

# Indexation de photos sociales par propagation sur une hiérarchie de concepts

Michel Crampes<sup>1</sup>, Jeremy de Oliveira-Kumar<sup>2</sup>, Sylvie Ranwez<sup>1</sup>, Jean Villerd<sup>1</sup>

<sup>1</sup> LGI2P, Ecole des Mines d'Alès

Site EERIE, Parc scientifique G. Besse, F – 30 035 Nîmes, France  
{Michel.Crampes, Sylvie.Ranwez, Jean.Villerd}@ema.fr

<sup>2</sup> School of Computer Science and Engineering, UNSW,  
Sydney NSW 2052, Australia  
jdok706@cse.unsw.edu.au

**Résumé :** Nous nommons ‘photos sociales’ les photos qui sont prises lors d’événements familiaux ou de soirées entre amis et qui représentent des individus ou des groupes d’individus. Leur indexation consiste à repérer l’événement et les personnes présentes sur les photos. Dans cet article nous présentons une méthode et des outils pour faciliter cette tâche.

De nouvelles photos sont indexées à partir de photos déjà indexées selon un procédé de ‘propagation’ qui se compose d’un ‘glisser-déposer’ suivi d’une fusion et d’une affectation des contenus. Il convient au préalable d’organiser sur l’écran les photos déjà indexées selon une disposition qui facilite l’identification des personnages. Dans ce but nous faisons appel aux techniques d’Analyse Formelle de Concepts et nous proposons un algorithme de construction incrémentale d’un Diagramme de Hasse pour à la fois faciliter le repérage, intégrer les photos nouvellement indexées dans le processus d’indexation et maintenir la représentation mentale de l’utilisateur.

**Mots-clés :** Ingénierie des connaissances, indexation, Analyse Formelle de Concepts, Diagramme de Hasse incrémental.

## 1 Introduction

Les appareils photographiques numériques et les téléphones portables dotés de capacités de prise de vues sont très utilisés pour conserver la mémoire d’évènements sociaux tels que les mariages, les soirées entre amis, les anniversaires, etc. Le rappel de ces photos sociales et leur partage nécessite souvent une phase préalable d’indexation, qui consiste à repérer l’événement (lieu, date, nature, etc.) et les personnes visibles sur les photos. Ces informations peuvent évidemment être saisies manuellement, comme dans ‘Facebook’, mais ce procédé est vite fastidieux quand il faut indexer des centaines de photos. En fait il existe peu d’outils qui assistent réellement un utilisateur dans cette tâche. Nous proposons ici une approche originale pour indexer les photos sociales en mettant en œuvre trois stratégies.

(i) le contenu de certaines photos déjà indexées sert à l’indexation de nouvelles photos selon un procédé de ‘propagation’ ;

(ii) pour retrouver certaines photos afin de propager leur contenu, nous faisons appel aux techniques de classification et de représentation de l'Analyse Formelle de Concepts (Ganter & Wille, 1999), et en particulier aux Diagrammes de Hasse ;

(iii) les photos nouvellement indexées sont automatiquement intégrées dans un Diagramme de Hasse de manière incrémentale tout en maintenant la carte mentale de l'utilisateur.

Toutes ces fonctionnalités ont été testées avec des étudiants de premier cycle. Selon ceux-ci et au vu d'observations que nous présentons à la fin, nos méthodes et nos outils se révèlent plus attractifs et plus performants que les solutions proposées actuellement dans la littérature et dans des applications comme Facebook, ou Flickr. Bien que limitée pour l'instant aux photos sociales, la méthode pourra être envisagée par la suite pour d'autres types de photos avec sans doute quelques adaptations.

## 2 Etat de l'art

Face à l'enjeu qu'elles présentent, de nombreuses solutions sont explorées pour indexer les photos numériques, et en particulier les photos sociales.

- Les techniques d'analyse d'images ont fait de grands progrès avec certaines applications de reconnaissance de visages déjà arrivées au stade commercial comme dans l'application iPhoto d'Apple. Mais les photos doivent avoir été prises dans un environnement très contrôlé (éclairage, orientation, etc.), ce qui est rarement le cas des photos sociales telles que nous les avons précédemment définies (voir par exemple un état de l'art en la matière dans (NSTC report, 2006). Notre méthode pourrait venir en complément de la reconnaissance de visages pour les photos prises dans de mauvaises conditions.

- L'alternative la plus évidente à l'indexation automatique est la saisie au clavier des noms des personnes présentes sur les photos. Mais cette méthode présente deux inconvénients majeurs. Tout d'abord l'utilisateur doit décider des mots à saisir, ce qui, même quand on considère le cas limité des photos sociales, pose le problème du choix des mots et de leur orthographe. Par ailleurs, ce procédé est long et fastidieux. Beaucoup d'utilisateurs se découragent et les photos restent non classées, ce qui ne facilite pas les échanges personnalisés.

