10/18 16:04" Duree="2:0" Raison="SplitDistance" Camera="M1" />
337 109" PK_fin="60003" Video="M2_231018_1606_216000_V1_50009.avi" Date="23/10/18 16:06" Duree="2:1" Raison="SplitDistance" Camera="M2" />
338 109" PK_fin="60003" Video="M1_231018_1606_216000_V1_50009.avi" Date="23/10/18 16:06" Duree="2:1" Raison="SplitDistance" Camera="M1" />
339 108" PK_fin="70004" Video="M2_231018_1608_216000_V1_60008.avi" Date="23/10/18 16:08" Duree="2:0" Raison="SplitDistance" Camera="M2" />
340 108" PK_fin="70004" Video="M1_231018_1608_216000_V1_60008.avi" Date="23/10/18 16:08" Duree="2:0" Raison="SplitDistance" Camera="M1" />
341 109" PK_fin="80002" Video="M2_231018_1610_216000_V1_70009.avi" Date="23/10/18 16:10" Duree="2:0" Raison="SplitDistance" Camera="M2" />
342 109" PK_fin="80002" Video="M1_231018_1610_216000_V1_70009.avi" Date="23/10/18 16:10" Duree="2:0" Raison="SplitDistance" Camera="M1" />
343 107" PK_fin="90008" Video="M2_231018_1612_216000_V1_80007.avi" Date="23/10/18 16:12" Duree="2:0" Raison="SplitDistance" Camera="M2" />
344 107" PK_fin="90008" Video="M1_231018_1612_216000_V1_80007.avi" Date="23/10/18 16:12" Duree="2:0" Raison="SplitDistance" Camera="M1" />
345 103" PK_fin="100003" Video="M1_231018_1614_216000_V1_90003.avi" Date="23/10/18 16:14" Duree="2:0" Raison="SplitDistance" Camera="M1" />
346 103" PK_fin="100003" Video="M2_231018_1614_216000_V1_90003.avi" Date="23/10/18 16:14" Duree="2:0" Raison="SplitDistance" Camera="M2" />
347 1004" PK_fin="110001" Video="M2_231018_1616_216000_V1_100004.avi" Date="23/10/18 16:16" Duree="3:33" Raison="SplitDistance" Camera="M2" />

```

Fichier XML

Pour associer les données à un instant précis avec une précision jusqu'à la milliseconde, le programme utilise la détection de la luminescence de la diode rouge

marquant le début de chaque seconde. En effet, cette diode fonctionne comme une horloge de précision et est utilisée comme référence temporelle. Toutefois, étant donné la précision de la caméra (60 frames/sec), une méthode plus précise pour déterminer le temps de début de la seconde (à la milliseconde près) est nécessaire. Pour cela, on va réaliser une interpolation sur la luminescence de cette diode, puisqu'on passe d'une frame où la diode est éteinte à une frame où celle-ci est allumée. En suivant l'évolution de la luminescence sur les différentes frames (du moment où elle s'allume faiblement, jusqu'à sa luminosité maximale) il sera possible de déterminer à quel instant de la seconde la frame a lieu avec une précision de l'ordre de la milliseconde.



Image extraite d'un fichier vidéo de la caméra 360°, on y voit la diode rouge allumée sur cette image

Quant au traitement vidéo ; deux fonctionnalités seront présentes. Dans la première, un dossier éphémère sera créé sur le bureau de l'utilisateur. Ce dernier devra y coller une copie de la vidéo qui sera traitée. La possibilité d'y mettre plusieurs vidéos sera intégrée. Dans la seconde fonctionnalité, l'utilisateur indique s'il souhaite traiter toutes les vidéos de l'analyse en suivant la chronologie du fichier XML.

Fonctionnalités complémentaires :

Disposer d'une fonction qui permet de charger les frames associées à chaque caméra à cet instant donné et toutes les caractéristiques à partir d'un temps précis.

L'utilisateur dispose d'une estimation du temps de traitement lorsqu'il utilise le programme. Pour réaliser cette estimation une fonction d'interpolation est utilisée. En interpolant le temps de calcul au fil de l'analyse, la fonction peut déterminer une estimation de la durée du traitement restant.

