



HAL
open science

Etude et intégration du module d'acquisition d'images couleur sur la plateforme Tracilogis

Marc Gusmerini, Benjamin Thabuis

► **To cite this version:**

Marc Gusmerini, Benjamin Thabuis. Etude et intégration du module d'acquisition d'images couleur sur la plateforme Tracilogis. Sciences de l'ingénieur [physics]. 2012. hal-01873270

HAL Id: hal-01873270

<https://hal.univ-lorraine.fr/hal-01873270v1>

Submitted on 13 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-memoires-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Projet fin d'étude
-ENSTIB 3-

**Etude et intégration du module
d'acquisition d'images couleur sur la
plateforme Tracilogis**

**GUSMERINI Marc
THABUIS Benjamin
2011/2012**

Enseignants responsables : André Thomas, Vincent Bombardier

Avant-propos

La prise en main de logiciels inconnus ne comportant que très peu d'aide est très difficile. De plus, la mise en place d'un système de vision nous permet d'enrichir nos connaissances. Nous tenons donc à remercier Vincent Bombardier pour les réponses qu'il a pu nous apporter sur la recherche de forme, Jeremy Jover, Julien Lallemand et Arnould Guidat pour leur aide et leur présence durant la totalité du projet et Jérôme Pannetier du service technique de Stemmer Imaging qui a su nous sortir des impasses informatiques que nous avons pu rencontrer.

Contenu

Avant-propos	1
Introduction	1
Partie 1 : Recherche bibliographique	2
1.1 Le contrôle non destructif.....	2
1.2 La vision industrielle	3
1.3 Méthodologie.....	4
1.3.1 Segmentation par seuillage	5
1.3.2 Reconnaissance utilisant une base de données	5
1.4 Les différentes normes de classification du bois	6
1.4.1 Les nœuds	7
1.4.2 Les flaches	9
1.4.3 Les poches de résine	9
1.4.5 Le bleuissement	9
1.5 Conclusion sur la recherche bibliographique	10
Partie 2 : Le projet.....	11
2.1 Introduction Présentation Tracilogis	11
2.1.1 Nos objectifs	12
2.1.2 Matériel utilisé	12
2.1.3 Les logiciels utilisés.....	13
2.2 L'expérimentation	15
2.2.1 Mode opératoire	15
2.2.2 Choix du classificateur.....	17
2.2.3 Présentation de Manto Teach.....	18
2.3 Conclusion.....	30
Partie 3 : Analyse des résultats.....	31
3.1 Procédés utilisés	31
3.1.1 Classificateur retenu.....	31
3.1.2 Prétraitement de l'image	33
3.1.3 Détection des nœuds sains et des poches de résine.....	37
3.2 Exploitation du système de vision en dynamique.....	39

3.2.1 Problèmes rencontrés	40
3.2.2 Résultats des tests dynamiques	41
3.3 Améliorations envisagées	42
3.3.1 Amélioration du classificateur	42
3.3.2 Détection des fentes	43
3.3.3 Recherche de la position de la singularité.....	44
3.3.4 Amélioration de l'éclairage.....	45
Conclusion.....	47
Bibliographie.....	48
Annexe	49
Tutorial Manto.....	49
Résumé.....	56

Introduction

Notre projet de fin d'étude a pour intitulé « étude et intégration du module d'acquisition d'images couleur sur la plateforme Tracilogis ». Le sujet étant vaste, nous avons abouti, après une réunion avec nos encadrants, sur le sujet suivant : « détection de singularité par système de vision ». Ce sujet est très intéressant car aujourd'hui l'intégration de systèmes de contrôle non destructifs est nécessaire dans l'industrie du bois. Cela permet d'augmenter le rendement, d'améliorer la performance et la compétitivité d'une entreprise. Le système de vision est une méthode qui permet d'identifier différentes singularités que nous pouvons trouver dans le bois. Cependant, appliquer un outil de vision pour classifier le bois est un exercice compliqué à cause de la grande diversité de ces singularités.

Ce rapport a pour but d'expliquer la mise en place de ce système de contrôle non destructif qui permet l'identification de défaut par vision sur des planches de bois. L'objectif est d'arriver à classer ces planches suivant le défaut détecté. Ce système est mis en place sur la plateforme Tracilogis du CRAN.

Partie 1 : Recherche bibliographique

L'utilisation de systèmes informatiques pour la détection des singularités du bois n'est pas nouvelle. En effet, de nombreuses thèses et articles ont déjà été réalisés sur ce sujet. Cette recherche bibliographique nous permettra de prendre conscience de l'avancée des différentes techniques utilisées actuellement et celles qui ont été employées par le passé. Nous pourrons ensuite les reproduire à notre échelle et les adapter en fonction de notre matériel disponible.

1.1 Le contrôle non destructif

Le contrôle non destructif représente l'ensemble des techniques et des procédés aptes à fournir des informations sur les caractéristiques d'une pièce ou d'une structure sans l'endommager pour son utilisation future. Ce procédé met en évidence les défauts susceptibles d'altérer le produit à l'usage auquel il est destiné. Aujourd'hui, ces systèmes sont très utilisés dans l'industrie automobile, l'industrie pétrolière, l'industrie navale, l'aéronautique, l'inspection alimentaire ou encore l'industrie de l'énergie.

Le contrôle non destructif dans l'industrie du bois permet de caractériser des planches, des grumes ou encore des avivés sans les détruire. Ces contrôles peuvent s'intégrer au circuit de fabrication d'une pièce dans l'industrie. Dans le souci de réaliser un contrôle qualité efficace, les CND sont indispensables. Ils permettent de caractériser un produit qui est par définition anisotrope et hétérogène, chaque produit est donc différent. Il nous faut pourtant les classer pour pouvoir leur affecter une valeur dans le but de les vendre.

Il est intéressant de comprendre les différences entre les méthodes employées. En effet, on ne recherchera pas la même chose si l'on utilise un système de vision avec l'aide de camera, un système de rayon X, un laser etc...Le Tableau 1 présente les différents types de capteurs utilisés pour faire du contrôle non destructif ainsi que leurs utilisations principales.

Type de capteur	Technologie	Utilisations
Capteurs de forme	Mécanique	détermination de la longueur, largeur, volume, forme, présence de flache
	Caméra	
	Laser	
	Couplage laser/caméra	
Capteurs de caractéristiques	Ondes mécaniques	Détection et classification des singularités diminuant la résistance mécanique
	Méthode vibratoire	Analyse des propriétés mécaniques
	Vision	Identification et localisation des singularités
	Rayons X	Sciage virtuel
	Micro-ondes	Mesure de propriétés affectant la résistance mécanique
	Infra rouge	Estimation de la présence de bois de cœur

Tableau 1

Pour notre étude, nous cherchons uniquement à détecter les singularités du bois situées en surface en vue d'un classement visuel de la planche. Nous n'avons donc pas besoin de connaître les caractéristiques intrinsèques qui sont utiles à un classement mécanique du bois. Nous avons donc le choix entre : les caméras couleurs (utilisation de capteurs CCD) ou les systèmes multi-capteurs utilisant les rayons-X ou les lasers.

Les capteurs CCD offrent un excellent rapport prix/efficacité, ils permettent d'obtenir une bonne résolution sans distorsion géométrique, leur vitesse de lecture est importante. De plus, ils sont robustes, ce dernier point est important dans l'optique de travailler dans le domaine de la première transformation du bois. La caméra couleur semble donc toute indiquée dans notre cas.

1.2 La vision industrielle

L'examen visuel par l'homme est le premier des procédés de contrôle, c'est le plus simple et le plus général. Les productions de masses à haute cadence, le souci constant d'amélioration de la qualité et la recherche de gain économique poussent de plus en plus les industriels à automatiser les moyens de production. La vision industrielle est une réponse à ces préoccupations pour les opérations de contrôles de la production. En effet, les machines de vision industrielle permettent un contrôle de la production à haute cadence et assurent une bonne répétabilité du contrôle (à la différence d'un opérateur, une machine n'est jamais fatiguée et ses critères de décision ne varient pas). La vision industrielle peut aussi être utilisée pour gérer des flux d'objets. Par exemple, la lecture optique d'un code à barres ou d'une adresse postale sur un colis pour l'orienter dans un centre de tri. Elle peut aussi être

utilisée afin de contrôler l'aspect extérieur d'un produit, par exemple la présence de tâches sur un fruit ou la présence de singularités sur une planche de bois.

Typiquement, la vision industrielle consiste à prendre une ou plusieurs images de l'objet à contrôler ou visualiser. Les images sont obtenues grâce à une ou plusieurs caméras. Ensuite, elles sont numérisées pour être utilisables par un logiciel de traitement d'images le plus souvent dédié au contrôle considéré. Le traitement de l'image ayant été réalisé, une décision est prise par rapport à des critères prédéfinis et une action est effectuée. Par exemple, si on détecte la présence de nœuds noirs sur une pièce de bois, elle sera orientée vers un bac « nœuds noirs ».



Figure 1 : Présentation du contrôle par vision [2]

1.3 Méthodologie

Pour le traitement des images, les différents articles et thèses utilisent exclusivement des programmes qui ont été spécialement édités pour leurs besoins, [1], [2], [3], [4]. Dans notre cas, n'étant des spécialistes en programmation, nous utilisons des logiciels disponibles pour les industriels permettant de faire l'acquisition et le traitement des images.

Le travail de détection par vision se divise en deux parties : La segmentation et la reconnaissance.

La segmentation d'images est une opération de traitement qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond.

Il existe plusieurs techniques de segmentation, dans une étude réalisée par J.W. Funck et al. , *Image segmentation algorithms applied to wood defect detection* [1], neuf algorithmes différents sont comparés pour la détection de singularités sur des placages en douglas. Ces algorithmes ont été testés sur des images en niveaux de gris et sur des images en couleur. Parmi les neuf algorithmes, cinq font appel à la méthode de seuillage, trois algorithmes font appel à la technique de détection de contours et le dernier fait appel à une technique combinant croissance et fusion des régions.

1.3.1 Segmentation par seuillage

La segmentation par seuillage présente l'avantage d'être simple et de faire ressortir les singularités. Il est aisé d'extraire les zones intéressantes pour pouvoir procéder à la reconnaissance du défaut. Cette technique est l'une des premières utilisées [2].

La Figure 2 montre les différentes étapes à réaliser.

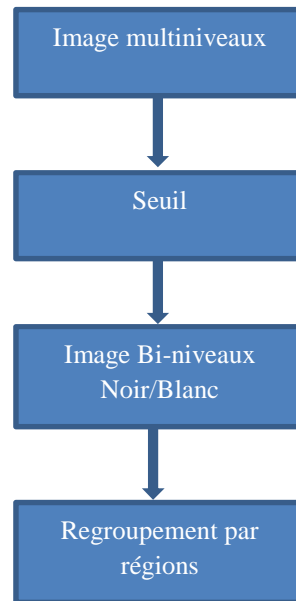


Figure 2 : La segmentation pas seuillage

Le regroupement en régions se fait par un changement de résolution. Les niveaux de gris de chaque pixel adjacent sont analysés et suivant si ces niveaux sont proches, les pixels sont inclus dans le contour ou exclus. On obtient donc des contours fermés autour des singularités. Cette phase conditionne la suite du traitement, elle est très importante.

On peut noter cependant qu'en utilisant des images bi-niveaux on perd de nombreuses informations comme la texture par exemple. Il est donc impossible de faire la détection entre différentes catégories de nœuds (noir, sain, adhérent...). Il faut donc réaliser un traitement différent sur les images en niveaux de gris ou en couleur pour relever leurs textures, etc.

Dans [3], la paramétrisation des régions obtenues se fait avec l'aide d'une analyse géométrique des formes. Plusieurs algorithmes ont été introduits pour réaliser cette étape. En effet, les fentes sont des défauts étroits et les nœuds sont des défauts larges. On peut donc utiliser des paramètres de classement simples comme l'aire et le périmètre.

1.3.2 Reconnaissance utilisant une base de données

Dans [4], il a été mis en place une méthode de reconnaissance pour des singularités du pin en utilisant une base de données et un classificateur. La première étape est la constitution de la base de données. Il faut un nombre assez important d'images par catégorie, plus il y a d'images plus la détection est efficace. Pour cette étude [4], 17 748 images ont été

collectées et près de 70 000 singularités ont été sélectionnées.

Le traitement se compose en trois phases :

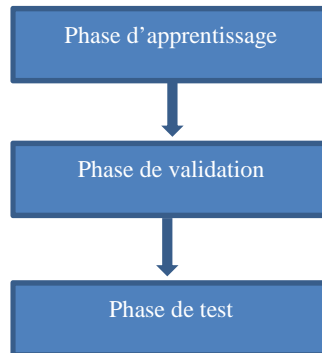


Figure 3 : La reconnaissance en utilisant une base de données

Le plus compliqué dans cette méthode est de créer les algorithmes permettant de faire le rapprochement entre une image appartenant à la base de données et l'image de test. Le temps de traitement dépend directement de ces algorithmes. Pour cette étude [4], un algorithme de type génétique a été utilisé. Les résultats obtenus semblent satisfaisants avec un taux de réussite de 87% sur 10 catégories de singularités différentes.

1.4 Les différentes normes de classification du bois

Le but final de la détection de singularités sur une planche de bois est d'aboutir à un classement visuel selon le nombre, le type et la taille des singularités détectées. Nous avons donc recherché les différentes normes de classement visuel existantes.

La norme NF EN 1611 [9], décrit les différents critères de classement pour les bois de résineux. Le but de cette norme est de classer un avivé en fonction du nombre et de la taille des singularités sur les quatre faces de cet avivé. Cinq choix sont possibles : 0 ; 1 ; 2 ; 3 ; 4. Dans le cas de pièces de sections rectangulaires, il y a une notion de rive et de face. Si la rive est d'un choix inférieur au choix établi par le classement des faces, la pièce est du choix de la plus mauvaise rive. Dans le cas de pièces de sections carrées, le choix de la pièce est du choix immédiatement supérieur au choix du côté le moins beau.

Ce système de classement est explicité dans le Tableau 2.

Pièce	Face	Face	Rive	Rive	Choix	
	1	2	1	2	pièces	
A	1	2	1	1	1	1
B	1	3	2	2	2	2
C	1	1	2	1	1	2
D	1	1	2	3	1	3

Tableau 2

Les singularités prises en compte pour le classement sont les suivantes (Figure 4) :

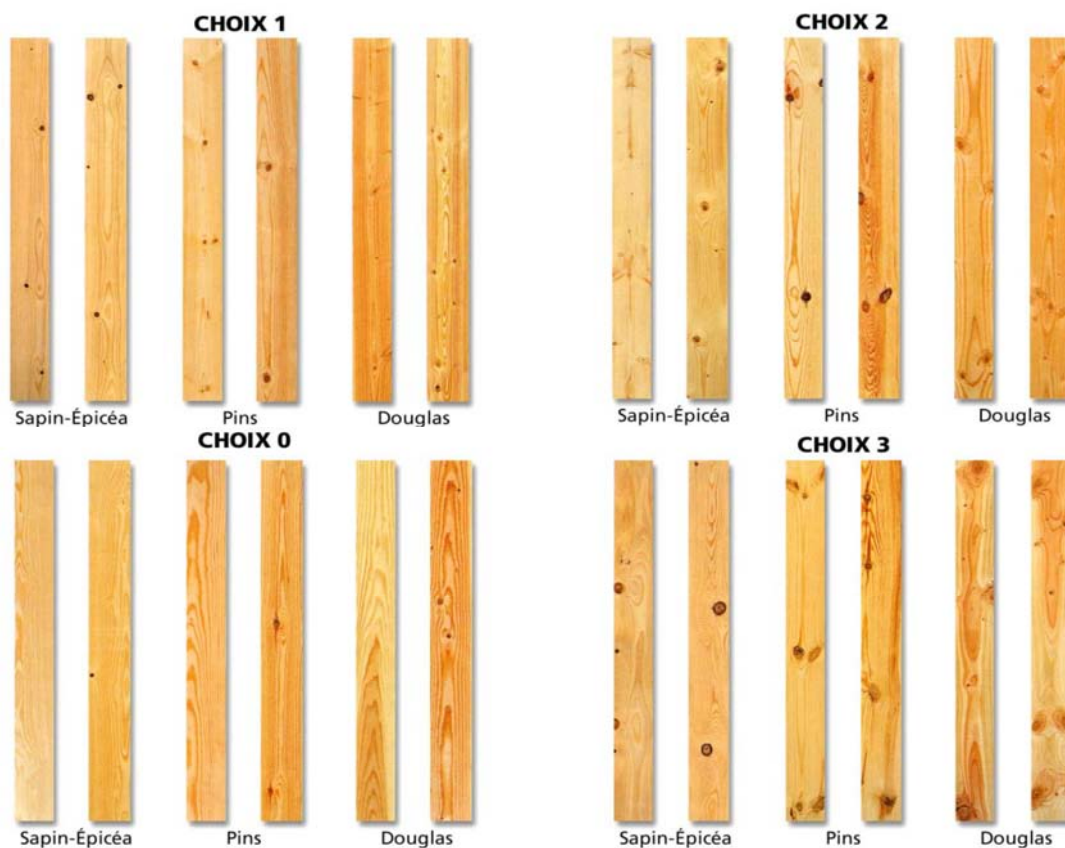


Figure 4 : Les différents choix esthétiques du bois [10]

1.4.1 Les nœuds

Ils sont mesurés sur le côté où la branche, origine du nœud sur le bois scié, a été coupée transversalement par rapport à son axe longitudinal. Dans le cas contraire, ils sont pris en compte par leur présence (admis avec limitation de nombre ou exclus). Lorsqu'elle est mesurée, la dimension du nœud prise en compte sur les faces est la moyenne entre la plus petite dimension du nœud et sa plus grande dimension. Sur les rives, la dimension prise en compte est la distance entre les tangentes aux nœuds parallèles aux arêtes de la pièce.