Pour éviter les saisies répétées et pour mieux contrôler le vocabulaire, il est possible de faire appel à une liste incrémentale des noms des personnes. Cette approche facilite effectivement le travail de l'indexeur, mais affecter des noms à partir d'une liste sur des photos reste encore une tâche fastidieuse comme nous avons pu l'observer lors des tests présentés dans la conclusion.

- Certains chercheurs proposent d'exploiter des index observables lors de la prise de photos (localisation géodésique, date) accompagnées d'informations sociales pour suggérer des lieux et des personnages (Monaghan & O'Sullivan, 2007). Cette approche pourrait aussi compléter notre méthode, cette dernière permettant de passer de la suggestion à l'indexation précise de chaque photo dans sa dimension humaine.

- L'indexation partagée ("social tagging") constitue une réponse astucieuse à la pénibilité de l'indexation. Les index créés par des utilisateurs peuvent être utilisés par

d'autres. Mais le problème du choix des mots et le caractère fastidieux de l'indexation mot à mot de centaines de photos en limite la portée pour le cas des photos sociales.

- A notre connaissance, l'indexation par propagation qui consiste à indexer de nouveaux éléments à partir d'éléments anciens n'a été proposée que dans le domaine de la musique (Crampes et al., 2007). Cependant, la méthode ne fait pas appel à un support formel comme un Diagramme de Hasse représentant une sous-hiérarchie de Galois.

- Certaines applications récentes utilisent une Hiérarchie de Galois pour naviguer dans une collection photo comme dans (Eklund et al., 2006), (Ferré, 2007) et (Loisant et al., 2003). Mais les auteurs n'utilisent pas un Diagramme de Hasse pour indexer les photos et ne proposent pas une technique d'intégration des photos nouvellement indexées de manière incrémentale dans la sous-hiérarchie de Galois.

### **3 Indexer des photos sociales par propagation**

Afin d'introduire notre méthode, nous prenons l'exemple réel d'une personne qui veut indexer un ensemble de photos avec notre application expérimentale. Marie, notre personnage, est allée à un anniversaire et a pris environ 100 photos qu'elle veut indexer avec les noms des personnes afin de les partager de manière personnalisée.

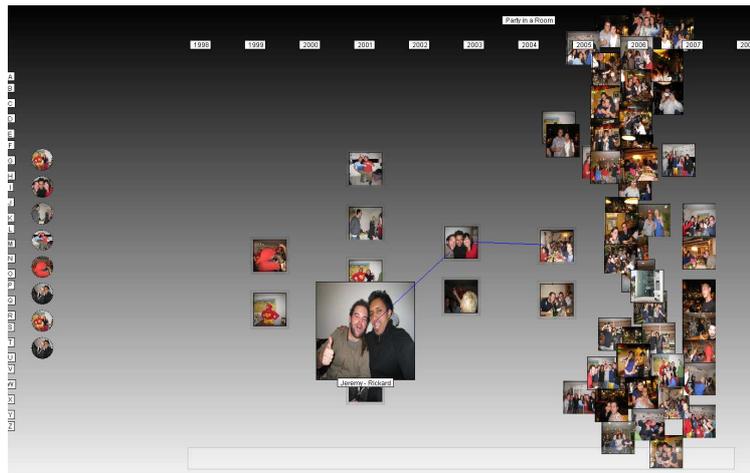
Elle a la possibilité de disposer toutes ses photos sur l'écran, que nous appelons la "planche photo" ("photo-board"). Dans un premier temps, elle peut caractériser l'événement avec son lieu, sa date, sa nature, etc. Toutes ces données sont rangées sous le même événement qui apparaît à l'écran. Quand celui-ci est sélectionné, toutes les photos qui seront indexées hériteront de ses caractéristiques.

La tâche suivante qui consiste à saisir les personnes présentes sur les photos est la plus longue et la plus fastidieuse. Une photo peut montrer un, deux ou de nombreux personnages dans différentes situations lors de l'événement. Notre méthode va permettre de faciliter cette saisie et de la rendre plus sûre et plus complète en évitant la saturation de l'utilisateur. Elle peut être mise en œuvre après qu'un certain nombre de photos concernant l'événement ont été préalablement indexées ou bien qu'il existe déjà un jeu de photos indexées venant d'un événement similaire avec un sous-ensemble commun de personnes. Cette phase d'initialisation sera présentée plus loin. Pour l'instant nous considérons qu'il existe un ensemble de photos déjà indexées et une liste alphabétique des personnes présentes sur ces photos. La planche photo est présentée sur la Figure 1 avec les photos à indexer à droite et la liste des personnes à gauche.

