UNIVERSITE
BRETAGNE
LOIRE

THÈSE / Télécom Bretagne
sous le sceau de l'Université Bretagne Loire

pour obtenir le grade de Docteur de Télécom Bretagne
En accréditation conjointe avec l'Ecole Doctorale Sicma
Mention : Sciences et Technologies de l'Information et de la Communication

présentée par
Ala Aboudib

préparée dans le département Electronique
Laboratoire Labsticc - IAS

Neuro-inspired Architectures for the Acquisition and Processing of Visual Information

Thèse soutenue le 2 décembre 2016
Devant le jury composé de :

Gérard Berry
Professeur, Collège de France, Inria - Paris / président

Jean Ponce
Professeur, Ecole Normale Supérieure, Inria - Paris / rapporteur

Olivier Le Meur
Maître de conférences (HDR), Université de Rennes 1, Irisa / rapporteur

Fan Yang
Professeur, Université de Bourgogne, LE2I / examinatrice

Claude Berrou
Professeur, Télécom Bretagne, Labsticc / examinateur

Vincent Gripon
Chargé de recherche, Télécom Bretagne, Labsticc / examinateur

Gilles Coppin
Professeur, Télécom Bretagne, Labsticc / directeur de thèse

N° d'ordre : 2016telb0419

Sous le sceau de l'Université Bretagne Loire

Télécom Bretagne

En accréditation conjointe avec l'Ecole Doctorale Sicma

Neuro-inspired Architectures for the Acquisition and Processing of Visual Information

---

Thèse de Doctorat

Mention : STICC

Présentée par **Ala Aboudib**
Département : ELEC
Laboratoire : Lab-STICC Pôle : IAS

Directeur de thèse : Gilles Coppin
Encadrant de thèse : Vincent Gripon

Soutenue le 2 Décembre, 2016

Jury :
M. Jean Ponce, professeur, ENS / INRIA, (rapporteur)
M. Olivier Le Meur, maître de conférence HDR, Université de Rennes 1 / IRISA, (rapporteur)
M. Gérard Berry, professeur, Collège de France / INRIA, (examinateur)
Mme Fan Yang, professeur, Université de Bourgogne / LE2I, (examinatrice)
M. Claude Berrou, professeur, Télécom Bretagne / Lab-STICC, (examinateur)
M. Vincent Gripon, chargé de recherche, Télécom Bretagne / Lab-STICC, (encadrant)
M. Gilles Coppin, professeur, Télécom Bretagne / Lab-STICC, (directeur)

Résumé

Depuis de nombreuses années, la conception de machines intelligentes est demeuré un sujet de recherche majeure. Il s'est avéré que la compréhension de l'intelligence humaine est l'un des plus grands défis que l'humanité a rencontré. Cette quête pour "réinventer" notre propre intelligence est motivée non seulement par la pure curiosité intellectuelle qui a toujours caractérisé l'homme, mais aussi par le potentiel d'une profonde transformation de notre civilisation et de domaines tel que le diagnostic médical, la communication, le transport, l'art, la recherche scientifique, la médecine, le monde des affaires...

Cette dernière décennie a connu un progrès remarquable dans une variété de domaines de l'intelligence artificielle, notamment en vision par ordinateur et en traitement du son et du langage. Ces avancées ont amorcé l'émergence de nouvelles technologies qui promettent d'apporter des changements fondamentaux à nos sociétés tels que les véhicules autonomes, la traduction automatique du texte et de la parole, l'éducation personnalisée et le commerce électronique.

Une limitation majeure des systèmes actuels de l'IA, c'est d'être restreints par une tâche spécifique et limitée; un modèle conçu pour reconnaître un objet dans une image ne peut pas être adapté à de nouvelles tâches comme la reconnaissance vocale par exemple. Bien que l'architecture d'un tel modèle soit suffisamment flexible pour reconnaître des objets qu'il n'a jamais appris, cette flexibilité est limitée à la tâche de reconnaissance d'objets sans pouvoir aller au-delà. Ce type d'IA est connu sous le terme d'*Intelligence Artificielle Faible*. Un ingrédient clé pour franchir cette limite et atteindre un niveau d'intelligence plus proche de celui de l'homme, ce que l'on appelle l'*Intelligence Artificielle Forte*, est de résoudre le problème de l'apprentissage

non supervisé, où la connaissance devrait être acquise sans l'intervention d'un expert humain, ce qui demeure une question ouverte aujourd'hui.

Cependant, certains comportements que l'on considère intelligents chez l'homme peuvent être accomplis par une machine sans que celle-ci soit pourtant considérée "intelligente". Par exemple, des tâches difficiles qui exigent notre intelligence, telles que les calculs mathématiques formels, peuvent être facilement programmés et réalisées par un ordinateur à des vitesses qui nous dépassent. En réalité, les ordinateurs sont régulièrement utilisés pour ce type de calculs sans avoir recours aux techniques de l'IA. En revanche, des tâches qui nous paraissent simples et banales telle que la reconnaissance d'un visage ou répondre à une question, sont parmi les plus difficiles à effectuer par une machine et nécessitent une recherche intensive en IA.

Ce contraste pour ce que l'on qualifie d'acte intelligent entre l'homme et la machine est du probablement à une différence fondamentale dans leurs structures ainsi que dans leurs architectures. La nature distribuée et ultra-connectée des neurones et les zones corticales dans le cerveau sont plus favorables aux tâches nécessitant un traitement massivement parallèle telle que la reconnaissance d'images ou de vidéos. Par contre, l'architecture von-Neuman de l'ordinateur basée sur un traitement successif des instructions est mieux adapté aux tâches intrinsèquement séquentielles comme les calculs formels.

Une leçon importante à tirer de la recherche en intelligence artificielle au vingtième siècle est le fait que si certaines tâches cognitives telles que la compréhension du langage ou d'une pièce de théâtre nous paraissent simples et évidentes, cela n'implique pas la simplicité des algorithmes et des circuits neuraux qui se cachent derrière.

Le cerveau est un organe extrêmement sophistiqué. Il est principalement composé d'un réseau de neurones dense massivement interconnecté. Comprendre les interactions complexes entre ces neurones est l'un des défis majeurs pour la communauté scientifique. Selon David Marr dans son célèbre ouvrage sur la vision (Marr, 1982a), l'analyse d'un système complexe tel que le cerveau devrait procéder selon trois niveaux principaux. Au premier niveau ou le niveau *computationnel*, le problème est défini d'une manière grossière. Les entrées du système sont identifiées ainsi que les sor-

ties attendues. Au deuxième niveau dit *algorithmique*, un algorithme pour effectuer la transition entre les entrées et les sorties devrait être élaboré. Une architecture matérielle pour implémenter cet algorithme devrait être réalisée dans le troisième niveau, le niveau de *conception matériel*, pour obtenir une solution analogue à celle du deuxième niveau.

Dans le travail présenté ici, nous nous intéressons au troisième niveau d'analyse de Marr. Nous nous inspirons de l'architecture du cortex cérébral, notamment de celle du cortex visuel, pour proposer des modèles de traitement d'images. Ce choix est motivé par le fait que le cortex visuel est parmi les aires cérébrales les plus étudiées et par l'existence de ressources abondantes à notre disposition, fournies par la communauté scientifique.

Nous nous focalisons sur le problème de l'acquisition et le traitement de l'information visuelle dans les réseaux de neurones artificiels. L'acquisition est la première étape dans la chaîne de traitement visuel qui commence dans la rétine. Dans cette étape, la lumière issue de la scène extérieure est capturée et transformée en un signal électrochimique que le cerveau est capable de manipuler. Le traitement de ce signal comprend une multitude de mécanismes sophistiqués tels que l'extraction de caractéristiques visuelles ou l'organisation de l'information acquise dans la mémoire. Dans les premières couches du cortex visuel, plus proches de la rétine, les neurones représentent des concepts simples; un neurone ne peut être excité que par des motifs visuels rudimentaires tels que l'apparition d'un contour orienté ou d'une couleur spécifique dans une zone très limitée du champ visuel. En s'éloignant de la rétine, cette zone devient de plus en plus étendue alors que les stimulis associés deviennent de plus en plus complexes. Dans les couches supérieures, tel que dans le lobe frontal, nous constatons que les concepts représentés par les neurones sont plus abstraits, et peuvent être invoqués non seulement par des stimulis extérieurs mais également par l'activité interne du cerveau.

Dans cette thèse, nous abordons le système visuel selon trois niveaux différents de traitement. Nous commençons par le niveau le plus bas qui représente la rétine et les premières couches de la voie ventrale. Nous examinons l'architecture de ces

couches et nous proposons un nouveau modèle d'acquisition qui intègre des propriétés importantes observées et souvent négligées par les modèles d'acquisition classiques.

Nous nous concentrons ensuite sur le problème de la mémoire au niveau le plus haut. Plutôt que proposer un nouveau modèle de mémoire à ce niveau, nous nous intéressons à un modèle de mémoire associative existant basé sur une architecture neuro-inspirée, et nous proposons d'améliorer les algorithmes de récupération d'informations associés.

Nous traitons finalement le niveau intermédiaire faisant le lien entre les deux niveaux précédents : le niveau d'acquisition du signal visuel, et le niveau de la mémoire. Trouver l'algorithme qui permet de transformer le signal rudimentaire acquis dans la rétine en une représentation robuste et abstraite dans la mémoire est une des questions fondamentales à laquelle se confronte la recherche en vision. Nous souhaitons contribuer à une meilleure compréhension de ce problème en proposant un nouveau réseau de neurones artificiels pour traiter le problème d'appariement de caractéristique d'image qui est un problème essentiel de la vision par ordinateur.

Un bref historique

De nombreuses architectures ont été proposées dans la littérature pour la modélisation des différents niveaux de traitement du cortex visuel. Certains modèles se sont limités à la modélisation d'un seul niveau alors que d'autres ont essayé d'en traiter plusieurs.

En ce qui concerne l'acquisition visuelle, certains modèles ont proposé d'approximer les calculs complexes effectuées par la rétine, notamment celui de (Wohrer and Kornprobst, 2009) surnommé *la rétine virtuelle*. Ce modèle applique une convolution linéaire pour imiter la fonction des cellules ganglionnaires, suivie par l'application d'une non-linéarité statique et d'une procédure de génération des potentiels d'action. D'autres modèles comme celui de (Lorach et al., 2012) ont proposé de modéliser les propriétés spatio-temporelles de la rétine en utilisant des capteurs dynamiques asynchrones ou (DVS). Ces capteurs dits événementiels sont directement inspirés de la fonction de la rétine biologique qui suit un régime asynchrone d'acquisition.

Certains modèles d'acquisition vont au-delà de la rétine pour accomplir des tâches visuelles spécifiques telle que la modélisation de l'attention visuelle et la prédiction des mouvements oculaires. Le célèbre modèle d'Itti introduit dans (Itti et al., 1998) a été parmi les premiers à offrir une méthode efficace pour estimer la saillance dans les images naturelles en se basant sur le modèle théorique de (Koch and Ullman, 1987). Pour ce faire, l'image est traitée selon trois modalités perceptuelles différentes: les contrastes des niveaux d'intensité, les orientations locales des contours et les couleurs. Cela a été directement inspiré par un mécanisme de décomposition similaire effectué dans les premières couches de la voie ventrale grâce à l'existence d'une grande variété des cellules ganglionnaires.