Dimensions à prendre en compte (Figure 5) :

Sur les faces (Nœuds A; B; C) = $(a+b)/2$

Sur les rives (Nœuds D; E; F) = entre les tangentes parallèles

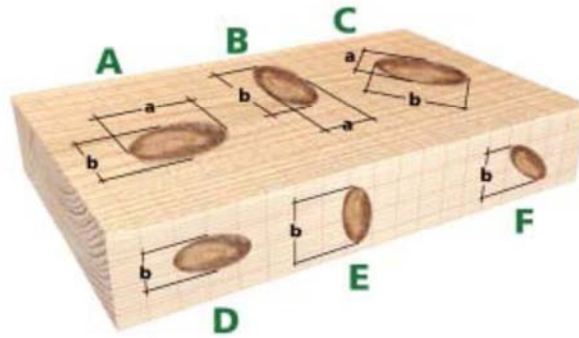


Figure 5 : Prise en compte des nœuds dans le classement d'une pièce [10]

Il existe différents types de nœuds :

- Nœuds sains
- Nœuds noirs
- Nœuds adhérents

Cependant aucune norme ne semble exister à ce sujet. La classification se fait donc selon l'appréciation de l'opérateur.

Il y a un lien entre la taille, le nombre et la position des nœuds et le classement structural du bois [11]. Plus le nœud est grand, plus la classe du bois sera grande et plus il est important de le repérer pour pouvoir l'enlever (Tableau 3).

Essences	Localisation	Classement structural		
		ST I	ST II	ST III
Sapin épicéa douglas	sur la face	$\varnothing < 1/6$ et $\varnothing < 30\text{mm}$	$\varnothing < 1/2$ et $\varnothing < 50\text{mm}$	$\varnothing < 3/4$ et $\varnothing < 100\text{mm}$
Pin	sur la face	$\varnothing < 1/10$ et $\varnothing < 15\text{mm}$	$\varnothing < 1/3$ et $\varnothing < 50\text{mm}$	$\varnothing < 2/3$ et $\varnothing < 100\text{mm}$

Tableau 3 [11]

Ensuite, le contraste et la couleur vont pouvoir déterminer s'il s'agit d'un nœud sain, d'un nœud sautant ou encore d'un nœud noir.

1.4.2 Les flaches

Particularité liée au débit, due aux irrégularités de forme de la grume ou à sa conicité, le flache est la portion visible de la surface de la grume d'où provient la pièce. Les flaches sont caractérisés par :

- leur longueur qui est la plus grande distance entre les extrémités du flache. En cas de présence de plusieurs zones flacheuses sur la pièce, les longueurs des flaches individuels s'additionnent,
- leur largeur sur les faces,
- leur épaisseur sur les rives.

Dimension à prendre en compte (Figure 6) :

Sur les faces: $(a-a_1) / a$ et noter la longueur $l_1 + l_2$

Sur les rives: $(b-b_1) / a$ et noter la longueur $l_1 + l_2$

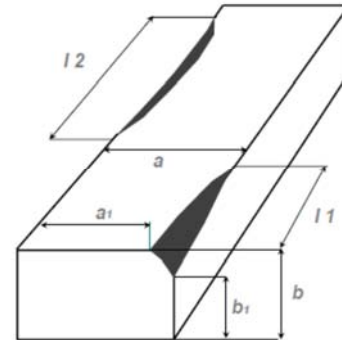


Figure 6 : Importance des flaches [10]

1.4.3 Les poches de résine

Il s'agit de cavités allongées qui contiennent de la résine. On distingue les petites poches de résine de longueur inférieure à 50 mm et les grosses poches de résine de longueur supérieure à 50 mm (Figure 7).



Figure 7 : Poche de résine

1.4.5 Le bleuissement

Coloration bleuâtre, superficielle ou profonde, provoquée par un champignon. Le bleuissement n'entraîne pas de changement appréciable de la consistance du bois et de ses propriétés mécaniques (Figure 8).



Figure 8 : Bleuissement

1.5 Conclusion sur la recherche bibliographique

Au terme de cette recherche bibliographique, nous pouvons commencer à proprement parler la phase de création. Nous réaliserons notre projet en deux temps, avec une phase de segmentation du défaut et une phase de classification. D'après la norme NF EN 1611 [10], Les singularités les plus importantes pour le classement visuel du bois sont les nœuds. Nous nous focaliserons donc, dans un premier temps, sur la détection de cette singularité.

Le Tableau 4 est un récapitulatif des différentes singularités rencontrées et les caractéristiques qui leurs sont attachées, il a été réalisé par Emmanuel Schmitt durant ses travaux de recherche.

Nom des singularités	Forme	taille	position	contraste	couleur	texture
nœud sain	X	X	-	X	-	-
nœud cerné	X	X	-	X	X	-
nœud noir	X	X	-	X	X	-
fente	X	X	-	X	-	-
flache	X	X	X	X	X	X
entre écorce-écorce	X	-	-	X	X	X
trou	X	X	-	X	-	-
bois de compression	-	-	-	X	X	X
aubier	-	-	X	X	X	-
moelle bois de cœur	X	X	X	X	X	-
poche de résine	X	X	-	X	X	-
bleuissement	-	X	-	X	X	-
coloration	-	-	-	X	X	X

Tableau 4

Partie 2 : Le projet

2.1 Introduction Présentation Tracilogis

Notre projet d'étude se déroule sur la plateforme TRACILOGIS. Cette plateforme expérimentale est utilisée par l'équipe de recherche Système-Produit du Centre de Recherche en Automatique de Nancy.

Elle réunit différentes compétences qui servent à traiter les problèmes inhérents à la traçabilité, à l'identification et au contrôle par le produit pour les chaînes logistiques. On retrouve, d'une part, des « instruments électroniques » embarqués, capteurs, senseurs et autres technologies (RFID) ou, d'autre part, des moyens « naturels » en s'appuyant sur les caractéristiques intrinsèques de la matière (mesures biométriques).

Quatre points sont particulièrement ciblés :

- Maîtrise et industrialisation des techniques d'instrumentation.
- Modèles de données et d'informations inhérents aux processus de traçabilité et d'éco-certification.
- Cohérence et coordination entre les différents flux de décisions.
- Amélioration de l'efficacité globale pour le système logistique et de production.

Dans le cadre de la détection de singularité du bois, nous utilisons la plateforme de vision.

La Figure 9 : La plateforme Tracilogis présente l'implantation générale de la plateforme Tracilogis. La Figure 10 : Le système de vision industrielle représente la partie vision



Figure 9 : La plateforme Tracilogis



Figure 10 : Le système de vision industrielle

industrielle de la plateforme.

2.1.1 Nos objectifs

Pour notre projet, nous voulons détecter trois singularités du bois : Les poches de résine, les nœuds sains et les nœuds noirs. Nous avons décidé de réunir les nœuds sautants et les nœuds cernés avec la catégorie des nœuds noirs. En effet, après quelques tests, il était trop difficile de faire la distinction entre les nœuds noirs, les nœuds sautants et les nœuds cernés car la différence de contraste n'est pas assez prononcée. Nous voulons ensuite faire cette reconnaissance de manière dynamique, en utilisant un convoyeur pour les échantillons, pour ensuite les trier suivant leurs singularités dans quatre bacs différents. L'autre intérêt de ce projet est de ne pas programmer un utilitaire spécifique à nos besoins, mais d'utiliser des logiciels déjà existants. Ce dernier point est important car les logiciels que nous utilisons ne sont pas forcément adaptés à la recherche de singularités sur du bois. Une part importante du projet est la prise en main des différents outils à notre disposition et également la recherche de nouveaux outils si nécessaire.

2.1.2 Matériel utilisé

Nous utilisons un convoyeur à bande sur lequel nous mettons nos échantillons de bois. La vitesse d'amenage est aux alentours de 10 mètres par minute (Figure 12).

La caméra utilisée est une Spyder 3 qui provient du fabricant Dalsa (Figure 11). C'est un modèle linéaire couleur qui a une résolution de 2048 pixels de large et un pixel de haut. Cela implique que l'information est envoyée sous forme de ligne. Il faudra ensuite assembler toutes ces lignes afin d'obtenir une image complète de la pièce.

La camera linéaire nécessite que la pièce soit fortement éclairée car le temps d'acquisition est très court. Nous utilisons une lampe halogène de 230W.

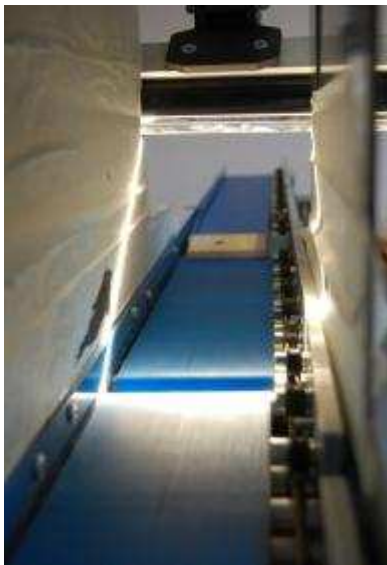


Figure 12 : Système de convoyage



Figure 11 : Caméra Spyder 3

Pour permettre une bonne synchronisation entre la bande convoyeuse et l'acquisition de la caméra, nous utilisons un codeur incrémental. Ainsi, la caméra démarre son acquisition uniquement si le codeur envoie un signal. De plus, le temps d'acquisition est directement lié au codeur, si celui-ci tourne vite, la vitesse d'acquisition de la caméra est augmentée.

La Figure 13 présente la chaîne d'acquisition de notre projet avec les différents flux d'information, les outils et les logiciels utilisés.

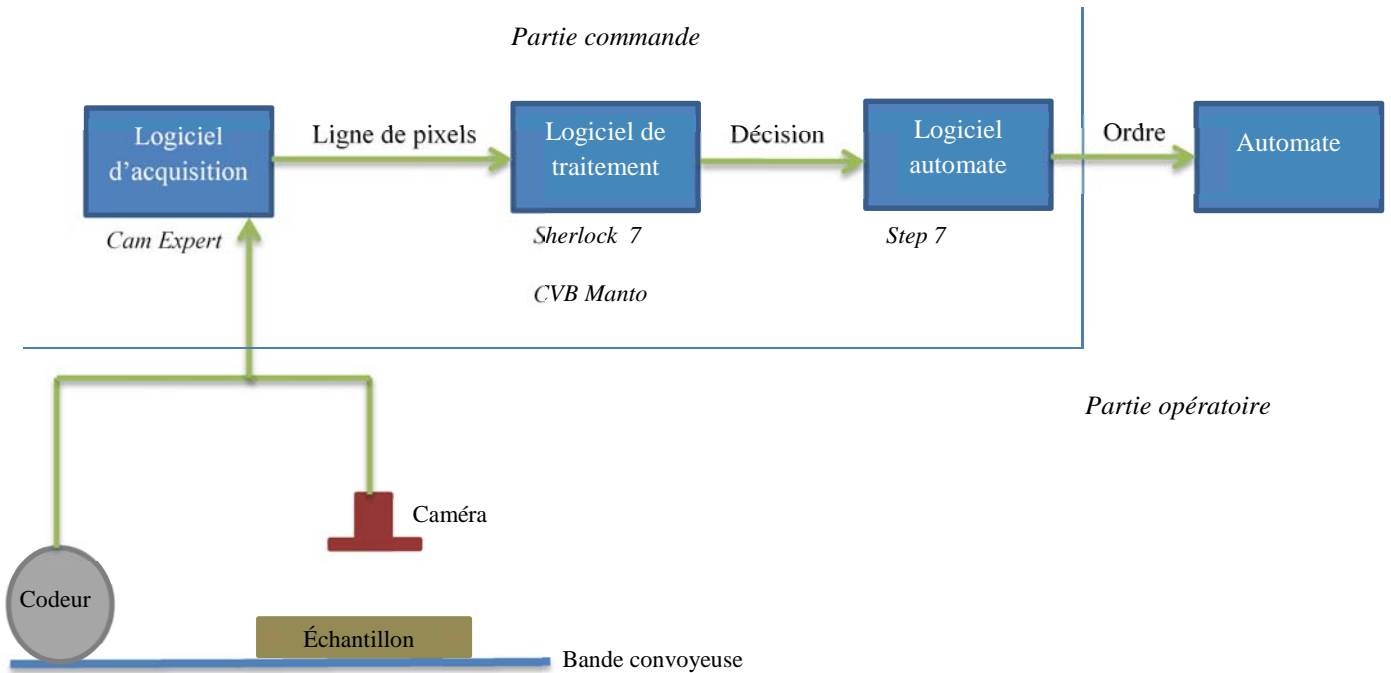


Figure 13 : Présentation de la chaîne d'acquisition

2.1.3 Les logiciels utilisés

Cam expert est le logiciel qui nous permet de faire l'acquisition de l'image. Il est directement lié à la caméra et nous permet de rentrer plusieurs paramètres comme le trigger (élément déclencheur de l'acquisition), l'intensité des différents plans de couleurs (Red Green Blue), la largeur de l'image, la vitesse d'acquisition, le gain, l'éclairage.

Nous n'avons que très peu utilisé ce logiciel, car le projet d'étude précédent avait déjà fourni une configuration acceptable pour notre étude [12].

Afin de mettre en place une détection des singularités du bois, une partie importante du projet consiste à prendre en main le logiciel de traitement. Sherlock est plus qu'un simple logiciel, c'est en réalité une interface de vision destinée à l'industrie. La prise en main s'avère assez complexe car son ergonomie est limitée, de plus la rubrique d'aide est très peu utile car elle ne donne ni d'informations ni d'exemple pratiques. Cette interface est compatible avec d'autres logiciels plus spécialisés. Ainsi l'acquisition se fait via Cam-expert mais l'image est directement envoyée à Sherlock. La bibliothèque d'outils additionnels Common Vision Blox est également compatible avec Sherlock, ce qui nous permet d'obtenir des outils supplémentaires très performants.

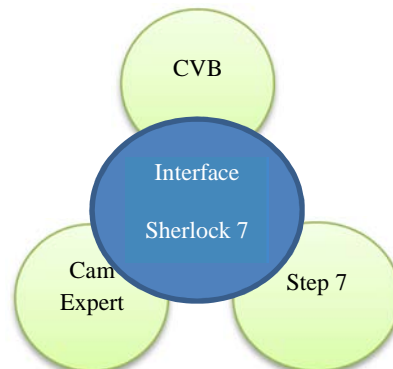


Figure 14 : L'interface Sherlock 7

Sherlock permet de faire du traitement d'image en utilisant des ROI (Region Of Interest). Les images à traiter peuvent être amenées manuellement, on peut donc utiliser des images préenregistrées. Cette fonction est très pratique pour mettre au point le programme de traitement car elle permet de tester notre programme sur de nombreux échantillons rapidement. Il est également possible d'acquérir les images directement depuis la caméra. Sherlock offre un bon nombre d'outils permettant de traiter les images à partir de nombreux algorithmes différents. Ainsi, il est possible d'augmenter les contrastes ou au contraire de flouter l'image, de détecter des arêtes, de reconnaître des formes....

