

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département d'Electronique

Mémoire de projet de fin d'études

pour l'obtention du diplôme d'ingénieur d'état

en Eletronique

Moniteur Différentiel de Mouvement d'Image

(Differential Image Motion Monitor)

Réalisation d'un logiciel de gestion et de commande d'un DIMM

Achouri.Ahmed_Redouane

Habchi.Mohamed_Khalil

Encadré par: M.Nassim Seghouani & M.Adel Belouchrani

Présenté et soutenue publiquement le (06/07/2019)

Composition du jury:

Président M. Mohamed Ouassaid TAGHI, Mr. ENP

Promoteur M.Adel BELOUCHRANI, Pr. ENP

Promoteur M.Nassim SEGHOUANI, Pr. CRAAG

Examineur M.Mourad ADNANE, Dr. ENP

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département d'Electronique

Mémoire de projet de fin d'études

pour l'obtention du diplôme d'ingénieur d'état

en Eletronique

Moniteur Différentiel de Mouvement d'Image

(Differential Image Motion Monitor)

Réalisation d'un logiciel de gestion et de commande d'un DIMM

Achouri.Ahmed_Redouane

Habchi.Mohamed_Khalil

Encadré par: M.Nassim Seghouani & M.Adel Belouchrani

Présenté et soutenue publiquement le (06/07/2019)

Composition du jury:

Président M. Mohamed Ouassaid TAGHI, Mr. ENP

Promoteur M.Adel BELOUHRANI, Pr. ENP

Promoteur M.Nassim SEGHOUANI, Pr. CRAAG

Examineur M.Mourad ADNANE, Dr. ENP

Dédicace

Nous dédions ce travail à nos chers parents,

A nos frères et sœurs,

A nos familles,

Ainsi qu'à tout ceux qui ont collaboré de près ou de loin à l'élaboration de ce travail.

Remerciement

En premier lieu, nous remercions Dieu le tout puissant de nous avoir donné le courage, la volonté et la patience pour réaliser ce travail.

Nous adressons aussi nos remerciements à Mr NASSIM SEGHOUANI pour son encadrement, sa compréhension, et ses encouragements.

On remercie aussi notre encadreur, Mr BELOUACHRANI ADEL de nous avoir aidé dans notre travail, et surtout d'avoir cru à nos compétences.

Un grand merci à nos parents pour nous avoir inlassablement encouragés à explorer tous les sujets qui nous intéressaient.

Nous remercions enfin l'ensemble de nos proches qui nous ont aidé et motivé durant ce cursus rempli de challenges. Nous les remercions pour l'aide qu'ils nous ont apporté dans la réalisation de ce travail.

Ce travail, fruits de notre cursus et notre volonté, n'a été possible que grâce au tous nos enseignants dont nous louons les efforts qu'ils ont consentis durant toutes ces années.

ملخص

كجزء من مشروع مرصد الأوراس، تُعد خطوة تأهيل الموقع ضرورية لتقدير عوامل الاضطرابات الجوية، والتي تُحد من دقة الملاحظات الفلكية. لهذا الغرض، تم إنشاء مراقب تفاضلي لحركة الصور في جبال الأوراس. الهدف من العمل هو تحقيق برنامج من أجل التحكم في المراقب و حساب العامل الأساسي للاضطراب في الغلاف الجوي. النتائج التي تم الحصول عليها فيها في كفاية مع تلك المنشورة في الأدب

: الكلمات الدالة رصد حركة الصورة التفاضلية الاضطرابات الجوية

Abstract

As part of the Aurès observatory project, a site qualification step is necessary in order to estimate the parameters of the atmospheric turbulence, which limits the resolution of astronomical observations. For that purpose, a Differential Image Motion Monitor (DIMM) has been set up in the Aurès mountains. The goal of this work is the realization of a software in order to control the DIMM and calculate in real-time the main parameter of the atmospheric turbulence: The seeing. The obtained results are in adequation with those published in the literature.

Key Words: atmospheric turbulence DIMM

Resumé

Dans le cadre du projet de l'observatoire des Aurès, une étape de qualification du site est primordiale afin d'estimer les paramètres de la turbulence atmosphérique qui a pour effet de limiter la résolution des observations astronomiques. Pour ce faire un moniteur de mouvement d'image différentiel (DIMM) a été mis en place dans les montagnes des Aurès. Le but de ce travail est la réalisation d'un logiciel permettant de piloter le DIMM et de calculer en temp-réel le principal paramètre de la turbulence atmosphérique : Le seeing. Les résultats obtenus sont dans l'ordre de grandeur de ceux publiés dans la littérature.

Mots clés : turbulence atmosphérique DIMM

Contents

Table des figures

Introduction Générale

1	Introduction Générale	9
I	État De L'art Et Concept	12
2	La Turbulence Atmosphérique	13
2.1	Introduction	14
2.2	Rappel Historique	14
2.3	La Turbulence dans les Couches Atmosphériques	15
2.4	Caractéristiques Spécifiques de la Turbulence Atmosphérique	15
2.5	Fluctuation de la Température et l'Indice de Réfraction	16
2.6	Les paramètres théoriques	16
2.6.1	L'angle d'Isoplanétisme	16
2.6.2	Le paramètre de Fried r_0	16
2.6.3	Le Seeing	17
2.6.4	Les Autres Paramètres	18
2.7	Réponse du Système Optique en Présence de la Turbulence atmosphérique	19
2.8	Techniques de Mesure de la Turbulence	20
2.8.1	Differential Image Motion Monitor	20
2.8.2	Multiple Aperture Scintillation Sensor MASS	21
2.9	Conclusion	21
3	Differential Image Motion Monitor	22
3.1	Introduction	23
3.2	L'instrumentation	23
3.2.1	Télescope	24
3.2.1.1	Eléments Constitutifs	24
3.2.1.2	L'Objectif	24
3.2.1.2.1	La Monture	24
3.2.1.3	Caractéristiques Techniques	25
3.2.2	Caméra CCD	26
3.2.3	Le Masque	27
3.3	La technique DIMM	28
3.3.1	Algorithme DIMM	29

3.3.1.1	Traitement de l'image	29
3.3.1.1.1	Le seuillage :	29
3.3.1.1.2	Calcul des Centroïdes	30
3.3.1.1.3	Méthodes de calcul du centroïde	30
3.3.1.1.4	Choix de la méthode de calcul	32
3.3.1.2	Calcul du seeing	33
3.4	Conclusion	34
II Implémentation		35
4	Modélisation Conceptuelle et Organisationnelle	36
4.1	Introduction	37
4.2	L'environnement de programmation Qt	37
4.2.1	Qt Designer	38
4.2.2	La version Qt5	39
4.3	Choix du Langage de Programmation	39
4.3.1	Caractéristiques du Langage Python	39
4.3.2	Python et la Programmation Orientée Objet	40
4.3.3	Python et la Programmation Temps-réel	41
4.4	Les Bibliothèques	42
4.4.1	OpenCV	42
4.4.1.0.1	Le Module OpenCV Python	43
4.4.2	Bibliothèque Numpy	43
4.4.2.0.1	OpenCV et Numpy	44
4.4.3	Bibliothèque Matplotlib	44
4.5	Diagramme de cas d'utilisation	45
4.6	Description des Scénarios	46
4.6.1	Cas de la commande START	47
4.6.2	Cas de la commande Select ROI	47
4.6.3	Cas de la commande SETTINGS	47
4.6.4	Cas de l'instruction SIMULATION	47
4.6.5	Cas de la commande IMPORT	48
4.6.6	Cas de la commande EXPORT	48
4.6.7	La commande PAUSE/RESUME	48
4.7	Conclusion	48
5	Réalisation & Résultats	50
5.1	Introduction	51
5.2	La classe TIS_CAM	51
5.2.1	Méthodes Principales Constituant la classe TIS_CAM	51
5.3	Acquisition des données	52
5.4	Le Contrôle du Flux de Données par le Mode d'Interruption	53
5.4.1	La fonction SnapImage	53
5.5	Traitement d'image	54
5.5.1	Convertir l'image en niveaux de gris	54
5.5.2	Thresholding	54
5.5.3	Détection des Etoiles	55
5.6	Calcul du seeing	56

5.6.1	Barycentre d'une étoile	56
5.6.1.1	Calcul des barycentres par la méthode des moments . . .	56
5.7	Repérer les étoiles sur l'interface graphique	57
5.8	Affichage graphique de la variation du seeing en temps-réel	57
5.9	Principales Interfaces Graphiques	57
5.9.1	Fenêtre d'accueil	58
5.9.1.1	Bouton Start	59
5.9.1.2	Bouton Settings	59
5.9.1.3	Bouton Simulation	61
5.9.1.4	Bouton Import	61
5.9.1.5	Bouton Export	62
5.9.1.6	Case à cocher 'Enable Seeing Monitoring'	63
5.9.1.7	Choix de l'étoile et de ses caractéristiques	63
5.9.1.8	Bouton Select ROI	63
5.9.1.9	Champs 'Saving File Path' et 'Saving File Name'	64
5.9.1.10	Champs 'Start Recording' et 'End Recording'	64
5.10	Résultats Expérimentaux	64
5.10.1	Test de mise au point	64
5.11	Conclusion	66
	Conclusion Générale	67
	References	69
	Annexe	70

Table des figures

1.1	Photo Altimétrique de l'Algerie, avec au centre la région des Aurès. . . .	12
2.1	Figure: Formation de l'image dans les deux cas, avec ou sans turbulence.	20
3.1	télescope complet avec sa monture, et la caméra CCD reliée	23
3.2	DMK 51AU02.AS	27
3.3	Le masque avec double ouverture	28
3.1	Explication du principe de l'utilité du prisme relié à l'une des ouvertures	29
4.1	Logique de fonctionnement du logiciel	46
5.1	Fenêtre d'accueil	58
5.2	Selection de la caméra	59
a	Reglage des paramètres fondamentaux	60
b	Réglage du gamma de l'image (correction de la brillance)	60
c	Coffee.	60
5.4	The same cup of coffee. Two times.	60
5.5	Lecture depuis fichier video	62
5.6	Sauvegarde vers un fichier video	63
5.1	Calcul en temps-réel du Seeing sur le site d'Alger	65

Chapitre 1

Introduction Générale

L'Algérie est en phase de construire le premier observatoire astronomique post-indépendance. En effet, le seul observatoire qui existe à l'heure actuelle est l'observatoire d'Alger, construit pendant la période coloniale en 1890. Si le site d'Alger était favorable aux observations et objectifs scientifique d'autre fois, il ne peut plus aujourd'hui effectuer des observations astronomiques de qualité.

En effet, l'expansion de la ville et le développement technologique ont fait que la pollution lumineuse est devenue actuellement très importante, et un microclimat défavorable aux observations s'est installé. De plus les exigences des instruments actuels (Télescope, Caméra CCD, ...) sont devenus plus importants en matière de qualité du ciel.

Les Astronomes Algériens ont donc été conduits à rechercher un autre site plus favorable pour y construire un nouvel observatoire.

Les observatoires astronomiques sont souvent construits à des altitudes importantes afin de fuir la pollution environnante mais surtout de diminuer les gradients de températures, principale source de la turbulence atmosphérique qui fait « danser » les étoiles au foyer d'un télescope, diminuant ainsi la qualité des observations. Si on observe une carte altimétrique de l'Algérie (voir fig 1.1), deux régions élevées se dégagent. La région du Hoggar avec le montagne Tahat (2800 m) et la région des Aurès avec le montagne Chélia (2300 m). La région du Hoggar était trop éloigné des universités algériennes, les chercheurs se sont intéressés surtout à la région des Aurès qui présente de nombreux sites susceptibles d'accueillir un observatoire (mont-Kelthoum, Chechar), ce dernier a été choisi car il présente moins de pollution lumineuse, le mont-Kelthoum qui est plus exposé à la lumière des

grandes villes. Une étude de la nébuleuse a déjà été effectuée, mais il reste à quantifier la turbulence atmosphérique.

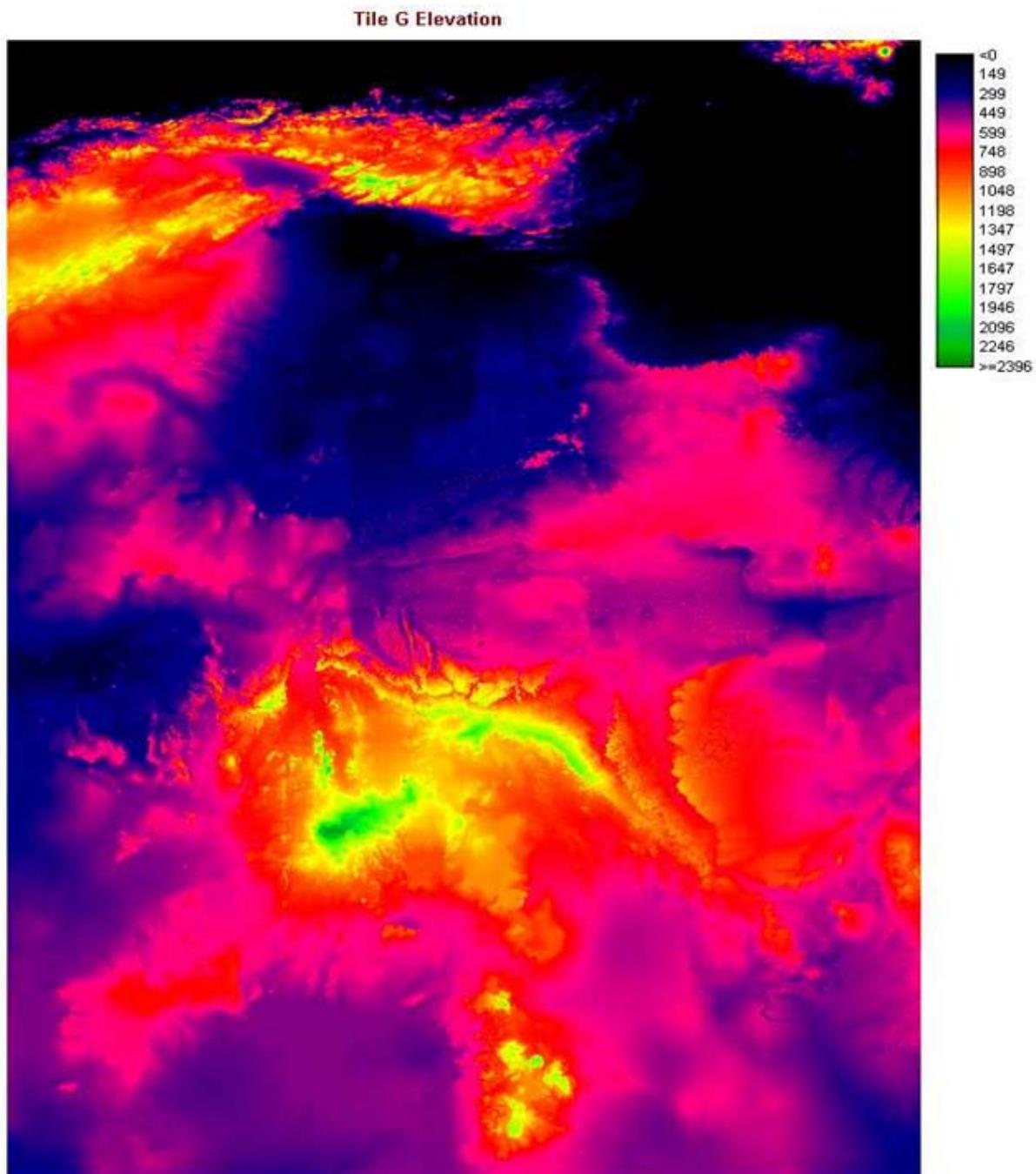


FIGURE 1.1: Photo Altimétrique de l'Algérie, avec au centre la région des Aurès.