La recherche sur la modélisation de l'attention visuelle et l'estimation de la saillance a été abondante ces dernières années. la plupart des modèles proposés se sont appuyés sur le modèle théorique de (Koch and Ullman, 1987) et son implémentation dans (Itti et al., 1998). Un formalisme générique pour unifier ces modèles a été présenté dans (Walther and Koch, 2007). Dans ce travail, Walther a proposé une méthode pour combiner les mécanismes attentionnels avec les techniques de reconnaissance d'objets s'appuyant sur la fameuse théorie de (Hubel and Wiesel, 1959). Ce modèle générique a inspiré celui que nous proposons dans le chapitre 2. De la même manière, nous définissons une approche générique pour représenter la rétine et les premières couches du système visuel, mais nous allons plus loin pour incorporer d'autres propriétés qui rendent l'architecture de notre modèle plus proche de celle du cortex visuel.

De nombreuses recherches ont été consacrées à l'investigation du niveau supérieur de la chaîne de traitement visuel, le niveau de la mémoire. Le but est de trouver des représentations de l'information visuelle d'une manière associative, similaire à celle utilisée dans le cerveau. Pour cela, de nombreux modèles de réseaux de neurones artificiels ont été proposés. Le réseau de Hopfield (Hopfield, 1982) est un fameux exemple d'une architecture capable de stocker des messages de taille fixe à travers un réseau de connexions synaptiques denses entre les neurones. Chaque neurone est connecté à tous les autres neurones, et un poids synaptique est associé à chaque

connexion. Le modèle de McCulloch-Pitts (McCulloch and Pitts, 1943) a été utilisé pour représenter ces neurones, et la règle d'apprentissage de Hebb (Hebb, 1949) a été utilisé pour syntoniser les poids synaptiques selon une procédure d'optimisation d'une fonction de coût prédéfinie.

La machine de Bolzmann (BM) introduite dans (Ackley et al., 1985) est un autre modèle célèbre de mémoire associative. ce réseau, doté de la même architecture que celle du réseau de Hopfield, applique un modèle stochastique pour l'activation des neurones inspiré par la nature stochastique des neurones biologiques. Cela rend la procédure d'optimisation plus résistante aux minima locaux lors de l'apprentissage ce qui constitue une limitation majeure du réseau de Hopfield. Une autre variante de la BM, c'est la machine de Boltzmann restreinte (RBM). Elle consiste de deux couches de neurones, une couche dite *cachée* et une autre *visible*, connectées d'une manière bipartie. Cette configuration permet au RBM de stocker l'information d'une manière plus robuste. Elle permet également d'empiler plusieurs couches cachées ce qui donne accès à des meilleures représentations de l'information et donc à des performances supérieures.

Une architecture similaire à celle de la RBM est appelée l'auto-encodeur (Hinton and Salakhutdinov, 2006; Bengio, 2009). C'est un modèle de mémoire qui consiste typiquement en trois couches de neurones: une couche d'entrée, une couche cachée et une couche de sortie. Lors de la phase d'entraînement, les neurones de la couche cachée apprennent à encoder l'entrée d'une façon à pouvoir la reconstruire à la sortie. En plus d'être utilisés comme modèles de mémoire, les auto-encodeurs sont généralement utilisés pour la réduction de la dimension lorsque les signaux d'entrée sont projetés dans un espace de dimension inférieure. Le préfixe 'auto-' vient du fait qu'un auto-encodeur vise à reconstruire l'entrée d'une manière non supervisée. Comme dans une RBM, les couches cachées d'un auto-encodeur peuvent être empilées pour réaliser des architectures plus 'profondes'.

Plus récemment, Gripon et Berrou ont proposé un nouveau modèle de mémoire associative (Gripon and Berrou, 2011) qui peut être vu comme une généralisation du réseau de Palm-Willshaw (Schwenker et al., 1996). Son architecture a été inspirée

par l'organisation en micro-colonnes dans le cortex visuel où les synapses inhibitrices dominent les connexions à courte distance, tandis que les synapses excitatrices prédominent sur les connexions à longue distance. Ce modèle offre également une capacité de stockage plus élevée que celle du réseau de Hopfield ou la machine de Boltzmann. Nous allons aborder ce modèle dans le chapitre 3 où nous proposons un nouvel algorithme pour améliorer sa performance de récupération de données.

La représentation de l'information visuelle dans le niveau intermédiaire, entre le niveau d'acquisition et celui de la mémoire, demeure un problème difficile. Plusieurs étapes de traitement sont nécessaires à ce niveau pour transformer les données brutes acquises en mémoires abstraites. La théorie phare de Hubel et Wiesel dans (Hubel and Wiesel, 1959) proposant une succession de couches composées de cellules simples et complexes dans le cortex visuel a ouvert la voie à une meilleure compréhension de ces transformations. Elle propose que les cellules neurales dans le cortex visuel soient organisées en plusieurs couches de manière hiérarchique. Cependant, certaines questions importantes n'ont pas encore été entièrement traitées telle que la nature exacte des transformations effectuées à chaque couche de la hiérarchie.

L'appariement stéréoscopique est un des problèmes fondamentaux du niveau intermédiaire. Son objectif est de comprendre comment les deux images acquises par les yeux sont aperçues comme une seule scène. Parmi les nombreux modèles qu'on trouve dans la littérature, Marr et Poggio ont proposé une architecture neuro-inspirée pour traiter ce problème (Marr and Poggio, 1976). Cette architecture s'appuie sur un algorithme coopératif où chaque neurone n'interagit qu'avec ses voisins les plus proches au travers de connexions synaptiques excitatrices et d'autres inhibitrices. Ce modèle a inspiré notre travail sur le problème d'appariement de caractéristique d'image dans le chapitre 4.

Nos contributions

Dans cette thèse, nous apportons trois contributions principales qui concernent les trois niveaux de traitement visuel mentionnés ci-dessus : Nous introduisons d'abord

une architecture neuro-inspirée pour l'acquisition du signal visuel. Nous proposons ensuite un nouvel algorithme pour la restitution d'information du modèle de mémoire associative proposé dans (Gripon and Berrou, 2011) sur lequel nous nous appuyons pour introduire notre dernière contribution qui concerne le problème d'appariement de caractéristiques visuelles. Voici une liste plus détaillée décrivant ces contributions :

- Dans le **chapitre 2**, nous examinons le problème de l'acquisition et de la représentation des caractéristiques visuelles de bas niveau. Nous étudions la structure de la rétine et les premières couches de la voie ventrale afin de proposer une architecture neurale pour acquérir et représenter l'information visuelle. Ce modèle se distingue par la flexibilité de son architecture. Nous montrons que cette flexibilité permet d'imiter des propriétés importantes du système visuel telle que l'échantillonnage rétinien, la magnification corticale et les mouvements oculaires, ainsi que la notion de champ visuel qui permet de modéliser la distance entre une image et un spectateur. Nous présentons ensuite une étude de cas dans laquelle nous utilisons le modèle proposé pour atteindre l'état de l'art sur le problème d'estimation de la saillance dans les images statiques.

- **Le chapitre 3** concerne la représentation à haut niveau de l'information dans la mémoire. Nous étudions le réseau de neurones surnommé *sparse clustered network (SCN)* proposé dans (Gripon and Berrou, 2011), modèle de mémoire associative. Nous proposons une formulation générique des différents algorithmes de restitution d'information associés à ce modèle, ainsi qu'un nouvel algorithme qui donne une meilleure performance sur la récupération des données à partir des versions partiellement effacées.

- Dans le **chapitre 4**, nous traitons le problème d'appariement de caractéristiques visuelles. Nous proposons un nouveau modèle pour le résoudre en s'appuyant sur l'architecture du SCN présenté dans le chapitre 3. Ce modèle exploite le concept de *grappes* du SCN pour renforcer les contraintes du problème d'appariement.

Nous évaluons ensuite la performance de l'algorithme proposé en la comparant avec l'état de l'art.

Abstract

Computer vision and machine learning are two hot research topics that have witnessed major breakthroughs in recent years. Although advances in these domains have not been exclusively dependent on ideas and principles suggested by neuroscience and neurophysiology, key historical contributions were, however, the fruits of many years of research on the visual cortex and brain function in a more general sense. Examples of such contributions are numerous. This includes the famous work by Hubel and Wiesel on the cat's visual cortex which had and still have a profound influence on machine learning and computer vision research, and the Hebbian theory on synaptic plasticity and learning in the cerebral cortex that was fundamental to the development of artificial neural networks and learning algorithms, which have been at the heart of cutting edge machine learning research in recent years. Studying lateral interactions among neural cells in the retina contributed to designing better image processing and compression techniques, and the study of neural representations of the visual scene in the primary visual cortex has led to the design of interesting mathematical models for image representation such as the log-polar model, or for receptive fields such as Gabor and Difference of Gaussians models.

In this thesis, we follow this line and, thus, we focus on designing neuro-inspired architectures and improving existing ones for processing visual information along three different stages of the visual cortex. At the lowest stage, we propose a neural model for the acquisition of visual signals. This model is closely inspired by the functionality and the architecture of the retina and early layers of the ventral stream. Eye movement, which is a fundamental property of vision in mammals, is built at the heart of the model we introduce, which is one of its main differences from most exist-

ing acquisition models. Designing an acquisition model with eye movement in mind called for the incorporation of other associated properties. Retinal sampling caused by the non-uniform distribution of photo-receptors, and the associated cortical magnification phenomenon amplified at later layers by a similar non-uniform distribution of receptive field sizes and spatial positions, are two ubiquitous properties that our model was inherently adapted to implement. By introducing our acquisition model, we aspire to provide a new framework for implementing vision tasks that need to experiment with eye movement and its associated properties.

On the highest stage, we address the memory problem. At this stage, many questions are raised such as the choice of the appropriate model to use for a specific task, and the best retrieval scheme to implement. In our case, we focus on an existing associative memory model based on a neuro-inspired architecture. This model, called the Sparse Clustered Network, offers a large storage capacity when used as a memory. Moreover, its architecture borrows interesting properties from the visual cortex such as the existence of short-range inhibitory connections, and local competition between neural cells, in addition to long-range excitatory synapses used to store information in a distributed fashion making it robust to many sorts of noise that might affect it, especially the partial loss of information during retrieval. Our main contribution at this stage consists in suggesting improvements on an existing algorithm used to retrieve stored information from partially erased versions of it. Furthermore, we suggest a generic formulation within which all existing retrieval algorithms can fit. It can also be used to guide the design of new retrieval approaches in a modular fashion.