Une fois que l'image est traitée et qu'une décision est prise, l'information est envoyée à l'automate. Cette communication se fait par l'intermédiaire d'un agent codé en Java qui transmet la décision prise par Sherlock au logiciel Step 7 qui commande l'automate.

Pour le traitement, Sherlock n'a pas tous les outils nécessaires à notre application, on utilise Common Vision Blox, qui est une bibliothèque d'outils additionnels, en complément. Donc chaque outil est spécifique et demande l'acquisition d'une licence supplémentaire. Pour notre application, nous utilisons l'outil Manto. Il permet de faire de la reconnaissance à partir d'une base de données que l'utilisateur paramètre. Il se décompose en deux parties. La première se déroule sur l'application Manto Teach. Elle consiste à créer une base de données et à la paramétrer. La seconde est l'exploitation à proprement parler, elle se déroule sur l'interface de Sherlock.

2.2 L'expérimentation

2.2.1 Mode opératoire

Suite à la recherche bibliographique, nous avons décidé d'utiliser Sherlock pour faire l'étape de segmentation des défauts. On utilise ensuite l'outil CVB Manto pour faire la reconnaissance.

L'étape de segmentation est compliquée à réaliser. Plusieurs choix s'offrent à nous, comme par exemple garder les images en couleur, les utiliser en niveaux de gris, augmenter les contrastes, les diminuer.

Nous avons choisi de tester cinq bases de données avec des traitements différents afin de retenir la plus probante pour l'utilisation finale de reconnaissance en dynamique (Figure 15). Dans la première base de données, nous travaillons avec des images en RGB. Dans les quatre autres, nous travaillons avec des images en MONO8. Une avec un fond noir, une autre avec un fond blanc et une à laquelle nous avons appliqué un traitement préliminaire, filtre Sharpen suivit d'un filtre Normalize dans le but d'améliorer les contrastes et la netteté. Enfin, la dernière est une base de données où nous avons appliqué un traitement de façon à isoler le défaut.

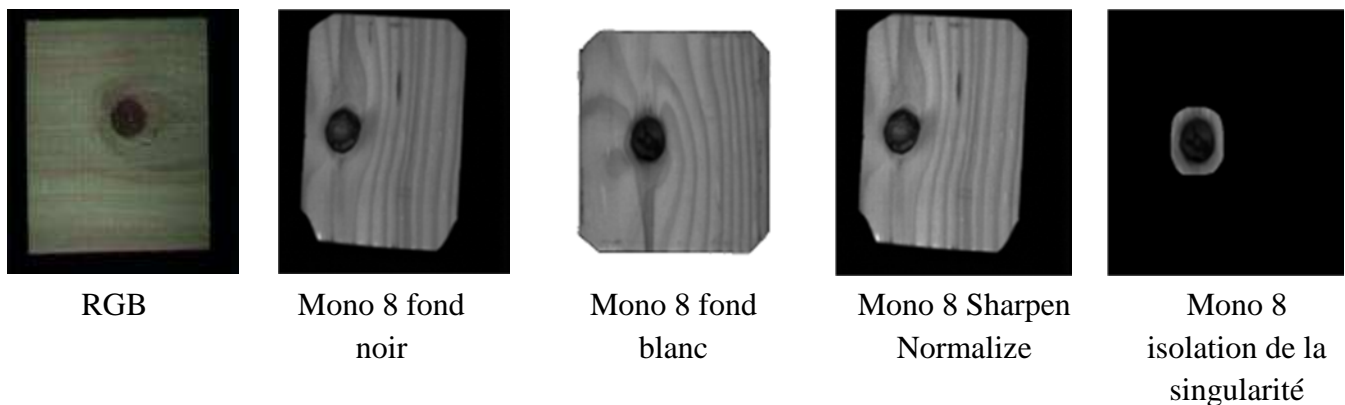


Figure 15 : Les différents prétraitements comparés

Afin de constituer une base de données, nous avons usiné des échantillons avec les singularités qui nous intéressaient. Nous ne nous sommes pas cantonnés à une seule essence de bois. Nous obtenons une base de données hétérogène se composant d'éprouvettes d'épicéa, de sapin, de pin maritime et de douglas, sur lesquels nous avons repéré 50 singularités (Tableau 6). Nous avons utilisé la notation du Tableau 5.

Base de données	
Désignation	Notation
nœud	N
poche de résine	P
noir	n
sain	s

Tableau 5 : notation utilisées

Singularités	Description	nombre singularités
Nn1	non-adhérents	2
Nn2	non-adhérent	1
Nn3	sautant	1
Nn4	non-adhérent	1
Nn5	sautant	1
Nn6	cerné	1
Nn7	non-adhérent	1
Nn8	sautant	1
Nn9	non-adhérent	1
Nn10	fendu presque sain	1
Nn11	sautant	1
Nn12	sautant	1
Nn13	sautant	1
Nn14	sautant	1
Nn15		1
Nn16	sautant	1
Nn17	sautant	1
Nn18		1
Nn19		2
Ns1		1
Ns2		1
Ns3	faible contraste	1
Ns4		1
Ns5	fendu	1
Ns6		1
Ns7	faible contraste	1
Ns8	fendu	1
Ns9		1
Ns10		1
Ns11	noir + sains	2
Ns12	fendu	1
Ns13	cerné + sain	2
Ns14	fendus	2
Ns15		1
Ns16	fendu	1

Ns17	Faible contraste	1
P1		1
P2	Poche+Nn	2
P3	Poche +Nn	2
P4	Poche+Nn	2
P5		1
P6		1
	Tot singularités	50

Tableau 6 : Description des éprouvettes

2.2.2 Choix du classificateur

Choisir le classificateur est une des étapes clés du projet. En effet, c'est grâce à lui que nous allons pouvoir détecter les singularités que nous souhaitons. Si le classificateur n'est pas bon nous ne retrouverons pas le défaut souhaité. Nous n'avons malheureusement pas eu accès à l'algorithme de traitement qu'utilise Manto pour la reconnaissance, nous ne connaissons donc pas son mode de fonctionnement. Cependant, après plusieurs essais sur des bases de données différentes, nous pensons qu'il fonctionne par « pattern matching ». C'est-à-dire que lorsque nous lui rentrons une image en base de données, il l'analyse et relève une liste de caractéristiques comme par exemple la forme, le contraste, ou encore l'orientation. Lorsque que nous lançons une phase de test, il évalue chaque parcelle de l'éprouvette et il compare ses caractéristiques à celle de la base de données. Nous pouvons régler sur le logiciel un taux de ressemblance compris en 0.5 et 1. Plus ce taux de ressemblance est élevé, moins notre classificateur pourra détecter une image qui n'est pas dans sa base de données.

En travaillant de façon dynamique, le but serait de faire un autoapprentissage. Par exemple, lorsqu'un nœud noir est détecté, ce nœud est automatiquement enregistré dans la base de données pour la compléter. En appliquant cette méthode, il faut prendre un facteur d'oubli en considération. En effet, plus la base de données est petite, plus l'analyse est difficile car il y a peu de comparaisons à faire. En revanche, si la base de données est grande, le besoin de mémoire informatique est élevé et l'analyse deviendra très longue, d'où l'importance de prendre en compte ce facteur. Le plus judicieux est de lui faire désapprendre les images qui sont rentrées en premier dans sa base. De plus dans une grande base de données on risque de rencontrer des images qui se ressemblent. Donc il faut trouver le juste milieu pour avoir une base de données performante mais pas trop grande. Malheureusement, nous n'avons pas eu le temps de mettre en place cette technique.

2.2.3 Présentation de Manto Teach

Manto Teach est un logiciel qui permet de créer un classificateur à partir d'une base de données.

Réaliser la base de données

En premier lieu, il convient de définir la taille de notre objet à détecter (Figure 16 et Figure 17). En effet, une fois la dimension du cadre jaune choisie, il est impossible de la changer pour la suite de la création de la base. Cette fonction a certainement été créée pour faciliter le travail de recherche du logiciel et raccourcir le temps de traitement. Il nous pose néanmoins des problèmes lorsque qu'il s'agit de reconnaître des objets qui peuvent avoir une taille très variable. Par exemple entre un petit nœud noir et un gros nœud.

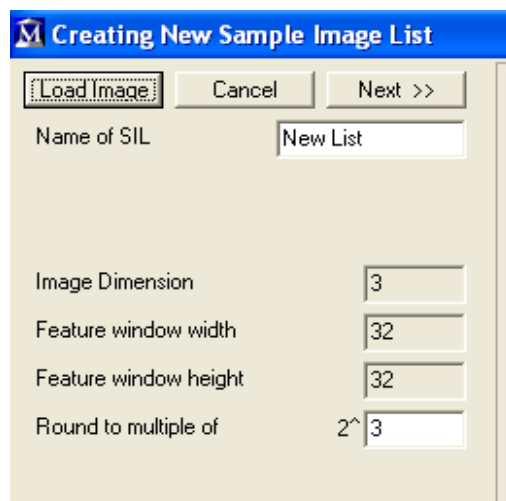


Figure 16 : Choix de la taille de la zone d'enregistrement

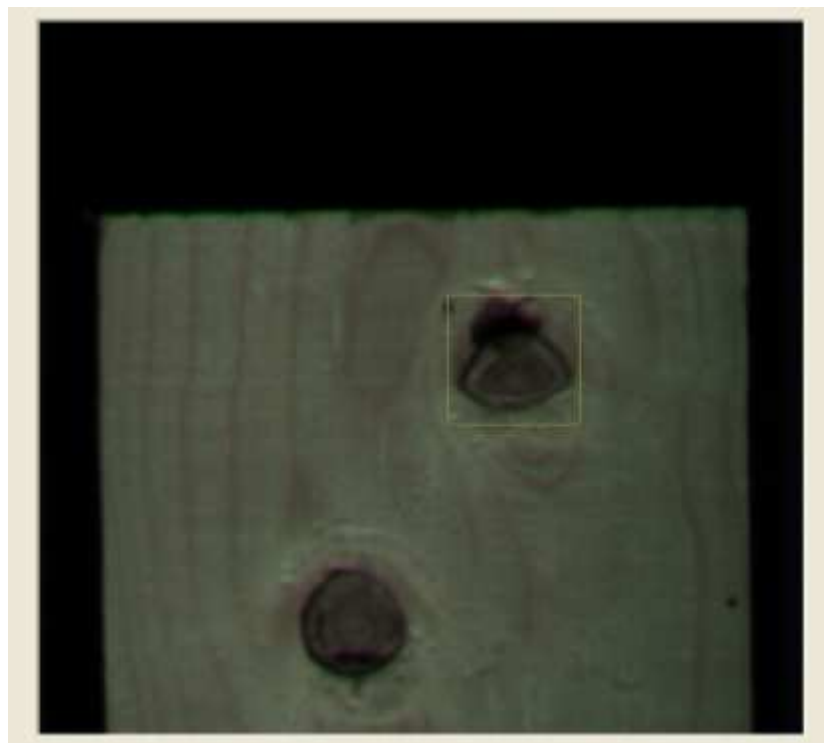


Figure 17 : La zone d'enregistrement

Pour créer la base de données, il faut sélectionner les singularités que nous désirons ajouter en déplaçant le cadre jaune de sélection. Nous décidons ensuite de la catégorie de la singularité, par exemple un nœud. Le classificateur enregistre alors les caractéristiques du nouveau nœud (Figure 18). Il faut recommencer cette opération sur d'autres images afin d'obtenir une base de données assez conséquente.

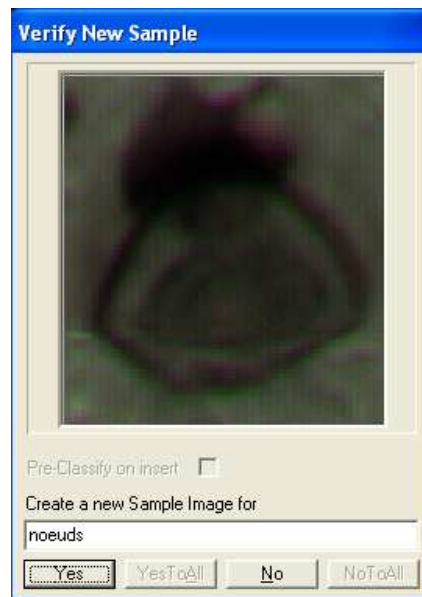


Figure 18 : Image d'une singularité dans la base de données

Afin d'aider notre classificateur, il est possible de lui donner comme information des images à ne pas prendre en compte, par exemple le veinage de l'éprouvette. Une classe « exclu » est alors créée (Figure 19). Il faut rentrer un maximum d'images dans cette classe si l'on veut obtenir un classificateur efficace.



Figure 19 : Image « exclu » de la base de données

Une fois que toutes les images qui constituent la base de données ont été rentrées, il convient de la faire apprendre au logiciel. Manto Teach nous propose alors plusieurs « preprocessing code » (Figure 20). Le « preprocessing code » est le paramètre le plus important de la génération du classificateur. Avec ce paramètre, nous avons une influence immédiate sur la qualité et la vitesse de la classification. Le « preprocessing code » est en réalité une simple chaîne de caractères (string) composée de lettres 'P' et 'A'. Cette chaîne de caractères définit quels filtres doivent être appliqués et dans quel ordre. Les filtres 'P' réduisent la quantité de données par un facteur 4. Les filtres 'A' améliorent les informations relatives aux contours. Les filtres 'P' vont, de manière générale, augmenter la vitesse de traitement du classificateur mais ils vont dégrader le résultat de la classification. Les filtres 'A' vont améliorer les résultats de la classification. Cependant, le temps de traitement sera augmenté. On choisira alors le code qui obtient le meilleur taux d'apprentissage. Suivant la taille de la base de données et la complexité des éléments enregistrés, le programme nous proposera un choix entre plusieurs « preprocessing code ».

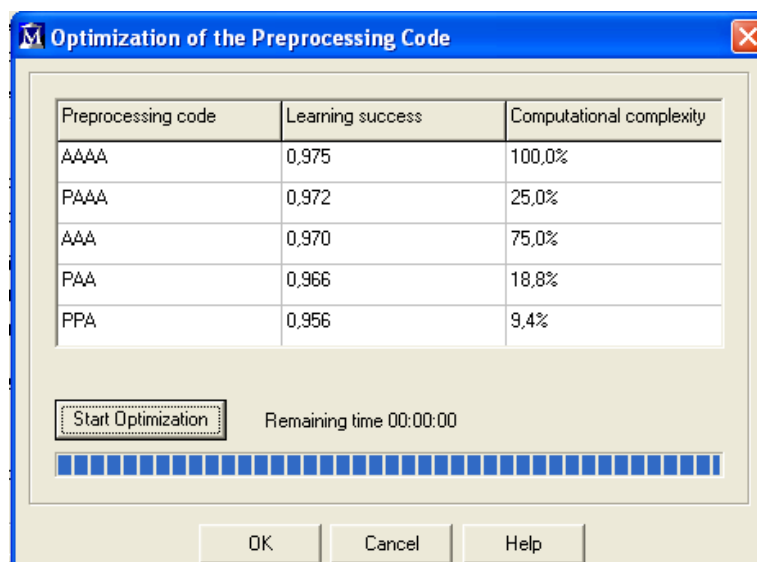


Figure 20 : Choix du preprocessing code

Phase de test

Pour juger de l'efficacité de notre classificateur, nous avons accès à une phase de test. On rentre alors des images qui ne sont pas dans la base de données enregistrée. Une recherche est ensuite lancée (Figure 21).