Une première station a donc été construite sur le site choisi dont le but est de mesurer les paramètres météo (à l'aide d'une station météo) et les paramètres de la turbulence atmosphérique grâce à un télescope dédié.

C'est dans ce cadre que s'inscrit notre travail, qui consiste à réaliser un logiciel permettant

de mesurer la turbulence atmosphérique à partir des images issus d'un appareil dédié plus connu sous le nom DIMM (Differential Image Motion Monitor). Ce dispositif qu'on va présenter en détail au deuxième chapitre, nous permet de mesurer l'un des paramètres le plus important de la turbulence en temps réel qui est le Seeing. Ce mémoire est divisé en deux parties :

la première partie sera consacrée à la théorie de la turbulence atmosphérique qu'on va la présenter au premier chapitre , dans le deuxième chapitre on va parler du dispositif DIMM et la technique de mesure du Seeing.

La deuxième partie sera consacrée à notre logiciel.

dans le troisième chapitre nous parlerons de la technologie derrière ce travail ainsi qu'à la modélisation conceptuelle du logiciel. Le quatrième chapitre sera consacré au logiciel réalisé ainsi qu'aux résultats obtenus.

Nous finirons par une conclusion générale et des perspectives.

Première partie

État De L'art Et Concept

Chapitre 2

La Turbulence Atmosphérique

2.1 Introduction

L'atmosphère est responsable de tous les jeux de lumière observés dans le ciel, en perturbant de façon spectaculaire les directions et les couleurs des rayons lumineux qui nous parviennent du soleil ou de tout autre astre. Le but de ce chapitre est de rappeler les résultats les plus importants des travaux qui ont été fait sur l'étude de la turbulence atmosphérique et la caractérisation de ce phénomène.

2.2 Rappel Historique

Leonardo da Vinci, l'artiste ingénieur de la Renaissance, fut le premier à avoir utiliser le mot « turbolenza », en français 'turbulence', pour décrire les mouvements complexes de l'eau et de l'air en observant de plus près les tourbillons, les jets et les écoulements des fleuves [?]. Les premiers expérimentateurs et théoriciens de la turbulence, parmi lesquels Poiseuille, Reynolds, puis Taylor et Von Karman ont donné la première définition de la turbulence comme étant « un mouvement irrégulier qui apparaît dans les fluides quand ceux-ci rencontrent des obstacles solides ».

Ce n'est qu'au début du 19ième siècle que Claude Navier a écrit les équations de base régissant l'évolution dans le temps de l'écoulement d'un fluide en turbulence [A., 1976]. Il existe deux types d'écoulements, on a l'écoulement laminaire et l'écoulement turbulent. Pour le premier type, il s'agit d'un mouvement régulier où tous les points du fluide ont la même vitesse, alors qu'on a un mouvement tourbillonnaire irrégulier dans l'espace et dans le temps pour les écoulements turbulents. La turbulence est donc une propriété des écoulements d'un fluide et non du fluide lui-même et la complexité du phénomène a fait qu'aucune définition rigoureuse n'a été établie mais on peut indiquer le caractère aléatoire de la turbulence (l'irrégularité), le caractère rotationnel aléatoire de la vitesse, en plus de la Diffusion et la Dissipation.

2.3 La Turbulence dans les Couches Atmosphériques

L'atmosphère terrestre, piégée entre le vide qu'est le milieu interstellaire et la surface composée de l'hydrosphère et la croûte terrestre, cette enveloppe constituée principalement de gaz, est responsable du transfert énergétique entre ces deux milieux. Le rayonnement du soleil d'un côté et la vapeur d'eau, sel, ... etc de l'autre. Compte tenu de ces échanges, des inhomogénéités apparaissent dans la répartition de la température et des teneurs en vapeur d'eau de l'atmosphère, d'autant plus que les écarts de densité en conjonction avec la gravité, la rotation et la géométrie terrestre, sont finalement les moteurs de l'ensemble des mouvements atmosphériques.

2.4 Caractéristiques Spécifiques de la Turbulence Atmosphérique

L'air atmosphérique est un mélange de gaz contenant des particules liquides et solides en suspension (gouttelettes d'eau, cristaux salins et les poussières provenant de la surface de la Terre). Chose qui a amené les météorologistes à considérer l'air comme un mélange de deux gaz, l'air sec (composé de plusieurs gaz dont les proportions restent sensiblement constantes) et la vapeur d'eau (l'humidité). Les nombreuses mesures in situ effectuées dans la haute atmosphère permettent de se faire une idée de la répartition verticale des températures, cela a permis de définir un profil vertical type de l'atmosphère. La stratification dans la répartition des couches dépend aussi du moment de l'observation, c'est à dire le jour ou la nuit. Le jour, on remarque une importante turbulence localisée au sol à cause de l'échauffement des structures et du sol par le Soleil en plus de l'effet de cisaillement dû au vent, alors que la nuit la turbulence est localisée en altitude. Puisque on a inversion de température dans la basse atmosphère (le sol se refroidit plus vite que l'air), l'air chaud stagne au-dessus des masses d'air froid, ce qui crée une stabilité dans la basse atmosphère.

2.5 Fluctuation de la Température et l'Indice de Réfraction

La turbulence donne lieu à des mouvements tourbillonnaires qui vont entraîner des mélanges de masses d'air différentes en température et pression. Ces fluctuations seront responsables des fluctuations de l'indice de réfraction de l'air entraînant les perturbations d'une onde électromagnétique lors de son passage à travers les couches atmosphériques.

2.6 Les paramètres théoriques

On caractérise la turbulence optique en introduisant quatre paramètres :

2.6.1 L'angle d'Isoplanétisme

L'angle isoplanétique est fondamental en optique adaptative. Il correspond à l'angle de corrélation de la turbulence, i.e. la distance angulaire maximale entre deux points sources affectés par la même turbulence. Il est exprimé en secondes d'arc (1 seconde d'arc = 0,000277778 degré). Il vaut, pour un temps d'exposition 4 milliseconde de environ 0.5 à 2 secondes d'arc.

2.6.2 Le paramètre de Fried r_0

Ce paramètre est également fortement utilisé dans les équations par de nombreuses équipes de recherche ; il correspond au diamètre qu'aurait un télescope qui fournirait, en l'absence de turbulence, une tache d'Airy (qui est la figure de diffraction résultant de la traversée d'un trou circulaire par la lumière) de même taille que le télescope, en présence de turbulence. Comme la turbulence, le seeing et le paramètre de Fried varient notamment en fonction de la température, de la pression et de la vitesse du vent. Il se mesure en secondes d'arc (arcsec). Typiquement, le seeing peut prendre des valeurs autour d'une à une seconde d'arc et demi. Pour le paramètre de Fried, l'ordre de grandeur est d'une

dizaine de centimètres [see,]. Pour un site d'observation, une turbulence et une longueur d'onde donnée, le r est égal au diamètre d'un télescope qui ne subirait pas la turbulence et dont la qualité d'image serait équivalente au diamètre infini qui lui la subirait.

2.6.3 Le Seeing

Le seeing est le paramètre principal dans la description de la turbulence atmosphérique. Il rend compte de la dégradation de la résolution des images.

Si on faisait des observations astronomiques, un télescope serait limité par le phénomène de diffraction exclusivement. L'angle de diffraction de la tache d'Airy est fonction du diamètre du télescope par la formule suivante :

$$\theta_{tot} = 2 \frac{1.22\lambda}{D}$$

(2.6.1)

avec D comme étant le diamètre de l'ouverture du télescope et une longueur d'onde définie.

Cette formule nous montre que si nous augmentons le diamètre du télescope, nous diminuons l'angle de diffraction. Ceci nous indique dans un premier temps que l'on a tout intérêt à choisir un télescope de grande ouverture. D'ailleurs, par le critère de Rayleigh, si la séparation de deux points objets est telle que le maximum de la figure de diffraction de l'un tombe sur le premier minimum de l'autre, alors nous pouvons distinguer ces deux points. Dans le cas contraire, nous n'en observons qu'un seul.

La résolution angulaire du télescope correspond donc au demi diamètre du disque d'Airy :

$$\theta_{res} = \frac{1.22\lambda}{D}$$

(2.6.2)

mais la résolution du télescope dépend d'un autre phénomène. A partir d'un certain diamètre du télescope, la limite de diffraction devient négligeable par rapport à la perturbation engendrée par l'atmosphère. De manière plus précise, ce n'est plus la limite de diffraction qui définit la résolution du télescope mais la perturbation engendrée par les différentes couches de l'atmosphère.

C'est pour cela qu'une grandeur a été instaurée : le Seeing qui caractérise les perturbations engendrée par l'atmosphère. Celui-ci indique la largeur angulaire à mi-hauteur de l'image d'une source ponctuelle d'une longueur d'onde donnée. Cette valeur est donnée en général en [arcsec]. On parle alors de : FWHM qui signifie Full Width at Half Maximum (voir [Tokovinin, 2002], [M. Sarazin,]). Le Seeing dépend non seulement de la longueur d'onde mais aussi du r_0 (paramètre de FRIED) :

$$FWHM = \frac{0.98\lambda}{r_0} \quad (2.6.3)$$

Cette valeur est en radian mais elle est le plus souvent exprimée en [arcsec]. Au niveau de la résolution, les télescopes de grand diamètre n'apportent pas forcément d'amélioration dû au Seeing mais ils offrent tout de même un facteur non négligeable : qui dit grand diamètre dit grande ouverture. Nous avons donc plus de lumière qui rentre dans celle-ci et il est alors possible de voir des étoiles de magnitude plus élevée (d'intensité plus faible).

2.6.4 Les Autres Paramètres

Les deux autres paramètres définis sont : · l'échelle externe également appelée échelle de cohérence spatiale correspond à la longueur caractéristique des plus grandes structures des fluctuations de la phase du front d'onde. Elle est liée à la taille des plus grands tourbillons dans un volume d'atmosphère considéré. le temps de cohérence correspond au temps durant lequel, en première approximation, le front d'onde turbulent n'aura pratiquement pas évolué.

2.7 Réponse du Système Optique en Présence de la Turbulence atmosphérique

A la traversée de l'atmosphère, les fronts d'ondes subissent une dégradation causée par les fluctuations de l'indice de réfraction. Celles-ci, étant liées à la phase, donnent lieu à des variations aléatoires de cette dernière. Cela se traduit par l'agitation et la distorsion de l'image qui prend une forme granulaire dans le plan image de l'instrument d'observation), la figure suivante présente les tavelures, en anglais «speckles» .

Chaque grain du speckle a pour dimension la tache d'Airy λ/D , où D est le diamètre de la pupille et l'extension spatiale de cette figure de speckle est de dimension égale à λ/r , elle est appelée étalement de l'image dû à l'effet de la turbulence atmosphérique caractérisé par son paramètre de Fried r_0 - [F, 1981] [?].

Du fait de l'évolution aléatoire et rapide de la turbulence atmosphérique dans le temps et dans l'espace rendant l'approximation de l'invariance par translation non appropriée. Les observations en astronomie et les estimations théoriques ont montré que ce domaine est de l'ordre de quelques secondes d'arc.

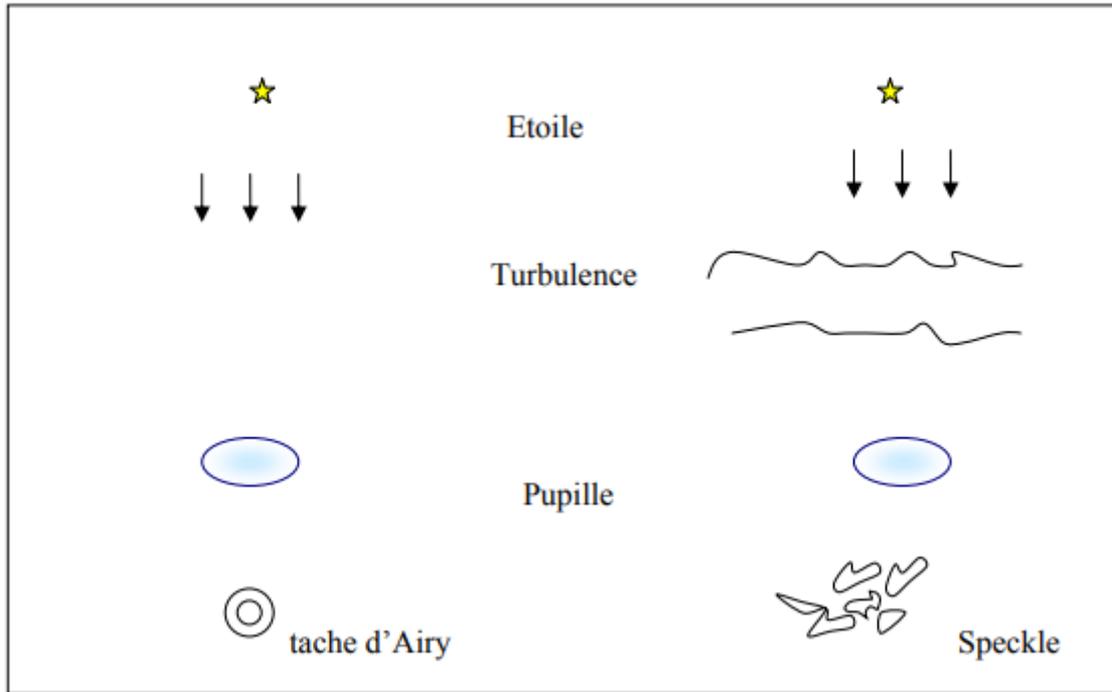


FIGURE 2.1: Figure : Formation de l'image dans les deux cas, avec ou sans turbulence.

