

Utilisation des treillis de Galois pour la caractérisation d'ensembles d'images

Characterization of Image Sets : The Galois Lattice Approach

E. Zenou^{1,2}

M. Samuelides¹

¹ École Nationale Supérieure de l'Aéronautique et de l'Espace

² LAAS - CNRS

ENSAE - SupAéro, 10 avenue édouard Belin, BP 4032, 31077 Toulouse, France

LAAS - CNRS, 7 avenue du Colonel Roche, 31077 Toulouse cedex 4, France

zenou@supaero.fr, samuelid@supaero.fr

Résumé

Ce papier présente une nouvelle méthode pour la classification supervisée d'ensembles d'images. À chacun de ces ensembles est attaché, pour le caractériser et le distinguer des autres, une entité appelée amer. Différentes caractéristiques, déduites de primitives visuelles, sont extraites de chaque image, et la relation qui lie certaines images à certaines caractéristiques sont structurées et hiérarchisées grâce à un formalisme mathématique appelé treillis de Galois (ou treillis de concepts). De tels treillis permettent de sélectionner des combinaisons de caractéristiques qui serviront d'amers visuels pour chaque ensemble d'images. Cette approche s'inscrit dans le cadre de la robotique mobile autonome et de la localisation topologique dans un environnement structuré, pour laquelle chaque ensemble d'images est issu de d'une même pièce qu'il est nécessaire de caractériser. Ce papier expose cette démarche originale et les résultats obtenus après expérimentation.

Mots Clef

Treillis de Galois, Treillis de concepts, Amers visuels, Robotique mobile autonome, Navigation, Localisation.

Abstract

This paper presents a new method for supervised image classification. To each image set is attached a landmark with the aim to characterize and to distinguish it from each others. The different properties, deduced from image primitives and their relationships with the sets of images are structured and organized into a hierarchy thanks to an original method relying on a mathematical formalism called Galois (or Concept) Lattices. Such lattices allow us to select properties as landmarks of specific classes. This approach belongs to the mobile robotic field and topological localization in a structured environment, for which each set of images is issued from one place that must be characterized. This paper exposes this original process and re-

sults after experimentation.

Keywords

Galois Lattices, Concept lattices, Visual Landmarks, Autonomous Mobile Robotics, Navigation, Localization.

1 Introduction

Caractériser et reconnaître un lieu dans un environnement structuré comprenant plusieurs pièces, en utilisant seulement quelques images issues de chaque pièce, est un exercice difficile pour une machine en général et un robot en particulier. Pour cela, il est nécessaire que la machine trouve "quelque chose" qui lui permette, d'une part, de caractériser chaque pièce, et d'autre part, de les distinguer les unes par rapport aux autres. Ce "quelque chose", sous certaines conditions, s'appelle un *amer*, et plus précisément dans notre cadre un *amer visuel*. Mais qu'est-ce qu'un amer ? Comment le trouver ? Comment le sélectionner ? Cet article présente une nouvelle méthode pour répondre à ces questions. Toutes les images issues d'une même pièce sont regroupées dans un ensemble, et on obtient ainsi plusieurs ensembles d'images à caractériser. Dans un premier temps, pendant la phase d'apprentissage, les relations entre des ensembles d'images et des caractéristiques sont structurées et hiérarchisées à travers un formalisme appelé *treillis de Galois* (ou *treillis de concepts*). L'utilisation d'une telle structure permet à la machine de déterminer ses propres amers attachés à chaque ensemble (chaque pièce). Dans un deuxième temps, en utilisant de nouvelles images issues du même environnement, la machine sera capable de retrouver à quel ensemble (quelle pièce) elles appartiennent.

Dans notre application, le lien entre des ensembles d'images d'un côté et des images issues de pièces d'un environnement structuré de l'autre côté est évident. Cependant la technique développée ici peut être appliquée à tous ensembles d'images pour d'autres types d'applications.

Ce papier est organisé comme suit : les sections 2 et 3 introduisent les amers, les primitives et les caractéristiques ; section 4 présente la théorie des treillis de Galois, et dans les sections 5 et 6 on expose comment ce formalisme est utilisé. La section 7 montre nos résultats, suivis d'une discussion générale et d'une conclusion section 8.

2 Les amers visuels

Comme défini dans le dictionnaire Hachette, un **amer**, terme maritime, est *tout point des côtes très visible (clocher, balise, etc.), porté sur une carte, servant de repère pour la navigation*. Mais dans la langue de Shakespeare, le terme de "*landmark*" revêt un sens plus général, comme un bâtiment important, un lieu à visiter [17] voire même une date importante sur le plan historique. Dans le monde de la robotique mobile, la définition la plus proche est celle du *point de repère* au sens strict, que le robot n'a pas spécialement besoin de visiter, mais surtout de reconnaître. Ainsi trouve-t-on, dans la littérature consacrée, différentes définitions d'amers telles que "des signes distinctifs d'une image qui peuvent être aisément reconnus dans une seconde image obtenue d'un point de vue différent" [19], ou plus simplement "un objet visuel identifiable dans un environnement" [3]. Habituellement les amers ne sont pas introduits par une définition formelle mais plutôt à travers des propriétés spécifiques comme "facilement distinguable" (saillant) [9] ou "localement unique". Dans un sens plus concret, un amer peut être un objet [10], une couleur [28], des points d'intérêt [26], etc.

Plusieurs classifications d'amers existent, notamment celle qui différencie les amers naturels des amers artificiels, mais nous introduisons ici une autre classification liée aux capacités de perception et de reconnaissance de la machine. Nous séparons les amers en trois catégories :

- les amers entièrement prédéfinis : une base d'images représentant des objets est introduite dans le robot [19, 22] qu'il "n'a plus qu'à" reconnaître ;
- les amers partiellement prédéfinis : ces amers potentiels ont une structure commune. Par exemple, dans [16], les amers sont des formes quadrangulaires planes (type affiches murales) que le robot apprend à identifier et à reconnaître ;
- les amers non prédéfinis : aucune hypothèse n'est formulée *a priori* pour les amers potentiels. Les principaux travaux ayant une telle approche sont essentiellement d'inspiration biologique [18, 1, 13]

Notre démarche s'inscrit dans cette dernière catégorie : le robot doit pouvoir choisir dynamiquement les amers les plus pertinents de façon autonome.

3 Primitives, caractéristiques et propriétés

Différentes images sont extraites de chaque pièce d'un environnement structuré ; ainsi, à chaque pièce, est attaché un ensemble d'images qu'il est nécessaire de caractériser.