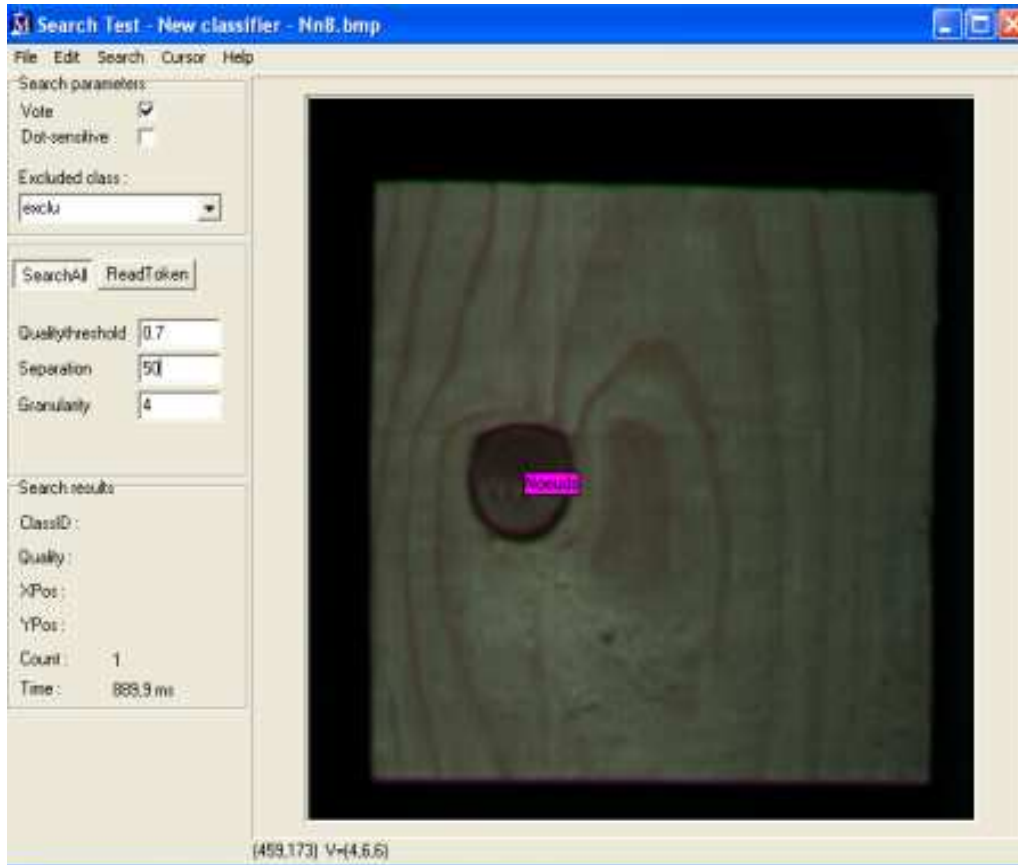


Figure 21 : Test de validation du classificateur

Plusieurs paramètres sont configurables en fonction des éléments que nous voulons rechercher.

Le paramètre « Qualitythreshold » est un seuil de qualité qui définit la limite en dessous de laquelle les résultats seront ignorés, il est compris entre 0,5 et 1. Plus ce taux est élevé, moins le classificateur reconnaitra des nœuds qui ne sont pas dans sa base de données. Cependant si ce seuil est trop bas, nous prenons le risque de reconnaître des objets qui ne sont pas des nœuds. Après de nombreux essais, la valeur 0,7 offre un bon compromis.

Le paramètre « Separation » permet de spécifier une distance minimale entre deux résultats positifs. Cela élimine les nuages inévitables des résultats qui se forment autour des centres des objets dans l'image. Avec la séparation, la meilleure correspondance dans le rayon spécifié est choisie pour être l'objet rapporté par la suite (Figure 22).

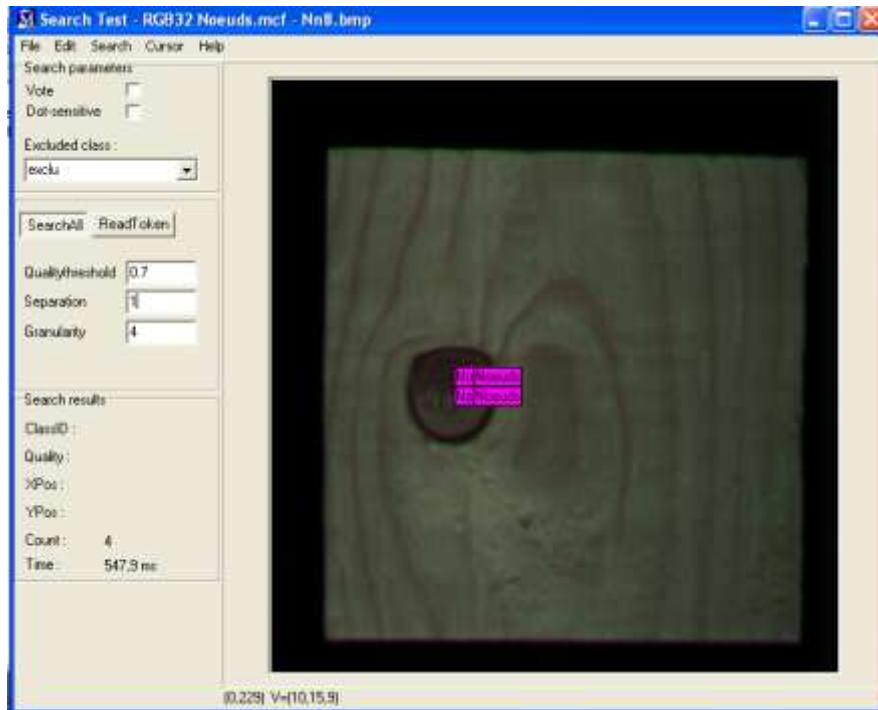


Figure 22 : Importance du paramètre separation

Expression des résultats

Pour exprimer nos résultats, nous avons choisi de les numéroter de quatre manières différentes (Tableau 7).

Résultat	
1	Singularité détectée vrai
2	Singularité détectée faux
3	Singularité non détectée vrai
4	Singularité non détectée faux

Tableau 7

Le résultat obtenu est '1' (singularité détectée vrai) si une singularité est détectée et si le logiciel Manto l'attribue à la bonne base de données. Par exemple l'image est un nœud noir et Manto l'identifie comme tel (Figure 23).

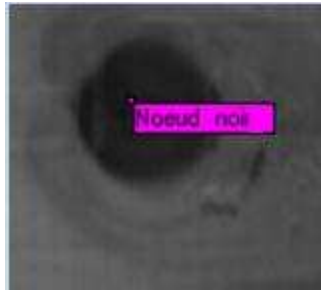


Figure 23 : Résultat '1'

Le résultat obtenu est '2' (singularité détectée faux) si une singularité est détectée et si Manto l'identifie avec la mauvaise base de données. Par exemple l'image est un nœud noir et Manto l'identifie comme un nœud sain (Figure 24).

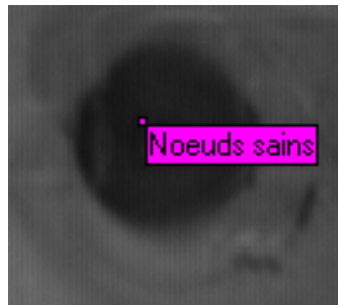


Figure 24 : Résultat '2'

Le résultat obtenu est '3' (singularité non détectée vrai) si la singularité présente ne doit pas être détectée, car elle ne fait pas partie d'une base de données et Manto ne l'identifie à aucune base de données. Par exemple l'image est une poche de résine mais il n'existe pas de base de données pour ce défaut donc le logiciel ne donne aucun nom (Figure 25)

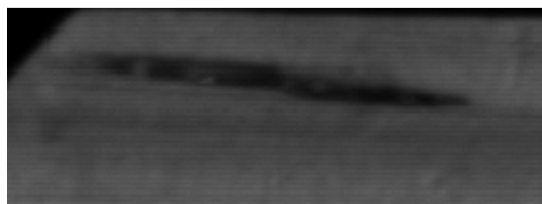


Figure 25 : Résultat '3'

Le résultat obtenu est '4' (singularité détectée faux) si la singularité présente ne doit pas être détectée car elle ne fait pas partie d'une base de données et Manto l'identifie à une base de données. Par exemple l'image est une poche de résine mais il n'existe pas de base de données pour ce défaut. Cependant le logiciel l'appelle nœud noir (Figure 26).

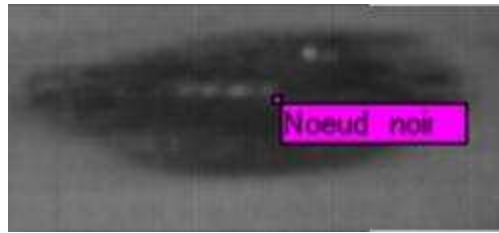


Figure 26 : Résultat '4'

Pour évaluer la qualité et l'influence des prétraitements des images sur un classificateur, il a été décidé de prendre en considération le taux d'erreur et le taux de mémorisation. Le temps de traitement est également relevé car il s'agit d'un indicateur important permettant de différencier les méthodes.

Le classificateur a été testé dans trois configurations différentes :

- Recherche uniquement des nœuds noirs à partir d'une base de données N_n
- Recherche uniquement des nœuds sains à partir d'une base de données N_s
- Recherche des nœuds noirs et des nœuds sains à partir d'une base de données N_n+N_s

Le taux de mémorisation est obtenu en ne recherchant les singularités que sur les images appartenant à la base de données.

Le taux d'erreur est obtenu en additionnant les singularités détectées faux '2' et les singularités non détectées faux '4'. Pour le calculer la recherche est lancée uniquement sur des images n'appartenant pas à la base de données.

Méthode RGB :

Dans cette base de données, les images sont enregistrées en couleur (Figure 27).

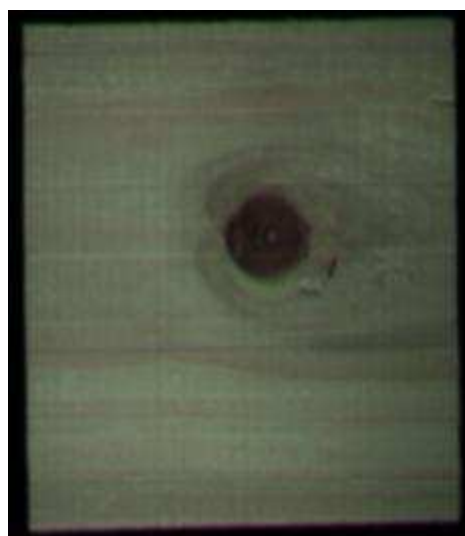


Figure 27 : Image base de données RGB

Les résultats que nous avons obtenus avec notre classificateur RGB sont représentés dans le Tableau 8.

Méthode RGB			
	Nn	Ns	Nn+Ns
Singularité détectée vrai (1)	32,50%	17,07%	44,12%
Singularité détectée faux (2)	12,50%	56,10%	29,41%
Singularité non détectée vrai (3)	45,00%	12,20%	8,82%
Singularité non détecté faux (4)	10,00%	14,63%	17,65%
Taux de réussite (1+3)	77,50%	29,27%	52,94%
Taux d'erreur (2+4)	22,50%	70,73%	47,06%
Temps de traitement	561 ms	560 ms	854 ms
Type de singularités	Taux de mémorisation		
Nn	90,48%		
Ns	88,89%		
Poche de résine	100%		

Tableau 8

On note que cette méthode possède un bon taux de mémorisation (90.5%) pour les nœuds noirs. Cela semble logique car chaque nœud enregistré dans la base de données possède de nombreuses caractéristiques (couleur). Cependant, le taux de réussite est trop faible. Avec la couleur il y a trop d'informations différentes. Le logiciel apprend par cœur la base de données et les éprouvettes proviennent d'essences de bois différentes, ainsi si notre classificateur ne retrouve pas exactement les mêmes couleurs, le résultat sera faux. On remarquera que le temps de traitement est relativement long par rapport aux techniques décrites ensuite (560 ms).

Méthode MONO8 avec fond noir :

Dans cette base de données, les images sont enregistrées en niveaux de gris. Par rapport à la couleur nous perdons les informations du plan rouge, vert et bleu (Figure 28).

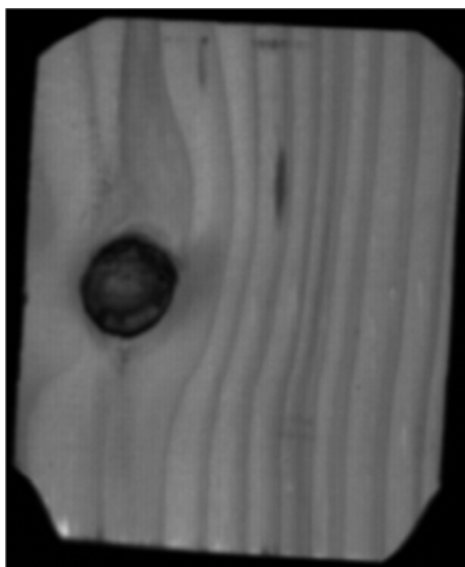


Figure 28 : Image base de données Mono8

Les résultats obtenus sont présentés dans le Tableau 9.

Méthode MONO8			
	Nn	Ns	Nn+Ns
Singularité détectée Vrai (1)	31,71%	4,76%	45,45%
Singularité détectée faux (2)	2,44%	50,00%	9,09%
Singularité non détectée vrai (3)	53,66%	23,81%	12,12%
Singularité non détectée faux (4)	12,20%	21,43%	33,33%
Taux de réussite (1+3)	85,37%	28,57%	57,58%
Taux d'erreur (2+4)	14,63%	71,43%	42,42%
Temps de traitement	186 ms	187 ms	292 ms
Type de singularités	Taux de mémorisation		
Nn	85,71%		
Ns	77,78%		
Poche de résine	40,00%		

Tableau 9

Cette base de données offre un taux de réussite accru par rapport à une base de données RGB, tout en gardant un taux de mémorisation acceptable (85.7%) sur les nœuds noirs. Cette méthode atténue la précision tout en gardant un effet de contraste suffisant. De cette manière les nœuds inconnus ressemblent plus à ceux de la base de données. Cependant

le fond noir induit parfois le classificateur en erreur, car le contraste est proche de certains nœuds noirs. Si l'on rentre le fond noir dans la catégorie « exclu » de notre classificateur, cela peut rentrer en conflit avec le contraste des nœuds noirs. On remarque que le temps de traitement est presque trois fois inférieur au temps de traitement en RGB.

Méthode MONO8 avec fond blanc :

Cette base de données est donc la même que la MONO8 classique, mais on a ajouté un prétraitement « Fil » qui permet de rendre le fond blanc (Figure 29).

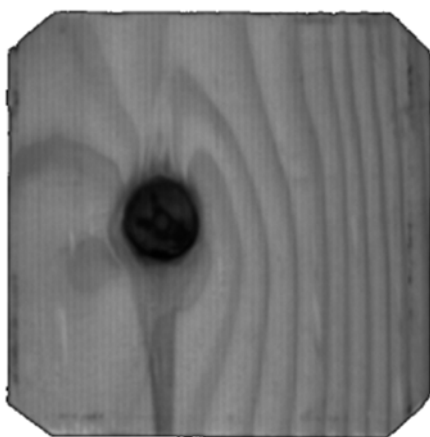


Figure 29 : Image base de données Mono8 fond blanc

Les résultats obtenus sont présentés dans le Tableau 10.

Méthode MONO8 fond blanc			
	Nn	Ns	Nn+Ns
Singularité détectée Vrai (1)	30,00%	20,51%	40,63%
Singularité détectée faux (2)	0,00%	58,97%	12,50%
Singularité non détectée vrai (3)	60,00%	7,69%	9,38%
Singularité non détectée faux (4)	10,00%	12,82%	37,50%
Taux de réussite (1+3)	90,00%	28,21%	50,01%
Taux d'erreur (2+4)	10,00%	71,79%	49,90%
Temps de traitement	175 ms	180 ms	270 ms
Type de singularités	Taux de mémorisation		
Nn	85,71%		
Ns	77,78%		
Poche de résine	82,05%		

Tableau 10

Cette base de données corrige certains problèmes obtenus avec la précédente, nous obtenons des meilleurs taux de réussite, 90% pour les nœuds noirs.

Méthode MONO8 sharpen normalize :

Cette base de données est en fait une base de données MONO8 classique à laquelle nous avons effectué deux prétraitements, Sharpen et Normalize, dans le but d'augmenter le contraste pour mieux faire ressortir les singularités (Figure 30).

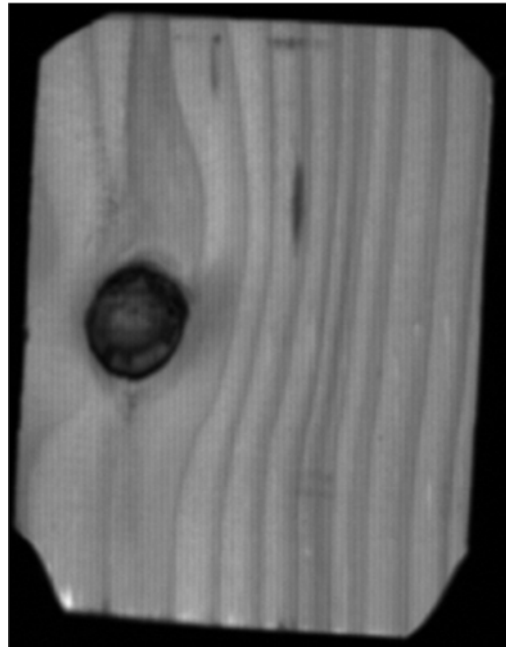


Figure 30 : Image base de données Mono8 sharpen normalize

Les résultats sont présentés dans le Tableau 11.