2.8 Techniques de Mesure de la Turbulence

Deux instruments de mesure de la turbulence sur source unique, généralement utilisés de façon complémentaire. Le Differential Image Motion Monitor, permettant d'accéder à une mesure de la turbulence intégrée sur la ligne de visée, à partir des fluctuations de l'angle d'arrivée. Le Multiple Aperture Scintillation Sensor, utilisant des mesures de fluctuations d'intensité pour reconstruire le profil de turbulence, sans toutefois estimer la turbulence proche du sol.

2.8.1 Differential Image Motion Monitor

Le Differential Image Motion Monitor (DIMM) [M. Sarazin,] estime les variances de l'angle d'arrivée différentiel entre deux sous-pupilles observant une source unique. Généralement, la configuration DIMM est réalisée en masquant un télescope de plus grand diamètre pour ne conserver que deux sous-pupilles. Comme il estime des variances différentielles, le DIMM n'est pas sensible aux erreurs de pointage ni aux vibrations. A partir de cette

estimation différentielle, il est possible de mesurer le paramètre de FRIED (r_0) et le seeing. Le DIMM est aujourd'hui communément utilisé dans les campagnes de tests de sites astronomiques [Wang et al.(2007)] [Kornilov,]. Le DIMM ne permet cependant pas de reconstruire un profil de turbulence en fonction de l'altitude.

2.8.2 Multiple Aperture Scintillation Sensor MASS

Le Multiple Aperture Scintillation Sensor (MASS) [Tokovinin, 2002] [A.Voziakova,] mesure la scintillation d'une étoile simple à travers une série d'ouvertures annulaires concentriques et d'une ouverture centrale circulaire. Le MASS ne permet pas de sonder la turbulence proche du sol. Cela est dû au fait qu'il utilise la scintillation, dont l'amplitude diminue avec la distance de propagation. Le MASS n'échantillonne pas la turbulence de façon linéaire. Dans son fonctionnement le plus classique, il reconstruit six couches turbulentes, aux altitudes 0,5 ; 1 ; 2 ; 4 ; 8 et 16 km. Afin de mesurer la turbulence proche du sol, cet instrument est souvent utilisé en combinaison du DIMM [A.Voziakova,].

2.9 Conclusion

Dans ce chapitre nous avons discuté de la turbulence atmosphérique en tant que phénomène physique, et on a parlé de ses caractéristiques et son influence sur la réponse d'un système optique.

Dans le prochain chapitre nous allons voir le dispositif DIMM et ses éléments constitutifs.

graphicx amsmath

Chapitre 3

Differential Image Motion Monitor

3.1 Introduction

La manière courante de caractériser la dégradation de l'image consiste à mesurer la largeur à mi-hauteur (FWHM) de l'Intensité d'une étoile (en arcsec) à la position focale, qui est en réalité son diamètre angulaire. La méthode qui est acceptée comme la plus précise est le moniteur de mouvement d'image différentielle, en anglais *Differential Image Motion Monitor*.

3.2 L'instrumentation



FIGURE 3.1: télescope complet avec sa monture, et la caméra CCD reliée

3.2.1 Télescope

Un télescope (du grec *tele* signifiant « loin » et *skopein* signifiant « regarder, voir ») est un instrument optique qui permet d'augmenter la taille apparente des objets observés et surtout leur luminosité. Son rôle d'amplificateur de lumière étant aussi, voire plus important que son grossissement optique, il permet d'apercevoir des objets célestes ponctuels difficilement perceptibles ou invisibles à l'œil nu. Les télescopes sont principalement utilisés en astronomie, car leurs réglages ne les rendent propices qu'aux observations d'objets très éloignés.

3.2.1.1 Éléments Constitutifs

Les instruments d'observation astronomique sont généralement constitués de deux systèmes optiques complémentaires : l'objectif et l'oculaire.

3.2.1.2 L'Objectif

Dans un télescope l'objectif est un miroir concave, le plus souvent parabolique. La face réfléchissante est située en avant, de sorte que la lumière ne traverse pas le verre qui sert uniquement de support à une pellicule d'aluminium de quelques centièmes de micromètres. La lumière étant simplement réfléchiée et non réfractée, contrairement à ce qui se passe dans une lunette astronomique, l'achromatisme des télescopes est total.

La lumière est ensuite focalisée en un point appelé foyer image. Le faisceau convergent peut être renvoyé vers un oculaire à l'aide d'un second miroir qui est plan dans le cas d'un télescope de Newton. Ce petit miroir provoque inévitablement une obstruction, c'est-à-dire une perte de luminosité ce qui n'est pas grave, mais aussi une légère perte de contraste sans gravité si elle ne dépasse pas 20%

3.2.1.2.1 La Monture

La monture est la partie mobile, celle qui permet d'orienter l'instrument. On distingue deux principaux types de montures :

- *altazimutale* : La monture altazimutale : Elle est constituée d'un axe vertical et d'un axe horizontal. Elle permet le suivi d'un astre car elle est équipée d'un moteur sur chacun de ses axes. Elle peut être pilotée par un ordinateur intégré dans le télescope, ou extérieur, avec positionnement automatique sur un astre sans mise en station, mais après indication de la position de deux étoiles en début de séance d'observation.
- monture équatoriale : L'usage de cette monture est rendue pratique en raison de la rotation de la sphère céleste. Elle permet de suivre le même astre en faisant pivoter l'instrument sur un seul axe, l'autre étant parallèle à l'axe de rotation de la terre. Pour cela, elle possède quatre axes dont deux permettent de régler. Les deux autres servant à orienter l'instrument selon les coordonnées célestes données par la déclinaison et l'ascension droite. C'est la monture généralement utilisée sur les télescopes.

3.2.1.3 Caractéristiques Techniques

- *Diamètre* : Le diamètre de l'objectif, en l'occurrence le miroir primaire, est la caractéristique la plus importante de l'instrument car la plupart des propriétés optiques de l'instrument en dépendent. Plus il est grand, plus il autorise de forts grossissements et permet d'observer des étoiles lointaines. Le diamètre est généralement exprimé en millimètre pour les instruments du commerce, quelquefois en pouces ($1'' = 25,4mm$).
- *La longueur focale* : Il peut s'agir de la longueur focale du miroir primaire ou de celle des oculaires. La longueur focale de l'instrument à proprement parler correspond à celle de l'objectif, elle est exprimée en millimètre ou bien doit être calculée à partir du rapport f/D .
- *Le rapport f/D* : Le rapport focale/diamètre est le rapport de la longueur focale du miroir primaire et de son diamètre, exprimés bien sûr dans la même unité. Un faible rapport f/D donne un instrument compact, donc stable et facile à manier et

transporter. Néanmoins, la précision de collimation croit comme $(D/f)^2$. En d'autres termes, un télescope ouvert à $f/D = 5$ sera deux fois plus difficile à collimater qu'un télescope ouvert à $f/D = 7$. Un rapport supérieur ou égal à $f/D = 5$ est très satisfaisant, au-delà de $f/D = 10$, l'instrument a un champ limité mais une faible obstruction, ce qui est favorable en planétaire. En outre, les oculaires pouvant être de focale plus longue, le recul d'œil et donc le confort sera meilleur. Pour faire de l'astrophotographie un f/D de 4 sera acceptable surtout si l'on améliore le champ avec un correcteur de Ross à deux lentilles [tel,].

le télescope qu'on a utilisé est le cELESTRON C11 avec un tube Schmidt-Cassegrain de diamètre 279,4 mm, une focale de 2800mm et une longueur de 610 mm.

3.2.2 Caméra CCD

Dès leur introduction, les caméras CCD ont très rapidement remplacé les plaques photographiques pour l'observation en astronomie. Plus sensibles, de réponse plus linéaire, fournissant un signal digitalisé, avec une réponse spectrale plus modulable.

Cette nouvelle technique mise au point en 1970 permet d'allier électronique et informatique pour un rendu extrêmement impressionnant. CCD signifie "Charge Coupled Device", ce qui veut dire en français "dispositif à transfert de charge". La caméra CCD n'est rien d'autre qu'un appareil photo dont on a remplacé la pellicule par un capteur électronique sensible à la lumière.

Le récepteur CCD est une plaque de silicium qu'on appelle aussi matrice, et qui ne fait que quelques millimètres de côtés mais qui est très sensible à la lumière. Lorsque cette plaque est frappée par un photon, des électrons sont libérés. En se déplaçant, ces électrons engendrent un signal électrique. La plaque de silicium, étant séparée en plusieurs milliers de pixels, on peut savoir où la plaque a reçu le photon et en fonction de l'intensité de ce signal, on en déduit la quantité de lumière que l'on doit renvoyer. Plus il y a de pixels, plus la définition de l'image finale sera grande. Toutes ces informations sont interprétées et transformées en langage numérique pour qu'un ordinateur puisse les lire.

La caméra utilisée est une DMK 51AU02.AS du constructeur Imaging Source. Elle

est dotée d'une puce CCD de 1600 x 1200 pixels et d'une plage dynamique de 8 bits. la sensibilité spectrale s'étend sur toute la plage visible avec un pic à une longueur d'onde de 500 nm (la valeur quantique), l'efficacité à 500 nm est de 50%. La taille de pixel est de $5.31 \mu\text{m} \times 5.66 \mu\text{m}$.

Le temps d'exposition est réglable à partir de $10 \mu\text{s}$ à plus de 60 mn, la cadence maximale est de 12 images par seconde en pleine résolution.



FIGURE 3.2: DMK 51AU02.AS

3.2.3 Le Masque

Le masque est destiné à être utilisé avec le télescope pour fournir deux images de la même étoile. Le masque standard est composé de deux trous en aluminium plat anodisé noir et découpé au laser. L'un des trous est équipé d'un prisme, permettant d'obtenir les deux images de la même étoile à une distance donnée, dans le champ de vision de la caméra. les caractéristiques du masque qu'on utilise sont :

- diamètre des deux ouvertures est : 50mm
- la séparation centre au centre entre les deux ouvertures est : 215 mm



FIGURE 3.3: Le masque avec double ouverture

3.3 La technique DIMM

La technique DIMM consiste à mesurer le mouvement relatif de deux images de la même étoile, et en se basant sur ce mouvement, on peut estimer le seeing atmosphérique ainsi que le paramètre de Fried. L'avantage de cette méthode est qu'elle élimine les mouvements erratiques du télescope, car elle mesure les différences angulaires entre les deux images et n'est affectée par aucun autre mouvement. L'idée est que la lumière provenant d'une étoile

passant par les deux ouvertures voyage dans des conditions atmosphériques légèrement différentes, produisant une inclinaison du front d'onde de l'une par rapport à l'autre. Pour obtenir les deux images au bon foyer, il est nécessaire d'utiliser un mince prisme optique sur l'une des ouvertures afin de produire une petite déviation de l'un des faisceaux.

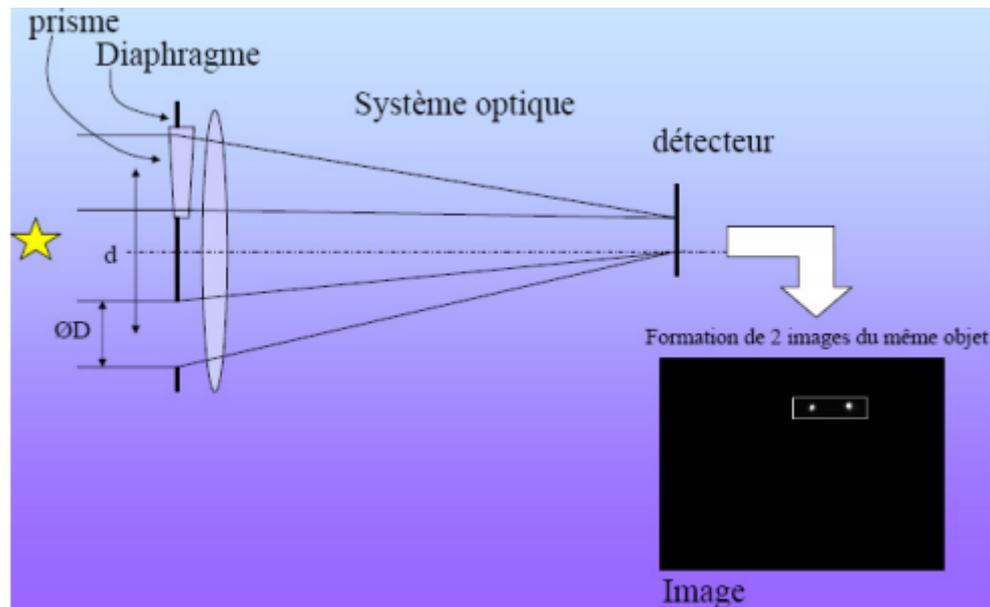


FIGURE 3.1: Explication du principe de l'utilité du prisme relié à l'une des ouvertures

3.3.1 Algorithme DIMM

3.3.1.1 Traitement de l'image

3.3.1.1.1 Le seuillage :

Pour appliquer la méthode DIMM, il faut d'abord passer par une phase de traitement de l'ensemble des expositions récupérées.

Cela est dû au fait que tous les pixels d'un capteur CCD n'ont pas le même gain de conversion, et ne possèdent pas la même sensibilité pour une luminance identique. Même une très faible variation de cette sensibilité entre deux pixels se voit nettement sur une image. On rappellera que le bruit de fond varie en fonction de la température du CCD ainsi que la durée d'exposition. Il faut voir ce bruit comme un certain nombre d'électrons générés en l'absence de lumière. Plus l'exposition est longue, plus il y aura d'électrons

généérés.

On peut comparer ce bruit à un offset (non identique pour chaque pixel), que l'on peut soustraire à l'image finale afin d'obtenir une image corrigée. Etant donné que pour notre application DIMM, nous réalisons des poses de très faible durée, le bruit de fond est relativement peu gênant pour le calcul de la position des divers centroïdes. Pour résoudre ce problème, chaque exposition doit passer par un seuillage qui consiste à soustraire un niveau de seuil à l'ensemble de l'image.