We further extend Sparse Clustered Networks at the intermediate stage. We propose a new architecture adapted to deal with the feature correspondence problem, which is a fundamental problem in computer vision. Most approaches in literature use optimization methods to solve a quadratic assignment problem whose solution represents matches that respect the underlying spatial configuration among features to some degree. The model we propose deploys the structure of Sparse Clustered Networks, especially the local competition property among groups of neurons inspired by short-range inhibitory interactions in the visual cortex. The matching performance

obtained by the proposed network attains state-of-the-art and provide a useful insight on how neuro-inspired architectures can serve as a substrate for implementing various vision tasks.

Acknowledgments

First and foremost, I would like to thank the European Research Council who supported our work via the European Union's Seventh Framework Program (FP7/2007-2013) / ERC grant agreement n° 290901.

I would like to express my gratitude to my thesis advisor Gilles Coppin and my supervisor Vincent Gripon for their continuous help and useful advises throughout the three years of my PhD thesis.

I would also like to address special thanks to Claude Berrou for his effort in providing the best conditions to our research team at Télécom Bretagne, and to his useful ideas and advices.

I am very thankful for the precious time and all the moments I spent with my team collaborators, and for all the enriching discussions I had the chance to have with them.

I also wish to thank Thomas Serre for giving me the opportunity to spend two months in his laboratory at Brown University in the USA. This was a great experience that enriched my brain and my spirit.

Many thanks to the reviewers, Jean Ponce and Olivier Lemeur, for their effort in reading and commenting my thesis, and to all members of the jury including Fan Yang and Gérard Berry, for expressing interest in our work.

To Mama.. To Baba..

To Fadi.. To Laure.. To Kamal..

To Elodie...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem statement

Designing intelligent machines has been the topic of a large body of research for many years especially in the last decade. As it turned out, solving the human intelligence problem is one of the hardest challenges we have ever encountered. This quest to 'reinvent' our own intelligence is motivated by the pure intellectual curiosity that has always characterized *homo sapiens*, as well as by the fact that it has the potential to profoundly transform human civilisation as we know it. This transformation is expected to affect all aspects of human activity including business, social communication, transportation, scientific research, medicine, finance, and the list goes on.

Remarkable progress has been already achieved in last decades especially in the domain of vision, voice and language understanding. These advances are heralding the emergence of new technology that will bring fundamental changes to our societies such as autonomous vehicles, automatic real-time translation of text and speech, chatting bots, personalized education and e-commerce.

Until the present day, all AI models are designed to accomplish one specific task. A network designed to recognize objects in images or videos cannot adapt itself to learn a new task such as understanding speech or English text. Although its architecture is flexible enough to be able to recognize objects it has never learned, this flexibility is limited to the object recognition task and cannot go beyond it. This is

sometimes called Artificial Narrow Intelligence (ANI) or weak AI as opposed to Artificial General Intelligence (AGI) or Strong AI which is the human-level intelligence. We are arguably still a long way from attaining AGI. A key ingredient for that is to solve the unsupervised learning problem, where an AI can discover patterns and structures within acquired data with little or no intervention from human trainers which is still a limited capability in current unsupervised systems.

However, what is considered as intelligent behaviour might sometimes be different between humans and machines. For example, some complicated tasks that require human intelligence such as performing formal mathematical calculations, can be easily achieved by computers at speeds highly surpassing the human brain limits. Actually, computers have been doing this kind of tasks for many years without requiring what we designate today as AI. On the other hand, tasks that seem to be effortless to us, such as summerizing the plot of a movie or recognizing a face are still hard problems for artificial intelligence research. This constrast was nicely illustrated by computer scientist Donald Knuth (Nilsson, 2009):

> I'm intrigued that AI has by now succeeded in doing essentially everything that requires "thinking" but has failed to do most of what people and animals do "without thinking" - that, somehow, is much harder! I believe the knowledge gained while building AI programs is more important than the use of the programs...

One reason for this contrast in what we consider as intelligent behaviour between humans and machines might be the fundamental structural and architectural differences between the brain and computers that are von Neumann machines. The highly distributed and interconnected structure of the brain might be more favorable to massively parallel tasks such as image and video recognition, while the von-Neumann architecture which based on sequential processing is more adapted to tasks such as formal calculations.

The intricate structure of the brain has been refined and finely tuned throughout millions of years of evolution. The algorithm implemented by neural cells has also

been increasing in complexity, slowly and progressivly, with the increased complexity of the brain structure culminating at human-level intelligence. An important lesson we learned from AI research in the twentieth century is the fact that the apparent simplicity with which we are able to achieve sophisticated tasks such as image and language understanding does not imply the simplicity of the underlying algorithm or neural mechanisms.

The brain is a remarkably complex organ. It is mainly composed of a densely interconnected network of neural cells or neurons. Understanding the nature of neural interactions that are taking place and giving emergence to intelligent behavior remains a major challenge. According to David Marr in his famous work on vision (Marr, 1982b), analysis of complex systems such as the brain should better proceed along three levels. The first one is the *computational* level, in which the problem is described and specified in a generic manner. The final goal of the computation is also identified at this level but no solution should be found. The *algorithmic* level comes next. It is where the input and output representations are specified, and the algorithm of transformation between them is elaborated. The third level of analysis is the *architecture design*, in which a physical architecture should be designed to obtain the same solution described by the algorithm.

In the work we present here, we are more concerned with the third level of analysis. We consider the neural architecture of the brain as our primary source of inspiration. We think that this might be helpful for implementing functions that the brain is known to excel in. We choose to put more focus on studying the visual cortex and processing visual information. The visual system is one of the most explored structures of the brain. Extensive research has been done on this subject and abundant resources are available including computational models and datasets.

We mainly focus on the problem of acquistion and processing of visual information in neural networks. *Acquisition* is the first step in the visual processing pipeline which is typically achieved by the retina. It is where light reflected by an object is captured and transformed into neural signals that the brain is able to manipulate. *Processing* includes mechanisms such as feature extraction as well as organizing pieces

of information into memory. Neurons in early layers of the visual system usually represent simple 'concepts'; they are stimulated by very specific visual patterns such as oriented edges and color contrast. They also span a little fraction of the visual field. As we go further from external stimuli, neurons would represent increasingly complex patterns that can be invoked by stimuli coming from increasingly larger zones of the visual field. In higher cortical layers, as in the frontal lobe, neurons represent abstract concepts independent from any external stimulus. These concepts can be invoked by external stimuli as well as by internal brain activity. Memory is the word used typically to refer to neural assemblies in these layers.

We tackle the visual system on three different stages that we think useful for understanding its function. We start at the lowest stage which is the retina and early layers of the ventral stream where acquisition of the visual signal takes place. We peer into that stage and propose a computational model for visual acquisition that captures some ubiquitous properties of the visual system that were overlooked by most classical visual acquisition models.

After that, we focus on the highest stage which is memory. Rather than proposing a new memory model at this stage, we proposed to improve the performance of retrieval algorithms of an existing neural network model originally designed to function as an associative memory.

We finally study the intermediate stage meant to bridge the gap between the two previous stages: low-level acquired visual information and high-level information stored in memory. A fundamental question in vision research is, what are the operations, and what kind of interactions are happening among low-level information captured in early visual layers that are allowing them to evolve, or to be transformed or mapped to more abstract concepts residing in higher cortical areas in the form of memories. We try to contribute to the understanding of this problem by proposing a neural network model for matching image features or for solving the graph matching problem in a more general sense.

1.2 A brief background

Many computational architectures have been proposed to model the low, intermediate and high stages of processing of visual information mentioned in the previous section. Some of them addressed a single stage while others spanned several ones.

Various computational models for visual acquision have been proposed in recent decades. Some of these models aimed at approximating the complex computations performed by the retina. The *virtual retina* developed by (Wohrer and Kornprobst, 2009) is one such example. It models the function of some ganglion cell (GC) types in the retina by applying a linear convolution on image frames followed by a static non-linearity and a spike generation process. Other models like in (Lorach et al., 2012) aimed at a more faithful emulation of spatio-temporal properties of the retina. It used event-based, asynchronous dynamic vision sensors (DVS) to implement a computational model that mimics biological visual acquision which is asynchronous in nature.

Other neural models of visual acquisition go further than the retina. Some of these models were proposed to accomplish specific tasks such as predicting the eye movement behaviour driven by visual attention. One of the earlist computational models of saliency is that of (Itti et al., 1998). It introduced one of the first implementations of the (Koch and Ullman, 1987) model of bottom-up saliency prediction. It used three separate acquisition channels for intensity contrast, local orientation and color to capture a visual input. This was directly inspired by a similar decomposition procedure that takes place in early layers of the ventral stream thanks to different types of ganglion cells. It also applied some other biological properties such as center-surround antagonism of receptive fields of ganglion cells in their acquisition of the input signal.

Research on designing neuroinspired models of visual attention and saliency prediction has been abundant in recent years. Many were based on the same theoretical model of (Koch and Ullman, 1987) and its implementation in (Itti et al., 1998). In an attempt to provide a unifying base for these models, a framework for visual ac-

quisition, saliency prediction and object recognition was proposed in (Walther and Koch, 2007). It suggested a generic formalism in terms of which most saliency-based attention models can be described. It also proposed a method for smoothly incorporating attention mechanisms, and object recognition models based on the simple and complex cell theory of (Hubel and Wiesel, 1959) such a the Hmax, into a single generic processing pipeline. The visual acquisition framework we propose in chapter 2 is inspired by the one presented in (Walther and Koch, 2007). Similarly, we design a generic scheme to represent the retina and early layers of the visual system. However, we incorporate other properties that allow for a more biologically plausible emulation of the acquisition process.

At the highest stage, memory, much research has also been devoted to investigating architectures for representing information in an associative fashion as biological neural networks do. Many artificial neural networks were proposed as models for human memory known to be associative or content-addressable. This is one of the essential properties sought by all neural models. The Hopfield network (Hopfield, 1982) is one early example. It was suggested as a potential architecture for organizing and interconnecting neurons to memorize fixed size messages. It used the McCulloch-Pitts neuron (McCulloch and Pitts, 1943) and the Hebbian learning rule (Hebb, 1949) to achieve that. Each neuron is connected to all other neurons, and synaptic weights are associated with each connection. These weights are set during the information storing process. Memorization is achieved by optimization of an energy function that could lead to being trapped in local minima, which is a serious problem for Hopfield networks.

The Boltzmann machine introduced in (Ackley et al., 1985) is another landmark associative memory model. It has the same recurrent neural architecture of a Hopfield network in which each neuron is connected to all other neurons. It provides a method for escaping local minima during the optimization procedures by introducing stochastic dynamics in neurons activation inspired by the stochastic nature of biological nerve cells. A known variant of the network is the restricted Boltzmann machine (RBM). It consists of two layers of neurons, a hidden and visible one, with

a bipartite connectivity. This configuration makes learning and storing information more efficient than in the original model. It also allows for stacking several hidden layers in order to learn better representations of the stored data and thus a better retrieval performance.

Autoencoders are also memory models with significant similarities to RBMs (Hinton and Salakhutdinov, 2006; Bengio, 2009). An autoencoder has an input layer, a hidden layer, and an output layer. It is trained to copy or reconstruct its input on its output using a code learned by the hidden layer. This code represents the input by capturing its most useful properties and avoiding redundancy. In addition to being used as models of memory, autoencoders are typically used for dimensionality reduction where input signals are projected into a lower dimensional space. The prefix 'auto-' in the word 'autoencoder' refers to the fact that learning of the code is achieved in an unsupervised fashion, in which the only information needed is the input signal. As in an RBM, hidden units can be stacked to obtain 'deeper' and better representations of the inputs.