Pour indexer de nouvelles photos, Marie va utiliser un principe de 'propagation'. Nous appelons 'indexeurs' les photos déjà indexées qui propageront leur contenu, et nous utilisons le néologisme "indexables" pour désigner les photos à indexer sur lesquelles seront propagés les contenus des indexeurs.

Dans la Figure 2 l'utilisatrice a sélectionné un indexable encadré de bleu. Elle peut visuellement identifier les personnages sur la photo en utilisant éventuellement une lentille "fish eye" grossissante (Furnas, 1986). Marie peut alors amener l'indexable près d'un ou plusieurs indexeurs qui, selon elle, contiennent les personnages de la

photo. En relâchant et en cliquant sur le bouton de la souris, elle peut sélectionner plusieurs indexeurs qui affichent alors leur contenu et sont entourés d'un disque bleu. Dans la Figure 2 l'indexable comporte trois personnages. L'indexeur au dessus a déjà été sélectionné et l'indexeur au dessous qui vient d'être sélectionné affiche la liste de ses personnages. Un double clic sur la souris déclenche la propagation qui opère de la manière suivante.



**Fig. 1** – La planche photo

Formellement nous désignons par  $N$  l'ensemble des personnages qui sont visibles sur l'indexable, par  $S_i$  l'ensemble des personnages qui sont présents dans l'indexeur  $i$  (par exemple dans l'indexeur 2 sur la Figure 2,  $S_2 = \{\text{Jeremy, Maria}\}$ ). Le contenu de l'indexable après propagation est égal à l'union des contenus des indexeurs sélectionnés :  $N = \cup S_i$

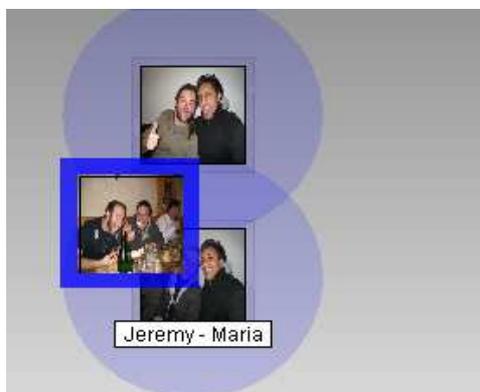
Par exemple, sur la figure 2,

$$N = S_1 \cup S_2$$

Comme  $S_1 = \{\text{Jeremy, Willey}\}$  et  $S_2 = \{\text{Jeremy, Maria}\}, \{\text{Jeremy, Maria, Willey}\}$  est propagé vers  $N$ .

Dans certains cas des personnages visibles sur l'indexable ne figurent pas dans le contenu de tous les indexeurs disponibles. Ils peuvent alors être créés à la main et intégrés dans la liste des personnages, et en même temps au contenu de l'indexable.

A l'inverse il peut aussi y avoir des cas où il y a des intrus dans les indexeurs sélectionnés, c'est-à-dire des personnages en plus de ceux qui figurent sur l'indexable. Si on propage l'union des contenus des indexeurs sélectionnés, l'indexable risque de se retrouver avec des personnages en trop. Or ceux-ci sont nécessairement dans la liste des personnages puisqu'il a fallu les créer pour qu'ils figurent comme contenu d'un ou plusieurs indexeurs. Ils peuvent être retirés du contenu de l'indexable en cliquant sur leur image dans la liste des personnages. L'expérience a montré lors des tests que ce procédé de soustraction est généralement peu nécessaire, mais peut s'avérer utile.



*Fig. 2 – Indexation d’une photo avec deux indexeurs*

## **4 Organisation des photos en un Diagramme de Hasse**

### **4.1. Le treillis de concepts des photos indexées**

La mise en œuvre de la propagation présentée ci-dessus suppose que les indexeurs soient organisés sur la planche photo de manière à rendre leur contenu clair et facile d’accès. A cette fin nous utilisons les techniques de l’Analyse Formelle de Concepts (AFC) et nous nous proposons de les organiser selon un Diagramme de Hasse.

Un Diagramme de Hasse est une représentation pratique d’un treillis de concepts (ou treillis de Galois). Dans l’AFC, un ensemble d’objets dotés de propriétés (ou attributs) peut être organisé en un treillis de concepts ; un concept contient l’ensemble des objets qui possèdent un même sous-ensemble de propriétés, en retenant le plus grand des sous-ensembles de propriétés communes entre les objets. Dans notre cas, nous considérons les photos comme des objets et les personnages sur les photos comme les propriétés de ces objets. Le processus d’organisation débute avec la construction d’un contexte formel, ou plus simplement contexte, qui est une table avec les objets disposés en lignes et les propriétés en colonnes. Chaque case est marquée (par exemple avec la valeur 1) si l’objet en ligne possède la propriété en colonne ; elle n’est pas marquée (évaluée à 0) dans le cas contraire.