Si on ne soustrait pas le bruit à une image relativement bruitée, nous aurons tendance, lors du calcul du centroïde, à diminuer le poids des points qui définissent le spot. De ce fait, il se peut que la variation de position de ce dernier soit minimisée et donc nous n'en ressortons pas la bonne valeur. Le bruit peut noyer la véritable position du spot sur l'image. Par contre, il faut faire attention à ne pas avoir un niveau de seuil trop important, de façon à ce que l'information résultante soit éloignée de la réalité.

3.3.1.1.2 Calcul des Centroïdes

Le centre de gravité d'une forme est la moyenne arithmétique (c'est-à-dire la moyenne) de tous les points. Supposons qu'une forme se compose de n points distincts X_1, X_2, \dots, X_n alors le centre de gravité est donné par :

$$C_n = \frac{\sum_{i=1}^n X_i}{n} \quad (3.3.1)$$

Dans le contexte du traitement des images et de la vision par ordinateur, chaque forme est composée de pixels et le centre de gravité est simplement la moyenne pondérée de tous les pixels constituant la forme.

3.3.1.1.3 Méthodes de calcul du centroïde

- *Gaussian Fitting* : Une des méthodes de calcul du centroïde d'une étoile est l'algorithme d'ajustement gaussien (GF), car la distribution de l'intensité d'un point en étoile est conforme à la fonction gaussienne.

Le modèle d'imagerie du moniteur de mouvement d'image différentielle peut être formulé comme suit :

$$I_i = S_i + N_i \quad (3.3.2)$$

où I_i , S_i et N_i représentent respectivement l'intensité observée, la source, et l'intensité du bruit du pixel i .

l'intensité de la source de chaque pixel dans le point étoile peut être exprimée par une fonction Gaussienne :

$$S(X_i, Y_i|v) = A \exp\left(-\frac{(x_i - x_c)^2}{2\sigma_x^2} - \frac{(y_i - y_c)^2}{2\sigma_y^2}\right) \quad (3.3.3)$$

où X_i et Y_i représentent les coordonnées du pixel i , v représentent les paramètres gaussiens (A , x_c , y_c , σ_x , σ_y) tel que

A représente le niveau de luminosité de l'étoile,

x_c et y_c représentent le centroïde de l'étoile Y ,

et σ_y représente l'écart type de la fonction.

Le Gaussien fitting estime le centroïde en ajustant l'intensité observée d'une étoile à la fonction gaussienne. La fonction objectif du GF est :

$$Z = \underset{v}{\operatorname{Argmin}} \sum_{i \in U} [Z_i]^2 \quad (3.3.4)$$

avec $Z_i = I(X_i, Y_i) - S(X_i, Y_i|v)$ et U un ensemble formé par les pixels dans la tâche étoile.

Selon la théorie de l'erreur, le centroïde en étoile estimé à l'aide de l'équation précédente est l'estimation optimale au sens du maximum de vraisemblance lorsque

le bruit de chaque pixel est conforme à une distribution Gaussienne et qui peut généralement être satisfaite pour les images en étoile.

- *Methode des moments* : Le moment d'une image est une certaine moyenne pondérée des intensités des pixels de l'image, ou une fonction de tels moments, généralement choisie pour avoir une propriété ou une interprétation intéressante.

Pour $i, j = 0, 1, 2, \dots$ et adaptation de ceci aux images scalaires (niveaux de gris) avec des intensités de pixels $I(x, y)$, les moments bruts de l'image M_{ij} sont calculés comme suit :

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \quad (3.3.5)$$

Le barycentre est donné par la formule suivante :

$$centroid(x, y) = \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \quad (3.3.6)$$

3.3.1.1.4 Choix de la méthode de calcul

La complexité informatique de l'ajustement gaussien est trop élevée pour les applications temps réel, ainsi la solution du système est un problème de carrés non linéaires. Des algorithmes complexes sont nécessaires pour le résoudre. Ces algorithmes partent généralement d'une hypothèse initiale des paramètres et s'approchent de la solution optimale dans une procédure d'itérations multiples qui prend beaucoup de temps. D'autre part, une estimation initiale incorrecte des paramètres peut faire converger les algorithmes vers des paramètres optimaux locaux, ce qui entraîne une instabilité des résultats.

La deuxième méthode est donc plus adaptée aux exigences du temps-réel car le calcul va être très rapide par rapport au GF ce qui nous permet de travailler avec une grande cadence.

3.3.1.2 Calcul du seeing

Le code DIMM est basé sur les équations mathématiques utilisées par ESO / DIMM (Sarazin Roddier 1990). Les paramètres qui doivent être spécifiés dans le code sont :

1. Le diamètre des deux trous du masque (D)
2. La séparation des deux trous centre à centre (d)
3. La taille en pixels de la combinaison télescope / caméra CCD dans les deux dimensions x , y (en arcsec)

La partie fondamentale consiste à calculer exactement les centres des deux images de l'étoile dans chaque exposition. Les distances angulaires Δx et Δy du centre de gravité de chaque image en étoile sont calculées dans les deux sens (parallèle à la base des ouvertures). Les variances longitudinale et transversale (σ_l et σ_t respectivement) de Δx et Δy sont estimés sur une séquence de N captures instantanées (N étant typiquement plusieurs centaines). Le Seeing (en radian) est calculé en utilisant les formules suivantes :

$$\epsilon_l = 0,98 \left(\frac{D}{\lambda}\right)^{0.2} \left(\frac{(\sigma_l)^2}{K_l}\right)^{0.2} \quad (3.3.7)$$

$$\epsilon_t = 0,98 \left(\frac{D}{\lambda}\right)^{0.2} \left(\frac{(\sigma_t)^2}{K_t}\right)^{0.2} \quad (3.3.8)$$

Avec :

$$K_l = 0,364 \cdot (1 - 0.532b)^{-\frac{1}{3}} - 0,024b^{-\frac{7}{3}} \quad (3.3.9)$$

$$K_t = 0,364 \cdot (1 - 0.798b)^{-\frac{1}{3}} - 0,018b^{-\frac{7}{3}} \quad (3.3.10)$$

Tel que : ϵ_l est le seeing longitudinal (calculé selon les abscisses x)

ϵ_t est le seeing transversal (calculé selon les ordonnées y)

d est la distance entre les sous-ouvertures, $b = d/D$

λ la longueur d'onde, définie traditionnellement à 500 nm comme norme.

3.4 Conclusion

A travers ce qu'on a cité ci-dessus, il est clair que le logiciel qu'on doit concevoir est un outil dédié pour l'étude de la turbulence atmosphérique "seeing" .

Dans la prochaine partie, nous allons parler sur l'aspect technique du projet et on termine par la présentation des interfaces graphiques du logiciel.

Deuxième partie

Implémentation

Chapitre 4

Modélisation Conceptuelle et Organisationnelle

4.1 Introduction

Dans ce chapitre nous allons décrire les logiciels et les langages de programmation utilisés, qui nous ont permis la réalisation de ce travail. Pour l'implémentation, nous avons utilisé le framework Qt5 et le langage de programmation Python. Nous allons aussi identifier le diagramme de cas d'utilisation réalisé pour mettre en œuvre l'architecture proposée et les scénarios des principaux cas d'utilisation.

La motivation fondamentale de la modélisation est de fournir une démarche antérieure afin de réduire la complexité du système étudié lors de la conception, ainsi que d'organiser la réalisation du projet en définissant les modules et les étapes de la réalisation.

4.2 L'environnement de programmation Qt

Qt est un framework orientée objet qui est développée en C++ conjointement par The Qt Company et Qt Project. Qt offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. Par certains aspects, elle ressemble à un framework lorsqu'on l'utilise pour concevoir des interfaces graphiques ou que l'on conçoit l'architecture de son application en utilisant les mécanismes des signaux et slots par exemple.

Qt permet la portabilité vers différents systèmes d'exploitation des applications qui n'utilisent que ses composants par simple recompilation du code source. Les environnements supportés sont les Unix (dont GNU/Linux) qui utilisent le système graphique X Window System ou Wayland, Windows de Microsoft, Mac OS X, Tizen et également Genode.

Qt supporte des bindings avec plus d'une dizaine de langages autres que le C++, comme Ada, C, Java, Python, Ruby, Visual Basic, etc [qt,].

Qt Development Frameworks fournit un ensemble de logiciels libres pour faciliter le développement d'applications Qt :

- Qt Designer est un concepteur d'interface graphique, il enregistre les fichiers ".ui".
- Qt Assistant permet de visualiser la documentation complète de Qt hors-ligne
- Qt Linguist est un outil destiné aux traducteurs, il leur permet d'ouvrir les fichiers

".ts" qui contiennent les chaînes de caractères à traduire, et d'entrer ensuite leur traduction.

- Qt Creator est l'environnement de développement intégré destiné à Qt et facilite la gestion d'un projet Qt. Son éditeur de texte offre les principales fonctions que sont la coloration syntaxique, le complètement, l'indentation, etc. Qt Creator intègre en son sein les outils Qt Designer et Qt Assistant .
- Qt Demo présente un ensemble de petits programmes (sous forme d'exécutables et de sources) qui montrent les fonctionnalités offertes par la bibliothèque.
- Qt Installer Framework est un outil permettant de créer facilement des programmes d'installations.

4.2.1 Qt Designer

Qt Designer est l'outil Qt pour la conception et la construction d'interfaces utilisateur graphiques (GUI) avec Qt Widgets. Nous pouvons composer et personnaliser nos fenêtres ou nos boîtes de dialogue de la manière que nous voulons, et les tester à l'aide de différents styles et résolutions. Les widgets et les formulaires créés avec Qt Designer s'intègrent de manière transparente au code programmé, à l'aide du mécanisme de signaux et d'emplacements de Qt, afin que nous puissions facilement attribuer un comportement aux éléments graphiques. Toutes les propriétés définies dans Qt Designer peuvent être modifiées dynamiquement dans le code. En outre, des fonctionnalités telles que la promotion des widgets et des plug-ins personnalisés nous permettent d'utiliser nos propres composants avec Qt Designer. Nous avons la possibilité d'utiliser Qt Quick pour la conception d'interface utilisateur plutôt que pour les widgets. C'est un moyen beaucoup plus simple d'écrire de nombreux types d'applications. Il permet une apparence entièrement personnalisable, des éléments tactiles et des transitions animées fluides, renforcés par la puissance de l'accélération graphique OpenGL.

4.2.2 La version Qt5

Qt 5 est la dernière version de Qt. Il permet aux développeurs de développer des applications avec des interfaces utilisateur intuitives pour plusieurs systèmes d'exploitation, plus rapidement. Qt 5 facilite le traitement des derniers changements de paradigme d'interface utilisateur que les écrans tactiles et les tablettes nécessitent. On peut donc être plus productif et plus flexible grâce à la prise en charge améliorée de Python et de QML dans Qt 5, tout en conservant la prise en charge de la fondation C++ et du widget Qt.

4.3 Choix du Langage de Programmation

Pour implémenter notre logiciel, le langage de programmation Python est le mieux adapté. En effet, Python s'annonce comme une des évolutions majeures de la programmation. Pour la première fois, un langage efficace, performant, standard et facile à apprendre est disponible.

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions.

Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Il est également apprécié par certains pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas niveau, permet une initiation aisée aux concepts de base de la programmation.

4.3.1 Caractéristiques du Langage Python

- La syntaxe de Python est très simple et, combinée à des types de données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles. À fonctionnalités égales, un programme Python (abondamment commenté et présenté selon les canons standards) est souvent de 3 à 5 fois plus court qu'un

programme C ou C++ (ou même Java) équivalent, ce qui représente en général un temps de développement de 5 à 10 fois plus court et une facilité de maintenance largement accrue.

- Python gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du programmeur, par un mécanisme de comptage de références (proche, mais différent, d'un ramasse-miettes).
- Python est orienté-objet. Il supporte l'héritage multiple et la surcharge des opérateurs. Dans son modèle objets, et en reprenant la terminologie de C++, toutes les méthodes sont virtuelles.
- Python est dynamique (l'interpréteur peut évaluer des chaînes de caractères représentant des expressions ou des instructions Python), orthogonal (un petit nombre de concepts suffit à engendrer des constructions très riches), réflexif (il supporte la métaprogrammation, par exemple la capacité pour un objet de se rajouter ou de s'enlever des attributs ou des méthodes, ou même de changer de classe en cours d'exécution) et introspectif (un grand nombre d'outils de développement, comme le debugger ou le profiler, sont implantés en Python lui-même).
- Comme Scheme ou SmallTalk, Python est dynamiquement typé. Tout objet manipulable par le programmeur possède un type bien défini à l'exécution, qui n'a pas besoin d'être déclaré à l'avance.

4.3.2 Python et la Programmation Orientée Objet

Chaque langage de programmation appartient à une "famille" de langages définissant une approche ou une méthodologie générale de programmation. Par exemple, le langage C est un langage de programmation procédurale car il suppose que le programmeur s'intéresse en priorité aux traitements que son programme devra effectuer. Un programmeur C commencera par identifier ces traitements pour écrire les fonctions qui les réalisent sur des données prises comme paramètres d'entrée.

En Python, la liberté est plus grande. Des blocs d'instructions manipulant des données. C'est ce qu'on appelle la méthode de programmation orientée procédure. Il existe un autre moyen d'organiser un programme, qui consiste à combiner des données et des fonctionnalités

et à les envelopper dans un objet. C'est ce qu'on appelle le paradigme de la programmation orientée objet. La plupart du temps, nous pouvons utiliser la programmation procédurale, mais lorsque nous écrivons des programmes lourds ou rencontrons un problème mieux adapté à cette méthode, nous pouvons utiliser des techniques de programmation orientée objet.

Les classes et les objets sont les deux aspects principaux de la programmation orientée objet. Une classe crée un nouveau type où les objets sont des instances de la classe. Une analogie est que vous pouvez avoir des variables de type " int ", ce qui signifie que les variables qui stockent des entiers sont des variables qui sont des occurrences (objets) de la classe "int". Les objets peuvent stocker des données à l'aide de variables ordinaires appartenant à l'objet. Les variables qui appartiennent à un objet ou à une classe sont appelées FIELDS.

Les objets peuvent également avoir des fonctionnalités en utilisant des fonctions appartenant à une classe. Ces fonctions sont appelées méthodes de la classe. Cette terminologie est importante car elle nous aide à différencier les fonctions et les variables indépendantes de celles appartenant à une classe ou à un objet. Ensemble, les champs et les méthodes peuvent être appelés attributs de cette classe. Les champs sont de deux types : ils peuvent appartenir à chaque instance / objet de la classe ou à la classe elle-même. On les appelle respectivement variables d'instance et variables de classe [pyt,].