Afin de perfectionner les zones à intérêt et minimiser le temps d'analyse, un code complémentaire est mis à disposition de l'utilisateur. Après avoir placé la vidéo de son choix dans un dossier éphémère, il peut sélectionner sur la première image de la vidéo

la zone à intérêt à l'aide de clics. Plus la zone à traiter est petite, plus le temps d'analyse sera court.

Cas d'utilisations / scénarios :

Cas d'utilisation 1 : L'utilisateur peut choisir d'analyser un ou plusieurs fichiers vidéo (pour extraire les données à chaque instant t)

Scénario 1 correspondant : il déplace ces fichiers dans un document temporaire, puis lance le code. Il récupérera en sortie un fichier csv contenant : dans la 1^{ère} colonne, le numéro de la frame, dans la 2^{nde} colonne, l'instant t (HH : MM : xx,xxx), dans la 3^{ème} colonne, la vitesse (en km/h), dans la 4^{ème} colonne, le kilométrage (en km). Ce fichier csv se placera dans ce même document temporaire avec le fichier vidéo brut. Pour résumer, l'utilisateur peut avoir accès pour tous les fichiers (en supposant qu'il donne bien le fichier vidéo de la caméra arrière ET le fichier vidéo de la diode rouge) au nombre de frame, à l'instant, la vitesse, ou la position (kilométrage).

Cas d'utilisation 2 : L'utilisateur veut redéfinir l'espace de reconnaissance des données sur le fichier vidéo de la caméra motrice arrière car celles-ci ont changées.

Scénario 2 correspondant : si pour une certaine raison, les données à extraire (vitesse, heure, etc.) ne sont pas exactement au même endroit d'un fichier à l'autre, l'utilisateur a la possibilité de redéfinir cet espace avec une fonction à part. Après avoir lancé ce code en question, il dispose d'un curseur pour définir l'espace de reconnaissance souhaité. Il peut ensuite réutiliser le code d'analyse des fichiers sans problème.

Cas d'utilisation 3 : L'utilisateur peut également avoir accès à n'importe quelle frame d'un fichier vidéo à l'instant t précisé, grâce à une classe regroupant toutes ces données (sans tout ouvrir dans un fichier csv)

Scénario 3 correspondant : L'utilisateur met en entrée une heure arbitraire, il obtiendra en sortie un dossier avec 3 capture d'écrans (une image de la caméra motrice avant, une image de la caméra motrice arrière, une image de la caméra de la diode rouge).

Cas d'utilisation 4 : L'utilisateur ne fournit qu'un fichier sur les 2 nécessaires pour une analyse correcte de ces fichiers (celui de la caméra motrice arrière pour accéder aux différentes données, et celui de la diode rouge correspondant pour avoir une référence temporelle)

Scénario 4 correspondant : Après le lancement du code, l'utilisateur se voit renvoyer une erreur lui disant que tel ou tel fichier n'a pas été renseigné. Dans le cas où les 2 fichiers vidéo ont été renseignés mais ne correspond pas au même trajet, un message d'erreur lui est aussi renvoyé pour vérifier que les 2 fichiers vidéo sont bien le même morceau de trajet.

Tableau récapitulatif des priorités des différentes fonctions du programme

Nom de la fonction	Tâche réalisée par la fonction	Justification du niveau de priorité	Niveau de priorité
Génération fichier csv	Cette fonction permet de générer le fichier csv sur lequel figurent les caractéristiques du train	Si cette fonction n'était pas implémentée les données extraites par le programme ne seraient pas recueillies. Le	

		besoin du client ne serait alors pas rempli.	
Interpolation sur la diode	Cette fonction permet de réaliser l'interpolation qui associe un instant précis de l'ordre de la ms aux données recueillies.	Sans l'instant qui correspond à la mesure d'une donnée de cette dernière est sans intérêt et ne peut être exploitée.	
Sélection de vidéos	Cette fonction permet d'analyser une vidéo particulière au sein d'un dossier de vidéos	Cette sélection permet à l'utilisateur de gagner du temps et de ne pas manipuler la vidéo en elle-même, elle ne risque donc pas d'être perdue ou endommagée. Cependant cette fonction n'est pas indispensable pour l'utilisation du programme.	