More recently Gripon and Berrou proposed a new associative memory architecture (Gripon and Berrou, 2011) as a generalisation of Palm-Willshaw networks (Schwenker et al., 1996). This model was directly inspired by the principle of error correcting codes in information theory and by the origanization of microcolumns in the visual cortex where inhibitory synapses dominate short-range connections, while excitatory synapses prevail on long-range connections. It provides a higher storage capacity than Hopfield or Boltzmann networks. We focus on this model in chapter 3 and propose a new algorithm to enhance its data retrieval performance.

Visual information processing at the intermediate stage that comes after acquisition and before high-level abstract memory is still a hard problem. Actually, this stage comprises many steps, and modest progress has been achieved on exploring the transformation process of raw acquired information into abstract memories. Hubel and Wiesel's influential theory on simple and complex cells in the visual cortex (Hubel and Wiesel, 1959) provided an overture for a better understanding of these transformations. It suggested a hierarchical organization of nerve cells into visual layers where

representation of the visual world gets more complex and abstract as we go up that hierarchy. However, some imporatant questions were not fully answered such as the nature of the exact tranformations performed at each layer of the hierarchy.

Another important question was about the stereo matching problem. This is a fundamental vision phenomenon in which the two images received by both eyes are perceived as a single scene with additional information about depth. In an effort to figure out neuro-compatible solutions for this problem, David Marr and Tomaso Poggio proposed an interesting neural architecture in (Marr and Poggio, 1976) that implemented a cooperative algorithm in which each neuron needs only to interact with a few of its neighbors through excitatory and inhibitory synapses. They applied this method successfully on matching random-dot stereograms. This approach provided a useful insight on how local interactions known to dominate neural activity in the cerebral cortex can be powerful in solving or approximating complex vision problems.

Another fundamental vision problem that we attribute to the intermediate stage between acquisition and memory is the feature correspondence problem. The simplest case of this problem consists in matching each feature extracted from one image called the query image to one feature in a destination image. The goal of this matching procedure is to check whether an instance of an object whose features are extracted from the query image can also be found in the destination image or not. This can be useful for applications such as object tracking, search or unsupervised object category discovery. Inspired by Marr's stereo matching alogrithm, we propose a new approach to solve the feature correspondence problem in chapter 4. In that approach we design a variant of the neural network of (Gripon and Berrou, 2011) and use it to perform the matching procedure. As in the original network, local inhibitory synapses and long excitatory connections are at the heart of the proposed matching procedure.

There have been many attempts to build more complete visual processing pipelines spanning the three modeling stages that start by performing signal acquisition and go as far as incorporating a memory model to perform object recognition. The Hmax originally proposed in (Riesenhuber and Poggio, 1999) and later extended in (Serre et al., 2007) is one example of such attempts. It is a model of object recognition

whose architecture was inspired by Hubel and Wiesel's work (Hubel and Wiesel, 1959). Acquisition of an image signal is first achieved by a set of simple cell layers. Each simple cell responds to the existence of a simple pattern in a limited region of an image such as an oriented edge. A set of complex cell layers follows simple cell layers. Each complex cell applies a max-pooling operator over a few simple cells, in which the maximum response among these cells is taken as that of the complex cell. Simple and complex cell layers then alternate as suggested in (Hubel and Wiesel, 1959) in order to reach a higher level, more abstract description of the image content, which emulates hierarchical feedforward information processing in the ventral stream.

Deep learning with convolutional neural networks (CNNs) (LeCun et al., 1998) is another example of effort seeking object recognition by modeling the whole visual processing pipeline starting from acquisition. As in Hmax, CNNs alternate between simple cell (convolution) layers and complex cell (pooling) layers. However, unlike Hmax, the convolution coefficients in CNNs are not predetermined according to neurophysiological experiments. They are 'learned' during a supervised training process. They allow for building networks with much deeper hierarchies than what is possible with Hmax or other handtuned network. This also allows them to be more adapted to the task they are designed to achieve and thus reach a better performance in object recognition as demonstrated in (Krizhevsky et al., 2012) as well as in various other tasks. However, while CNNs borrow some of their key architectural aspects from that of visual cortex such as convolution an hierarchical processing, the need of strong supervision and a very large number of training examples during the training process suggest that the learning principle it uses is fundamentally different from its biological counterpart.

1.3 Contributions of the thesis

This thesis brings three main contributions to the three stages of the visual processing pipeline discussed above: we introduce a neuro-inspired architecture for visual acquisition and representation, and a neural network model for solving the feature

corresponding problem. We also propose a new algorithm for retrieving information from associative memories. Here is a more detailed list of our main contributions:

- In **chapter 2** we examin the visual acquisition problem and low-level representation. We study the human retina and early layers of the ventral stream, then we propose a neural architecture as a framework for acquiring and representing visual information. This framework differs from other models in the flexibility with which it can be configured to represent visual layers. We will show that this flexibility allows to emulate important properties of the visual system such as retinal sampling, cortical magnification, saccadic eye movements, as well as the visual field which is used to represent the distance parameter between a viewer and the scene. After that, we propose a model of bottom-up saliency estimation based on the proposed visual framework, and demonstrate its performance compared to state-of-the-art models on a standard benchmark.

- **Chapter 3** is about high-level representation of information in memory. More precisely, we study the sparse clustered network (SCN) which is a neural network model used typically as an associative memory (Gripon and Berrou, 2011; Aliabadi et al., 2014). We propose a generic formulation of the different algorithms used to retrieve stored information. We also suggest a new algorithm that gives a better performance at retrieving stored data from partially erased queries.

- In **chapter 4**, we deal with the correspondence problem of image features. We propose a new matching algorithm based on the architecture of the Sparse Clustered Network presented in chapter 3. This matching algorithm harnesses the structural and functional properties of SCNs by using them as matching constraints for the correspondence problem. The performance of this matching model is then evaluated experimentally and compared to state-of-the-art matching models.

Chapter 2

A new framework for visual acquisition

An emerging trend in visual information processing is toward integrating some interesting properties of the ventral stream in order to account for some limitations of machine learning algorithms. Retinal sampling and cortical magnification are two such important features that have been the subject of a large body of research in recent years. In this chapter, we focus on the lowest stage of information processing in the ventral stream. We propose a new framework for visual information acquisition and representation that emulates the architecture of the primate visual system by integrating features such as retinal sampling and cortical magnification while avoiding spatial deformations and other side effects produced by classical models that tried to implement these two features. It also explicitly integrates the notion of visual angle, which is rarely taken into account by vision models. We argue that this framework can provide the infrastructure for implementing vision tasks such as object recognition and computational visual attention algorithms. It also raises important questions about the role of the newly integrated features on vision behavior. Moreover, we propose an algorithm for bottom-up visual attention implemented using the proposed framework, and show that it can attain state-of-the-art performance, and provide a better insight on the significance of studying the role of the visual angle more closely.

2.1 Introduction

Vision and the visual system have been an active area of research for many centuries. Interest in exploring this territory has been motivated by a wide variety of applications. Ophthalmology was one of the first domains to benefit from such discoveries. More recently, that interest has been widely driven by the desire to learn more about the brain and decipher its neural code. A better understanding of the neural code has enabled to design better machine learning algorithms for computer vision, and for artificial intelligence in a more general sense.

The discovery of simple and complex cells in the famous work by Hubel and Wiesel on receptive fields in the cat's visual cortex (Hubel and Wiesel, 1959) marked a new era in vision research. It revolutionized the way the visual system is studied and understood, and allowed for the emergence of 'computational neuroscience', a new field founded by David Marr whose theory on vision is still very influential (Marr, 1982b).

More recently, deep learning networks have achieved an unprecedented performance on many visual tasks such as image categorization (Krizhevsky et al., 2012). The architecture of these networks has been inspired by the multi-layered structure of the visual system and the hierarchical organization of simple and complex cells.

Some criticism of deep learning includes its limited performance on tasks such as unsupervised object discovery and localization, multiple instance recognition (MIL) (Zhu et al., 2015; Ray et al., 2010), recognizing spatial relationships between objects and its limited ability to generalizing to variable-scale representations of the learned classes without increasing the size of the training set (Lake et al., 2015). Another important problem of deep learning, according to (Ranzato et al., 2015), is its computational cost, which renders it impractical for very high resolution images. This called some researchers to get a closer look at the visual system and some of its overlooked properties to address these limitations.

On such property is selective visual attention that guides covert processing biases and saccadic eyes movements. The study of this property is an emerging trend in

visual information processing. It finds its root in Treisman's Feature Integration Theory (FIT) (Treisman and Gelade, 1980). This theory provided a strong evidence of the fundamental role of attention for object recognition. This role was later explored by many researchers including (Koch and Ullman, 1987; Itti et al., 1998; Walther et al., 2004; Bonaiuto and Itti, 2005; Borji et al., 2014). It also motivated the recent emergence of attention-based recognition as in (Larochelle and Hinton, 2010; Zheng et al., 2015).

Cortical magnification is another ubiquitous feature of the visual system (Gattass et al., 1981, 1988). In addition to its role in reducing the amount of visual information entering the brain, Poggio has proposed that it might be a key property for enabling scale-invariant learning of objects (Isik et al., 2011; Anselmi et al., 2015).

In this chapter, we propose a new framework for visual information acquisition that incorporates these important features of the ventral stream. Our contributions are the following:

1. Introducing a new bio-inspired framework for visual information acquisition and representation that offers the following properties:

   - Providing a method for taking the distance between an image and the viewer into account. This is done by incorporating a visual angle parameter which is ignored by most visual acquisition models.
   - Reducing the amount of visual information acquired by introducing a new scheme for emulating retinal sampling and the cortical magnification effects observed in the ventral stream.

2. Providing a concrete application of the proposed framework by using it as a substrate for building a new saliency-based visual attention model, which is shown to attain state-of-the-art performance on the MIT saliency benchmark (Borji et al., 2013a).

The rest of this chapter is organized as follows: In section 2.2, we provide a background of the work we are presenting. Section 2.3 introduces a brief anatomy

of the visual system and its function, paving the way to section 2.4 where a new vision framework is proposed that captures some interesting properties of the visual system. In section 2.5, a new model of visual attention is proposed using the proposed framework. We show in section 2.6 that coupling the proposed vision framework with the proposed attention model gives interesting results that motivates the utility of the framework. A discussion of some of the model's properties is also discussed in section 2.6. Section 2.7 is the chapter conclusion.

2.2 Related work

Computational and mathematical modeling of the visual system have been the focus of many works in literature in recent years. The scope of such models ranges from mathematical models of single neurons (McCulloch and Pitts, 1943) and receptive fields (Rodieck, 1965; Marčelja, 1980) to modeling complete visual layers, especially the retina (Wohrer and Kornprobst, 2009), or even modeling a succession of layers representing early areas of the visual cortex such as the Hmax model (Serre et al., 2007) or the one proposed by David Marr in his famous work on vision (Marr, 1982b).