4.3.3 Python et la Programmation Temps-réel

En informatique, on parle d'un système temps réel lorsque ce système est capable de contrôler (ou piloter) un procédé physique à une vitesse adaptée à l'évolution du procédé contrôlé. La programmation temps réel se différencie à des autres programmations par la prise en compte de contraintes temporelles dont le respect est aussi important que l'exactitude du résultat, autrement dit elle ne doit pas simplement délivrer des résultats exacts, elle doit les délivrer dans des délais imposés. Pour garantir le respect de limites ou contraintes temporelles, il est nécessaire que les différents services et algorithmes utilisés s'exécutent en temps borné mais peuvent être interrompus par un processus plus prioritaire et que les différents enchaînements possibles des traitements garantissent que ceci ne dépasse pas les contraintes temporelles.

Faire en sorte que Python fonctionne dans des situations en temps réel où les délais d'exécution sont courts peut être délicat. La philosophie de Python en matière de vitesse est fondamentalement la suivante : «si elle fonctionne suffisamment bien, alors tout va bien». la programmation python en temps réel est utilisée dans plusieurs domaines en voici quelques exemples :

- Systèmes embarqués.
- Machine Learning.
- Analyse des données.
- Développement d'interface graphique.
- Développement de jeu.
- Informatique scientifique et numérique.
- Exploration Web et mise au rebut Web.

Python s'est avéré si puissant et facile à utiliser qu'il parvient même à se frayer un chemin dans les systèmes temps réel en tant que langage de programmation. Il existe de nombreuses bibliothèques libres et puissantes pour développer des applications Python. Python peut être utilisé pour recevoir des données issues d'un système critiques qui peuvent ensuite être stockées dans une base de données ou localement pour analyse. on peut ensuite ensuite utiliser Python pour développer des visualisations en temps réel présentant des paramètres critiques, ou pour stocker et sauvegarder ces paramètres en vue d'une analyse ultérieure. La bonne partie de l'analyse des données à l'aide de Python est que le travail préparatoire a déjà été fait. La fonctionnalité est simplement en baisse.

4.4 Les Bibliothèques

4.4.1 OpenCV

OpenCV (Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société

de robotique Willow Garage et la société ItSeez se sont succédé au support de cette bibliothèque. Depuis 2016 et le rachat de ItSeez par Intel, le support est de nouveau assuré par Intel [ope,].

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques. Elle propose la plupart des opérations classiques en traitement bas niveau des images comme :

- Lecture, écriture et affichage d'une image.
- Calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs.
- Lissage, filtrage.

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

- Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra).
- Détection de droites, de segment et de cercles par Transformée de Hough.

4.4.1.0.1 Le Module OpenCV Python

Une caractéristique importante de Python est qu'il peut être facilement étendu avec C / C ++. Cette fonctionnalité nous aide à écrire des codes gourmands en calculs en C / C ++ et à créer un "wrapper" Python pour pouvoir les utiliser en tant que modules Python. Cela nous donne deux avantages : premièrement, notre code est aussi rapide que le code original C / C ++ (puisque'il s'agit du code C ++ réel travaillant en arrière-plan) et deuxièmement, il est très facile de coder en Python. C'est ainsi que fonctionne OpenCV-Python, c'est un "wrapper" Python autour de l'implémentation C ++ originale.

4.4.2 Bibliothèque Numpy

Numeric est un paquet développé à l'origine par Jim Hugunin. C'était l'ancêtre de NumPy, une librairie Python et un projet open-source créé par Travis Oliphant, contraction de

Numerical Python. Travis a conçu NumPy en incluant les caractéristiques du paquet Numarray de Numeric [ope,].

Numpy est une bibliothèque numérique apportant le support efficace de larges tableaux multidimensionnels, et de routines mathématiques de haut niveau (fonctions spéciales, algèbre linéaire, statistiques, etc.). Elle fournit une interface pour stocker et effectuer des opérations sur les données. D'une certaine manière, les tableaux Numpy sont comme les listes en Python, mais Numpy permet de rendre les opérations beaucoup plus efficaces, surtout sur les tableaux de large taille. Les tableaux Numpy sont au cœur de presque tout l'écosystème de data science en Python. Elle fournit des techniques pour charger, stocker et manipuler efficacement les données. Elles peuvent venir de sources très variées, mais nous pouvons toujours les considérer comme des tableaux de nombres. Par exemple, une image peut être considérée comme un tableau de deux dimensions (une matrice) où chaque nombre représente l'intensité lumineuse d'un pixel.

4.4.2.0.1 OpenCV et Numpy

Numpy est une bibliothèque hautement optimisée pour les opérations numériques. Il donne une syntaxe de style MATLAB. Toutes les structures de tableau OpenCV sont converties en une structure numpy à partir de tableaux Numpy. Ainsi, quelles que soient les opérations que nous pouvons effectuer dans Numpy, nous pouvons les combiner avec OpenCV, ce qui augmente le nombre d'armes dans notre arsenal. En plus de cela, plusieurs autres bibliothèques comme SciPy, Matplotlib qui supporte Numpy peuvent être utilisées avec cela.

4.4.3 Bibliothèque Matplotlib

Matplotlib est une idée originale de John Hunter (1968-2012) qui, avec ses nombreux contributeurs, a consacré une quantité incalculable de temps et d'efforts à la création d'un logiciel utilisé par des milliers de scientifiques du monde entier [mat,]. C'est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy.

Plusieurs points rendent cette bibliothèque intéressante :

- Export possible en de nombreux formats matriciels (PNG, JPEG...) et vectoriels (PDF, SVG...)
- Forte communauté très active.
- Interface pylab : reproduit fidèlement la syntaxe MATLAB.
- Bibliothèque haut niveau : idéale pour le calcul interactif.

Matplotlib possède un module appelé pyplot qui facilite le traçage en fournissant une fonctionnalité permettant de contrôler les styles de ligne, les propriétés de police, les axes de formatage, etc. Il prend en charge une très grande variété de graphiques et de tracés, à savoir - histogrammes, diagrammes à barres, spectres de puissance, diagrammes d'erreur, etc. Il est utilisé avec NumPy pour fournir un environnement qui constitue une alternative efficace au code source ouvert pour MatLab. Il peut également être utilisé avec des outils graphiques tels que PyQt et wxPython.

4.5 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisations représentent un intérêt pour l'analyse des besoins métier ce qui nous permettra de démarrer l'analyse orientée objet. Le diagramme suivant représente le cas d'utilisation de notre logiciel. Il décrit le comportement du système du point de vue utilisateur.

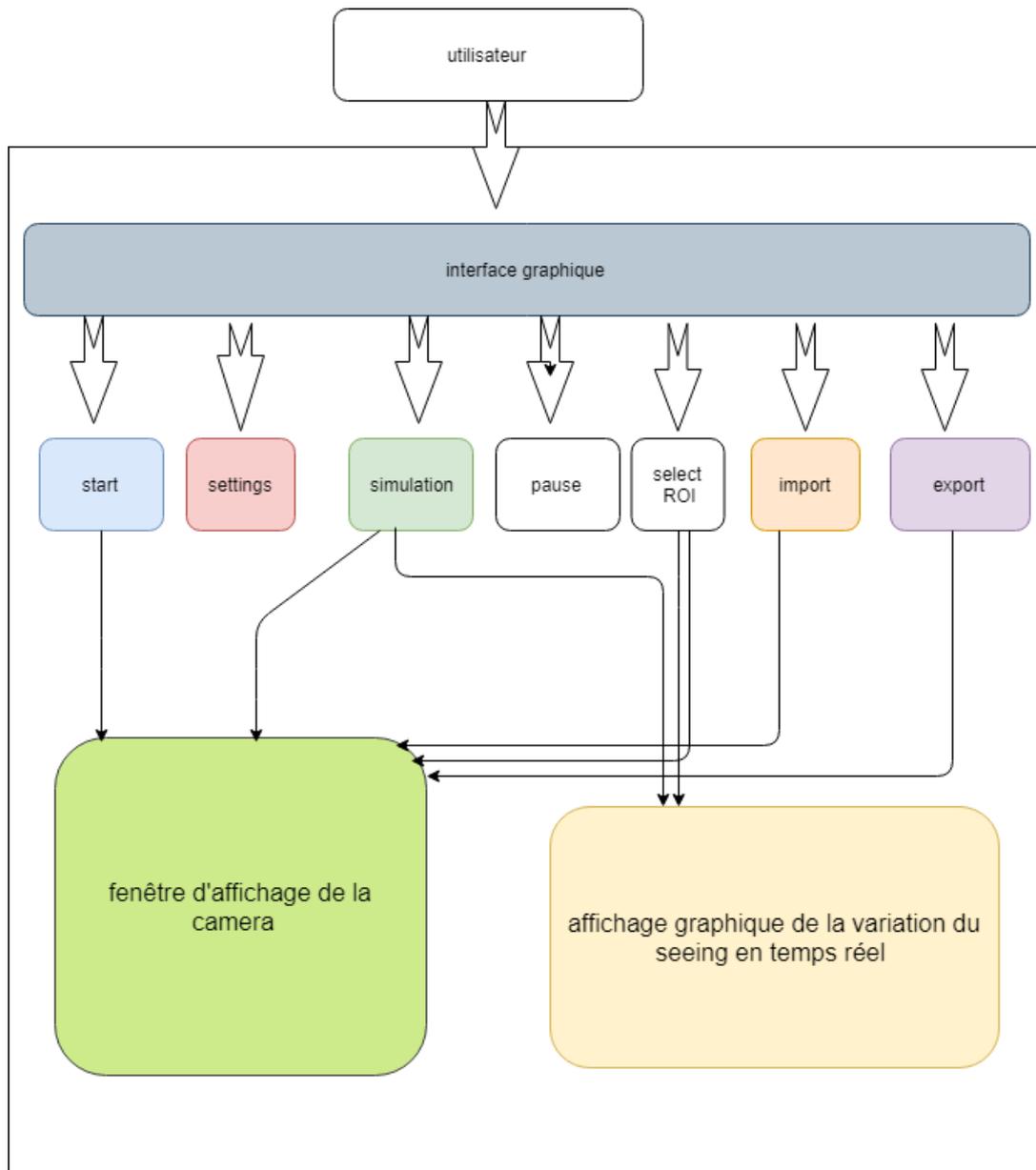


FIGURE 4.1: Logique de fonctionnement du logiciel

4.6 Description des Scénarios

Dans le but de mieux comprendre notre système et l'interaction avec l'utilisateur, dans cette partie nous allons détailler les scénarios de principaux cas d'utilisation.

4.6.1 Cas de la commande START

Dans ce cas une nouvelle fenêtre va apparaître à l'utilisateur pour sélectionner le périphérique de capture vidéo ainsi le format de la vidéo et le FPS (Frame Per Second) du périphérique.

Une fois terminé le logiciel commence l'acquisition des données. L'utilisateur peut dans ce cas lancer le calcul du seeing.

4.6.2 Cas de la commande Select ROI

Cette commande (qui veut dire 'Select Regions Of Interest') permet de sélectionner les zones où se trouvent les deux projections de l'étoile.

4.6.3 Cas de la commande SETTINGS

Pour régler les paramètres d'acquisition des images de la caméra on distingue deux états :

- Le périphérique n'est pas encore sélectionné : l'apparition d'une nouvelle fenêtre dans laquelle on demande à l'utilisateur la sélection de l'appareil en premier temps.
- Le périphérique est sélectionné : l'ouverture d'un dialogue contenant les propriétés de la caméra, dont l'utilisateur peut changer les valeurs ou bien les réinitialiser par les valeurs par défauts.

4.6.4 Cas de l'instruction SIMULATION

Ce service est donné à l'utilisateur pour pouvoir simuler des images acquises par la caméra en utilisant un générateur aléatoire de turbulence atmosphérique.

En premier, une image vidéo similaire à celle issue du DIMM apparaît dans la fenêtre d'affichage contient deux projections d'une étoile qui bougent au cours du temps. Ce qui nous donne une simulation du phénomène de la turbulence atmosphérique.

A partir de cette simulation on calcul le seeing des deux étoiles virtuelles et on affiche sa variation au cours du temps.

4.6.5 Cas de la commande IMPORT

Cette commande nous a permet la lecture d'une vidéo à partir d'un fichier. La vidéo doit avoir un format spécifié (parmi les formats proposés). Si cette vidéo est un enregistrement d'une observation avec un DIMM, dans ce cas le logiciel calcule le seeing et affiche son évolution au cours du temps.

4.6.6 Cas de la commande EXPORT

On l'utilise lorsque nous capturons donc une vidéo, la traitons image par image et nous voulons la sauvegarder. Cette fois, nous créons un objet « Video Writer ». Nous devrions spécifier le nom du fichier de sortie. Cette tâche peut être utilisée soit au cours d'une observation réelle ou une simulation.

4.6.7 La commande PAUSE/RESUME

Cette commande provoque un arret temporaire de la capture, soit en observation réelle ou une simulation. Dans ce cas le logiciel ne reçoit plus des données à partir du périphérique, ainsi le calcul du seeing va être en pause. Un autre clique sur le même bouton relance l'acquisition des données, le traitement des expositions successives et l'affichage du résultat.

4.7 Conclusion

Dans ce chapitre nous avons décrit l'environnement de travail de notre projet. Aussi on a justifié nos choix d'un point vue technique. Le choix des outils de développement influence énormément sur le coût en temps de programmation, ainsi que sur la flexibilité du produit à réaliser.

On a fini par identifier le diagramme de cas d'utilisation et les scénarios des principaux

cas d'utilisation pour faciliter la réalisation de notre prototype.

Dans le chapitre suivant, nous montrerons les étapes, plus en détails, que nous avons suivies pour implémenter et réaliser notre logiciel.

Chapitre 5

Réalisation & Résultats

5.1 Introduction

Il est évident que les méthodes et les outils choisis pour concevoir et développer une application doivent être en fonction de l'environnement et du domaine d'application de celle-ci. On consacra ce chapitre à la présentation et l'analyse des classes et méthodes de programmation utilisées. Nous présenterons également quelques interfaces graphiques du logiciel réalisé.

5.2 La classe TIS_CAM

C'est l'une des classes définies lors de notre approche orientée objet. Elle contient l'ensemble des fonctions qui permettent à l'utilisateur l'interaction avec les caméras existantes, l'extraction des propriétés de la caméra et le réglage des paramètres. Cette classe est basée sur le fichier *tisgrabber.dll* qui est une bibliothèque de liens dynamiques qui contient du code et des données pouvant être utilisés simultanément par les différentes classes.