De chacune de ces images, des primitives sont extraites afin d'en déduire des *caractéristiques* de l'image, pour aider le robot à trouver des *propriétés* du lieu. Ainsi les caractéristiques se distinguent des propriétés par le fait qu'elles appartiennent à l'image, alors que les propriétés appartiennent au monde réel.

Trois types de primitives sont extraites des différentes images :

- les primitives structurelles : des segments, avec leur taille et leur orientation (issus des approximations polygonales de contours), etc.
- les primitives colorimétriques : on extrait les pixels rouges, verts, bleus, jaunes, etc.
- les primitives photogrammétriques, venant de chaque pixel de l'image : textures, points d'intérêt, etc.

On construit et on définit des caractéristiques à partir de ces primitives. On peut noter que la définition des caractéristiques est générale et inclut toute caractéristique potentielle, qu'elle appartienne aux images d'intérêt ou non. leur pertinence relèvera de la connaissance *a priori* que nous avons sur l'environnement. Par exemple, le fait qu'"il y a beaucoup de rouge dans cette image" peut être une caractéristique, comme le fait qu'"il y ait telle texture". Cependant, nous nous limitons ici à des caractéristiques les plus invariantes possibles en rotation, translation, et échelle. C'est pourquoi la présence de tel ou tel segment n'est pas significative, mais une caractéristique potentielle sera plutôt le fait qu'"il y ait un grand nombre de segments parallèles et de même longueur" (typiquement pour une bibliothèque).

4 Les Treillis de Galois

Les treillis de Galois ont été largement utilisés en Intelligence Artificielle ces vingt dernières années. Cette théorie a été développée sous le nom d'*Analyse Formelle de Concepts*. Plusieurs algorithmes constructifs ont été mis au point depuis lors [20], et certaines applications concrètes sont apparues récemment, principalement en fouille de données [27, 21], dans le domaine aéronautique [7]. Nous proposons ici une application au traitement d'images dans le contexte de la robotique mobile autonome.

4.1 Formalisme mathématique (1) [2][12]

Définition 1 Un **treillis** est un ensemble ordonné dans lequel deux éléments quelconques ont une borne supérieure (BS, ou *sup*) et une borne inférieure (BI, ou *inf*). Un treillis complet est un treillis pour lequel tout sous-ensemble possède une BS et une BI.

Par exemple, l'ensemble $\mathcal{P}(\mathcal{O})$ des parties de $\mathcal{O}$ ordonné par l'inclusion est un treillis complet.

Définition 2 Un contexte $\mathcal{K}$ est un triplet $(\mathcal{O}, \mathcal{F}, \zeta)$ pour lequel $\mathcal{O}$ est un ensemble d'objets, $\mathcal{F}$ un ensemble d'attributs et ζ une application binaire de $\mathcal{O} \times \mathcal{F}$ dans $\{0, 1\}$.

Dans notre application, les objets sont les images appartenant aux différents ensembles situés, les attributs sont les caractéristiques et l'application ζ est définie par la présence d'une caractéristique $f \in \mathcal{F}$ dans une image $o \in \mathcal{O}$; on a alors $\zeta(o, f) = 1$.

Définition 3 Étant donné un contexte $\mathcal{K} = (\mathcal{O}, \mathcal{F}, \zeta)$, nous pouvons définir une application de $\mathcal{P}(\mathcal{O})$ dans $\mathcal{P}(\mathcal{F})$ et une application de $\mathcal{P}(\mathcal{F})$ dans $\mathcal{P}(\mathcal{O})$, notées identiquement par l'opérateur ' de la façon suivante :

$$\forall \mathcal{A} \subset \mathcal{O}, \mathcal{A}' = \{f \in \mathcal{F} \mid \forall o \in \mathcal{A}, \zeta(o, f) = 1\} \quad (1)$$

$$\forall \mathcal{B} \subset \mathcal{F}, \mathcal{B}' = \{o \in \mathcal{O} \mid \forall f \in \mathcal{B}, \zeta(o, f) = 1\} \quad (2)$$

Ces applications sont appelées **connexions de Galois** du contexte ; $\mathcal{A}'$ est appelé **dual** de $\mathcal{A}$, et de façon similaire $\mathcal{B}'$ est appelé **dual** de $\mathcal{B}$.

Pratiquement, $\mathcal{A}'$ est l'ensemble des attributs communs à tous les objets de $\mathcal{A}$, et $\mathcal{B}'$ est l'ensemble des objets qui possèdent en commun tous les attributs de $\mathcal{B}$. On peut trouver les propriétés des connexions de Galois dans [5]. Rappelons les propriétés essentielles suivantes :

Propriété 1 $\mathcal{A}_1 \subset \mathcal{A}_2 \Rightarrow \mathcal{A}_2' \subset \mathcal{A}_1'$;

Propriété 2 $\mathcal{A} \subset \mathcal{A}''$;

Propriété 3 $\mathcal{A}' = \mathcal{A}'''$;

Nous pouvons maintenant énoncer la définition d'un concept :

Définition 4 Étant donné un contexte $\mathcal{K} = (\mathcal{O}, \mathcal{F}, \zeta)$, le couple $\mathcal{C} = (\mathcal{A}, \mathcal{B})$ est appelé un **concept** de $\mathcal{K}$ si et seulement si $\mathcal{A}' = \mathcal{B}$ et $\mathcal{B}' = \mathcal{A}$.

Définition 5 $\mathcal{A}$ est appelé l'**extension** du concept $\mathcal{C}$ et $\mathcal{B}$ est appelé son **intension**. On note $\mathcal{A} = \text{extent}(\mathcal{C})$ et $\mathcal{B} = \text{intent}(\mathcal{C})$.

L'ensemble de tous les concepts d'un contexte $\mathcal{K}$ s'appelle $\mathcal{L}(\mathcal{K})$ ou simplement $\mathcal{L}$ pour abréger les notations s'il n'y a pas d'ambiguïté. On peut prouver le théorème suivant [12]

Théorème 1 Soient $\mathcal{C}_1 = (\mathcal{A}_1, \mathcal{B}_1)$ et $\mathcal{C}_2 = (\mathcal{A}_2, \mathcal{B}_2)$ un couple de concepts alors $\mathcal{C}_1 \vee \mathcal{C}_2 = ((\mathcal{A}_1 \cup \mathcal{A}_2)', \mathcal{B}_1 \cap \mathcal{B}_2)$ et $\mathcal{C}_1 \wedge \mathcal{C}_2 = (\mathcal{A}_1 \cap \mathcal{A}_2, (\mathcal{B}_1 \cup \mathcal{B}_2)')$ sont des concepts.