Most vision models are designed to accomplish a specific task. For instance, the Hmax model is a view-based object recognition processor. It is inspired by the description of simple and complex cells in the primary visual cortex by Hubel and Wiesel (Hubel and Wiesel, 1962). A similar model was proposed in (LeCun et al., 1998), which also provides an implementation of simple and complex cells. However, it uses supervised learning coupled with back-propagation to learn mathematical models of simple cells instead of fixing them beforehand. This allowed for unprecedented performance on many image classification tasks (Krizhevsky et al., 2012).

Some models have more general objectives. The virtual retina model in (Wohrer and Kornprobst, 2009) was proposed as a tool for researchers in neuroscience and neurophysiology to test their ideas and theories about visual function. Similarly, Walther and Koch proposed their model in (Walther and Koch, 2007) as a unified framework for implementing saliency-based visual attention and object recognition

algorithms.

Although vision models are very numerous in literature, some important and even ubiquitous properties of the visual system are still absent in most of them. Retinal sampling and cortical magnification are examples of such properties. Poggio has argued that cortical magnification might play a fundamental role in introducing scale invariance in recognition (Poggio et al., 2014). However, a few models have used foveal-like transformations as an approximation to the cortical magnification effect (Rybak et al., 1998; Isik et al., 2011). While this imitates magnification in the sense that foveal and parafoveal regions are modeled at a higher resolution than the periphery, they differ in that the number of pixels representing the periphery is the same as in the original image, so the number of input pixels is not reduced (see figure 2-1(b)).

At every layer of the visual system, an image zone that falls within the fovea is represented by more neurons than a zone with the same size falling within the periphery. One known method for emulating this is the log-polar representation (Schwartz, 1984). This method emulates retinal sampling very well by using a log-polar grid for sampling pixels of a given image. It then maps sampled pixels onto a rectangular-shaped image that has the drawback of having severe spatial deformations as shown in figure 2-1(c). While this deformed representation has the advantage of being invariant to certain rotation and scale transformations, it is difficult to use such images for subsequent spatial processing used in many models such as the Hmax.

A different retinal sampling method that attempted to avoid log-polar-style deformation was proposed by (Martínez and Robles, 2006). It used sampling points organized in concentric squares to sample an image. These points can then perfectly fit into a square-shaped 2D array like in figure 2-2. While this representation causes less deformation than the log-polar method, it still contains geometrical deformations along its diagonals as shown in figure 2-1(d).

Thus, most known methods for generating retinal images have one of two major drawbacks. The first drawback is constraining the size of the retinal image to be equal to that of the input image, such as the Gaussian blurring method in figure 2-1(b).

(a) Original image.

(b) Blurring.

(c) Log-polar.

(d) Square-sampling.

Figure 2-1: Some of the classical methods traditionally used for emulating retinal sampling and cortical images. Notice that blurring in (b) keeps the same number of pixel as in the original image. The log-polar and the square-sampling methods in (c) and (d) introduce severe spatial deformations that make further spatial filtering more challenging.

This dependency of the output image size on the input is not observed in the visual system where the number of photo-receptors does not depend on the number of image pixels. Moreover, one important property of retinal sampling that such methods do not exploit is the fact that having a constant number of photo-receptors fixes an upper bound on the amount of information allowed to enter the visual system. The second drawback is the deformation introduced by methods that try to avoid the first drawback as in figures 2-1(d) and (c).

We think that the main reason why such methods always have one of the above

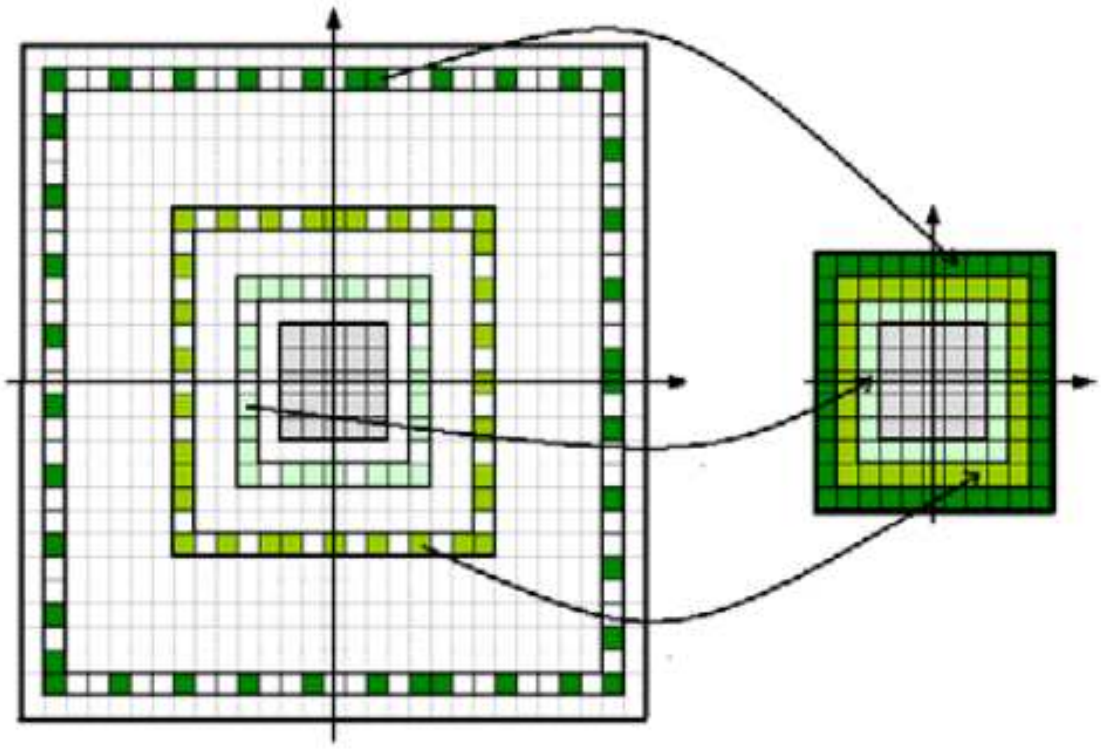


Figure 2-2: Retinal sampling is emulated by using a set of points organized in concentric squares (left). Sampled points can then fit into a smaller square-shaped image (right) that represent the retinal image. (Martínez and Robles, 2006).

drawbacks is that they are constrained to producing an output image with a 'regular' shape. The term 'regular' here means a circular or a rectangular shape. This constraint is set such that the output image is suitable for presentation to a human observer or to be compatible with available image processing tools.

In the vision framework we propose in section 2.4, no such shape constraints are fixed. Hence, we introduce a simple method for applying cortical magnification and retinal sampling in which the output is completely independent from the size of the input image without producing any geometrical deformations.

2.3 The visual system

2.3.1 Structure

The mammalian visual system is a highly intricate structure that exhibits a high level of organization. It starts at the eyes where light is trapped and transduced into neural signals. These signals are then conveyed through the optic nerve to the rest of the nervous system.

The first neural layer that processes visual signals is the retina. Although situated within the eyeball, the retina is considered as an integrate part of the brain. The retina is roughly composed of a three-layered feed-forward structure. The first layer is the Outer Nuclear Layer (ONL) containing photo-receptors. These photo-receptors synapse onto the bipolar cells that are found in the Inner Nuclear Layer (INL). Bipolar cells send their signals to ganglion cells in the Ganglion Cell Layer (GCL). An elaborate network of lateral connections are found between photo-receptors and are mediated by horizontal cells, while amacrine cells mediate lateral connections between bipolar cells (Dowling, 1987).

The optic nerve that is made of ganglion cell axons is the sole output of the retina. Most of the axons of the optic nerve project to the lateral geniculate nucleus (LGN) in the thalamus. An important part of axons in the optic nerve also project to the superior colliculus. The optic nerve is the first stream where visual signals take the form of action potentials. Action potentials conveyed to the LGN by the optic nerve continue their way through the optic radiation which is another axonal structure. The optic radiation projects to the primary visual cortex V1 in the occipital lobe (Hubel and Wiesel, 1962).

The occipital lobe is divided into two distinct layers called V1 and V2. The optic radiation coming from LGN terminates in V1. At this point and starting from V2, the visual stream starts to diverge into to two distinct pathways: the ventral and the dorsal pathways. It has been argued that these pathways act as two independent visual systems with distinct functions (Goodale and Milner, 1992): the ‘What’ and the ‘Where’ systems. The ‘What’ system is situated in the temporal lobe. It is

responsible for visual recognition tasks such as recognizing the identity of faces and other objects. In this system, higher visual areas such as V4, PIT, CIT and AIT are found. The 'Where' system, sometimes called the 'How' system, is found in the parietal lobe. It has been suggested that visually-guided behavior, such as reaching and grasping, is among the main functions of this system. Higher visual areas such as MT, LIP, MST and VIP are parts of this system. However, A later work by (Milner and Goodale, 2008) suggested the existence of a more complex interaction scheme than a simple separation into two independent systems. In addition to the feed-forward pathway of axons, a rich feedback stream also go down throughout all the stages described so far.

2.3.2 Function

Two major families of photo-receptors are found in ONL: rods and cones. Rods are sensitive to low light conditions and are mainly responsible for night vision. On the other hand, cones are less sensitive to light, which makes them more adapted to day vision when light is abundant. Rods' and cones' main function is to transduce incoming photons into neural signals. These signals are further processed by the network of horizontal, bipolar and amacrine cells. They finally arrive at ganglion cells which translate them into action potentials and send them via the optic nerve to other areas.

There are two major types of ganglion cells with distinct functions, Parasol and midget cells. Parasol cells, also called M,Y or β cells, have wider receptive fields (RFs). They are characterized by a lower spatial resolution and a transient response to persistent stimuli. They are associated with achromatic vision. On the other hand, midget cells, which are sometimes called P, X or α cells, have smaller RFs. They have a higher spatial resolution and a lower temporal resolution than parasol cells, and they are associated with color vision. Each type of the above ganglion cells is divided into two sub-types called ON or OFF cells, which have complementary response levels. Hence, we find parasol-ON, parasol-OFF, midget-On and midget-OFF cells (Salin and Bullier, 1995; Hubel and Wiesel, 1959).

Most ganglion cells are known for their center-surround configuration. ON ganglion cells are excited by the onset of light stimuli in their central region and inhibited by light in their surround region. The inverse holds for OFF ganglion cells. Rodieck was the first to propose an elegant mathematical model for spatial and temporal responses of ganglion cells in the form a difference of Gaussians (DoG) (Rodieck, 1965). This center-surround model is also involved in color vision. For example, some midget-ON cells encode the degree of red in their RFs; their center is excited by long wave (red) light and their surround is inhibited by medium wave (green) light. Another type is sensitive to blue, having a center excited by short wave (blue) light and a surround inhibited by medium and long wave light.