5.2.1 Méthodes Principales Constituant la classe TIS_CAM

- Chercher les cameras liées à l'ordinateur "GetDevices" : Le rôle de cette fonctions et de trouver tous les caméras liées à l'ordinateur et les identifier avec nom et numéro de série.
- Afficher la boîte de dialogue de sélection du périphérique "ShowDeviceSelectionDialog()" : Elle donne la main à l'observateur de voir le nom de tous les périphériques qui sont liées à l'ordinateur et de sélectionner une seule caméra selon son nom ou bien son numéro de série.
- Vérifier la disponibilité du périphérique choisis "IsDevValid" : Une fois la caméra sélectionnée, cette fonction vérifie si elle est en état de repos, c'est-à-dire qu'elle n'est pas liées à autre tache.

- Définir le format vidéo "SetVideoFormat" : On l'utilise pour choisir le format de la vidéo. A chaque format de fichier vidéo va correspondre un codec particulier pour réaliser la compression/ décompression du fichier. Chaque format a une limite en terme de taille d'image par pixels et de cadence.
- Récupérer les propriétés de la camera "GetCameraProperty" : Le role de cette fonction est de lire les propriétés de la caméra à chaque instant donné. Il faut savoir que juste après le choix de la caméra elle renvoie les valeurs par défauts.
- Afficher la boîte de dialogue des propriétés de la camera "ShowPropertyDialog()" : Cette fonction utilise les propriétés que la fonction "GetCameraProperty" nous renvoie, les affiche, et permet à l'utilisateur de changer ces propriétés en tenant en compte la capacité et les limites de la caméra.
- Changer les propriétés de la camera "SetCameraProperty" : Cette fonction est liée directement à la précédente. Elle s'occupe de lire les valeurs changées par l'utilisateur à chaque ouverture de la boîte des propriétés pour reconfigurer la caméra.
- Obtenir la description de l'image "GetImageDescription" : Elle renvoie des informations sur les types de trame qu'on va récupérer comme la longueur et la largeur de l'image, le nombre de bits par pixel et l'espace colorimétrique de l'image.
- Afficher la vidéo "GetImage" : Elle prend en considération les valeurs données par la fonction "GetImageDescription" qui représentent les caractéristiques des images de la vidéo et l'affiche dans la fenêtre d'affichage qu'on doit spécifier ultérieurement.

5.3 Acquisition des données

Pour récupérer les données en temps réel à partir de la camera CCD, on a créé la fonction STARTLIVE de la classe TIS_CAM. Son argument peut être l'index d'un périphérique ou le nom d'un fichier vidéo. L'index de périphérique est le numéro permettant de spécifier la caméra. Normalement, dans notre cas une seule caméra sera connectée. Donc on passera simplement 0 (ou -1). Il est également possible de sélectionner une deuxième caméra en passant 1 et ainsi de suite. Après cela on capte des trames à partir des vidéo sur lesquelles

nous allons faire notre traitement.

5.4 Le Contrôle du Flux de Données par le Mode d'Interruption

Une interruption est un arrêt temporaire de l'exécution normale d'un programme par le microprocesseur afin d'exécuter un autre programme. Le mode d'interruption peut être relativement simple, permettant de faire interrompre par exemple à la fin d'une opération d'entrée-sortie, ou lorsqu'un dispositif périphérique est prêt à exécuter ou achever une Instruction, ou encore lorsque une erreur arithmétique s'est produite (on parle alors d'exception). Ce mode peut être plus complexe et peut traiter les diverses interruptions par ordre de priorité.

Dans les cas ci-dessus, le CPU peut être inutilement occupé. Cette situation peut très bien être évitée en utilisant une méthode basée sur des interruptions pour le transfert de données. En utilisant une fonction d'interruption et des commandes spéciales pour informer l'interface de l'émission d'un signal de demande d'interruption chaque fois que des données sont disponibles à partir de n'importe quel périphérique. Entre-temps, la CPU peut procéder pour toute autre exécution du programme. Pendant ce temps, l'interface surveille toujours l'appareil. Lorsque le périphérique est prêt pour un transfert de données, il envoie un signal de requête d'interruption à l'ordinateur. Lors de la détection d'un signal d'interruption externe, la CPU arrête momentanément la tâche déjà en cours, se branche sur le programme de service pour traiter le transfert des données, puis revient à la tâche qu'il était en train d'exécuter.

5.4.1 La fonction SnapImage

Elle s'occupe de capter les trames en tenant en compte de la cadence de la caméra et du temps d'exposition. Elle renvoie un booléen (vrai / faux). Si l'image est lue correctement, ce sera vrai. On peut donc vérifier la fin de la vidéo en vérifiant cette valeur de retour. La fonctionnalité d'interruption permet de recueillir les images à chaque fois que la caméra

termine la capture. Elle est mise en place à travers une fonction de type *callback* qui s'active à la fin de chaque capture. Cette fonction *callback* fonctionne sur un thread séparé du thread principal du logiciel pour ne pas bloquer celui-ci et pour avoir une rapidité maximale en temps de réponse .

L'acquisition des données nécessite en premier la sélection de la caméra à partir du dialogue de sélection d'appareil et l'initialisation des propriétés de la caméra.

5.5 Traitement d'image

Une fois les trames récupérées, un prétraitement doit être fait avant de passer à la phase de calcul de seeing, cela correspond à la conversion de l'image en un autre espace colorimétrique, le thresholding et la detection des étoiles.

5.5.1 Convertir l'image en niveaux de gris

Pour ce faire, nous devons appeler la fonction *cvtColor*. La fonction convertit une image d'entrée d'un espace colorimétrique à un autre. En cas de transformation de l'espace colorimétrique RGB, l'ordre des canaux doit être spécifié explicitement (RGB ou BGR). Le format de couleur par défaut dans OpenCV est BGR.

En tant que première entrée, cette fonction reçoit l'image d'origine. En deuxième entrée, elle reçoit le code de conversion d'espace colorimétrique. Puisque nous voulons convertir notre image d'origine de l'espace colorimétrique BGR en gris, nous utilisons le code `COLOR_BGR2GRAY`. La fonction nous renvoie une image de sortie de la même taille et profondeur que l'image d'origine.

5.5.2 Thresholding

Le seuillage est la méthode la plus simple de segmentation d'image. Il s'agit d'une opération non linéaire qui convertit une image en niveaux de gris en une image binaire dans laquelle les deux niveaux sont attribués à des pixels inférieurs ou supérieurs à la valeur de seuil spécifiée. En d'autres termes, si la valeur de pixel est supérieure à une valeur de seuil, une

valeur lui est affectée (elle peut être blanche), sinon une autre valeur est attribuée (peut être noire). Dans OpenCV, nous utilisons la fonction `cv2.threshold()`

Cette fonction applique un seuillage de niveau fixe. Elle est généralement utilisée pour extraire une image à deux niveaux (binaire) d'une image en niveaux de gris (on peut aussi l'utiliser à cet effet) ou pour supprimer un bruit, c'est-à-dire filtrer les pixels trop petits ou trop grands valeurs. Les paramètres de la fonction sont l'image source, qui devrait être une image en niveaux de gris, valeur de seuil '*Threshold*' qui est utilisé pour classer les valeurs de pixel et type de seuillage. Elle renvoie la valeur de seuil calculée et l'image de seuil. Il existe plusieurs types de seuillage pris en charge par la fonction nous avons utilisé le type `THRESH_TOZERO` dont la logique est la suivante :

$$dst(x, y) = \begin{cases} src(x, y) & \text{si } src(x, y) \text{ est } > \text{ à } Threshold \\ 0 & \text{sinon} \end{cases} \quad (5.5.1)$$

Avec : $src(x, y)$ étant l'intensité du pixel définie par les coordonnées x et y dans l'image source

$dst(x, y)$ étant l'intensité du pixel définie par les coordonnées x et y dans l'image de sortie.

pour le calcul de la valeur seuil (`Threshold`) on donne la main à l'utilisateur pour sélectionner un carré vide sur la fenêtre d'affichage et le seuil sera donc la moyenne des intensités des pixels sélectionnés.

5.5.3 Détection des Etoiles

Une fois le fond de l'image supprimé, on peut facilement chercher les deux projections de l'étoile récupérées par le DIMM, Celles-ci correspondent aux groupes de pixels connectés dans l'image qui partagent une propriété commune qui est une valeur en niveaux de gris non nulle. Pour cela on a utilisé la fonction `findcontours()`. Cette fonction cherche les points continus ayant la même caractéristique d'un point de vue couleur ou intensité. Cette fonction a trois paramètres, le premier est l'image source, le deuxième est le mode de récupération des contours, le troisième est la méthode d'approximation des contours. Et elle renvoie en sortie les contours et la hiérarchie. Les contours est une liste Python de tous les contours de l'image. Chaque contour individuel est un tableau Numpy de

coordonnées (x, y) des points de l'objet.

5.6 Calcul du seeing

La partie fondamentale du programme consiste à calculer exactement les centroïdes des deux étoiles dans chaque exposition. Après certain nombre d'expositions ultérieures, toute variation entre les deux étoiles est calculée dans les deux directions (alignement parallèle et perpendiculaire à l'ouverture), puis après un nombre spécifique d'expositions, la variance longitudinale $\sigma(l)$ et transversale $\sigma(t)$ du mouvement différentiel de l'image peut être déduite.

5.6.1 Barycentre d'une étoile

Dans le contexte du traitement des images, le centre de gravité est simplement la moyenne pondérée de tous les pixels constituant la forme de l'objet. Donc le calcul des barycentres revient à calculer la moyenne pondérée sur les pixels de chacun des contours détectés.

5.6.1.1 Calcul des barycentres par la méthode des moments

Pour le calcul des moments on a utilisé la méthode des moments de la bibliothèque OpenCV. Cette fonction calcule les moments, jusqu'au 3ème ordre, d'une forme vectorielle ou d'une forme pixellisée.

En appliquant cette fonction sur les points qui définissent chaque étoile qu'on a trouvé auparavant et en utilisant la formule précédente on obtient les coordonnées des barycentres pour les deux étoiles. Après on calcule les différentielles des barycentres selon x et y qui vont être enregistré dans des conteneurs de type liste avec des ajouts rapides. L'étape prochaine consiste à calculer la variance sur les différentielles des barycentres selon x (σ_x) et y (σ_y) en utilisant la fonction *numpy.std*. Une fois obtenue, on applique la formule donnée dans les chapitres précédents (voir eq.5 [Tokovinin, 2002]).

5.7 Repérer les étoiles sur l'interface graphique

Cette phase permet de suivre en temps réel la fluctuation de la position de l'étoile qui est due à la turbulence atmosphérique, et ainsi de vérifier le bon fonctionnement de la méthode de détection des étoiles en prenant en considération le résultat du calcul précédent (calcul des barycentres). La fonction `drawMarker` qu'on a utilisé dessine un marqueur sur une position prédéfinie dans une image. Elle nécessite comme paramètre l'image récupérée et les coordonnées des deux barycentres.

5.8 Affichage graphique de la variation du seeing en temps-réel

Le tracé en temps réel du résultat de notre calcul est un outil puissant pour analyser les données lors de leur transmission dans le système d'acquisition. Pour créer un tracé en temps réel, on a utilisé un module d'animation de matplotlib (`FigureCanvasQTAgg`). Ce module sert à automatiser la mise à jour du graphique. La fonction chargée de cette tâche est `plot` dont le fonctionnement est le suivant : chaque fois que l'on obtient une valeur du seeing issue d'une nouvelle exposition, la fonction prend cette valeur et le temps auquel elle correspond et l'ajoute au graphe, tout en gardant les valeurs calculées pour les expositions précédentes.

5.9 Principales Interfaces Graphiques

La conception des interfaces utilisateur (User Interface) est une étape très importante puisque toutes les interactions avec le cœur du logiciel passent à travers ces interfaces. Dans cette partie, nous allons présenter les différentes interactions tout en les associant aux principales fonctionnalités du logiciel.

5.9.1 Fenêtre d'accueil

Au lancement, une fenêtre s'ouvre affichant les différentes parties du logiciel, tel que le widget d'affichage, le graphique, et les différents réglages, comme on peut le voir à la figure 5.1

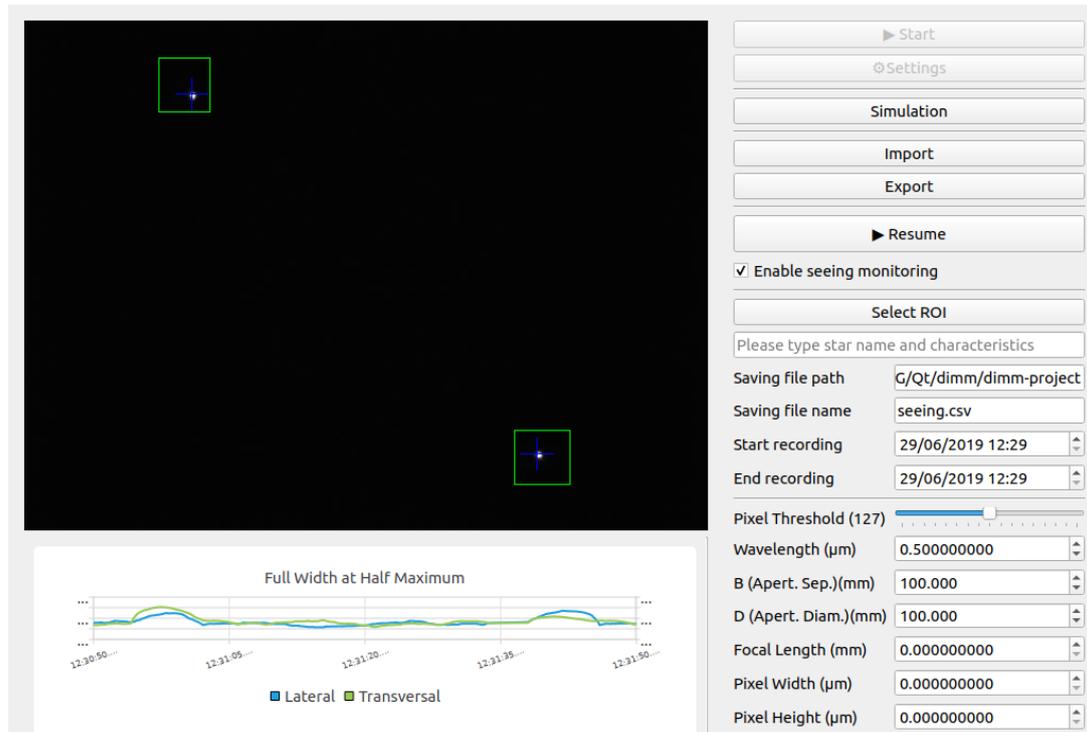


FIGURE 5.1: Fenêtre d'accueil

Dans la colonne de gauche, nous avons un affichage en temps-réel des images capturée avec le positionnement des barycentres, et juste au dessous on a l'affichage de la valeur du seeing calculée en temps-réel sur un graphe. L'axe des abscisses représente l'axe temporel et défile en même temps que l'acquisition se fait.