Ce résultat peut être étendu à tout ensemble $\mathcal{I}$ de concepts. On notera $\mathcal{C}_{\mathcal{I}} = (\mathcal{A}_{\mathcal{I}}, \mathcal{B}_{\mathcal{I}}) = \bigvee_{i \in \mathcal{I}} \mathcal{C}_i$ et de même $\mathcal{C}^{\mathcal{I}} = (\mathcal{A}^{\mathcal{I}}, \mathcal{B}^{\mathcal{I}}) = \bigwedge_{i \in \mathcal{I}} \mathcal{C}_i$. Alors l'ensemble des concepts $\mathcal{L}$ muni de la relation d'ordre $\subset$ (inclusion) sur ses extensions est un treillis complet et on peut poser :

Définition 6 Le treillis complet $\mathcal{L}(\mathcal{K})$ des concepts d'un contexte $\mathcal{K}$ est appelé le **treillis de Galois** ou le **treillis conceptuel**.

4.2 Contexte propre et disjonction

L'extension du plus petit élément de $\mathcal{L}$ est l'ensemble des objets qui partagent tous les attributs de $\mathcal{F}$. Ce n'est pas toujours l'ensemble vide. En fait, la définition formelle d'un contexte donnée ci-dessus est trop générale et englobe des redondances superflues. Aussi nous nous restreignons aux contextes propres dans le sens suivant :

Définition 7 Soit $\mathcal{K} = (\mathcal{O}, \mathcal{F}, \zeta)$ un contexte ; si $(\emptyset, \mathcal{F})$ et $(\mathcal{O}, \emptyset)$ sont des concepts, alors le contexte $\mathcal{K}$ est dit **propre**.

Dans le cas d'un contexte propre, $(\emptyset, \mathcal{F})$ est le plus petit concept (le plus restrictif), et $(\mathcal{O}, \emptyset)$ est le plus grand concept (le plus général). Il n'existe pas de complémentaire canonique dans les treillis de Galois : si $\mathcal{C} = (\mathcal{A}, \mathcal{B})$ est un concept, généralement $(\mathcal{A}^c, \mathcal{A}'^c)$ et $(\mathcal{B}^c, \mathcal{B}'^c)$ ne sont pas des concepts. Cependant la propriété de disjonction est très utile pour construire des classifieurs dans la théorie des classifications standard [29]. Ainsi pour l'apprentissage il est utile de poser :

Définition 8 Soit $\mathcal{C} = (\mathcal{A}, \mathcal{B})$ un concept quelconque d'un contexte $\mathcal{K}$. Si $(\mathcal{A}^c, \mathcal{A}'^c)$ (resp. $(\mathcal{B}^c, \mathcal{B}'^c)$) est un concept, alors $\mathcal{A}$ (resp. $\mathcal{B}$) est appelé ensemble d'objets (resp. d'attributs) de concept **disjonctif**.

Il est ainsi possible de soustraire un concept disjonctif de n'importe quel concept en soustrayant les extensions ou extensions selon la nature de la disjonction.

D'une manière générale, si $\mathcal{I}$ et $\mathcal{J}$ sont deux sous-ensembles de $\mathcal{L}$, il est évident que $\mathcal{B}_{\mathcal{I}} \cap \mathcal{B}_{\mathcal{J}}$ est l'intension de la BS de $\mathcal{C}_{\mathcal{I}}$ et $\mathcal{C}_{\mathcal{J}}$ ($\mathcal{C}_{\mathcal{I}} \vee \mathcal{C}_{\mathcal{J}}$). Le concept associé est appelé **conjonction** des concepts $\mathcal{C}_{\mathcal{I}}$ et $\mathcal{C}_{\mathcal{J}}$.

4.3 Les algorithmes de construction de treillis

Il existe deux familles d'algorithmes de construction de treillis : les algorithmes incrémentaux et les algorithmes non incrémentaux. Les algorithmes incrémentaux (Godin [15], Carpineto & Romano [6], Norris [25], ...) construisent le treillis au fur et à mesure que les objets arrivent, alors que les algorithmes non incrémentaux (Chen [8], Ganter [11], Bordat [4], ...) construisent le treillis une fois le contexte entièrement connu.

Tous ces algorithmes construisent des treillis sans considérer une quelconque spécificité du contexte. Leur complexité est exponentielle, et plusieurs techniques ont été développées afin de réduire les temps de calcul, notamment par simplification du contexte en amont, comme [24] à l'aide d'une fonction d'entropie.

4.4 Exemple

Considérons l'exemple suivant, à la fois simple et explicite, représenté par une *table de correspondance* (figure 1) et le treillis correspondant (figure 2).

		Ppty 1	Ppty 2	Ppty 3	Ppty 4	Ppty 5	Ppty 6	Ppty 7	Ppty 8	Ppty 9	Ppty 10	Ppty 11	Ppty 12	Ppty 13	Ppty 14	Ppty 15
1	Place 1	x		x												x
2	Place 1	x	x		x											x
3	Place 1		x	x							x		x			
4	Place 1		x	x	x			x								x
5	Place 1			x	x					x						x
6	Place 1	x	x		x											
7	Place 2					x	x	x	x				x			
8	Place 2					x		x								x
9	Place 3									x	x					
10	Place 3									x	x					x
11	Place 3		x				x				x		x		x	
12	Place 3			x						x	x					x
13	Place 3									x					x	
14	Place 4										x	x	x	x		
15	Place 4										x		x	x	x	
16	Place 4				x			x					x	x		
17	Place 4												x	x		x

Figure 1: Un exemple simple et explicite

En réalité, le schéma treillis figure 2 est un *treillis d'héritage* [14] dans le sens où il ne montre que les nouveaux objets (*resp.* les nouveaux attributs) d'un nœud par rapport à ses descendants (*resp.* ascendants). Cette représentation permet d'éviter la redondance d'éléments dans l'écriture des nœuds.

Tous les nœuds des treillis correspondent à des concepts. Comme remarqué précédemment, si un objet est inclus dans l'extension d'un concept, il est inclus dans l'extension de tous ses ascendants. De même, si une propriété est incluse dans l'intension d'un concept, alors elle est incluse dans l'intension de tout ses descendants. C'est pourquoi dans cette représentation du treillis (appelé **diagramme de Hasse**), les objets et les propriétés n'apparaissent qu'une seule fois.

5 Les amers

5.1 Utilisation des concepts formels pour la sélection d'amers

Le lien entre concept et amer se définit de la façon suivante :

Définition 9 Considérons un contexte propre $(\mathcal{O}, \mathcal{F}, \zeta)$ et un sous-ensemble d'objets $\mathcal{A} \subset \mathcal{O}$. On dit qu'un sous-ensemble $\mathcal{B}$ de $\mathcal{F}$ est un **amer** de $\mathcal{A}$ si et seulement si

- $\mathcal{B}'' = \mathcal{B}$,
- et $\mathcal{B}' \subset \mathcal{A}$.