Méthode MONO8 sharpen normalize			
	Nn	Ns	Nn+Ns
Singularité détecté e Vrai (1)	19,51%	21,95%	24,24%
Singularité détecté e faux (2)	2,44%	58,54%	3,03%
Singularité non détectée vrai (3)	56,10%	9,76%	18,18%
Singularité non détectée faux (4)	21,95%	9,76%	54,55%
Taux de réussite (1+3)	75,61%	31,71%	42,42%
Taux d'erreur (2+4)	24,39%	68,29%	57,58%
Temps de traitement	186 ms	187ms	295 ms
Type de singularités	Taux de mémorisation		
Nn	100,00%		
Ns	77,78%		
Poche de résine	100,00%		

Tableau 11

Avec ce classificateur, on retrouve des résultats comparables à la méthode RGB. Cela s'explique car en améliorant les contrastes, nous ajoutons plus de paramètres à chaque nœud enregistré dans la base de données. Ce n'est donc pas une bonne méthode.

Méthode isolation de la singularité :

Cette base de données est constituée à partir d'une base de données MONO8 classique à laquelle on applique plusieurs prétraitement afin de ne faire ressortir uniquement la singularité (Figure 31).



Figure 31 : Image base de données singularité isolée

Les résultats obtenus sont présentés dans le Tableau 12.

Méthode d'isolation			
	Nn	Ns	Nn+Ns
Singularité détecté e Vrai (1)	33,33%	28,57%	44,44%
Singularité détectée faux (2)	54,55%	62,86%	37,04%
Singularité non détectée vrai (3)	9,09%	5,71%	3,70%
Singularité non détectée faux (4)	3,03%	2,86%	14,81%
Taux de réussite (1+3)	42,42%	34,29%	48,15%
Taux d'erreur (2+4)	57,58%	65,71%	51,85%
Temps de traitement	161 ms	251 ms	260 ms
Type de singularités	Taux de mémorisation		
Nn	100,00%		
Ns	100,00%		
Poche de résine	100,00%		

Tableau 12

On obtient un taux de mémorisation parfait avec cette méthode. Mais le taux de reconnaissance est très mauvais. Cela peut s'expliquer par la taille de la région enregistrée par le classificateur. En effet, comme on ne peut pas changer la taille de cette région et si le nœud est plus petit, on obtiendra une image avec un fond noir qui sera unique. On ne peut donc pas étendre cette méthode sur des nœuds n'ayant pas été rentrés préalablement dans la base de données.

2.3 Conclusion

Le matériel mis à notre disposition pour ce projet nous permet tout à fait de procéder à une détection des singularités du bois par un système de vision industrielle. Cependant, les bases de données que nous avons créées demandent à être étoffées. Cette première phase de traitement statique (à partir de photos) nous donne satisfaction et nous a permis de comparer l'influence des prétraitements sur les images.

Partie 3 : Analyse des résultats

3.1 Procédés utilisés

3.1.1 Classificateur retenu

Nous remarquons que la détection de nœuds sains n'est pas évidente, aucun classificateur ne dépasse 35% de taux de réussite pour la détection des nœuds sains (Tableau 13). Plusieurs facteurs peuvent l'expliquer.

	RGB	MONO 8	MONO8 fond blanc	MONO8 Sharpen Normalize	Isolation singularité
Taux de réussite	29,27%	28,57%	28,21%	31,71%	34,29%

Tableau 13

- Il y a un manque de contraste entre la pièce et le nœud (Figure 32).

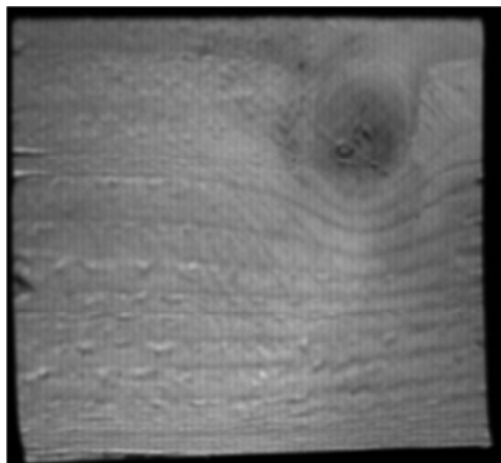


Figure 32 : Nœud sain à faible contraste

- Un nœud sain peut ressembler à un nœud noir. La Figure présente un nœud noir (en fait c'est un nœud sautant). On remarque qu'elle est très similaire à la Figure qui est un nœud sain.

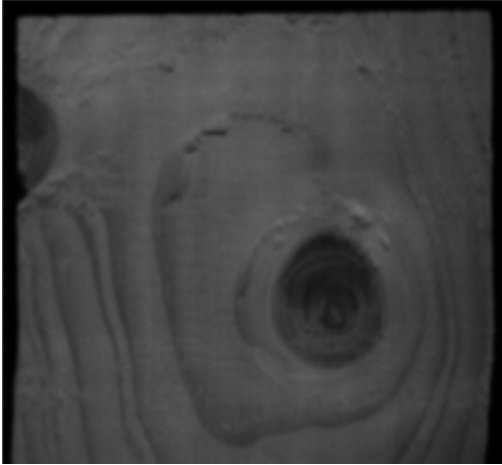


Figure 33 : Nœud noir

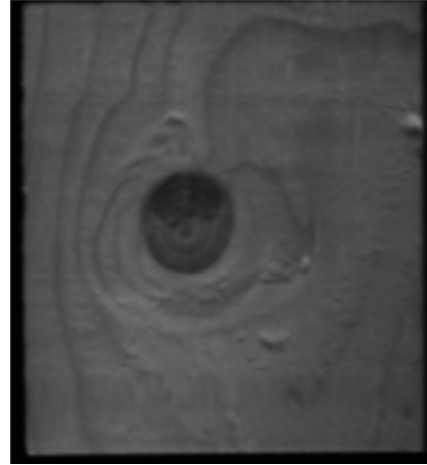


Figure 34 : Nœud sain

Nous avons donc décidé de choisir notre classificateur en fonction de son taux de réussite sur les nœuds noirs. Les nœuds sains ne peuvent pas être détectés avec cette technique.

Nous pouvons également remarquer que nous n'avons pas utilisé qu'une seule essence de bois pour faire notre base de données. Il serait judicieux de créer un classificateur par essence de bois. En effet les caractéristiques visuels du bois varient énormément d'une essence à une autre (veinage, traces vasculaires, nœuds sautants, nœuds sains, poches de résine).

Evaluation des performances :

Nous cherchons le taux d'erreur total pour évaluer la meilleure méthode à utiliser. Ce taux comprend le taux d'erreur de mémorisation calculé sur le lot d'apprentissage et le taux d'erreur de généralisation calculé sur un lot test « inconnu ». Donc le taux d'erreur total vaut :

$$\text{Erreur totale} = \text{erreur de mémorisation} + \text{erreur de généralisation}$$

Au vu des résultats précédents, le taux d'erreur total n'a été calculé que sur la base de données nœuds noirs. Les résultats sont présentés dans le Tableau 14.

Méthodes	Erreur Mémorisation	Erreur généralisation	Erreur totale
RGB	0%	22,50%	23%
MONO8	12,50%	14,63%	27%
MONO8 sharpen normalize	0%	24,39%	24%
MONO8 fond blanc	12,50%	10%	23%
Isolation de la singularité	0%	57,58%	58%

Tableau 14

Un taux d'erreur de mémorisation de 12.5% équivaut à une singularité non mémorisée sur un total de huit enregistrées.

La méthode de prétraitement qui présente le plus faible taux d'erreur pour notre classificateur est la MONO8 avec un fond blanc. Compte tenu de notre faible nombre d'échantillons, ces résultats sont à nuancer.

Actuellement, l'apprentissage effectué est fait de façon « par cœur ». Le problème avec cet apprentissage est de choisir une bonne taille de lot et de prendre des échantillons significatifs pour la classe enregistrée. Pour la détection de défauts cette méthode n'est pas intéressante, car chaque singularité est différente de par sa taille, sa couleur ou encore sa forme.

Pour réaliser l'analyse dynamique, nous avons opté pour une technique de seuillage pour isoler la singularité que nous avons couplée à notre classificateur pour la détection des nœuds noirs. Cette décision s'appuie sur différents essais réalisés avec le logiciel Sherlock7.

3.1.2 Prétraitement de l'image

Dans un premier temps l'image est reconstituée dans une fenêtre à partir des lignes de pixels envoyées par la caméra (Figure 33).

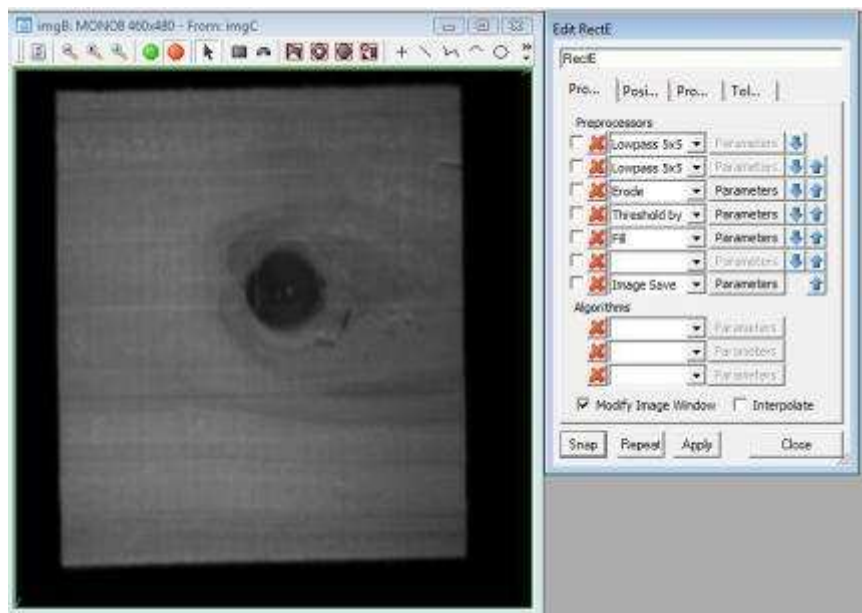


Figure 33 : Reconstitution de l'image dans une fenêtre

L'image est ensuite importée dans une nouvelle fenêtre où nous appliquons les traitements suivants :

Lowpass 5x5 : ce prétraitement permet d'atténuer les contrastes. On l'appliquera deux fois de suite pour obtenir un meilleur résultat (Figure 34).

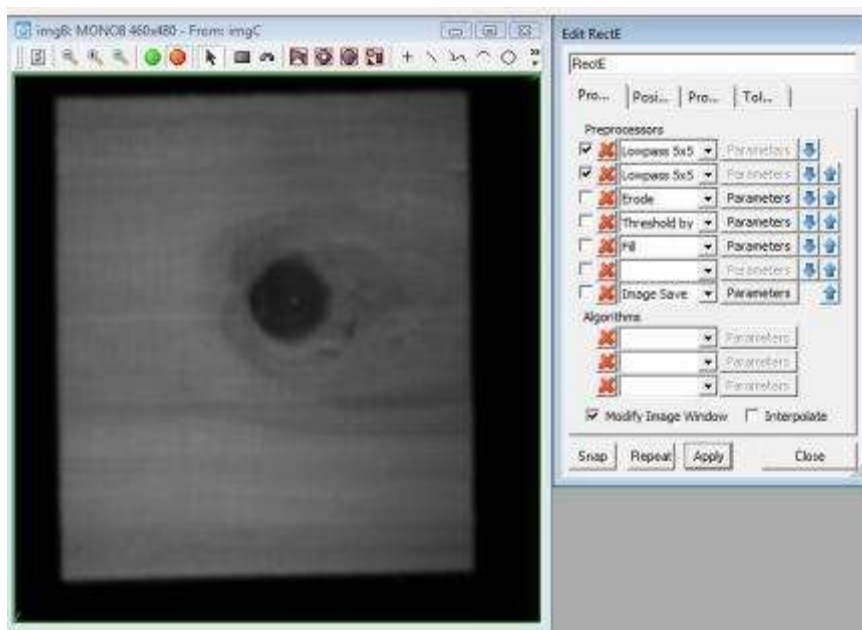


Figure 34 : Prétraitement Lowpass

Erode : L'érosion est un opérateur de base de la morphologie mathématique. Il permet de solidariser les pixels ayant un contraste proche (Figure 35).

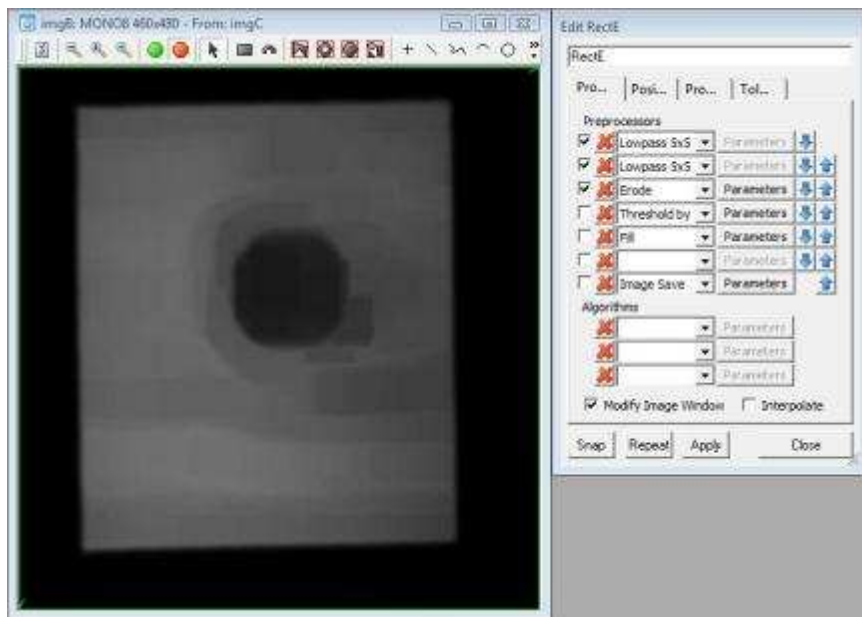


Figure 35 : Prétraitement Erode

Thershold : ce prétraitement est le seuillage, tous les pixels qui ont un contraste inférieur à une valeur paramétrée sont considérés comme étant noirs, les autres sont blancs. On obtient une image bi-niveaux (Figure 36).

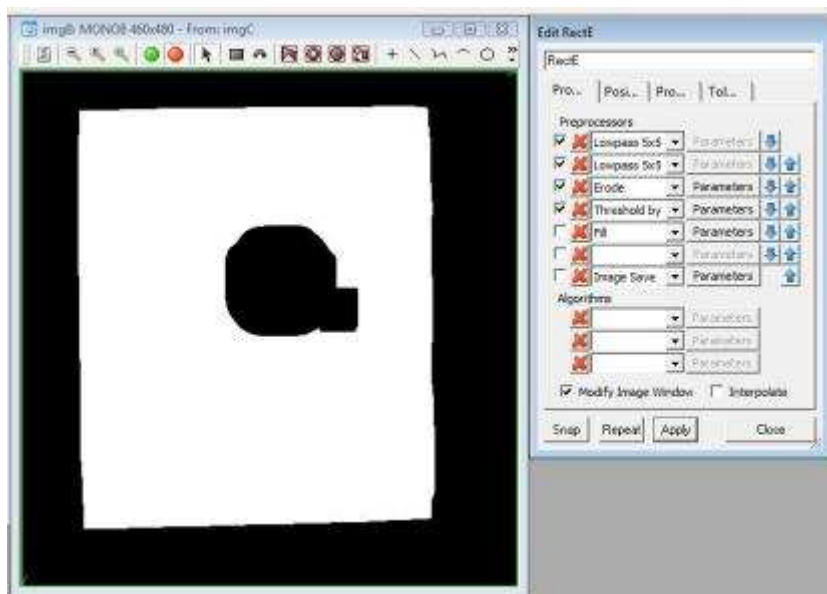


Figure 36 : Prétraitement Threshold

Fil : ce prétraitement nous permet d'éliminer le fond noir. Enfin, Image Save sauvegarde l'image finalement obtenue dans un des buffers disponible de Sherlock (Figure 37).