Dans la colonne de droite, nous avons les boutons de lancement de la caméra, de réglage des paramètres, et de manipulation des fichiers vidéo. Dans ce qui suit nous allons illustrer ces boutons et réglages.

5.9.1.1 Bouton Start

Ce bouton est relié directement à la caméra, c'est à dire dès que l'utilisateur presse ce bouton, une fenêtre s'affiche pour permettre de sélectionner la caméra, la dimension et le format des images, ainsi que le FPS (Frame per second). Dès que l'utilisateur a fini la sélection, il peut cliquer sur OK comme le montre la figure 5.2 pour débiter la capture.

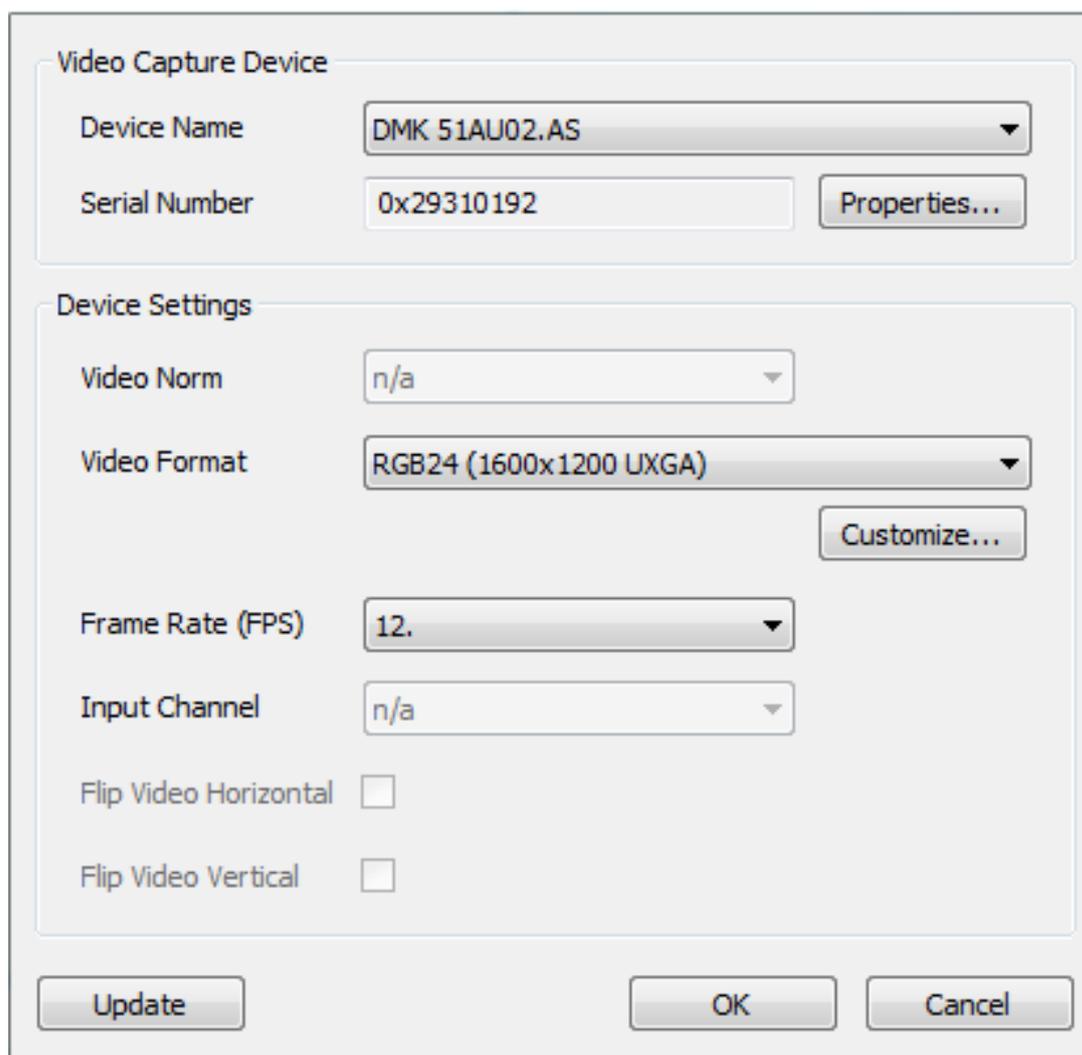
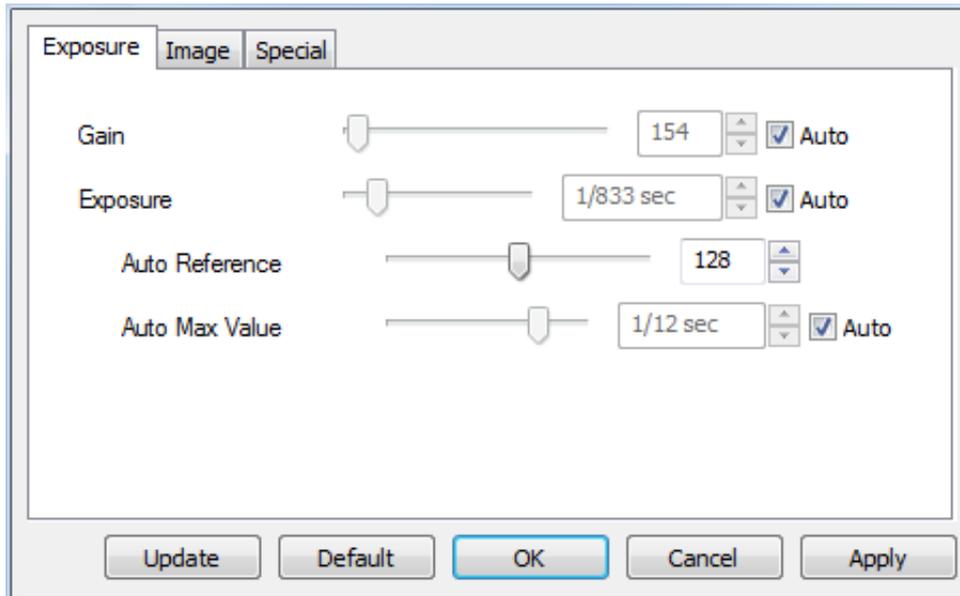


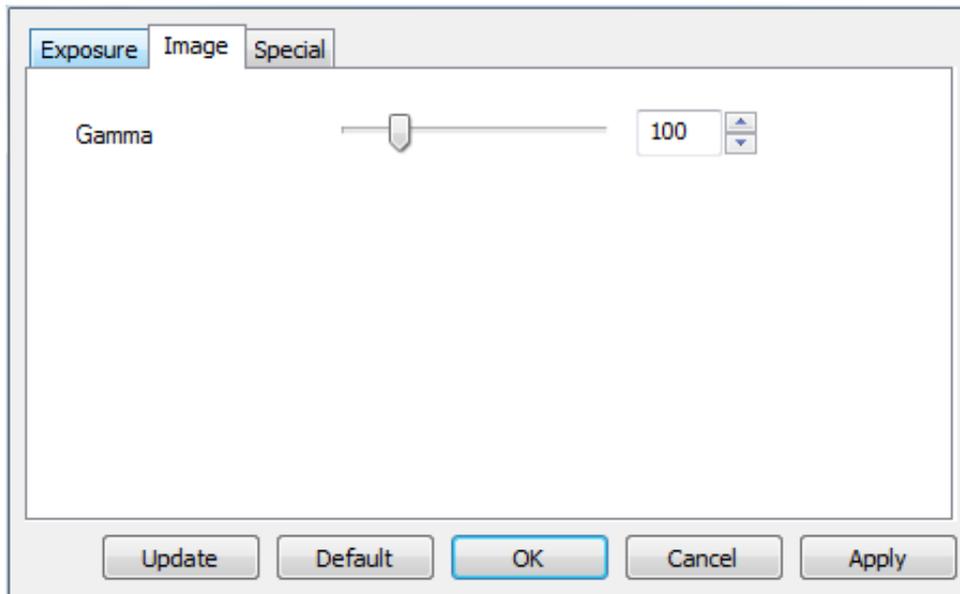
FIGURE 5.2: Selection de la caméra

5.9.1.2 Bouton Settings

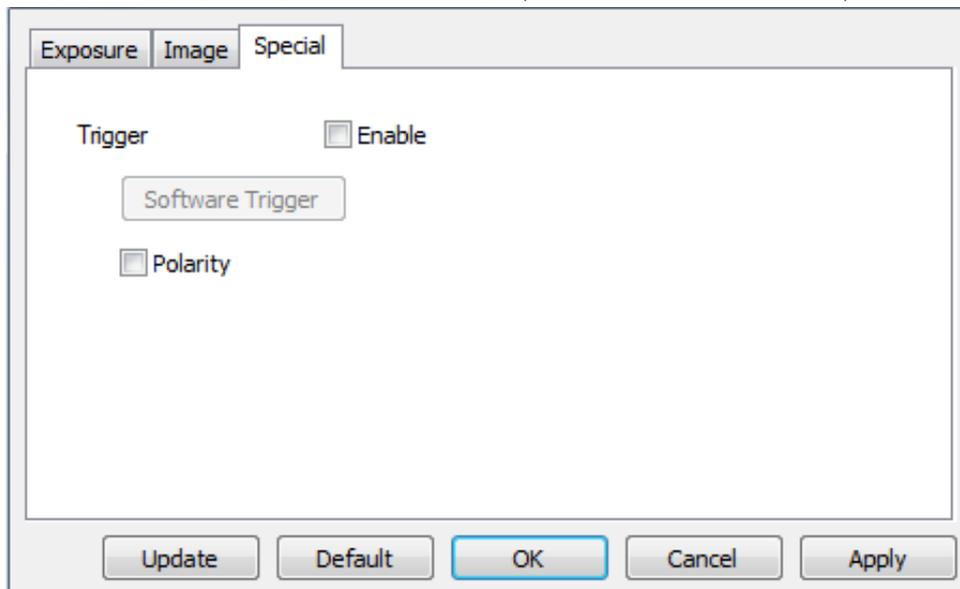
Ce bouton permet d'ouvrir une fenêtre qui permet de changer les réglages de la caméra comme le montre la figure 5.4.



Reglage des paramètres fondamentaux



Réglage du gamma de l'image (correction de la brillance)



Coffee.

FIGURE 5.4: The same cup of coffee. Two times.

Ces paramètres sont :

- Gain
- Exposition
- Gamma
- Trigger (Pour rajouter un déclancheur externe qui actionne la prise de capture par la caméra)

5.9.1.3 Bouton Simulation

Ce bouton permet de démarrer le simulateur de turbulence atmosphérique qui génère deux projections de l'étoile et qui bougent d'une manière aléatoire.

5.9.1.4 Bouton Import

Ce bouton permet de lancer une fenêtre de dialogue qui permet à l'utilisateur de sélectionner un fichier vidéo qui contient la capture des projections de l'étoile, comme le montre la figure 5.5

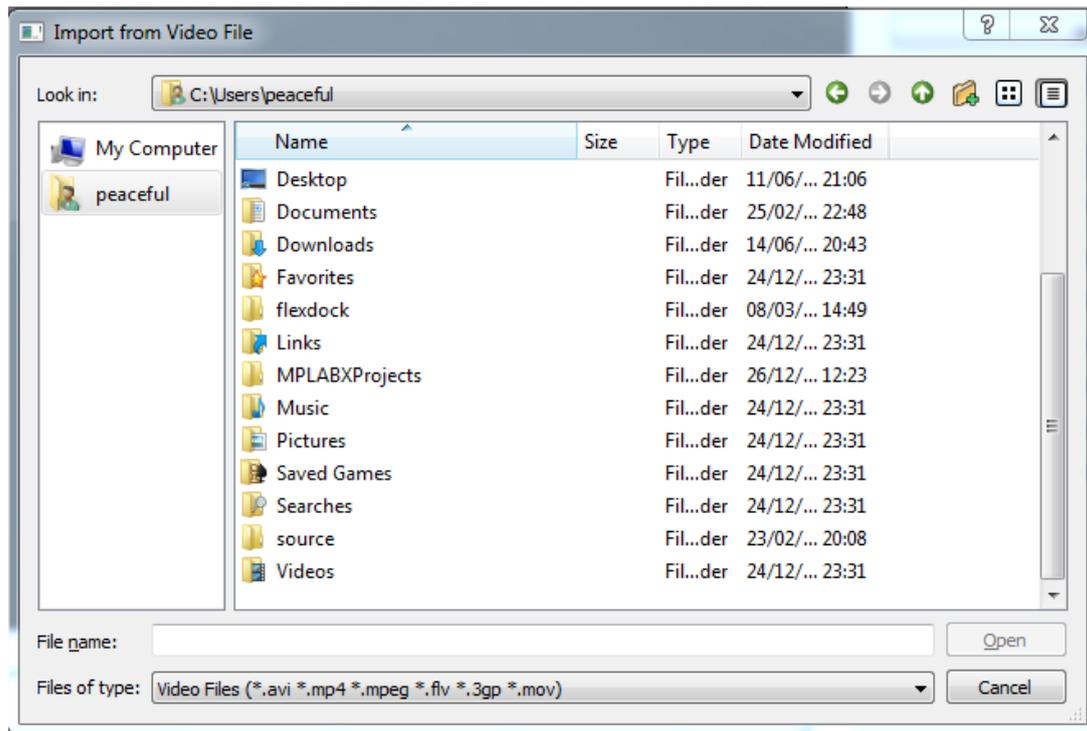


FIGURE 5.5: Lecture depuis fichier video

Le logiciel va détecter automatiquement les paramètres de la video tels que les dimensions, la durée, le FPS...etc.