Formellement, un contexte est un triplet  $(G, M, I)$  où  $G$  est un ensemble d’objets,  $M$  un ensemble de propriétés, et  $I$  une relation binaire entre les objets et les propriétés, i.e.  $I \subseteq G \times M$ .

La table 1 présente un contexte formel très simple construit à partir de notre exemple de photos déjà indexées, où  $G$  contient 6 photos et  $M$  contient quatre propriétés, c’est-à-dire quatre personnages potentiellement présents sur les photos.

Propriétés Objets	Maria	Willey	Jeremy	Peter
Photo P1	1	0	0	1
Photo P2	0	0	1	1
Photo P3	1	1	1	1
Photo P4	1	1	0	0
Photo P5	0	0	0	1
Photo P6	0	0	0	1

**Table 1.** Un contexte avec photos (objets) et personnages (propriétés)

La construction du treillis de concepts se poursuit avec la recherche des concepts. Un concept est défini par une paire de sous-ensembles : un sous-ensemble d’objets, appelé l’extension du concept, et un sous-ensemble de propriétés appelé l’intension, qui est le sous-ensemble maximal des propriétés que les objets du concept partagent.

Pour le contexte de la table 1, les concepts sont les nœuds du graphe présenté dans la Figure 3, comme par exemple le concept  $(\{P2, P3\}, \{\text{Jeremy}, \text{Peter}\})$  qui a pour extension  $\{P2, P3\}$  et pour intention  $\{\text{Jeremy}, \text{Peter}\}$ .

Pour un ensemble d’objets  $O \subseteq G$  et un ensemble de propriétés  $A \subseteq M$ , on définit l’ensemble des propriétés communes aux objets de  $O$  :

$$f : 2^G \rightarrow 2^M, f(O) = \{a \in A \mid \forall o \in O, (o, a) \in I\}$$

et l’ensemble des objets qui ont toutes leur propriétés dans  $A$  :

$$g : 2^M \rightarrow 2^G, g(A) = \{o \in O \mid \forall a \in A, (o, a) \in I\}.$$

La paire  $(f, g)$  est une correspondance de Galois entre  $(2^G, \subseteq)$  et  $(2^M, \subseteq)$ .

Un concept formel du contexte  $(G, M, I)$  est une paire  $(O, A)$  avec  $O \subseteq G, A \subseteq M$  et  $A = f(O)$  et  $O = g(A)$ .

Après l’identification des concepts, le but suivant est de construire le treillis dont les éléments sont les concepts. Nous définissons un ordre partiel dans l’ensemble des concepts, tel que toute paire de concepts a un supremum (le plus petit majorant commun aux deux concepts), et un infimum (le plus grand minorant commun). Formellement, soit  $L$  l’ensemble des concepts de  $(G, M, I)$  et soit  $\leq_L$ , l’ordre partiel est défini par :

$$(O_1, A_1) \leq_L (O_2, A_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow O_2 \subseteq O_1.$$

La paire  $(L, \leq_L)$  est appelée le treillis des concepts de  $(G, M, I)$ .

La Figure 3 montre le Diagramme de Hasse du treillis de concepts des photos de l’exemple. C’est un graphe dont les nœuds sont les concepts, ordonnés de haut en bas, selon leur ordre dans le treillis. Chaque concept montre son extension et son intension entre accolades (les éléments en gras seront explicités dans la suite). Les arêtes du graphe relient les concepts qui sont dans une relation d’ordre directe, sans concept intermédiaire. L’extension d’un concept est un groupe de photos  $P_i$  qui partagent les mêmes personnages ; ces derniers forment l’intension du concept. On peut voir qu’une photo peut apparaître dans plusieurs concepts. Il en est de même pour une propriété.

Pour simplifier les étiquettes des concepts, il est possible de ne mentionner que les extensions réduites (en gras sur la Figure 3). Une extension réduite d'un concept (O, A) est l'ensemble des objets qui appartiennent à O et n'appartiennent pas à un concept inférieur, c'est-à-dire l'ensemble des objets qui n'ont pas d'autres propriétés que celles appartenant à A.