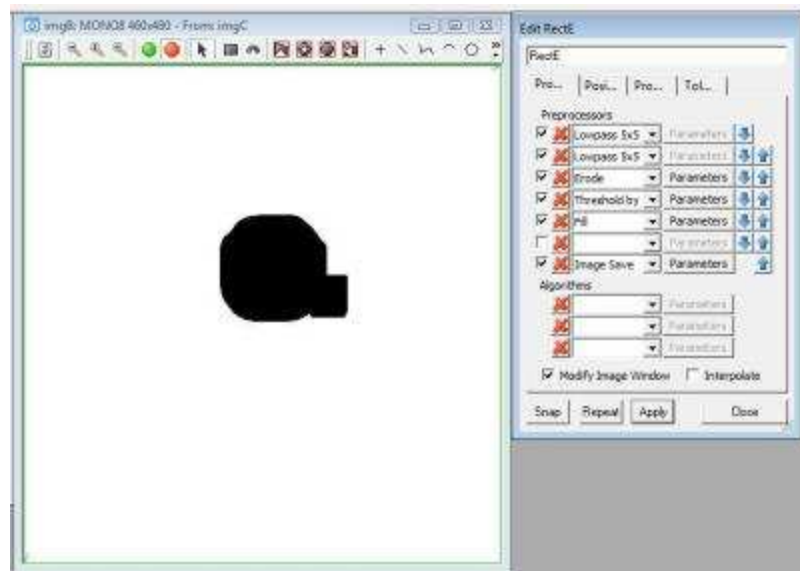


Figure 37 : Prétraitement Fil

Le but de ces prétraitements est de repérer la position des singularités de façon approximative. Les contours ne sont pas francs, il reste un peu de l'éprouvette autour des singularités.

L'image de base est alors envoyée dans une nouvelle fenêtre, une soustraction entre cette image et celle enregistrée dans le buffer en bi-niveaux est alors réalisée.

Nous obtenons alors la singularité en niveaux de gris sur un fond blanc (Figure 38). Nous allons appliquer sur cette image le classificateur pour la détection de la singularité. Ainsi de cette manière on ne perd pas toutes les informations et nous pouvons faire une différenciation entre nœuds noirs et nœuds sains. De plus le fond de l'image étant blanc, le domaine analysé est très réduit, on gagne ainsi du temps de traitement et les erreurs dues au contraste du veinage naturel du bois sont évitées.

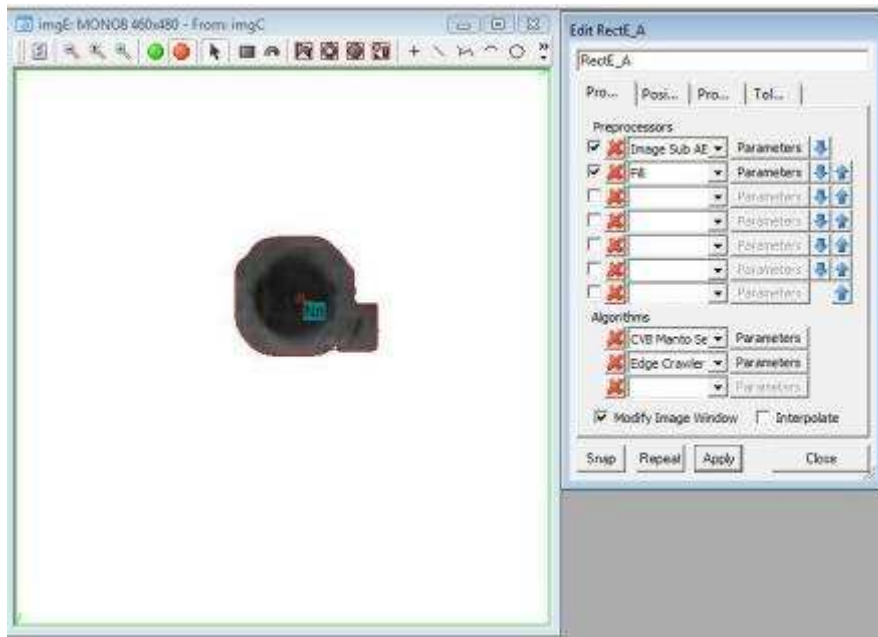


Figure 38 : Soustraction des deux images

3.1.3 Détection des nœuds sains et des poches de résine

Notre classificateur n'étant pas assez performant, nous ne pouvons pas détecter les nœuds sains de cette manière. Nous avons donc mis en place une technique pour pallier à ce problème :

- Nous arrivons à extraire tous les contours avec un seuillage
- Tous les nœuds noirs sont détectés
- Il est possible d'extraire les contours avec un algorithme « Edge Crawler »

Si le logiciel détecte un contour et que celui-ci n'est pas répertorié comme étant un nœud noir, c'est un nœud sain. C'est une technique détournée, mais elle offre un bon résultat final.

La Figure 39 décrit l'algorithme de traitement que nous avons utilisé pour détecter les nœuds sains.

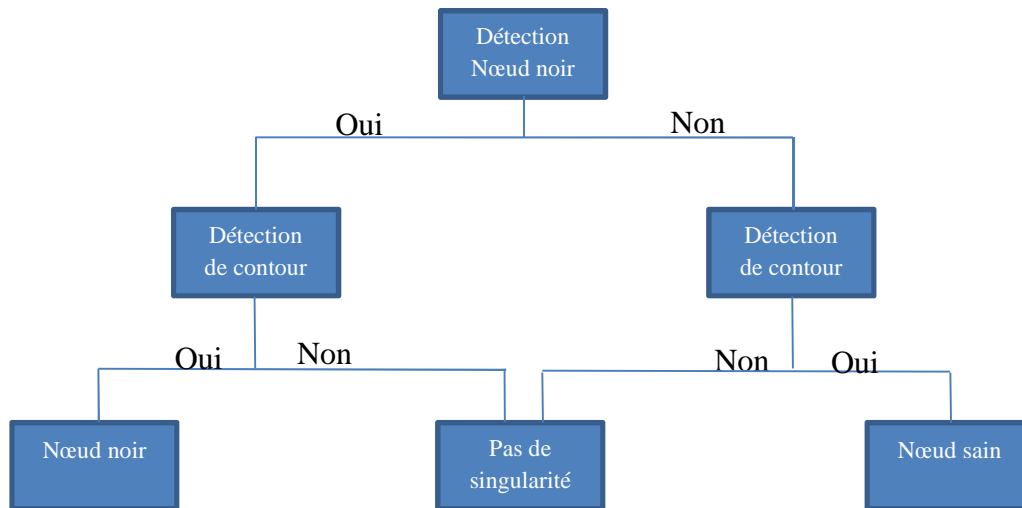


Figure 39 : Algorithme de traitement des nœuds sains

Cette décision est donnée au système par l'intermédiaire d'un simple scripte codé en Java script présentée dans la Figure 40.

```

Script
1  if ((Vars l >= Vars Noir))
2
3  == {
4
5  if (Vars Noeud_noir length > 0) // détection noeuds noirs.
6  == {
7  Vars Variable_Envoi = "1";
8  }
9  if (Vars nbr_contour > 0 && Vars Noeud_noir length == 0) // détection noeuds sains sans manto
10 == {
11 Vars Variable_Envoi = "2";
12 }
13 if (Vars nbr_contour == 0 && Vars Noeud_noir length == 0) // détection de pieces sans défauts
14 == {
15 Vars Variable_Envoi = "3";
16 }
17
18 Vars varA = Vars Noeud_noir length;
19 Vars Envoi_PLC = Vars Variable_Envoi;
20 }
21
22
23
  
```

Figure 40 : Script de décision

Le problème se pose maintenant pour la détection des poches de résine. Notre nombre d'échantillons étant beaucoup trop limité, nous ne sommes pas arrivés à utiliser l'outil CVB Manto avec une base de données. L'idée retenue est d'utiliser la différence géométrique en une poche de résine (Figure 41) et un nœud (Figure 42) pour définir de quelle singularité il s'agit. Ainsi nous pouvons utiliser le rapport entre la largeur de la forme détectée et sa longueur. Si ce rapport est proche de 1, il s'agit d'une forme circulaire donc d'un nœud, si son rapport est très différent, il s'agit d'une forme rectangulaire et donc d'une poche de résine (Tableau 15).

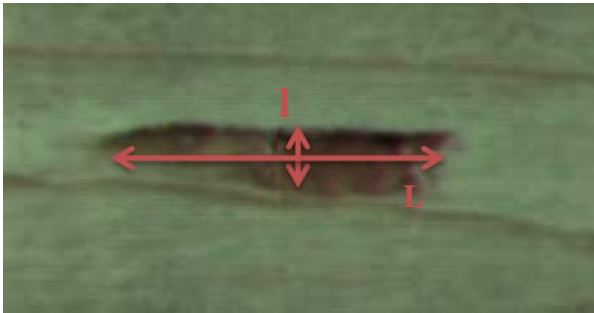


Figure 41 : Longueur et largeur d'une poche de résine

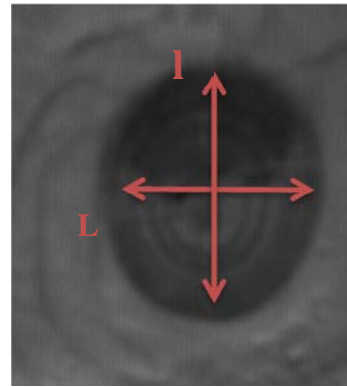


Figure 42 : Longueur et largeur d'un nœud

	Poches de résines	Nœuds sains
Rapport moyen I/L	0,19	0,89
Ecart type	0,08	0,15
Borne supérieure	0,32	1
Borne inférieure	0,07	0,6

Tableau 15

Nous remarquons que la borne supérieure du rapport largeur/longueur des poches de résine est inférieure à la borne inférieure du rapport largeur/longueur des nœuds sains. Il n'y a donc pas de risque de confusion. Cette méthode peut donc être utilisée.

3.2 Exploitation du système de vision en dynamique

Nous allons à présent évaluer la pertinence de nos choix en appliquant notre programme sur des éprouvettes en mouvement. Cette simulation est proche de ce que l'on pourrait rencontrer dans le monde industriel. La Figure 43 montre les différents bacs de



Figure 43 : Classement des éprouvettes

classements.

3.2.1 Problèmes rencontrés

Systeme de convoyage

Le système de convoyage par bande n'est pas satisfaisant. En effet nous rencontrons souvent un effet de « glissement » entre la bande et le rouleau entraineur. Ainsi, lors des acquisitions il nous arrive d'obtenir des images inexploitables (Figure 44). Pour éviter ce problème, l'acquisition a été asservie à un codeur incrémental. Cependant, si le tapis n'avance plus car le glissement est trop élevé, l'acquisition s'arrête. Ce système n'est pas suffisant pour résoudre notre problème et il faudrait envisager un autre système de convoyage, comme par exemple un entrainement à chaîne.

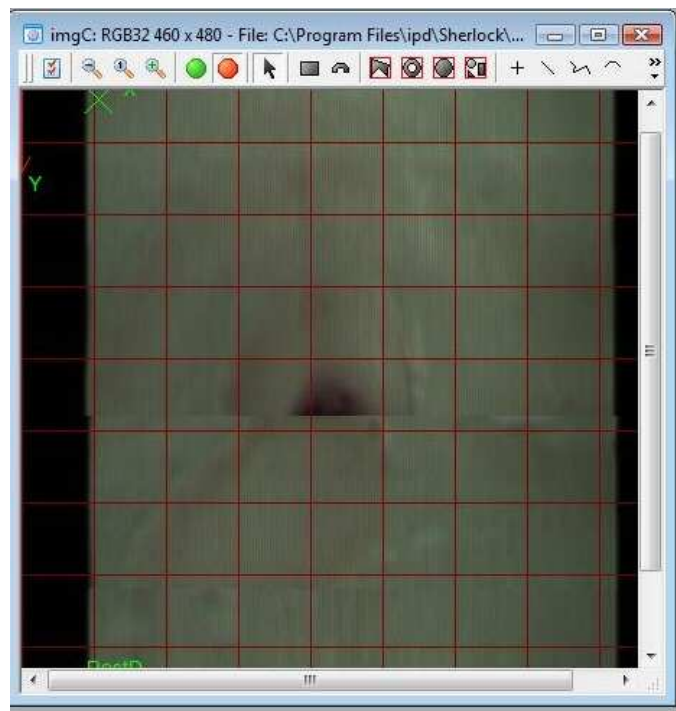


Figure 44 : Problème lié au système de convoyage

Eprouvettes

Les éprouvettes que nous avons usinées proviennent de chutes ainsi les dimensions ne sont pas toutes identiques. Cela nous a posé un problème pour la réalisation dynamique où les éprouvettes doivent être calibrées afin de correspondre aux différents éjecteurs.

3.2.2 Résultats des tests dynamiques

Pour effectuer notre test en dynamique nous avons lancé l'acquisition en continu et les éprouvettes se suivent les unes par rapport aux autres. Elles sont suffisamment espacées pour laisser le temps à l'automate de les orienter vers leurs bacs de réception. Nous avons ajouté des éprouvettes témoins sans la moindre singularité pour évaluer l'efficacité du classificateur.

Les résultats sont présentés dans le Tableau 16. La colonne « correcte » répertorie les fautes du programme. Si l'éprouvette est orientée vers le bon bac, nous notons « 1 » dans cette colonne, dans le cas contraire nous notons « 0 ». Pour déterminer le taux de réussite du système, nous comptabilisons le nombre de « 0 » que l'on ramène au nombre d'éprouvettes.

éprouvette	Orienté vers Bac	correcte
Nn1	1	1
Nn2	2	0
Nn3	1	1
Nn4	1	1
Nn5	1	1
Nn6	1	1
Nn7	1	1
Nn8	1	1
Nn9	1	1
Nn10	1	1
Nn11	1	1
Nn12	1	1
Nn13	1	1
Nn14	1	1
Nn15	1	1
Nn16	1	1
Nn17	1	1
Nn18	1	1
Nn19	1	1
Ns1	2	1
Ns2	2	1
Ns3	1	0
Ns4	2	1
Ns5	2	1
Ns6	1	0
Ns7	2	1
Ns8	2	1
Ns9	2	1
Ns10	2	1
Ns11	1	0
Ns12	2	1
Ns13	2	1
Ns14	2	1
Ns15	2	1

Ns16	2	1
Ns17	2	1
Témoin 1	3	1
Témoin 2	3	1
Témoin 3	3	1
Témoin 4	3	1
Témoin 5	3	1
Témoin 6	3	1

Tableau 16

Nous obtenons seulement quatre décisions fausses. Nous avons donc un taux de réussite de 90%. Ces résultats peuvent toutefois varier selon le nombre d'éprouvettes témoins que l'on choisit. Notre système est très performant pour détecter la présence d'une singularité, il a un peu plus de mal à les trier ensuite.

3.3 Améliorations envisagées

3.3.1 Amélioration du classificateur

Pour poursuivre ce projet, et améliorer le classificateur, il serait nécessaire de compléter la base de données. Il faudrait des échantillons plus caractéristiques. Notre choix de réunir les nœuds sautants et les nœuds cernés dans la même base de données est source d'erreurs. Peut-être faudrait-il envisager l'utilisation de la couleur pour pouvoir différencier les différents types nœuds entre eux. Après isolation de la singularité, il faudrait calculer la moyenne de la couleur à l'intérieur de chaque contour, cette étape peut être effectuée avec le logiciel Sherlock. Avec un nombre suffisant d'éprouvettes, il semble envisageable de dégager des seuils correspondants à chaque singularité. Comme nous pouvons le voir sur la Figure et la Figure , il y a une différence notable de couleur entre les nœuds noirs et les nœuds cernés.



Figure 47 : Nœud cerné



Figure 48 : Nœud noir

Il serait également intéressant de diversifier les essences de bois et de créer les bases de données par essences. Ainsi les erreurs ayant pour causes les différences de contrastes dues à la couleur du bois seraient atténuées.