5.9.1.5 Bouton Export

Ce bouton permet d'ouvrir une fenêtre de dialogue qui permet de selectionner le fichier cible pour la sauvegarde de la video, comme le montre la figure 5.6.

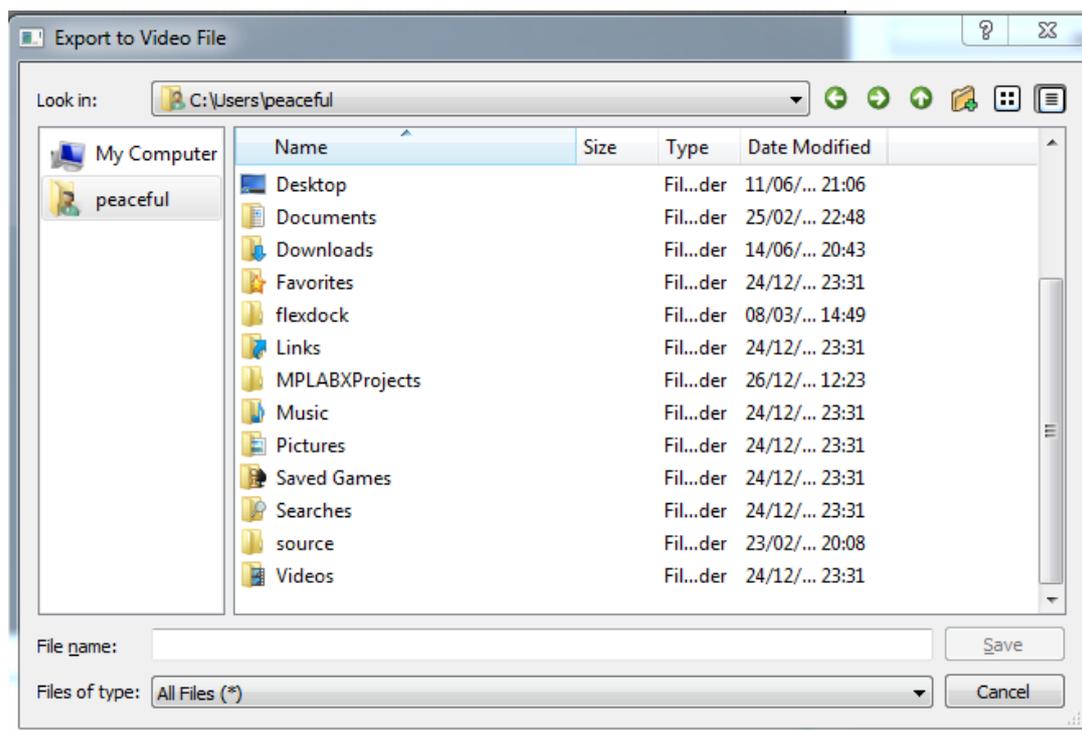


FIGURE 5.6: Sauvegarde vers un fichier video

5.9.1.6 Case à cocher 'Enable Seeing Monitoring'

Lorsque cette case est cochée, le logiciel calcul en temps-réel la valeur du seeing. On a ajouté cette option à l'interface pour donner le choix à l'utilisateur d'entreprendre le calcul du seeing ou non, suivant que les images capturées soit suffisamment intéressantes ou pas.

5.9.1.7 Choix de l'étoile et de ses caractéristiques

Le widget suivant permet à l'utilisateur de choisir une étoile particulière qu'il est en train d'observer, ainsi que ses caractéristiques. Ces paramètres vont être ensuite sauvegardés dans le même fichier CSV que pour les valeurs du seeing.

5.9.1.8 Bouton Select ROI

Ce bouton (qui signifie Select Regions Of Interest) permet de sélectionner avec la souris les deux zones qui vont contenir les deux projections de l'étoile. La figure 5.1 montre le résultat de la sélection avec deux rectangles verts.

5.9.1.9 Champs 'Saving File Path' et 'Saving File Name'

Dans ces deux champs, l'utilisateur peut entrer le chemin ainsi que le nom du fichier CSV sur lequel la valeur du seeing est sauvegardée (avec aussi le timestamp et les informations sur l'étoile).

5.9.1.10 Champs 'Start Recording' et 'End Recording'

Permet à l'utilisateur d'entrer le temps de démarrage de l'enregistrement vidéo et le temps d'arrêt.

5.10 Résultats Expérimentaux

5.10.1 Test de mise au point

Lieu : Observatoire d'Alger

Heure : de 22h à 00h00

Météo : Divers passage de nuage tout au long de la nuit

Lors de cette expédition sur le site astronomique, on a effectué les mesures suivantes afin de vérifier la plupart de nos algorithmes ainsi que l'interface de commande du télescope :

- Vérification du fonctionnement de la mesure du Seeing en temps réel, avec affichage des courbes et mémorisation à la fois des diverses valeurs de Seeing obtenues. Le fait de conserver ces valeurs donne la possibilité de refaire du post-processing à l'observatoire si l'on observe une divergence entre les valeurs obtenues et celles espérées.
- Vérification des diverses fonctions de commande de la caméra.
- Mesure du Seeing sur diverses étoiles et principalement sur diverses élévations.

Lors de cette soirée, la météo était assez désavantageuse. Nous sommes sortis installer le

télescope sur la terrasse aux alentours de 20h. Le ciel était un peu nuageux, l'humidité et la nébulosité assez élevées. Ainsi la ville d'Alger a une pollution lumineuse très importante et un micro-climat défavorable aux observations. Ces facteurs ont rendu le seeing comme considéré insuffisant pour des soirées d'observation normales.

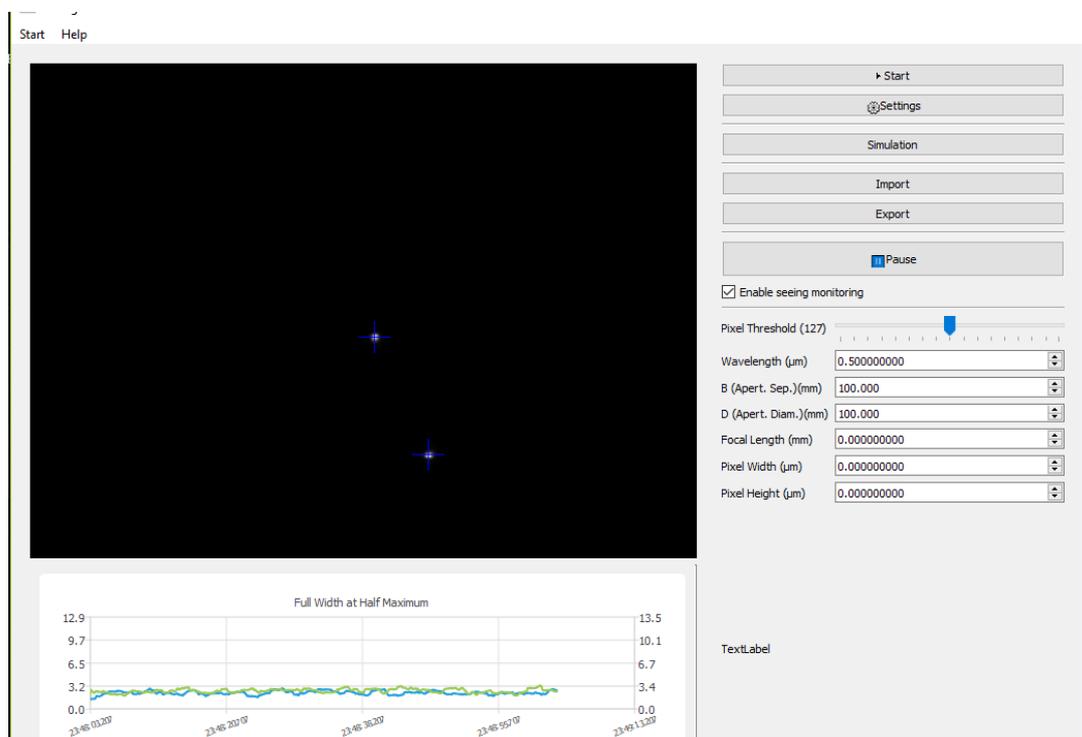


FIGURE 5.1: Calcul en temps-réel du Seeing sur le site d'Alger

Les valeurs du Seeing qu'on a trouvé en utilisant notre logiciel était entre 2,2 et 2,4. Ce sont des valeurs dans l'ordre de grandeur attendu. Elles sont proches de celles trouvées par un logiciel de référence (Seeing Monitor de cyrille Cavadore) dont le Seeing était entre 1,9 et 2,1. Cela est relatif au fait qu'on a pas pris en compte le problème de la saturation qui influe sur le calcul du centroïde, et qui peut induire des décalages allant jusqu'à 10%. Cette phase nous a permis de vérifier le bon fonctionnement du logiciel en partant de l'acquisition des données, puis le traitement des images, et le calcul en temps réel du Seeing et l'affichage de son évolution en temps-réel. Une amélioration future serait de prendre en considération le problème de la saturation lors du traitement des images.

5.11 Conclusion

Ce dernier chapitre de notre projet décrit la phase de réalisation et de test. Nous y avons précisé en particulier les outils de développement utilisés pour la conception du logiciel. Puis nous avons présenté les interfaces les plus importantes dans notre application. Nous pensons que ce travail répond à plusieurs règles de réalisation de projets qui consiste en la réalisation d'une interface graphique d'un logiciel.

En dernier lieu, nous avons fait un test de benchmarking de notre logiciel pour pouvoir l'étalonner sur un logiciel standard de quantification du seeing. Les résultats obtenus sont satisfaisants et permettent d'entrevoir les améliorations futures à apporter au logiciel.

Conclusion Générale

L'objectif de notre projet était de réaliser un logiciel qui fournit un service lié au domaine de l'astronomie en Algérie. Nous voulons, à travers ce travail, établir un moyen avec lequel on peut analyser la problématique de la déformation de l'image à travers la turbulence atmosphérique.

Pour ce faire nous avons commencé notre travail par une partie théorique dont le but est la bonne compréhension de la partie pratique de notre travail, ou on a expliqué le phénomène de la turbulence d'un point de vue physique. Nous avons exploré la théorie de la turbulence atmosphérique, ces caractéristiques et son influence sur un système d'observation optique. Nous avons également donné un aperçu général sur les instruments constitutifs du moniteur de mouvement d'image différentielle et expliqué la technique DIMM permettant de calculer le seeing atmosphérique.

Dans la deuxième partie nous avons décrit la phase d'analyse et de l'implémentation du logiciel. Nous avons d'abord présenté l'ensemble des moyens technologiques utilisés dans le développement de notre projet ainsi que le diagramme des cas d'utilisation afin de donner une vision globale du comportement fonctionnel du système. Nous avons ensuite donné une description des méthodes de programmation utilisées et des interfaces du logiciel réalisé, avant de terminer par une présentation et discussion des résultats pratiques obtenus.

Avec les contraintes temporelles et météorologiques, nous avons pu effectuer un test réel du logiciel sur le terrain au CRAAG. Les résultats obtenus montrent que : i) Le logiciel fonctionne en temps-réel et permet d'avoir un très bonne cadence d'acquisition, ii) Les mesures du seeing obtenues sont de valeur approximativement égales à celles attendues. les résultats obtenus sont dans l'ordre de grandeur de la littérature et restent comparables à ceux obtenus par un logiciel professionnel (Seeing Monitor de Cyrille Cavadore).

Cette première version du logiciel est très concluante et permet donc de traiter des images

en temps réel avec une cadence élevé, supérieure à celle atteinte par la caméra.

Ce travail nous a été très formateur, puisqu'il nous a permis de découvrir de nouveaux outils et concepts de développement, ainsi que de nous confronter à plusieurs contraintes à la fois : contraintes temporelle et météorologique, et technologique.

Ce fut un challenge très bénéfique !

Bibliographie

- [qt,] Cross-platform software developement for embedded desktop (<https://www.qt.io/>).
- [see,] Fried parameter.
- [ope,] Learning opencv, gary bradski and adrian kaehler, o'reilly.
- [mat,] Matplotlib (<https://matplotlib.org/>).
- [pyt,] Python official web site – comparing python to other languages.
- [tel,] T lescope - wikipedia (<https://fr.wikipedia.org/wiki/t%3%a9lescope>).
- [A., 1976] A., F. (1976). La turbulence en m canique des fluides.
- [A.Voziakova,] A.Voziakova. Restoration of turbulence profile from scintillation indices.
- [F, 1981] F, R. (1981). The effects of atmospheric turbulence in optical astronomy. *Progress in optics 19, North Holland, Amsterdam.*
- [Kornilov,] Kornilov, V. Combined mass-dimm instruments for atmospheric turbulence studies.
- [M. Sarazin,] M. Sarazin, F. R. The eso differential image motion monitor. *Astronomy and Astrophysics.*
- [Tokovinin, 2002] Tokovinin, A. (2002). From differential image motion to seeing. *The Astronomical Society of the Pacific.*

Annexe

Présentation de l'Organisme d'Accueil

Le CRAAG

Le Centre de Recherche en Astronomie Astrophysique et Géophysique CRAAG, est un Établissement Public à Caractère Scientifique et Technologique (EPST), régi par le décret 20-06 du Février 2006. Le CRAAG est issu de création de l'Observatoire d'Alger en 1890 et puis de l'IMPGA en 1931. En 1980, fut créé le CNAAG .En 1985, lors de la mise en place des statuts des centres de recherches, le CRAAG fut créé par décret 83-521 .Il fit partie du HCR en 1990 et fut placé en 1991 sous l'autorité du Ministère de l'Intérieur.

Historique du CRAAG :

Le CRAAG résulte de la fusion de deux organismes ayant existé durant l'époque coloniale à savoir l'Observatoire d'Alger créée en 1890 et l'Institut de Physique du Globe d'Alger l'IMPGA en 1931.

En 1980, fut créé le Centre National en Astronomie Astrophysique et Géophysique regroupant les activités de l'Institut de Météorologie et de Physique du globe d'Algérie et de l'Observatoire d'Alger, placé sous tutelle de l'ONRS. Ce regroupement s'impose dans le souci de redonner à ces deux axes scientifiques importants la place qui leur revenait et sauvegardait les possibilités de développement des études géophysiques et astronomiques en Algérie.

En 1985, la recherche scientifique bénéficia de l'attention des pouvoirs publics qui

décidèrent de la création de Centres de Recherche à la faveur du décret 83-521. C'est ainsi que fut créé le CRAAG qui reprit les prérogatives de l'ex CNAAG mais qui bénéficia de l'organisation et du mode de gestion des Centres de Recherche.