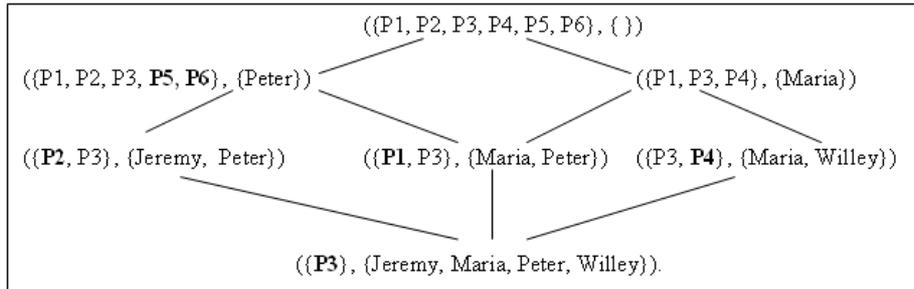


Fig. 3 – Treillis de concepts associé au contexte de la Table 1

Un concept dont l'extension réduite est non vide est appelé concept-objet. Pour chaque objet (photo), il existe un unique concept-objet qui constitue le concept le plus spécifique contenant l'objet. Dans la mesure où nous souhaitons qu'une photo ne soit associée qu'à une unique description, la plus spécifique, nous réduisons le treillis aux seuls concept-objets.

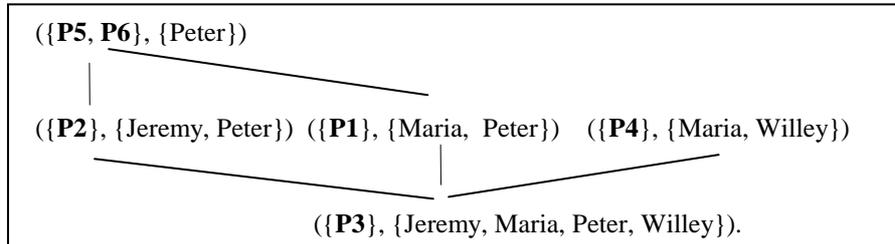


Fig. 4 – La Sous Hiérarchie de Galois associée au contexte de la Table 1

Ce principe d'élimination des nœuds dont les extensions réduites sont vides a été proposé dans (Godin et al., 1995) sous le nom PCL/X. On peut le voir appliqué à notre exemple dans la Figure 4. Nous obtenons une Sous Hiérarchie de Galois selon la terminologie employée dans (Godin & Chau, 1999) restreinte aux concepts objets. De manière plus explicite nous l'appellerons une Sous-Hiérarchie de Galois Objet (Object Galois Sub-Hierarchy – OGSH). La visualisation est plus claire puisque dans notre cas on se focalise sur les objets, à savoir les photos.

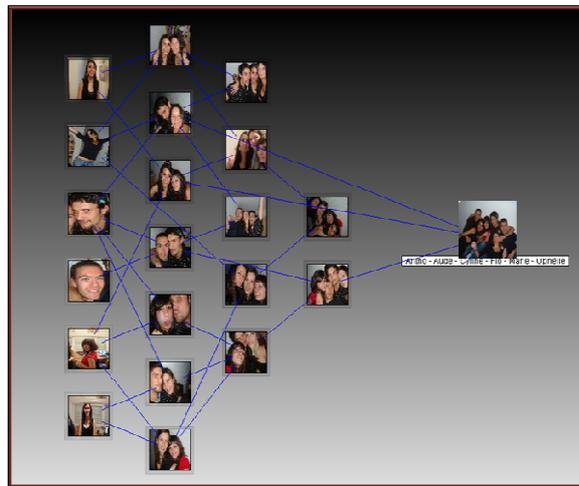
## 4.2. Construction incrémentale du Diagramme de Hasse

De nombreux algorithmes ont été proposés pour construire un treillis de Galois et son Diagramme de Hasse à partir d'un contexte. On trouvera un état de l'art succinct dans (Valtchev et al., 2000). Plus précisément (Arévalo et al., 2007) comparent trois applications principales qui ne considèrent que des OGS, ce qui est aussi notre cas. Ces algorithmes calculent en premier lieu la topologie du graphe, et ensuite génèrent le Diagramme de Hasse, ou au mieux construisent de manière incrémentale le Diagramme de Hasse à chaque pas de calcul de la topologie du graphe. Cette approche présente un inconvénient majeur pour notre cas.