Un amer est donc l'ensemble des caractéristiques d'un concept (intension) qui répondent aux conditions ci-dessus.

5.2 Définitions formelles des amers dans un contexte partitionné

Définition générale. On se donne dans ce qui suit un contexte propre $(\mathcal{O}, \mathcal{F}, \zeta)$, et une partition $(\mathcal{O}_\theta)_{\theta \in \Theta}$ de son ensemble d'objets. L'ensemble Θ est appelé **ensemble des sites**, ou plus généralement **ensemble des classes** dans le contexte d'une classification générale.

Définition 10 On dit qu'un sous-ensemble $\mathcal{B}$ de $\mathcal{F}$ est un **amer** de $\theta \in \Theta$ si et seulement si

- $\mathcal{B}'' = \mathcal{B}$,

- et $\mathcal{B}' \subset \mathcal{O}_\theta$.

Ainsi un amer est un ensemble de caractéristiques dont l'occurrence simultanée n'est effective que dans *certaines* images du site à caractériser.

Amers pleins et amers partiels. En particulier, si un amer $\mathcal{B}$ est un ensemble de caractéristiques simultanément présentes dans *toutes* les images du site à caractériser, et seulement dans celles-là, on parle alors d'**amer plein** :

Définition 11 On dit qu'un sous-ensemble $\mathcal{B}$ de $\mathcal{F}$ est un **amer plein** de $\theta \in \Theta$ si et seulement si

- $\mathcal{B}'' = \mathcal{B}$,
- et $\mathcal{B}' = \mathcal{O}_\theta$.

L'existence d'amers pleins rend la classification simple à mettre en œuvre, mais en général la propriété de présence dans toutes les images n'est pas vérifiée.

Définition 12 On dit qu'un sous-ensemble $\mathcal{B}$ de $\mathcal{F}$ est un **amer partiel** de $\theta \in \Theta$ si et seulement si

- $\mathcal{B}'' = \mathcal{B}$,
- $\mathcal{B}' \subset \mathcal{O}_\theta$;
- et $\mathcal{O}_\theta \not\subset \mathcal{B}'$.

Amer maximal. Dans le cas où il n'existe pas d'amer plein, afin de limiter le nombre d'amers partiels à considérer (et à retenir), il est intéressant d'introduire les amers maximaux :

Définition 13 Un **amer maximal** est un amer minimal pour l'inclusion dans l'ensemble des amers d'un site.

Couverture. La couverture d'un site se définit de la façon suivante :

Définition 14 Considérons les images $\mathcal{O}_\theta$ d'un site $\theta \in \Theta$ et l'ensemble N_θ des amers $\{\mathcal{B}_i\}_{i=1\dots N_\theta}$ de ce site. On dit que le site θ est **couvert**, ou que les images du site sont **entièrement couvertes**, si et seulement si $\bigcup_{i=1\dots N_\theta} \{\mathcal{B}'_i\} = \mathcal{O}_\theta$

Dans le cas où il existe un amer plein d'un site, la couverture est évidente. Dans le cas où il n'existe que des amers partiels, certaines images peuvent ne pas être couvertes par des amers. Il est clair que s'il existe une couverture du site, elle se fait par les amers partiels maximaux.

6 Construire un classifieur à l'aide d'amers

Dans cette partie, nous exposons le raisonnement complet relatif à l'extraction d'amers de chaque ensemble d'images dans un premier temps, puis l'utilisation des ces amers dans un deuxième temps. Il existe plusieurs types de classifieurs basés sur des treillis de Galois (voir [23]). La principale différence avec ces classifieurs étant le fait que notre classifieur se base sur des amers, qui sont des combinaisons de

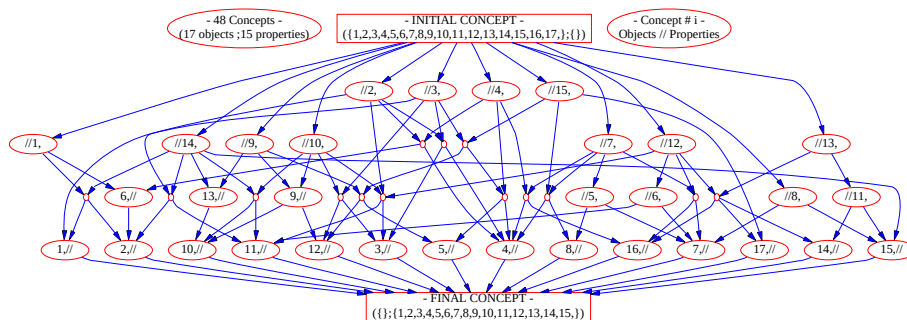


Figure 2: Le treillis correspondant

caractéristiques. De plus, il introduit l'approche décentralisée développée plus loin.

Détaillons quelque peu notre application. Nous disposons d'un ensemble d'images issues d'un environnement structuré. Chaque image appartient à une pièce particulière de cet environnement, lieu où elle a été prise par un robot équipé d'une caméra CCD. L'ensemble des images est donc partitionné en fonction du nombre de pièces. Notre objectif est de donner au robot les moyens de se repérer sur carte topologique¹ (et, dans un futur proche, de construire la carte) de cet environnement. Il s'agit, de façon basique, d'un problème de classification supervisée. La règle de décision est extraite à partir d'un ensemble d'images appartenant à une base de d'apprentissage. La règle est formalisée pour chaque sous-ensemble (un sous-ensemble correspondant à une pièce) en y associant un amer partiel maximal. Certaines images peuvent échapper à la règle de décision, et des erreurs d'apprentissage peuvent apparaître. Il existe classiquement deux phases dans notre processus : la phase d'apprentissage, qui étiquette à chaque pièce un amer, et la phase de généralisation (avec des images issues des mêmes pièces mais différentes des précédentes) qui utilise ces amers pour reconnaître telle ou telle pièce.

6.1 Phase d'apprentissage : extraction des amers

Primitives. Dans un premier temps, il est nécessaire d'extraire les primitives de chaque image. Les algorithmes utilisés sont très classiques ; par exemple, les contours sont issus des opérateurs de Canny-Deriche, et après approximation polygonale on en déduit les "segments de contour" présents dans l'image. D'autres primitives (couleur, texture, points d'intérêt, etc.) peuvent être extraits à l'aide des algorithmes classiques appropriés.

Caractéristiques. Ensuite, dans un deuxième temps, les éventuelles caractéristiques sont déduites de ces primitives, et l'on remplit ainsi la table de correspondance entre les images d'un côté, et les caractéristiques de l'autre.