3.3.2 Détection des fentes

La détection d'une fente dans le bois peut être faite comme la différence entre nœud noir et nœud cerné (expliqué précédemment). Une fente apparaît toujours en noir sur les images (Figure 45). Cela est causé par la profondeur du défaut.



Figure 45 : Fentes

On se base sur la méthode de détection envisagée pour les poches de résines. Sherlock détecte une forme rectangulaire. Il regarde ensuite la couleur de cette forme. Si la couleur est noire, c'est qu'il s'agit d'une fente. Si la couleur est plus claire, alors il s'agit d'une poche de résine.

3.3.3 Recherche de la position de la singularité

Pour chaque défaut identifié, il serait intéressant de pouvoir déterminer leur position. Actuellement avec Sherlock nous avons le point d'origine en haut à gauche de l'image. Mais ce n'est pas le coin en haut à gauche de la pièce. Sherlock fournit les coordonnées du centre du contour détecté. Il pourrait y avoir deux moyens de procéder pour déterminer le centre du défaut. La première (Figure 46), consisterait à déterminer les longueurs que représentent ici les flèches 1 et 2. Il suffirait de soustraire ces longueurs aux longueurs des flèches 3 et 4. On obtiendrait donc : coordonnée en $X=4-2$ et coordonnée en $Y=3-1$.

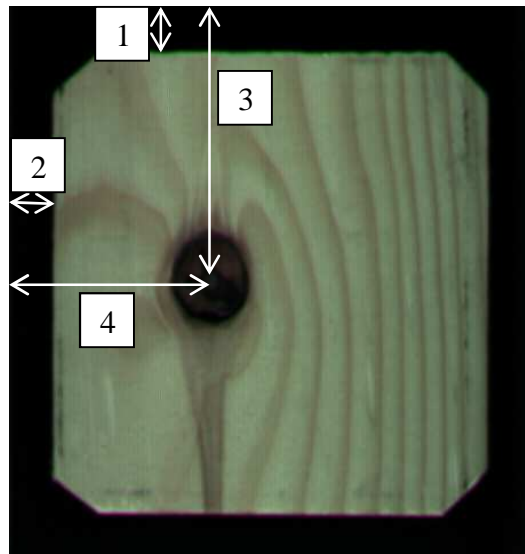


Figure 46 : Localisation de la singularité méthode 1

La deuxième méthode (Figure 47), consisterait à forcer le déplacement de l'origine par Sherlock. Ainsi, l'origine ne serait plus en haut à gauche de l'image mais en haut à gauche de la pièce. Une nouvelle origine serait faite pour chaque pièce qui passerait sous la caméra.

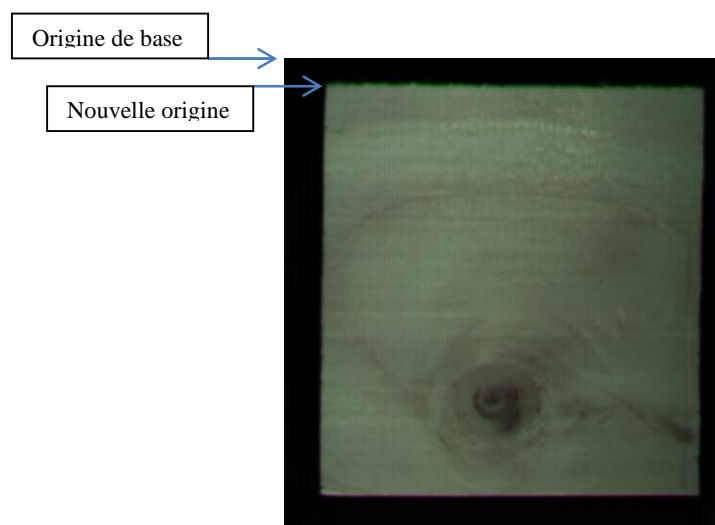


Figure 47 : Localisation de la singularité méthode 2

Dans les tous les cas, les coordonnées de la singularité resteraient approximatives. Il faudrait que la pièce soit parfaitement droite sur le tapis pour que les coordonnées soient précises.

3.3.4 Amélioration de l'éclairage

Un autre aspect positif qui pourrait être apporté est l'intégration de l'éclairage au support caméra. Actuellement l'éclairage a un angle incident sur les pièces (Figure 49 et Figure 48).

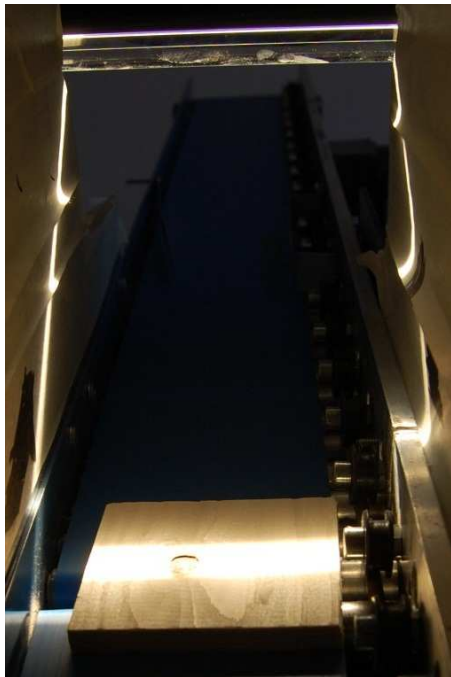


Figure 49 : Incidence de l'éclairage



Figure 48 : Inclinaison de système d'éclairage

Cela peut créer des zones d'ombres sur la pièce (Figure 50 : Ombre liée au système d'éclairage). Etant donné que le contraste a une grande importance lors du traitement, une zone d'ombre peut fausser les identifications faites par Manto (Figure 50).

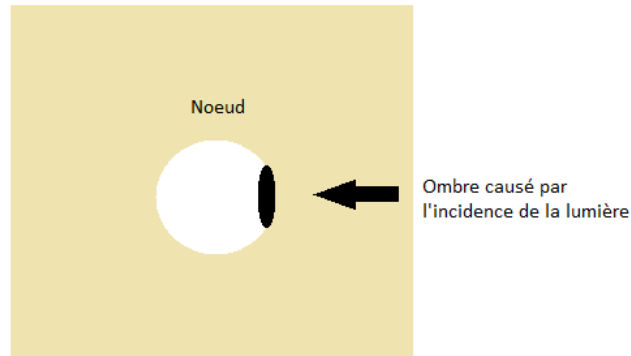


Figure 50 : Ombre liée au système d'éclairage

En positionnant l'éclairage perpendiculairement à l'éprouvette, ce problème disparaît. Cependant cette réalisation n'est pas facile à faire car le système de vision est déjà implanté.

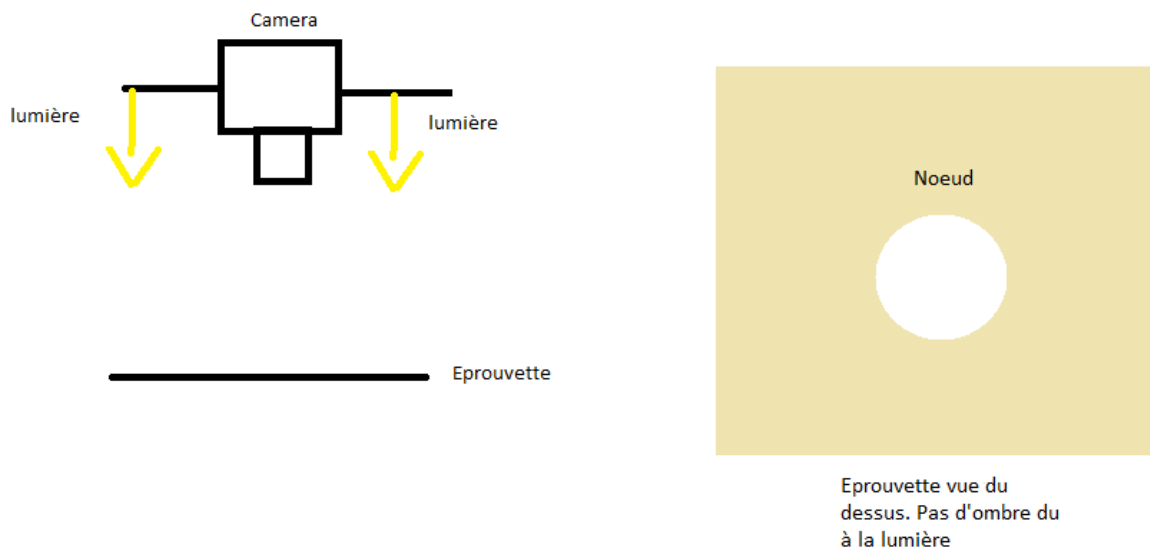


Figure 51 : Implantation du système d'éclairage perpendiculaire à la pièce

Conclusion

A travers ce projet, nous présentons une solution pour la détection de singularités du bois en utilisant exclusivement des logiciels non dédiés. Cependant, notre détection n'est pour l'instant efficace que pour des nœuds. Nous donnons quelques pistes d'amélioration pour rendre notre programme encore plus performant.

D'un point de vue personnel, ce projet a su mettre en pratique les connaissances que nous avons pu acquérir durant ces trois ans grâce au cours sur l'anatomie du bois, la reconnaissance de forme et le contrôle non destructif du bois.

Le projet de fin d'études est une conclusion de notre vie étudiante et une introduction à notre vie professionnelle en nous montrant ce qu'est le réel travail en autonomie.

Bibliographie

- [1] J.W. FUNCK, et al, *Image segmentation algorithms applied to wood defect detection.*- Corvallis: Oregon state University, 2003.-23p.
- [2] Cyril MAZAUD, et al, *Modélisation de connaissances « métier » pour le paramétrage d'un Système Flou d'identification de défauts sur des planches de bois.* Vandoeuvre les Nancy : CRAN, 2006.- 7p
- [3] Christine MALOIGNE-FERNANDEZ, Richard CROCHE. *Classification de défauts observés dans le bois par caméra.* sl, 1988.- 8p.
- [4] Pablo A. ESTÉVEZ, Claudio A. PEREZ, Eric GOLES. Genetic input selection to a neural classifier for defect classification of radiata pine boards. *Forest products journal*, 2003.- Vol. 53, No. 7/8.-p.87-94.
- [5] Vincent BOMBARDIER, Pascal LHOSTE, Cyril MAZAUD. *Modélisation et intégration de connaissances métier pour l'identification de défauts par règles linguistiques floues.* Vandoeuvre les Nancy : CRAN, 2004.- 21p.
- [6] Vincent BOMBARDIER. *Traitement Numérique de l'Image : Interprétation – Décision - Reconnaissance de Formes.* Vandoeuvre les Nancy, sl, sd.- 20p.
- [7] F.CAZET, V.BOMBARDIER, P-J MEAUSONNE. *Modelisation de connaissances appliquée à la caractérisation de la texture et de la couleur du placage.* Vandoeuvre les Nancy : CRAN, LERMAB, sd.-7p.
- [8] HUBER, A.H., S. RUDDELI et C.W. MCMILLIN. Industry standards for recognition of marginal wood defects. *Forest Products Journal*, 1990. vol. 40, n° 3, p. 30-34.
- [9] AFNOR. *Bois sciés - Classement d'aspect des bois résineux - Partie 1 : épicéas, sapins, pins et Douglas européens.* NF EN 1611-1/A1, Mars 2003, 8p.
- [10] CNDB BOIS CONSTRUCTION. *Les classements d'aspect et d'utilisation en structure.*- sl, sd.- 12p.
- [11] AFNOR. *Règles d'utilisation du bois dans la construction - Classement visuel pour l'emploi en structures des bois sciés français résineux et feuillus - Partie 1 : bois massif.* NF B52 001, Aout 2011,22p.
- [12] William UNDERWOOD. *Etude et intégration du module d'acquisition d'images couleur.*- Epinal : ENSTIB, 2011.-69p.

Annexe

Tutorial Manto

Dans un premier temps, il faut ouvrir le logiciel Manto teach (figure 56).

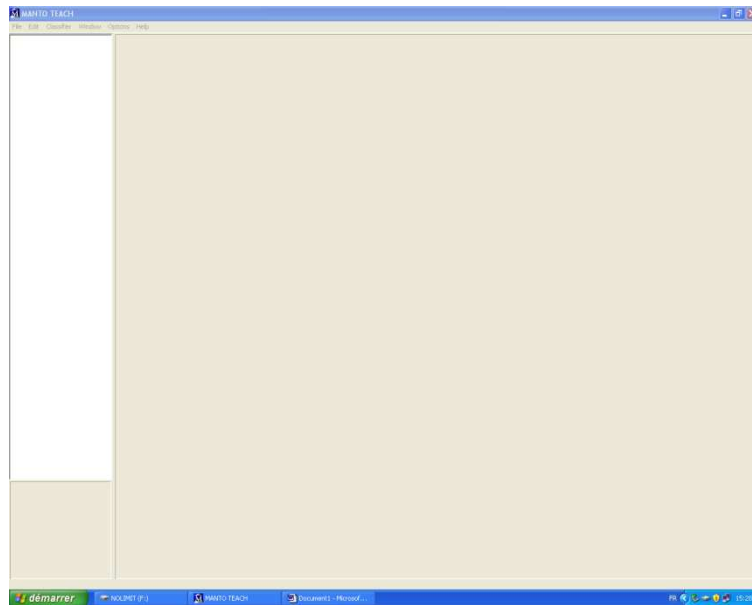


Figure 52

A l'ouverture, allez dans *fichier, new SIL* (figure 57). Entrez le nom de la base de données que vous allez rentrer. Sélectionnez *Load image* et choisissez une image référence. Sur cette image sélectionnez la taille du cadre. Sur cet exemple la taille du cadre sera le périmètre du nœud. Ensuite appuyer sur *Next*.

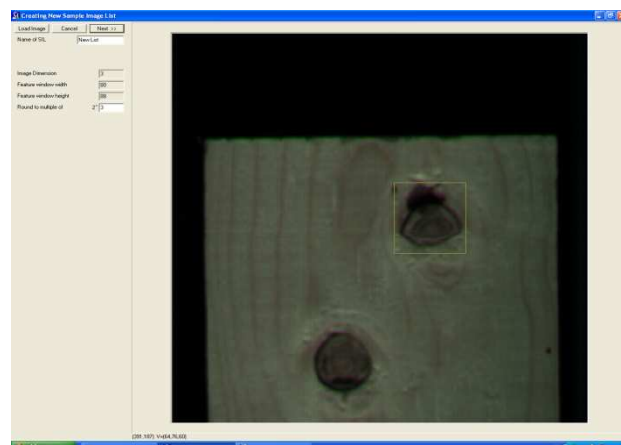
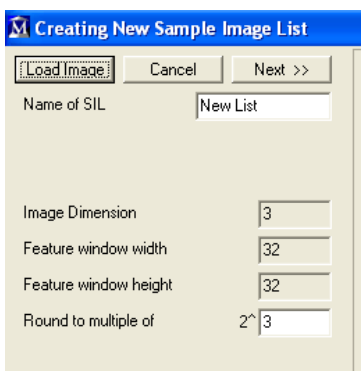


Figure 53

Il faut ensuite modifier le paramètre Left and right rotation DAlpha (figure 58). Il est préférable de le mettre à 90°. Cela veut dire que le logiciel prendra en compte les rotations du cadre dans ses calculs de 0° à 360°. Homogeneous texture ne doit pas être coché car sinon il ne recherchera pas à trouver un objet mais plutôt à identifier une texture.

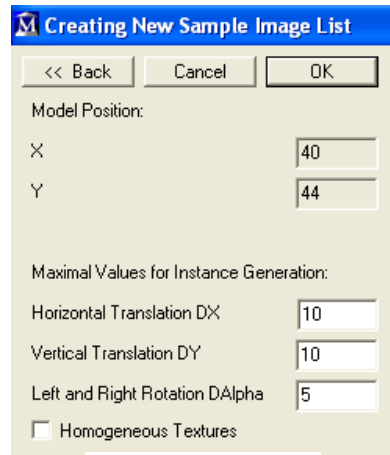


Figure 54

Il faut ensuite entrer la base de données. Dans un premier temps, placer le cadre jaune que vous avez défini juste avant sur la partie de l'image qui vous intéresse. Ensuite il faut faire un clic droit sur le SIL que vous venez de créer (figure 60) et faire *add a new sample*. Dans la fenêtre qui s'affiche rentrez le nom du type d'enregistrement (figure 59). Par exemple, nous donnons le nom « nœuds » à ce type. Il faut mettre un maximum d'images de type « nœuds noirs » pour faciliter le logiciel à les reconnaître. Vous pouvez ensuite rajouter d'autres types comme par exemple « nœuds sains ».