Lors de l'indexation d'une photo à l'aide des indexeurs organisés en Diagramme de Hasse, l'indexable, après propagation, doit pouvoir être intégré dans le diagramme qui doit être en conséquence mis à jour dynamiquement. Comme le diagramme est bâti à partir d'ensembles de photos représentant les mêmes groupes de personnes – ces ensembles forment les concepts – il est essentiel que l'utilisatrice trouve facilement les groupes de personnes pour indexer de nouvelles photos. En conséquence à chaque fois qu'une nouvelle photo vient d'être indexée elle doit être intégrée dans le diagramme avec un minimum de remaniements. Les anciens concepts doivent ne pas bouger, ou au pire bouger mais sans changer de voisinage. L'objectif d'une telle stratégie est le maintien de la carte mentale de l'utilisatrice, l'exigence que l'utilisatrice ne soit pas désorientée par des mouvements inattendus (Misue et al., 1995). Les algorithmes de visualisation de Diagrammes de Hasse qui calculent en premier le treillis ne sont pas applicables ici car le diagramme, à l'image de la sous-hiérarchie, n'est pas connu à l'avance puisqu'il se construit au fur et à mesure qu'il sert à indexer de nouvelles photos. Tous les autres algorithmes qui construisent le Diagramme de Hasse de manière incrémentale sont intéressants s'ils conservent les concepts à leur place, ou du moins ne les déplacent que légèrement en particulier en préservant au maximum leurs voisins sur le plan. Mais les algorithmes proposés dans la littérature ont d'autres priorités : la vitesse et le respect de certaines règles d'esthétique parmi lesquelles la limitation du nombre de croisements d'arêtes (Battista et al., 1999). Bien que ces objectifs soient justifiés, et même s'ils restent pertinents dans notre cas, ils nous conduisent à un dilemme. La construction du graphe est incrémentale et on ne peut donc pas prédire les concepts et les arêtes qui vont apparaître. Comme par ailleurs on veut maintenir au maximum la carte mentale de l'utilisatrice, on ne peut pas réorganiser les concepts sur le plan en permanence afin de limiter le nombre de croisements d'arêtes. Le maintien de la carte mentale est notre première priorité, même si elle implique de ne pas gérer les croisements d'arêtes. De plus nous verrons plus loin que, dans notre cas particulier, la contrainte esthétique sur les arêtes est moins exigeante. Il nous faut donc proposer un nouvel algorithme incrémental centré sur le maintien de la carte mentale.

L'algorithme que nous présentons met en œuvre deux techniques. La première fait appel à un jeu de forces, une simulation d'un jeu de ressorts bien connu dans la communauté 'dessin de graphes' (Eades, 1984). Un algorithme de ce type est par exemple appliqué pour dessiner des treillis dans (Freese, 2004) (Hannan & Pogel, 2006). Cependant dans ce dernier cas le treillis est calculé avant de mettre en œuvre

l'algorithme d'affichage, ce qui va à l'encontre d'une stratégie incrémentale avec maintien de la carte mentale. A l'inverse, la seconde approche originale que nous présentons ici consiste à faire émerger le Diagramme de Hasse en premier à l'aide du jeu de forces, et ensuite d'identifier l'apparition des concepts. L'OGSH est donc le résultat de l'organisation du Diagramme de Hasse, et non le résultat d'un calcul préalable.



*Fig. 5 – Un Diagramme de Hasse auto-organisé de photos*

L'algorithme est le suivant. Comme dans (Freese, 2004), une nouvelle propriété est affectée à chaque photo lors de son indexation. Elle représente son rang. C'est un entier dont la valeur est égale au nombre de propriétés de la photo, c'est-à-dire au nombre de personnages présents. Une force horizontale est appliquée à toutes les photos indexées avec une intensité proportionnelle à leur rang. Les photos se déplacent alors et se positionnent sur une ligne horizontale selon leur rang. Les photos qui ont le même nombre de personnages forment des colonnes. Nous les fixons sur l'axe des abscisses.

Une nouvelle force est appliquée entre tous les objets uniquement par colonne et selon l'axe vertical. La force entre deux atomes sur une même colonne est calculée en utilisant la distance de Hamming. C'est un entier dont la valeur est proportionnelle à la différence des propriétés entre deux objets. Suite à l'application de cette force, toutes les photos qui représentent les mêmes personnages forment des tas correspondant exactement aux concepts objets, et tous les concepts sont séparés verticalement sur chaque colonne. L'intension du concept est l'ensemble des individus présents sur les photos qui forment le tas. Une troisième force est finalement appliquée pour égaliser la distance entre concepts.

A chaque fois que les forces sont appliquées sur une photo nouvellement indexée, elle va soit rejoindre un concept existant qui contient le même groupe d'individus, soit créer un nouveau concept sur une colonne contenant le même nombre d'individus.

Pour se faire une place, elle repousse verticalement les concepts déjà présents sur la colonne. La nouvelle photo peut éventuellement créer une nouvelle colonne si elle possède un nombre de personnes différent de ceux des colonnes déjà en place. Ces dernières glissent à leur nouvelle position sans modifier l'organisation des concepts qui les composent.