¹Une carte topologique d'un environnement structuré est un graphe pour lequel (en général, mais pas nécessairement) chaque nœud représente une pièce de l'environnement et chaque arc un passage entre deux pièces.

Treillis. Dans un troisième temps, le treillis de Galois de la table est construit, hiérarchisant ainsi les caractéristiques les unes par rapport aux autres. Enfin, dans un dernier temps, les amers sont extraits de chaque treillis afin de caractériser chacun des ensembles d'images (chacune des pièces de l'environnement structuré).

Concepts. Il est nécessaire ensuite, dans un quatrième temps, de sélectionner les concepts "intéressants". Considérant tous les concepts C_θ relatifs à une même pièce θ , c'est-à-dire les concepts du treillis d'héritage (le treillis d'héritage est utilisé afin d'éviter les redondances de caractéristiques) dont les objets sont des images de cette pièce θ , l'ensemble des concepts sélectionnés pour les amers potentiels sont :

1. les concepts relatifs à la pièce θ ;
2. les concepts ascendants et descendants des concepts de cette pièce C_θ ;
3. et parmi ces concepts seulement, les concepts qui n'ont pas, comme descendant, un concept dont l'extension renferme au moins une image d'une autre pièce $\theta^* \neq \theta$.

Amers. Enfin, dans un cinquième temps, il est nécessaire d'extraire les amers de ces concepts. Supposons l'ensemble non vide des concepts précédemment sélectionnés ; les amers sont, s'ils existent, les intensions de ces concepts. Plusieurs cas peuvent se présenter :

- s'il existe un concept dont l'intension est non vide et qui couvre parmi ses descendants l'ensemble des concepts relatifs à une pièce θ , on parle alors d'amer plein (total, global, ...) ;
- dans le cas contraire, les intensions lorsqu'ils existent des concepts sélectionnés sont appelés amers partiels.

Dans certains cas où les concepts du treillis d'héritage ne donnent pas de résultat, il est alors nécessaire de revenir au treillis "classic" et d'extraire, parmi les concepts relatifs à la pièce θ , les intensions non vides de ces concepts.

Algorithme général. La démarche algorithmique se présente figure 3.

1. Extraire les primitives de chaque image
2. Déterminer la présence ou non de toutes les caractéristiques potentielles et remplir la table de correspondance
3. Construire le treillis de Galois correspondant
4. Pour chaque pièce :
 41. Sélectionner les concepts ayant pour extension uniquement des images de la pièce
 42. Sélectionner les ascendants et les descendants de ces concepts
 43. Éliminer les concepts qui ont pour descendants des concepts qui renferment des images d'autres pièces dans leur extension
 44. Éventuellement, pour sélectionner les amers maximaux parmi les concepts du groupe restant, éliminer les concepts qui ont un autre concept comme ascendant dans ce même groupe
 45. Enfin, si nécessaire, lire les intensions des concepts relatifs à la pièce θ .

Figure 3: Algorithme général de la sélection d'amers

6.2 Phase de généralisation : Reconnaissance de la pièce

Une fois les amers sélectionnés, nous considérons une nouvelle image issue des mêmes pièces mais différente de celles utilisées lors de l'apprentissage. Les primitives sont extraites et les caractéristiques déterminées. Plusieurs cas sont possibles :

- si l'ensemble des caractéristiques inclut un ou plusieurs amers d'une seule et unique pièce, alors il est possible de conclure que l'image appartient à cette pièce ;
- si l'ensemble des caractéristiques n'inclut aucun amers, on ne peut pas conclure et le robot a pour tâche de se déplacer et de prendre une autre image ;
- si l'ensemble des caractéristiques inclut un ou plusieurs amers de plusieurs pièces différentes, alors non seulement on ne peut conclure, mais également le processus d'apprentissage est remis en cause.

6.3 Résultats expérimentaux (1)

Nous avons mis en œuvre cette approche à partir d'images issues de pièces différentes d'un environnement structuré, en l'occurrence une partie du LAAS.

Un ensemble de 15 images ont servi de base d'apprentissage, réparties en 4 pièces différentes, et nous considérons pour l'instant 50 caractéristiques potentielles, numérotées de #1 à #50 :

- peu de pixels (> 50), des pixels (> 300), beaucoup de pixels de rouge (> 3000), vert, bleu, cyan, magenta, jaune ($3 \times 6 = 18$ caractéristiques), petit ($> 15 \times 15$),

moyen ($> 50 \times 50$) ou gros ($> 75 \times 75$) objet R-V-B-C-M-J (18) ;

- un grand nombre (> 50), un très grand nombre (> 100) de segments horizontaux, verticaux, ou de direction quelconque (3×2), un grand nombre, un très grand nombre de segment horizontaux, verticaux ou de même orientation ($2 + 2 + 2 = 6$), un grand nombre, un très grand nombre de segments de même taille (2).

Apprentissage. La table de correspondance a été remplie et le treillis correspondant construit (algorithme de Carpineto & Romano[6]), figure 6.3.

Pour chaque ensemble d'images, les concepts correspondants ont été trouvés et l'on obtient (table 1) :

- pour le premier ensemble (images 1,2), {#42 et #45} est un amer plein et {#22.#28.#39} un amer partiel ; {#42 et #45} (*un grand nombre et un très grand nombre de segments horizontaux*) est un amer maximal de couverture totale ;
- pour le deuxième ensemble (images 3,4,5,6), {#5} est un amer plein et {#3.#32} et {#6} sont des amers partiels ; {#5} (*objet rouge moyen*) est l'amer maximum qui couvre ce site ;
- pour le troisième ensemble (images 7,8,9,10), pas d'amer plein mais quatre amers partiels : {#19}, {#41}, {#38.#44.#49}, {#15.#17.#18} et {#14}. Parmi ces amers partiels, l'amer {#14} couvre les amers {#19} et {#15.#17.#18}, et l'amer {#41} couvre les amers {#19} et {#38.#44.#49}. Ces amers ({#14} : *peu de pixels bleus* et {#41} : *un grand nombre de segments verticaux*) sont donc maximaux et l'on peut montrer également qu'ils forment une couverture totale ;
- et pour le dernier ensemble (images 11,12,13,14,15), {#7} est un amer plein et {#8} un amer partiel. {#7} (*des pixels verts*) est un amer maximal qui couvre ce site.