Neurons in higher visual areas respond to progressively more complex stimulus patterns. In the primary visual cortex V1, for example, neurons are tuned to simple oriented contours and spatial frequencies. The response of V1 cells, also called simple cells by (Hubel and Wiesel, 1959), are typically modeled mathematically by a Gabor filtering process, which consists in convoluting an image with a Gabor kernel made of the product of a 2D Gaussian kernel by a 2D cosine grating. Some neurons in V2 respond to stimuli as simple as oriented edges, but they are also tuned to illusory edges and a slightly more complex shapes. However, the complexity of spatial and temporal response patterns of neurons grows in complexity in higher visual areas.

2.3.3 Information reduction

The visual field (VF) of a single eye spans about 160° horizontally and 174° vertically. While a tremendous amount of visual information could be extracted from such a wide span, the visual system uses intelligent tricks to reduce the amount of acquired information. This reduction starts as early as the ONL layer in the retina; the visual field is sampled by the photo-receptors in a non-uniform fashion. The density of cones is very high in the central region of the retina called the fovea, spanning about 1°, and decreases logarithmically toward the periphery as shown in figure 2-3. This distribution leads to what is called ‘retinal sampling’ (Salin and Bullier, 1995; Hubel and Wiesel, 1959).

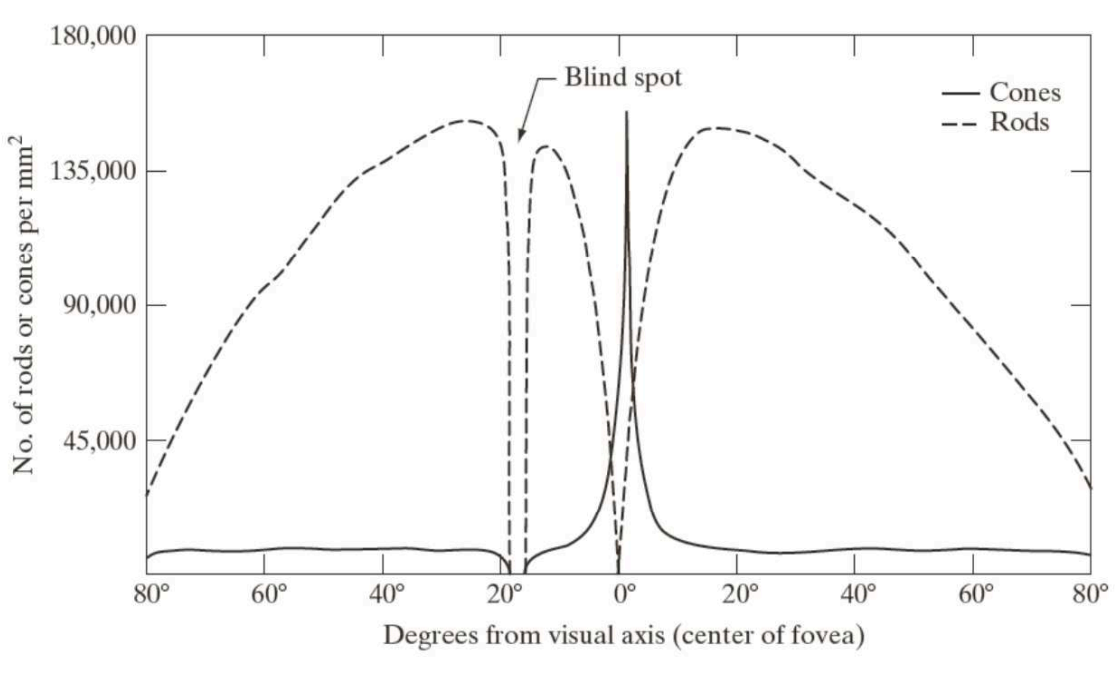


Figure 2-3: Spatial density distribution of rods and cones in the retina (Gonzalez and Woods, 2002).

The reduction of visual information by means of a 'privileged' fovea continues in subsequent areas of the visual system. For example, among ganglion cells of the same type, those which pool their inputs from photo-receptors near the fovea have smaller receptive fields than cells pooling their inputs from photo-receptors in the periphery. This phenomenon is known as the cortical magnification effect. This effect is also observed in LGN, V1, V2, V4 and even in higher visual areas. The ratio between the diameter of a given RF and its eccentricity stays relatively constant in a given visual layer and increases in higher areas as shown in figure 2-4.

2.4 The proposed vision framework

In this section, we propose a model for visual information acquisition and representation in early layers of the visual system. This model is meant to be used as a framework for implementing visual processing tasks such as visual attention modeling presented in section 2.5, or object recognition algorithms that need hierarchical

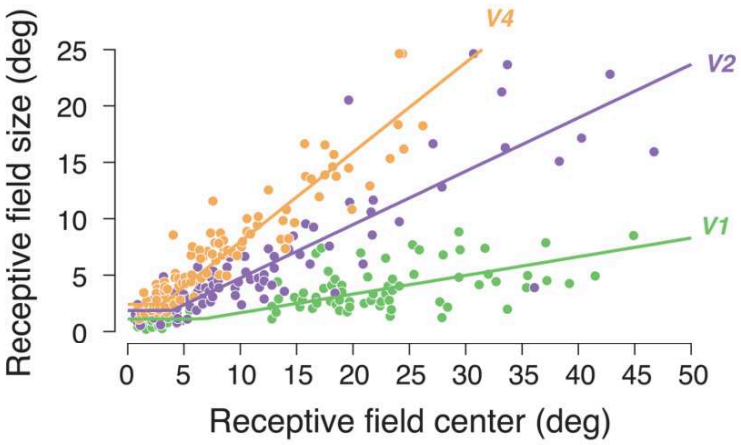


Figure 2-4: Cortical magnification factors in V1, V2 and V4 adapted from (Gattass et al., 1981) and (Gattass et al., 1988) by (Freeman and Simoncelli, 2011).

information processing. This model imitates the information reduction property of the visual system described in section 2.3. It does this by emulating retinal sampling and the cortical magnification effect. This leads to some interesting properties discussed later in section 2.6.

As we have seen in section 2.3, the early visual system can be functionally viewed as an arrangement of consequent layers. It starts at the photo-receptor layer (ONL) in the retina and continues through the GCL layer, the LGN, V1, V2 and so on. The transition from one layer into another can be viewed as a mapping mediated by synaptic connections constituting receptive fields.

The model we propose is made of two basic components: visual layers modeled as point clouds, and mapping functions between these layers in the feed-forward direction.

2.4.1 Notation

In this chapter, polar coordinates and their corresponding Cartesian coordinates are sometimes used interchangeably depending on the context. The radial component of a polar coordinate is always denoted by the roman letter r, while the angle is denoted by the Greek letter ω. The Cartesian version of (r, ω) is always denoted

by (x, y), where $x = r\cos\omega$ and $y = r\sin\omega$. If polar coordinates are written with super- and/or subscripts, these same super- and/or subscripts are attached to their Cartesian versions, and vice versa.

2.4.2 A generic model for visual layers

Visual layers as well as input stimuli are modeled as point clouds using a set representation. This representation can be used to instantiate any number of layers, which is a variable parameter between vision models, by providing a generic description that captures common properties of layers in the visual system as well as input stimuli, such as the visual field spanned by a layer, its fovea size, spatial distribution of cells and the distribution of associated receptive fields.

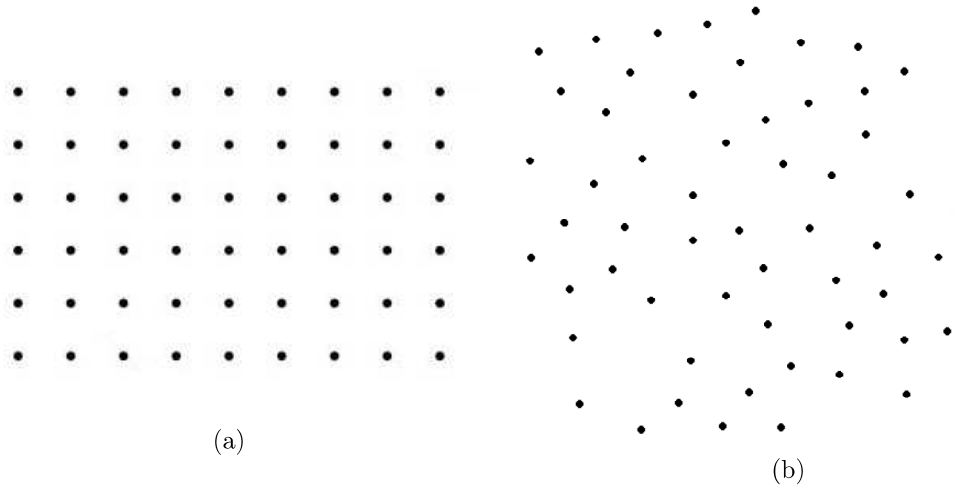
However, this representation focuses on two main properties of the visual system. First, it is adapted to implementing the information reduction properties in the form of retinal sampling and cortical magnification without introducing any deformations. Second, it implements the notion of visual angle which determines the visual field span associated with a given layer. The latter property is one main difference between the vision framework we propose and the one proposed in (Walther and Koch, 2007).

Hence, the structure of a given visual layer can be captured by our model using the following generic definition:

$$\begin{aligned}\mathcal{C}(\Theta^c, \psi^c, \mathcal{D}^c) = \{ f_k^c | f_k^c : \mathbb{R}^2 \rightarrow \mathbb{R}, \\ \sigma(\mathrm{diam}(\mathrm{dom}(f_k^c))) = \Theta^c, \\ k \in \{1, ..., K^c\}\},\end{aligned} \tag{2.1}$$

where $\mathrm{dom}(f_k^c)$ represents the domain of function f_k^c, which is the set of points in $\mathbb{R}^2$ on which f_k^c is defined, e.g., the coordinates of points in figure 2-5, the term $\mathrm{diam}(\mathrm{dom}(f_k^c))$ refers to the diameter of the set $\mathrm{dom}(f_k^c)$, e.g., the diameter of point clouds in figure 2-5, and $\sigma(\mathrm{diam}(\mathrm{dom}(f_k^c)))$ is the the visual angle Θ^c spanned by that diameter. Similarly, ψ^c is the visual angle spanned by the diameter of a central subset

of $\text{dom}(f_k^c)$ called the fovea. The parameter $\mathcal{D}^c$ is used to specify a two dimensional spatial distribution of points in $\text{dom}(f_k^c)$.



(a) (b)

Figure 2-5: Two example point clouds representing visual layers according to the definition in (2.1). A different distribution $\mathcal{D}^c$ is used for each cloud. In (a), the distribution $\mathcal{D}^c$ is chosen as a regular grid. This distribution is more adapted to representing images with a classical rectangular shape. In (b), this distribution is chosen at random. This shows that the representation of visual layers in the proposed framework is not limited to rectangular distributions as in most vision models.

As an illustrative example, the definition in (2.1) can be used to represent a classical two dimensional RGB image. In this case, the distribution $\mathcal{D}^c$ is chosen as a 2D rectangular grid corresponding to pixel positions of the image as in figure 2-5 (a), k refers to an image component (R, G or B) and f_k^c is the value of the component k at an index (i, j) in $\mathbb{N}^2$.