Quand une nouvelle photo crée un nouveau concept, des liens sont rajoutés avec les concepts immédiatement inférieurs et supérieurs qui contiennent des personnages en commun. Ce calcul est pour l'instant en cours d'optimisation. La Figure 5 montre un Diagramme de Hasse créé lors des tests qui sont décrits dans la conclusion ci-dessous. Pour des raisons pratiques, le diagramme est organisé hiérarchiquement de gauche à droite.

## 5 Utilisation du Diagramme de Hasse pour l'indexation

Il est maintenant possible d'utiliser le Diagramme de Hasse pour indexer de nouvelles photos à l'aide d'indexeurs bien organisés. Visible au dessus de chaque concept on trouve soit la photo la plus récemment indexée soit une photo choisie par l'utilisatrice parce qu'elle la trouve la plus représentative du groupe de personnes correspondant à l'intension. Sur la Fig. 1 les nouvelles photos à indexer sont entassées à droite, le Diagramme de Hasse des indexeurs est au centre ; une lentille "fish eye" est appliquée selon les principes décrits dans (Furnas, 1986) et la liste des personnages est située verticalement à gauche. Les liens entre les concepts en relation apparaissent localement et uniquement lorsqu'on fouille le graphe à l'aide de la souris. Ceci justifie que la minimisation du nombre de croisements d'arêtes, bien qu'intéressante, est une contrainte moins importante que le maintien de la carte mentale.

D'autres outils sont disponibles comme la possibilité de regrouper les indexables qui montrent un même groupe d'individus et de les indexer en une seule fois.

## 6 Tests

Deux séries de tests ont été menées avec un protocole bien établi portant sur des données objectives et subjectives. Le but était de mesurer les performances de notre application dénommée PhotoMap pour l'indexation de photos sociales face à Facebook et Flickr. Les populations concernées étaient des étudiants recrutés sur la base du volontariat. Ils avaient en moyenne 21 ans, et étaient aux trois quarts des hommes. Il est à noter que cette population correspond à un profil d'utilisateurs type pour l'application. Le premier test a permis de mettre au point la méthode d'évaluation avec 4 testeurs. Le second test avec 17 volontaires a permis de mesurer certaines performances et de recueillir des critiques. La plupart des testeurs connaissaient déjà ou avaient utilisé Facebook, quelques uns avaient utilisé Flickr, et évidemment aucun ne connaissaient Photomap. Il est intéressant de noter que

PhotoMap partait avec un léger handicap sur ce point parce que sa manipulation était ignorée de tous. Chaque test sur un logiciel commençait par une formation avec cinq photos à manipuler. Les testeurs devaient ensuite indexer seuls 40 photos, toujours les mêmes, prises lors de deux soirées auxquelles ils avaient participé. L'opération était répétée par chaque testeur sur les trois logiciels dans un ordre tiré au hasard.

**Résultats.** Deux indicateurs étaient particulièrement observés. 1) Le temps pour trouver une photo particulière après l'indexation d'un premier noyau de photos : il était jusqu'à deux fois inférieur en moyenne avec PhotoMap (environ 11 secondes) qu'avec les concurrents (environ 20 secondes). 2) Le temps moyen d'indexation observé pour les 15 dernières photos : les résultats donnaient en moyenne 3'10'' pour Facebook, 2'30'' pour Flickr et 2'35'' pour PhotoMap. Mais pour PhotoMap, le plus pénalisant était la création de nouveaux personnages alors que, inversement, le principe de propagation était très sollicité et vécu comme un facteur très productif.

Au plan qualitatif, la majorité des testeurs ont classé PhotoMap en première position seul ou à égalité avec un autre de ses deux concurrents sur tous les critères proposés : "simplicité", "qualité de l'indexation produite", "fonctions d'assistance", et surtout "caractère ludique". Le résultat le plus inattendu a été d'observer que 7 des 17 participants ont spontanément abandonné en cours de route par lassitude l'indexation des 40 photos avec Facebook et/ou Flickr. Par contre tous ont été jusqu'au bout avec PhotoMap, appréciant la révélation progressive des groupes humains dans le Diagramme de Hasse. Contrairement à une crainte préalable, la navigation dans le diagramme n'a jamais été qualifiée de complexe. La construction incrémentale et animée répond semble-t-il à l'objectif que l'on s'était fixé de maintien de la carte mentale de l'utilisateur. Seule l'utilisation des liaisons entre concepts a posé problème, les liens ayant été peu compris, peu utilisés, et causant même parfois de la gêne, non par leur croisement, mais par leur simple présence.

Sur les suites à donner, tous les testeurs étaient positifs, souhaitant disposer d'un tel environnement seul ou avec Facebook. Le point fort de la méthode est que l'effort d'indexation s'accompagne d'un retour d'information par la révélation de l'organisation des photos dans un Diagramme de Hasse.