Lieu	Amer plein	Amers partiels	Amers maximaux
Pièce n°1	{#42.#45}	{#22.#28.#39}	{#42.#45}
Pièce n°2	{#5}	{#6}, {#3.#32}	{#5}
Pièce n°3	$\emptyset$	{#19}, {#41}, {#38. #44.#49}, {#15.#17.#18}, {#14}	{#41}, {#14}
Pièce n°4	{#7}	{#8}	{#7}

Table 1: Les différents amers

Il est a priori inutile de retenir tous les amers possibles. Aussi, s'il existe des amers pleins dans une classe, on éliminera les amers partiels. Dans le cas contraire, parmi

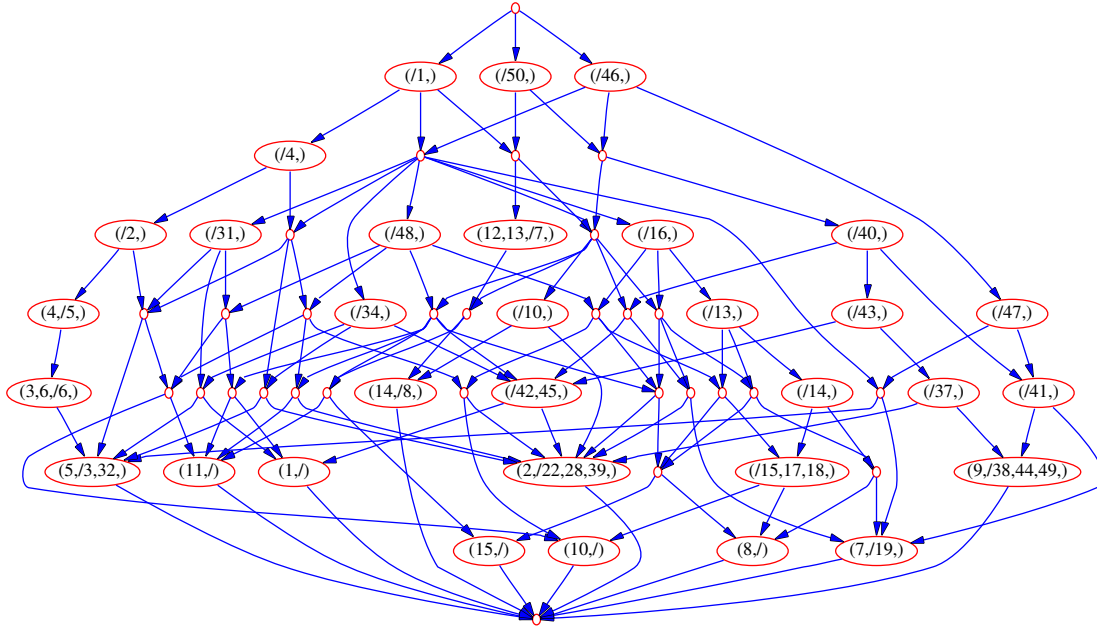


Figure 4: Treillis global correspondant à notre application.

les amers partiels, on ne retiendra que les amers partiels maximaux.

Test. Pour la phase de test, 20 nouvelles images ont été prises dans les quatre pièces différentes. 18 ont été correctement situées (c'est-à-dire que le système a su reconnaître le lieu où elles ont été prises), les deux dernières ne permettant aucune conclusion.

Dans l'application qui nous concerne, si une image n'est pas située, cela n'a aucune importance dans le sens où le robot est capable de "regarder ailleurs" et d'extraire d'autres images de la pièce où il se situe pour reconnaître cette pièce.

7 L'approche décentralisée

La précédente approche donne de très bons résultats pour des contextes de petite taille (moins de 25 objets / caractéristiques). Cependant, pour de gros contextes, la complexité exponentielle des algorithmes (voir [20]) ne permet pas de travailler dans de très gros contextes. De plus, l'exploitation du treillis risque d'être délicate de par sa complexité hiérarchique entre les concepts : il peut y avoir tant de concepts et tant de connexions entre eux qu'aucun amer ne pourrait être sélectionné.

C'est pourquoi nous introduisons ici une approche *décentralisée* pour gérer les données. L'idée principale est de construire non plus un seul treillis pour l'ensemble des images mais autant de treillis qu'il existe de pièces. Ces treillis seront de taille plus réduite, mais il sera alors néces-

saire de considérer les interactions entre treillis. Cette approche présente beaucoup d'avantages (voir section 7.4).

7.1 Formalisme mathématique (2)

Nous définissons ici les contextes, concepts et treillis locaux et locaux relatifs dont nous avons besoin plus loin dans notre application.

Définition 15 Le **contexte général** ou **contexte global** $\mathcal{K}$ est le contexte relatif à l'ensemble des objets $\mathcal{O}$, l'ensemble des attributs $\mathcal{F}$, et l'application $\zeta : \mathcal{O} \times \mathcal{F} \rightarrow \{0, 1\}$; on a alors $\mathcal{K} = (\mathcal{O}, \mathcal{F}, \zeta)$.

Le contexte général est alors divisé en sous-contextes, relatifs à chaque ensemble.

Définition 16 Soit $\mathcal{O} = \{\mathcal{O}_\theta\}_{\theta \in \Theta}$ les sous-ensembles d'objets du contexte général $\mathcal{K}$, on a alors $\mathcal{O} = \bigoplus_j \theta_j$. θ_j est appelé **classe d'objets** relatif au sous-ensemble d'objets concerné.

Définition 17 On appelle **application locale** ζ_θ la restriction de ζ à la classe d'objets θ .

On a alors : $\zeta = \bigoplus_{\theta \in \Theta} \zeta_\theta$

Définition 18 Soit $\mathcal{K}$ un contexte multi-classes, on appelle **contexte local** $\mathcal{K}_\theta$ relative à la classe θ le contexte $\mathcal{K}_\theta = (\theta, \mathcal{F}, \zeta_\theta)$

Une fois les contextes locaux définis, il est simple de définir les concepts locaux et treillis locaux :

Définition 19 On appelle **concept local** C_θ tout concept issu du contexte local K_θ .

Définition 20 On appelle **treillis local** L_θ le treillis issu du contexte local K_θ .

Cependant, comme nous allons le voir, les contextes locaux vont être modifiés par la sélection préalable d'attributs. Nous définissons donc :

Définition 21 Soit K_θ un contexte local et ζ'_θ une modification de l'application locale ζ_θ . On appelle **contexte local modifié** le contexte $K'_\theta = (\theta, \mathcal{F}, \zeta'_\theta)$.

De la même façon que précédemment, il est possible de définir les concepts locaux modifiés et treillis locaux modifiés :

Définition 22 On appelle **concept local modifié** C'_θ tout concept issu du contexte local modifié K'_θ .

Définition 23 On appelle **treillis local modifié** L'_θ le treillis local issu du contexte local modifié K'_θ .

Nous pouvons également définir des amers partiels et des amers pleins à partir de la structure des treillis locaux modifiés.