An interesting feature of using $\mathcal{C}$ to represent an image is that it associates a visual angle Θ^c with its diagonal. This emulates the fact that, in reality, an image is always associated with a certain visual angle when viewed from a certain distance. We argue that this is an important element for any model that aims at a faithful modeling of the visual systems. It allows to study the influence of the visual angle on the behavior of models performing visual tasks.

The cone receptors layer in the ONL can also be modeled using the definition in (2.1). In this case, $\mathcal{D}^c$ would be chosen to approximate cone distribution in the retina shown in figure 2-3. This means that the density of points in $\text{dom}(f_k^c)$ would

be higher in the foveal region defined by ψ^c, and decrease logarithmically towards the periphery. Each point f_k^c would represent a cone receptor whose type would be determined by the subscript k (a S, M or L cone). The angles Θ^c and ψ^c would represent the layer's visual field and the width of the fovea in degrees of visual angles, respectively.

In a much similar way to representing a cone-receptors layer, other visual layers such as the ganglion cell layer in the retina, LGN, V1 and higher layers can be modeled by (2.1) as we will see in section 2.5.

An optional modulation function m^c can be applied to a layer $\mathcal{C}$:

$$m^c : \mathbb{R}^{|\mathcal{C}|} \rightarrow \mathbb{R}^{|\mathcal{C}|}. \tag{2.2}$$

This function can be used to implement any operation that globally modifies values of points in a given layer. Example operations include non-linearities such as contrast gain control and intensity adaptation as in the retina, the Inhibition of Return (IOR) operation used in most models of visual attention, or any other operation.

2.4.3 Stacking layers

In the same way as a ganglion cell pools over a number of photo-receptors (mediated by bipolar cells), or a neuron in V1 pools over a number of axons seen by its RF in LGN to produce their output, layers of type $\mathcal{C}$ can be stacked to emulate the feed-forward path of the visual system. In this case, each point in a $\mathcal{C}$-type layer gets its value by pooling over a set of points belonging to the previous layer. More precisely, given two layers $\mathcal{C}_1$ and $\mathcal{C}_2$, a point $f_k^{c_2}(x_o, y_o) \in \mathcal{C}_2$ can be associated with a set of coordinates called its receptive field $\mathrm{RF}^{c_1 c_2}(f_k^{c_2}) \subseteq \mathrm{dom}(f_k^{c_1})$, where $f_k^{c_1} \in \mathcal{C}_1$:

$$\begin{aligned}\mathrm{RF}_k^{c_1c_2}(f_k^{c_2}(x_o,y_o)) = \{(x,y)|(x,y) \in \mathrm{dom}(f_{k'}^{c_1}),\\ (x_o,y_o) \in \mathrm{dom}(f_k^{c_2}),\\ \text{and } (x,y) \text{ satisfies some condition}\\ \text{guaranteeing its membership to the}\\ \text{receptive field of } f_k^{c_2}(x_o,y_o)\}.\end{aligned} \tag{2.3}$$

Determining whether a coordinate (x, y) is in the receptive field of a point $f_k^{c_2}(x_o, y_o)$ depends on the types of $\mathcal{C}_1$ and $\mathcal{C}_2$. For example, if both $\mathcal{C}_1$ and $\mathcal{C}_2$ model cortical layers, then a typical way of determining the RF membership is by looking whether (x, y) falls within a disk-shaped region around (x_o, y_o), given that (x, y) and (x_o, y_o) belong to the same space. When $\mathcal{C}_1$ is used to model a RGB image, and $\mathcal{C}_2$ models a cone-receptor layer, the process becomes similar to retinal sampling where a point in the receptors layer gets its value by sampling only one pixel in the image. In this case, determining the receptive field of $f_k^{c_2}(x_o, y_o)$ consists in finding its corresponding point in $\mathcal{C}_1$.

The input signal to the point $f_k^{c_2}(x_o, y_o)$ can be defined as follows:

$$\begin{aligned}s_k^{c_1c_2}(f_k^{c_2}(x_o,y_o)) = \{f_{k'}^{c_1}(x,y)|\\ (x,y) \in \mathrm{RF}^{c_1c_2}(f_k^{c_2}(x_o,y_o))\},\end{aligned} \tag{2.4}$$

and the value of $f_k^{c_2}(x_o, y_o)$ can be finally computed as:

$$f_k^{c_2}(x_o,y_o) = \phi_k^{c_1c_2}(s_k^{c_1c_2}(f_k^{c_2}(x_o,y_o))), \tag{2.5}$$

where $\phi^{c_1c_2}$ is a mapping defined as:

$$\phi_k^{c_1 c_2} : \mathbb{R}^{|s_k^{c_1 c_2}(f_k^{c_2}(x_o, y_o))|} \rightarrow \mathbb{R}. \tag{2.6}$$

This mapping can be linear as in the case of Gabor or DoG kernels. It can also be used to implement non-linearities for pooling functions.

In the next section, we propose a model for saliency-based visual attention that implements the proposed vision framework. This will shed the light on the framework's interesting properties and raises some insightful questions about their role in visual processing in section 2.6.

2.5 Application: modeling bottom-up visual attention

Many models have been proposed in litterature for modeling visual attention in recent years. This emerging field has been the subject of a large body of research in neuroscience as well as in computer vision. It has been useful in many applications including object recognition and video compression (Borji and Itti, 2013; Walther et al., 2004), object segmentation (Tu et al., 2016) and detection (Pan et al., 2016; Gao et al., 2015).

The Feature Integration Theory (FIT) introduced in (Treisman and Gelade, 1980) was probably the first to suggest a fundamental functional role for attention in visual recognition. A few years later, Koch & Ullman proposed a possible neural mechanism for driving attention (Koch and Ullman, 1987). This mechanism only considered low-level image features in which only color, intensity contrast and local intensity orientations are used to drive the focus of attention. The first working implementation of this mechanism was proposed by Itti and Koch in (Itti et al., 1998), and became a landmark for many saliency prediction models based on bottom-up visual attention (Le Meur et al., 2006; Navalpakkam and Itti, 2006; Murray et al., 2011).

The term bottom-up comes from the fact that only basic information about the

image signal such as color and intensity are involved in predicting saliency. Other models have attempted to enforce bottom-up biases with higher level information about the scene such as recognition of objects or proto-objects (Judd et al., 2009; Zhao et al., 2014), scene context and gist information (Goferman et al., 2012; Torralba et al., 2006), and by using fully convolutional neural networks (Kruthiventi et al., 2015), or statistics about oculomotor biases in human subjects as in (Le Meur and Liu, 2015) more recently.

The algorithm we propose here is based on the model of Itti and Koch. However, it shortcuts the first two steps consisting in Gaussian sub-sampling and across-scale subtraction. These steps are replaced by a filtering operation using kernels with eccentricity-dependent receptive fields emulating the cortical magnification effect. This allows us to reduce the number of feature maps to 9 maps instead of 42 maps in the original model.

The model we propose holds some similarity to the one the authors introduced in (Aboudib et al., 2015) with several major differences:

- The proposed model is implemented using the vision framework proposed in section 2.4.
- The proposed vision framework allows for a more plausible way for emulating retinal sampling and cortical magnification factors.
- Normalized feature maps are directly combined to form the final saliency map without computing conspicuity maps.

Figure 2-6 depicts the basic architecture of the attention algorithm based on the proposed vision framework.

2.5.1 The image layer $\mathcal{I}$

The attention model we propose consists of four $\mathcal{C}$-type layers called $\mathcal{I}$, $\mathcal{P}$, $\mathcal{U}$ and $\mathcal{L}$ defined according to (2.1). The first layer $\mathcal{I}$ represents an RGB image and is defined

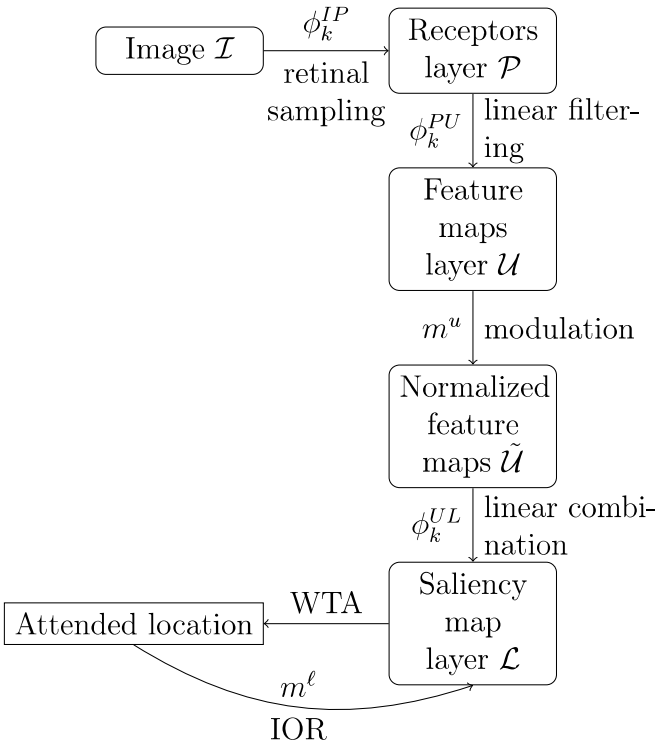


Figure 2-6: The basic architecture of the proposed saliency model.

as follows:

$$\begin{aligned}\mathcal{I}(\Theta^I, \psi^I, \mathcal{D}^I) = \{f_k^I | f_k^I : \mathbb{N}^2 \rightarrow [0,1], \\ \sigma(\text{diam}(\text{dom}(f_k^I))) = \Theta^I, \\ k \in \{1, 2, K^I = 3\}\},\end{aligned} \tag{2.7}$$

where $\text{dom}(f_k^I)$ is the set of all pixel indexes (i, j) in the image. A point f_k^I represents the value of the k component of the RGB image $\mathcal{I}$ at a given index in $\mathbb{N}^2$, where $k = 1$ stands for the R component, $k = 2$ for G and $k = 3$ for the blue component B.

2.5.2 The receptors layer $\mathcal{P}$

The second layer $\mathcal{P}$ is the receptors layer that samples the input image in the same way the ONL layer in the retina samples the visual scene, it can be similarly defined as:

$$\mathcal{P}(\Theta^p, \psi^p, \mathcal{D}^p) = \{f_k^p | f_k^p : \mathbb{R}^2 \to [0,1], \\ \sigma(\text{diam}(\text{dom}(f_k^p))) = \Theta^p, \\ k \in \{1, 2, K^p = 3\}\}, \tag{2.8}$$

where the distribution $\mathcal{D}^p$ is chosen to approximate the cone distribution in the primate retina as in figure 2-7. A point f_k^p represents a cone receptor of type L or red ($k = 1$), M or green ($k = 2$), S or blue ($k = 3$).