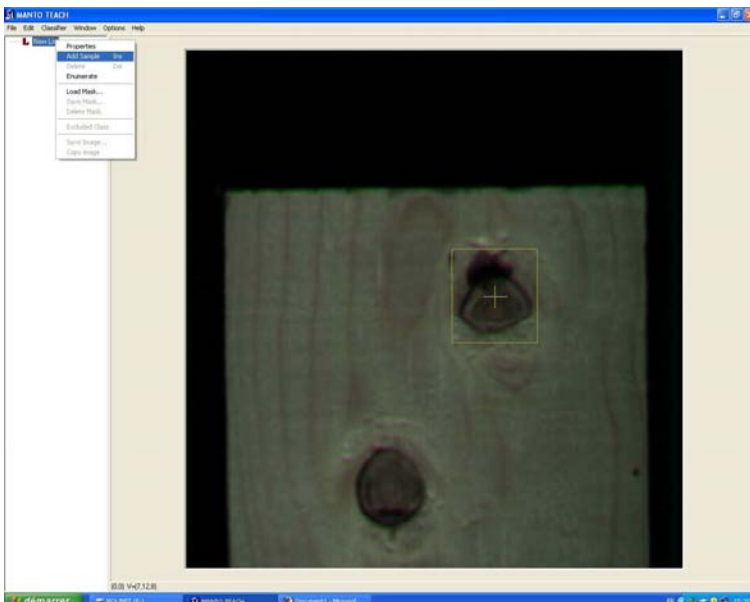


Figure 56



Figure 55

Un autre type doit être rajouté pour éviter les parasites. Par exemple la catégorie « exclu » (figure 61). Dans cet exemple nous enregistrons des images qui ne comportent pas de défauts. Plus cette base d'images est grande plus le logiciel sera précis.

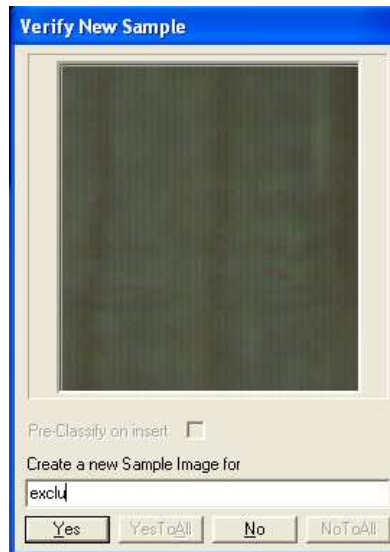


Figure 57

Une fois la base de données terminée, il faut l'apprendre au logiciel. Pour cela allez dans *Classifier, Learn*. Ensuite cliquez sur *code optimization* (figure 7), puis sur *Start optimization* (figure 8). Sélectionnez la ligne avec le learning succes le plus haut et validez. Si la valeur du learning succes est inférieure à 0.9 alors il ne faut pas continuer et compléter d'avantage la base de données. Cliquez sur *Learn* et, une fois l'apprentissage terminé, validez.

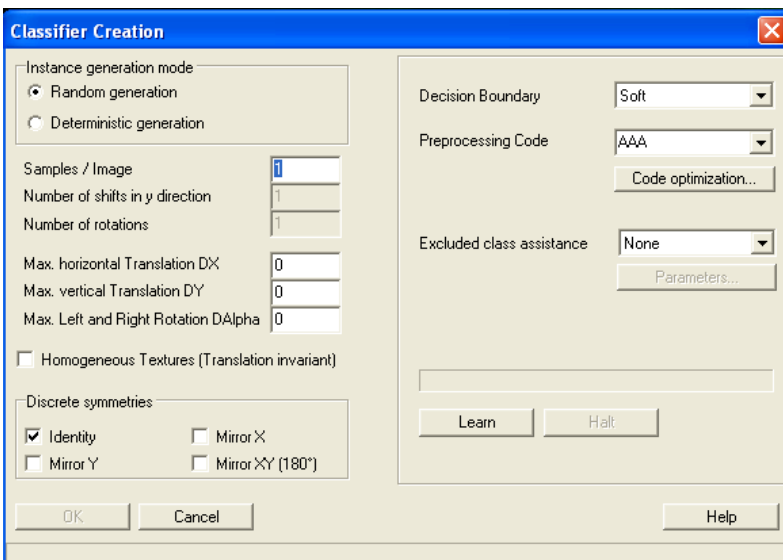


Figure 58

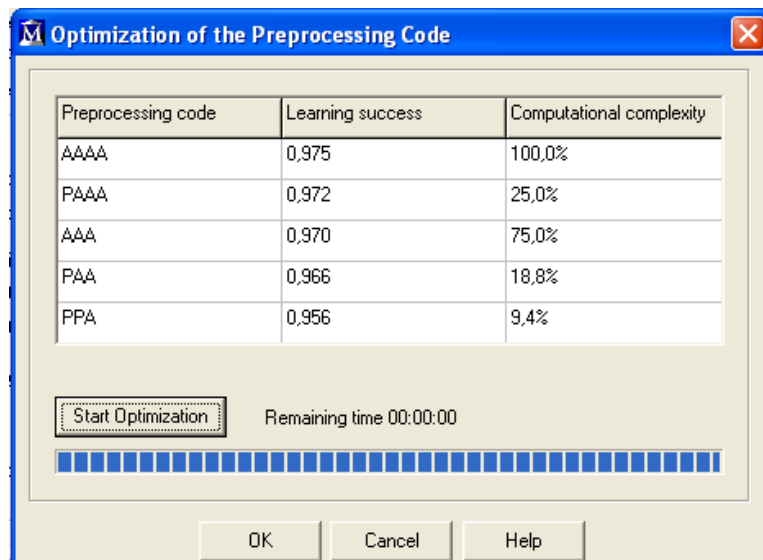


Figure 59

La partie qui suit est l'identification des objets appris. Pour cela allez dans classifier, *Search test*. Sélectionnez l'onglet *search all* (figure 9) puis cochez la case *vote*. Vote permet au logiciel de comparer les différents types entre eux et de sélectionner le meilleur. Le vote augmente le temps de recherche parce qu'il y a plus de possibilités à tester. Ensuite dans *excluded class* sélectionnez le type exclu pour que le logiciel ne note pas les endroits exclus sur l'image.

Il faut ensuite régler les paramètres *Qualitythreshold*, *Separation* et *Granularity*.

Qualitythreshold : le seuil de qualité définit la limite en dessous duquel les résultats seront ignorés.

Separation : La séparation vous permet de spécifier une distance minimale entre deux résultats positifs. Cela vous permet d'éliminer les nuages inévitables des résultats qui se forment autour des centres des objets dans l'image. Avec la séparation, la meilleure correspondance dans le rayon spécifié est choisie pour être l'objet rapporté par la suite.

Granularity : Ce paramètre est un peu compliqué donc nous laissons la valeur par défaut.



Figure 60

Pour finir, toujours sur cette même page, vous ouvrez l'image que vous voulez étudier et vous allez ensuite dans *search*, *search all*. Si des objets doivent être identifiés ils apparaîtront sur l'image (figure 10).

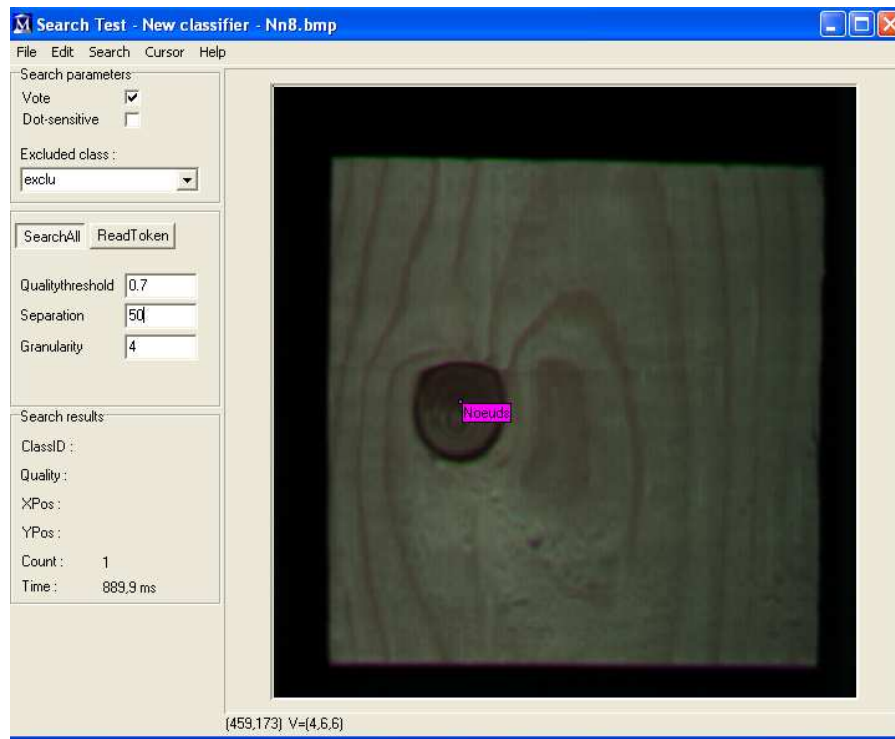


Figure 61

Intégration de l'application Manto sur le logiciel Sherlock

Après beaucoup de temps passé au téléphone avec le service technique de Stemmer Imaging, nous sommes arrivés à intégrer l'application Manto sur le logiciel Sherlock. La suite de ce tutorial n'explique pas comment programmer sherlock mais comment utiliser Manto sur Sherlock.

Une fois le logiciel Sherlock ouvert, créez un rectangle (ROI) sur l'image où vous souhaitez faire la reconnaissance. Ensuite double cliquez sur le rectangle une page s'ouvre où vous pouvez appliquer des filtres et des algorithmes (figure 66). Allez dans algorithme et sélectionnez CVB Manto.

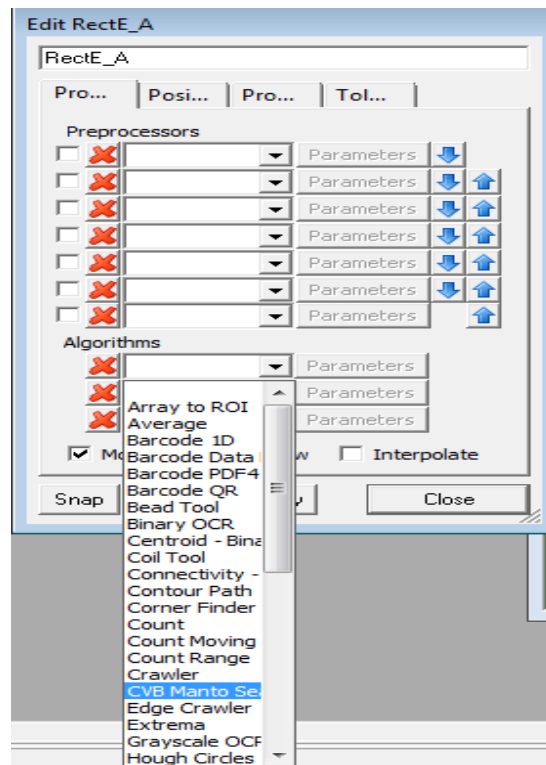


Figure 62

Ensuite cliquez dans *paramètres*. Une nouvelle page va s'ouvrir (figure 67).

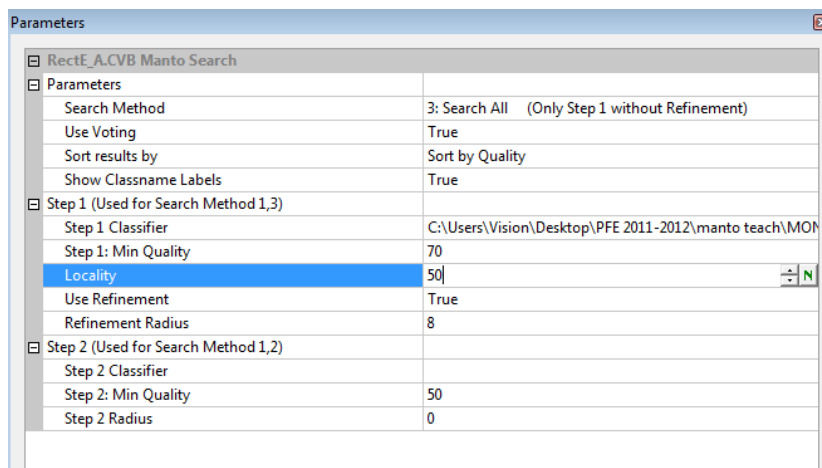


Figure 67

Search method : Trois méthodes existent. Nous utiliserons la méthodes search all car c'est cette méthode que l'on utilise avec l'application Manto seul.

Use voting : C'est le *vote* de l'application Manto seul. Mettre ça valeur sur True.

Sort result by : Sélectionnez sort by quality.

Show classname labels: le nom de la singularité apparaîtra sur l'image si cette valeur est mise sur true.

Step 1 classifier : il faut rentrer la localisation du fichier .mcf que vous avez créé sur l'application Manto seul. Il faut faire attention dans l'ordre où vous créez vos base de données sur Manto car ici, seul le premier onglet est pris en compte. C'est-à-dire que si vous avez créé deux classes (nœuds noirs et exclu par exemple) seul la première classe apparaîtra sur sherlock (soit nœuds noirs).

Step 1 min quality et localisation : Ce sont les mêmes paramètres que sur Manto. Nous avons choisi 70 et 50 pour ces paramètres.

Refinement et step 2 ne sont pas utilisés quand on utilise la méthode search all.

Une fois tout les paramètres rentrés, cliquez sur ok et vous pouvez lancer le processus en dynamique. Une fois l'image acquise le logiciel vous indiquera la singularité sur l'image.

Résumé

Aujourd'hui le contrôle non destructif est une méthode importante pour pouvoir classer le bois. Dans le but d'améliorer la compétitivité des entreprises, de nombreux systèmes ont été mis au point. Un certain nombre de ces méthodes sont effectuées avec l'aide de machines comme par exemple les ultrasons, les rayons X, les micro-ondes et la vision par caméra.

Ce rapport présente une méthode de détection de singularités du bois par l'intermédiaire d'un système de vision industrielle. Elle a été mise en place lors d'un projet de fin d'étude sur une plateforme expérimentale appartenant au Centre de Recherche en Automatique de Nancy (CRAN). Le but est de détecter les singularités du bois par le biais d'une caméra, pour pouvoir ensuite les classer, en apportant un maximum de précisions comme leurs types, leurs tailles et leurs emplacements.

Dans ce rapport nous détaillons les différents procédés que nous avons mis en place pour arriver à détecter une singularité. Deux méthodes ont été envisagées, la première utilisant un logiciel nommé *Sherlock*, permettant de faire l'acquisition et le traitement des images. La seconde utilisant *Sherlock* pour l'acquisition couplée à un autre logiciel nommé *CVB Manto* pour le traitement de haut niveau. Nous exposons les différences entre ces deux méthodes ainsi que les résultats obtenus.

Pour finir nous expliquons comment intégrer le logiciel *CVB Manto* dans *Sherlock* pour pouvoir travailler de façon dynamique.

Mots clés : Identification, traitement d'images, caméra linéaire couleur, nœud, poche de résine, bois

Abstract

Nowadays, non-destructive control is a very important method to classify wood. Many various systems have been developed aiming at improving companies' competitiveness. Some of these methods imply the use of computer devices such as ultrasonic sounds, X rays, micro-waves and computer vision.

This report outlines a methodology to detect wood singularities using a computer vision device. During a school project on an experimental platform, this very method has been introduced by the *Centre de Recherche en Automatique* of Nancy (CRAN). The aim is to detect wood singularities with a video camera, on order to classify them, giving a maximum of information such as their types, their sizes and their locations.

In this report we give details of the various techniques used to detect a singularity. Two different ways are considered: the first one using a software called *Sherlock* to proceed both to the acquisition and the treatment of the image. The second, using *Sherlock* for the acquisition coupled with another software called *CVB Manto* for the high level treatment. We explain the differences between these two methods and the results we obtain. Finally, we explain how to link *Sherlock* and *CVB Manto* in order to work in a dynamic way.

Key words: Identification, image treatment, linear color video camera, knot, pitch-pocket, wood