7.2 Sélection de caractéristiques

À chaque sous-ensemble est ainsi associé un treillis, qui sera simplifié en enlevant des caractéristiques *non essentielles* dans une classe. En effet, il est nécessaire de retirer des caractéristiques soit qui alourdissent inutilement un contexte local, soit que ne permettent pas de distinguer les classes entre elles.

Une caractéristique i sera considérée comme **essentielle** dans la classe θ si sa fréquence d'apparition $r_{i,\theta}$ dans la classe θ est bien supérieure à sa fréquence d'apparition $r_{i,\theta^* \neq \theta}$ dans toute autre classe θ^* du contexte général. Dans l'exemple de la figure 1, nous avons par exemple $r_{2,1} = 2/3$, $r_{5,2} = 1$, $r_{13,1} = 0$, etc.

Notre heuristique se base donc sur un taux relatif $\tau_{i,\theta,\theta^*} = r_{i,\theta}/r_{i,\theta^*}$. Si $\forall \theta^* \neq \theta$, $\tau_{i,\theta,\theta^*} \gg 1$, la caractéristique i est gardée pour la classe θ ; sinon, elle en est retirée.

Ainsi, l'ensemble des caractéristiques relatives à chaque classe, hiérarchisés dans le treillis, forment les amers de chaque sous-ensembles.

7.3 Algorithme général

L'algorithme général de la sélection d'amer se présente comme le montre la figure 5.

7.4 Résultats expérimentaux (2)

Nous avons fait deux types d'expérimentations : les premières montrent l'intérêt d'une approche décentralisée en terme de temps de calcul et nombre de concepts global, et les secondes dans le cas réel en travaillant avec des images différentes que précédemment, mais issues du même lieu.

1. Pour chaque site θ ,
2. Pour chaque caractéristique i ,
3. Calculer le taux de présence $r_{i,\theta}$
4. Fin Pour
5. Fin Pour
6. Pour chaque site θ ,
7. Pour chaque caractéristique i ,
8. Pour chaque site $\theta^* \neq \theta$,
9. Si $\tau_{i,\theta,\theta^*} \simeq 1$ or $\tau_{i,\theta,\theta^*} \ll 1$
10. Modifier le contexte local
 (càd retirer i du site θ)
11. Fin Si
12. Fin Pour
13. Fin Pour
14. Construire les treillis locaux modifiés
15. Extraire les amers
16. Fin Pour.

Figure 5: Algorithme général.

Temps de calcul et nombre de concepts. L'algorithme ci dessus a été testé sur des contextes carrés aléatoires de taille 5×5 à 35×35 , avec trois classes. L'algorithme λ est l'algorithme de Carpineto & Romano. La figure 6 montre l'avantage de l'approche décentralisée en terme de temps de calcul et de nombre de concepts global.

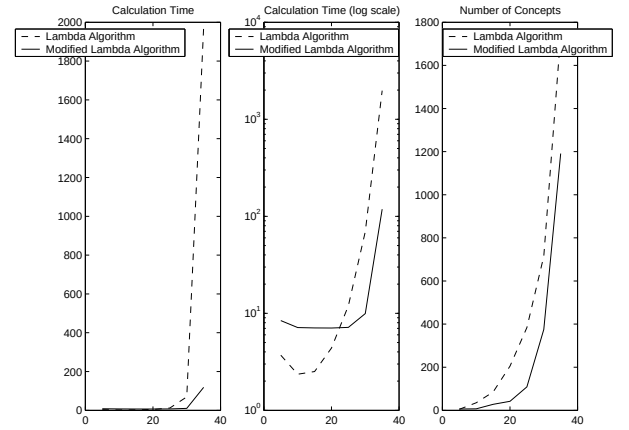


Figure 6: Nombre d'objets/propriétés, temps et nombre de concepts.

Application réelle. L'application réelle a permis de construire les treillis locaux présentés figure 7.4. Dans cette application, 15 images réelles ont été utilisées avec les mêmes caractéristiques, dans les mêmes pièces que précédemment.

Il suffit alors de lire les amers (table 2) à partir de ces treillis (figure 7.4), ce qui se fait de façon bien plus simple que précédemment. Il est à noter que les résultats sont exactement les mêmes.

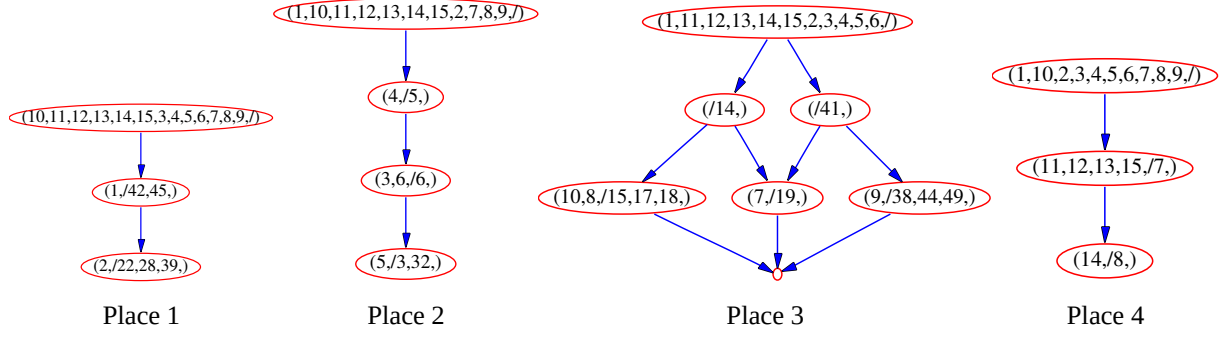


Figure 7: Les treillis locaux correspondants aux quatre pièces de l'environnement

Lieu	Amer plein	Amers partiels	Amers maximaux
Pièce n°1	{#42.#45}	{#22.#28#39}	{#42.#45}
Pièce n°2	{#5}	{#6}, {#3.#32}	{#5}
Pièce n°3	∅	{#19}, {#41}, {#38. #44.#49}, {#15.#17.#18}, {#14}	{#41}, {#14}
Pièce n°4	{#7}	{#8}	{#7}

Table 2: Les différents amers trouvés avec l'approche décentralisée.

8 Discussion et perspectives

Nous avons décrit dans ce papier une nouvelle application de la théorie des treillis de Galois, afin de caractériser par apprentissage des ensembles d'images. La motivation principale de ce travail vient du monde de la robotique mobile autonome : notre but est de permettre à un robot de caractériser les ensembles d'images issues des différentes pièces d'un environnement structuré, afin de reconnaître chaque ensemble et donc de savoir se situer dans l'environnement.