## **7 Conclusion**

Le Diagramme de Hasse et sa construction incrémentale semblent donc être de bons outils tant d'indexation que d'organisation de photos sociales, au moins en nombre limité (40 photos). La suite de nos travaux portera sur la montée en charge et sur l'assistance visuelle. En effet en nous basant sur les fondements théoriques bien établis de l'Analyse Formelle de Concepts nous pouvons avancer l'hypothèse que ce mode d'organisation est le plus à même de gérer la complexité d'une montée en charge avec plusieurs centaines de photos. Par contre, en termes de visualisation, il conviendra de mettre en œuvre des outils de zoom et de filtrage puissants et simples à manipuler sur l'ensemble du diagramme. La suite des travaux portera aussi sur la construction d'albums personnalisés qui pourront être partagés et collectivement enrichis sur un Diagramme de Hasse également partagé.

## Références

- ARÉVALO G., BERRY A., HUCHARD M., PERROT G., SIGAYRET A. (2007). Performances of Galois Sub-hierarchy-building Algorithms. *ICFCA*. p 166-180.
- BATTISTA G., EADES P., TAMASSIA R., TOLLIS I. (1999). Graph drawing. Algorithms for the visualisation of graphs. *Prentice Hall*.
- CRAMPES M., VILLERD J., EMERY A., RANWEZ S.. (2007). Automatic Playlist Composition in a Dynamic Music Landscape. In *Proceedings of the International Workshop On Semantically Aware Document Processing And Indexing, SADPI'07. ACM New York, vol. 259*, p. 15–20.
- EADES P. (1984). A heuristic for graph drawing. In *Proc. of the 13th Manitoba Conference on Numerical Mathematics and Computing. Utilitas Mathematica, vol. 2*. p. 149-160.
- EKLUND P., DUCROU J., WILSON T. (2006). An Intelligent User Interface for Browsing and Search MPEG-7 Images using Concept Lattices. In *Proc of the 4th International Conference on Concept Lattices and Their Applications, LNAI, Springer-Verlag*.
- FERRÉ S. (2007). CAMELIS: Organizing and Browsing a Personal Photo Collection with a Logical Information System. In *Proc. of the 5th International. Conference on. Concept Lattices and Their Applications*. p. 112-123.
- FREESE R. (2004). Automated Lattice Drawing. *Lecture Notes in Artificial Intelligence, 2961, Springer*. p. 112-127. Berlin.
- FURNAS G.W. (1986). Generalized Fisheye Views. In *Human Factors in Computing Systems (CHI'86) Mantei & Orbeton Eds, ACM*. p.16-23.
- GANTER B. & WILLE R. (1999). Formal Concept Analysis: Mathematical Foundations. *Springer*.
- GODIN R., MINEAU G., MISSAOUI R. (1995). Incremental structuring of knowledge bases. In *Proceedings of the International Knowledge Retrieval, Use, and Storage for Efficiency Symposium (KRUSE'95)*. p. 179-198. Santa Cruz
- GODIN R. & CHAU T. (1999). Comparaison d'algorithmes de construction de hiérarchies de classes. *L'objet 5(3/4)*.
- HANNAN T. & POGEL A. (2006). Spring-Based Lattice Drawing Highlighting Conceptual Similarity. In *Proc of ICFCA'06*. p. 264-279.
- LOISANT E., SAINT-PAUL R., MARTINEZ J., RASCHIA G., MOUADDIB N. (2003). Browsing Clusters of Similar Images. In *Proc. of BDA '03*. Lyon.
- MISUE K., EADES P., LAI W., SUGIYAMA K. (1995). Layout Adjustment and the Mental Map. In *Journal of Visual Languages and Computing, Vol. 6*. p. 183–210.
- MONAGHAN F. & O'SULLIVAN D. (2007). Leveraging Ontologies, Context and Social Networks to Automate Photo Annotation. In *2nd International Conference on Semantics and Digital Media Technologies*. p. 5-7. Genova
- NSTC REPORT, (2006). Face Recognition. *NSTC, Committee on Technology, Committee on Homeland and National Security, Subcommittee on Biometrics*.
- VALTCHEV P., GROSSER D., ROUME C., ROUANE HACENE M. (2003). Galicia: an open platform for lattices. In *Using Conceptual Structures: Contributions to the 11th Intl. Conference on Conceptual Structures ICCS'03*.
- VALTCHEV P., MISSAOUI R., LEBRUN P. (2000). A Fast Algorithm for Building the Hasse Diagram of a Galois Lattice. In *Proceedings of the Colloque LaCIM*. Montréal.