Chargement d'une frame	Cette fonction permet d'obtenir les informations d'une frame spécifiée.	L'obtention d'une frame particulière bien que très pratique pour éviter d'avoir à traiter l'entièreté d'une vidéo n'est pas une fonction des plus prioritaires car les données recueillies apparaissent dans le fichier csv que peut générer la première fonction	
Définir l'espace de reconnaissance	Cette fonction permet de définir un nouvel espace de reconnaissance dans le cas où la vidéo n'est pas du format pris en charge par défaut.	Cette fonction remplit un rôle utilitaire qui permet d'élargir la quantité de format que le programme peut prendre en charge. Cependant, elle ne participe pas directement à la réalisation du besoin.	
Estimation du temps de traitement restant	Cette fonction analyse les frames de la vidéo pour estimer la durée totale que le traitement prendra. Cette fonction détecte le temps de calcul des premières frames, et fait une extrapolation pour estimer la durée totale du traitement.	Cette fonction est seulement utilitaire et ne participe pas directement à remplir le besoin du client, mais participe à sa satisfaction.	

Légende :



Fonctions prioritaires



Fonctions peu prioritaires



Fonctions non prioritaires

4. VALIDATION DU BESOIN

Afin d'être certain que le produit délivré correspond bien aux attendus du client, nous pouvons vérifier que les fonctions présentées précédemment sont bien respectées.

Pour procéder à la validation, nous vérifierons d'abord que le lancement du code renvoie bien un fichier csv contenant les informations décrites auparavant (vitesse, position...), dans le bon ordre, dans la bonne unité et dans son intégralité (si plusieurs fichiers vidéo à traiter, alors plusieurs fichiers csv nommés en conséquence).

Nous pouvons également vérifier que le temps de traitement d'un fichier vidéo ne dépasse pas le temps de vidéo lui-même, par simple comparaison entre les 2 durées.

La vérification de la précision à la milliseconde près peut se faire en vérifiant d'une part qu'à chaque milliseconde, un positionnement est affecté, et que les données associées diffèrent d'une milliseconde à l'autre.

La redéfinition de l'espace à reconnaître peut faire l'objet d'un test avec le client, pour vérifier que le système est assez facile à prendre en main. On laisserait alors au client le choix de redéfinir l'espace de reconnaissance, et on vérifie ensuite que le fichier qu'il récupère contient bien les données de la nouvelle zone.

La validation de l'estimation du temps de traitement restant peut se faire en lançant simplement un traitement, en observant le temps estimé (1^{ère} vérification : celui-ci s'affiche) et en comparant avec le temps effectif pour le traitement (2^e vérification : précision correct de l'estimation).

En présence du client, nous pourrions effectuer un test où celui-ci pourra chercher précisément la frame qui l'intéresse et les données qui vont avec, à un instant donné via une classe. Ou encore un autre test durant lequel le client indique quel(s) vidéo(s) il souhaite traiter, puis lorsque le nom de ces vidéos est donné au code, les vidéos se mettent dans un nouveau document à part et le fichier csv correspondant se met dans ce même nouveau document.

5. CONCLUSION

Pour conclure, ce document définit les grandes lignes et les détails du système qui permettra de localiser avec précision le TGV IRIS 320 afin de mettre au point un nouveau modèle de télécommunication ferroviaire en France (projet FRMCS). En examinant le contexte, les besoins et les défis du projet, nous avons mis en place un plan pour le développement du produit.

Le système proposé, qui utilise l'analyse vidéo et l'interpolation de données en Python, est un outil important pour établir des modèles de canaux de transmission pour le déploiement du standard de télécommunication

Nous avons détaillé les fonctionnalités importantes et envisagé différents scénarios d'utilisation pour répondre au mieux aux besoins exprimés par notre encadrant, M. PAJUSCO, tout en tenant compte des situations difficiles qui pourraient se présenter. Tester ces besoins par des démonstrations et des analyses sera l'étape finale essentielle pour garantir que le produit final répond aux attentes.

Ce projet vise à innover et à améliorer continuellement, montrant l'engagement de tous les participants à aider au développement d'une nouvelle technologie. Nous sommes confiants que le travail d'équipe, en suivant les instructions de ce document et de notre encadrant, aboutira à un système qui répondra aux besoins stratégiques et opérationnels du projet.