Nous avons validé notre architecture générale à travers des expérimentations sur des ensembles d'images réelles du laboratoire. Les résultats sont très encourageants, mais seulement pour des petits contextes. C'est pourquoi nous avons développé une approche dite *décentralisée*, dans le sens où l'on ne construit plus un treillis général de l'environnement mais n treillis pour chaque pièce : on les appelle alors *treillis locaux* attachés aux différentes pièces. Les premières expérimentations dans ce sens donnent de bons résultats.

D'autres méthodes de classification supervisée pourrait utiliser les mêmes données initiales pour résoudre des problèmes de classification. Les réseaux de neurones ou les arbres de décisions semblent les plus indiqués pour cela, et seront testés dans un futur proche. Entre autre, les arbres de décision ont été utilisés pour caractériser les ca-

ractères manuscrits utilisant des *tags* [29] qui peuvent être comparés à nos caractéristiques. Ces techniques ont leurs propres limitations.

Habituellement il est préférable d'avoir un processus de navigation et d'orientation en ligne, afin de limiter les erreurs propres à tout approche statistique : si une erreur survient ou une indétermination apparaît, le robot doit pouvoir réagir en temps réel pour par exemple prendre de nouvelles images afin d'affiner sa décision, et éventuellement modifier sa base d'apprentissage, le (ou les) treillis correspondants et donc ses amers. Notre approche permet une certaine souplesse par la redondance d'informations, à travers les nombreux concepts. Et ce contrairement aux arbres de décision classiques. Dans le processus de construction des arbres de décision de [29], les nœuds pertinents des arbres sont sélectionnés en maximisant une fonction d'entropie. Il sera ainsi difficile pour le système de gérer des informations contradictoires. De fait il y a un compromis à trouver entre redondance et robustesse dans l'apprentissage symbolique des treillis de Galois d'un côté, et leur complexité d'autre part.

Enfin, les amers considérés dans ce papier sont des amers visuels, dans le sens où ils ne sont que des caractéristiques issues de la perception visuelle, et ne sont pas reliés à des objets physiques. Le passage de la vision (caractéristiques) au monde réel (propriétés) n'est pas évident à instaurer, mais pourra être considéré à plus ou moins long terme afin d'améliorer notre apprentissage et d'avoir une approche plus "humaine" des amers de l'environnement.

References

- [1] A. Arleo, F. Smeraldi, S. Hug, and W. Gerstner. Place cells and spatial navigation based on 2d visual feature extraction, path integration, and reinforcement learning. *Advances in Neural Information Processing Systems 13*, MIT-Press, 2001.
- [2] M. Barbut and B. Monjardet. *Ordre et Classification*. Hachette Université, 1970.
- [3] G. Bianco and A. Zelinsky. Biologically-inspired visual landmarks learning and navigation for mobile robots. *IEEE/RSJ, IROS*, 1999.

- [4] J. Bordat. Calcul pratique du treillis de galois d'une correspondance. *Mathématique, Informatique et Sciences Humaines*, 1986.
- [5] Marc Boyer. *Induction de régularités dans une base de connaissance, application au phénomène bruit/Gêne et ses extensions*. PhD thesis, École Nationale Supérieure de l'Aéronautique et de l'Espace, 2001.
- [6] C. Carpineto and G. Romano. A lattice conceptual clustering system and its applications to browsing retrieval. *Machine Learning*, 1996.
- [7] L. Chaudron, N. Maille, and M. Boyer. The (cube) lattice model and its applications. *Applied Artificial Intelligence*, vol 19, no 3, 2003.
- [8] M. Chen. Algorithme de recherche des sous-matrices premières d'une matrice. *Bulletin Mathématique de la Sociologie Scientifique de la R.S. de Roumanie*, 1969.
- [9] Zachary Dodds and Gregory D. Hager. A color interest operator for landmark-based navigation. In *AAAI/IAAI*, pages 655–660, 1997.
- [10] G. Dudek and D. Jugessur. Robust place recognition using local appearance based methods. *IEEE Int. Conf. on Robotics and Automation*, 2000.
- [11] B. Ganter. Two basics algorithms in concept analysis. Technical report, Technische Hochschule Darmstadt, 1984.
- [12] B. Ganter and R. Wille. *Formal Concept Analysis*. Springer-Verlag, 1999.
- [13] P. Gaussier, C. Joulin, J.P. Banquet, S. Leprêtre, and A. Revel. The visual homing problem : An exemple of robotics/biology cross fertilization. *Robotics and Autonomous Systems*, 2000.
- [14] R. Godin, G. Mineau, R. Missaoui, and H. Mili. Méthodes de classification conceptuelle basée sur les treillis de galois et applications. *Revue d'Intelligence Artificielle*, 9(2), 105-137, 1995.
- [15] R. Godin, R. Missaoui, and H. Alaoui. Learning algorithms using a galois lattice structure. *IEEE Int. Conf. on Tools for AI*, 1991.
- [16] J.B. Hayet, F. Lerasle, and M. Devy. A visual landmark framework for indoor mobile robot navigation. *Int. Conf. on Robotics and Automation*, 2002.
- [17] Jean-Bernard Hayet. *Contribution à la navigation d'un robot mobile sur amers visuels texturés dans un environnement structuré*. PhD thesis, Université Toulouse III, Toulouse, 2002.
- [18] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
- [19] M. Knappek, R.-O. Oropeza, and D. Kriegman. Selecting promising landmarks. *IEEE Int. Conf. on Robotics and Automation*, 2000.
- [20] S. Kuznetsov and S. Obiedkov. Algorithms for the construction of concept lattices and their diagram graphs. *Principles of Data Mining and Knowledge Discovery (PKDD 2001)*, Freiburg, Germany, 2001.
- [21] M. Liquière and J. Sallantin. Structural machine learning with galois lattice and graphs. *International Conference on Machine Learning*, 1998.
- [22] M. Mata, J. M. Armingol, A. de la Escalera, and M.A. Salichs. A visual landmark recognition system for topological navigation of mobile robots. *IEEE International Conference on Robotics and Automation (ICRA01)*, 2001.
- [23] E. Mephu Nguifo and P. Njiwoua. Supervised classification and formal concept analysis. *International Conference on Formal Concept Analysis, Technische Universität Darmstadt*, 2003.
- [24] Engelbert Mephu Nguifo and Patrick Njiwoua. Using lattice-based framework as a tool for feature extraction. In *European Conference on Machine Learning*, pages 304–309, 1998.
- [25] E.M. Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées* 23 (2), 1978.
- [26] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE TPAM*, 1997.
- [27] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with t. *Data Knowledge Engineering*, 42(2):189–222, 2002.
- [28] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localisation. *IEEE Int. Conf. on Robotics and Automation*, 2000.
- [29] Y.Amit and D.Geman. Shape quantization and recognition with randomized trees. *Neural computation*, 1997.