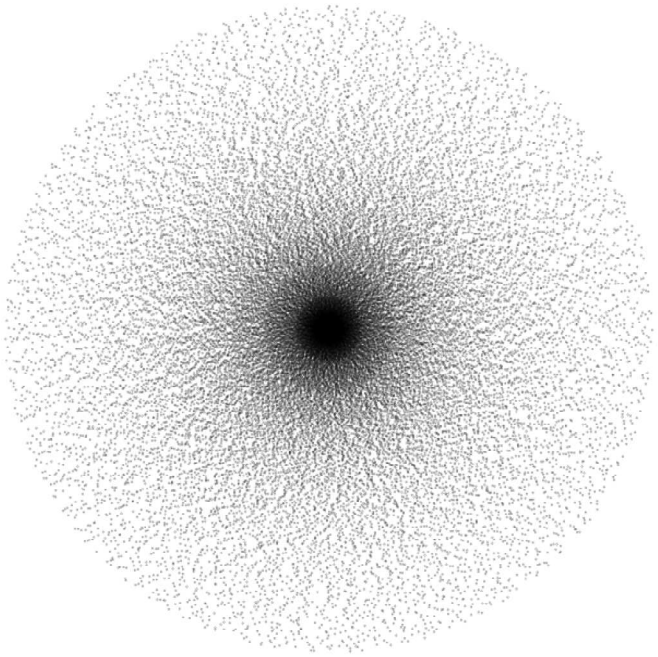


Figure 2-7: The distribution $\mathcal{D}^p$ used for the receptor layer $\mathcal{P}$. This distribution is inspired by the distribution of cone receptors in the retina, where the density is higher in the central fovea and decreases rapidly toward the periphery. In this figure, the span of the diameter of layer $\mathcal{P}$ is $\sigma(\text{diam}(\text{dom}(f_k^p))) = \Theta^p = 10^\circ$ and the span of the fovea diameter $\psi^p = 1^\circ$. The total number of points in this figure is 41284 of which 10000 are within the fovea.

Point coordinates (r, w) in $\text{dom}(f_k^p)$ are expressed in degrees, where r is the eccentricity relative to the center of the fovea measured in degrees of visual angles, which is a typical way of referring to cell positions in the retina. The coordinate w is the angle made between the horizontal line passing through the fovea's center and the

line between the fovea's center and (r, w). Parameters ψ^p and Θ^p are also expressed in visual angles. They refer to the diameter span of the fovea and the overall visual field of $\mathcal{P}$, respectively. Figure 2-7 depicts an example distribution of points in $\text{dom}(f_k^p)$. Notice that points are very dense toward the center where the fovea is found and get sparser toward the periphery.

In order to compute the value of a point $f_k^p \in \mathcal{P}$, a mapping ϕ_k^{Ip} is applied. This mapping can be viewed as a retinal sampling operation where each point in $\mathcal{P}$ is used to sample only one pixel of the image $\mathcal{I}$ at the corresponding location. Hence, given the distribution $\mathcal{D}^p$, the image is sampled at the highest resolution in the fovea, and at progressively lower resolutions toward the periphery.

We start by determining the set $\text{RF}_k^{Ip}(f_k^p(r_o, \omega_o))$ as:

$$\begin{aligned} \text{RF}_k^{Ip}(f_k^p(r_o, \omega_o)) = \{(i,j) | (i,j) \in \text{dom}(f_{k'}^I), \\ (r_o, \omega_o) \in \text{dom}(f_k^p), \\ \text{and } (i,j) = \text{proj}^{pI}(r_o, \omega_o)\}, \end{aligned} \tag{2.9}$$

where proj^{pI} is a mapping that associates with each coordinate (r_o, ω_o) in $\text{dom}(f_k^p)$ an index (i, j) in $\text{dom}(f_k^I)$:

$$\text{proj}^{pI} : \mathbb{R}^2 \rightarrow \mathbb{N}^2. \tag{2.10}$$

Since the radial coordinate r_o is expressed in degrees of visual angles, a natural graphical representation of layer $\mathcal{P}$ would be a spherical surface as in figure 2-8. This is close to the real shape of the primate retina, which is often modeled by a spherical surface centered around the nodal point of the eyeball. This spherical representation is used for all subsequent layers of the proposed model: $\mathcal{U}$ and $\mathcal{L}$. Given this representation, the mapping proj^{pI} can be determined from figure 2-8 as follows:

$$(r, \omega) = (d \tan(r_o), \omega_o),$$
$$(i, j) = \text{proj}(r_o, \omega_o) = ([x], [y]), \tag{2.11}$$

where [.] denotes a rounding operation to the closest integer value, and d emulates the distance between the photo plane and the nodal point of the eyeball as shown in figure 2-8:

$$d = \frac{\text{diam}(\text{dom}(f_k^I))}{\tan \Theta^I}. \tag{2.12}$$

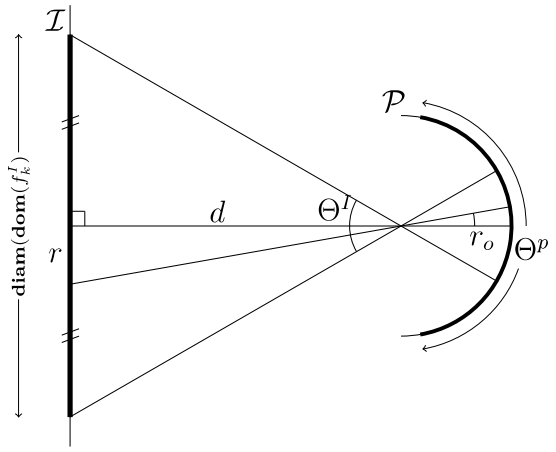
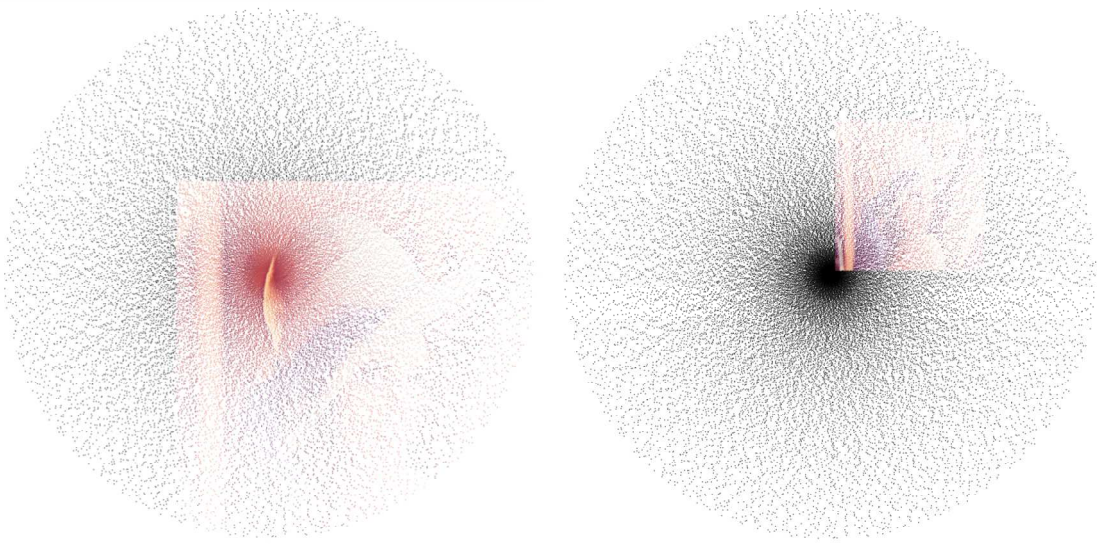


Figure 2-8: The projection of the image layer $\mathcal{I}$ onto the receptor layer $\mathcal{P}$ modeled by a hemisphere. The figure is a 2-dimensional cross section plane that passes through the diagonal of the image $\mathcal{I}$ and through the center of the sphere. The point of the image whose radial coordinate is r falls onto the image diagonal.

Notice from (2.11) that setting the value of ω to ω_o ignores the fact the the image is inverted on the surface of layer $\mathcal{P}$ as shown is figure 2-8. A more faithful way would be to set ω to $\omega_o + \pi$. However, this inversion can be safely ignored since it has no significance on the visual processing task in question.

Also notice that in our experiments, we always consider that the center of the fovea is fixated at the image's center as in figure 2-10. However, this model offers the

possibility to fixate the fovea at any arbitrary point of the image or even outside its borders as shown in figure 2-9, which is a useful property for designing models that needs to emulate saccadic eye movements.



(a) (b)

Figure 2-9: The acquisition of the signal of layer $\mathcal{P}$ is totally independent of the size, position and the resolution of the image $\mathcal{I}$. In (a), the visual angle of the image is set to $\Theta^I = 10°$, but the fovea falls onto the upper-left corner of the image so that a part of the image falls outside of the visual field of layer $\mathcal{P}$. In (b), the fovea center falls outside of the image borders, the value of Θ^I is set to $4°$.

The input signal to the point $f_k^p(r_o, \omega_o)$ is then defined as:

$$\begin{aligned} s_k^{Ip}(f_k^p(r_o,\omega_o)) = \{ & f_{k'}^I(i,j) | \\ & (i,j) \in \mathrm{RF}_k^{Ip}(f_k^p(r_o,\omega_o)), \\ & k = k', \\ & \text{and } f_{k'}^I(i,j) \in \mathcal{I}\}, \end{aligned} \tag{2.13}$$

and finally, sampling is applied by computing the value of each point $f_k^p(r_o, \omega_o)$ as follows:

$$f_k^p(r_o, \omega_o) = \phi_k^{Ip}(s_k^{Ip}(f_k^p(r_o, \omega_o))), \tag{2.14}$$

where ϕ^{Ip} is a mapping defined on a given set A as follows:

$$\forall A, \phi_k^{Ip}(A) = \begin{cases} A & if A \neq \phi. \\ 0 & \text{Otherwise,} \end{cases} \tag{2.15}$$

where ϕ is the empty set. Notice that the multi-part definition in (2.15) accounts for the fact that when $\text{proj}^{pI}(r_o, \omega_o)$ falls outside the image borders, the sampled value is considered as a zero. This is equivalent to considering that the image $\mathcal{I}$ is embedded into a black background. Figure 2-10 is an example of a retinal representation in $\mathcal{P}$ of an image $\mathcal{I}$ after applying (2.15).

Figure 2-10: The retinal image $\mathcal{P}$ after sampling image $\mathcal{I}$. Notice that the image is sampled at a higher resolution at the foveal center, and that the resolution decreases toward the periphery. No spatial deformations are introduced, and the number of sampled points depends only on the number of points in $\mathcal{P}$ not on the number of pixels in $\mathcal{I}$.

2.5.3 The feature map layer $\mathcal{U}$

The next layer, is the feature map layer $\mathcal{U}$. This layer is composed of 9 feature maps representing intensity contrast, color opponency and local orientation selectivity, which are the basic three feature dimensions originally used in (Itti et al., 1998):

$$\begin{aligned}\mathcal{U}(\Theta^u, \psi^u, \mathcal{D}^u) = \{f_k^u | f_k^u : \mathbb{R}^2 \rightarrow \mathbb{R}, \\ \sigma(\text{diam}(\text{dom}(f_k^u))) = \Theta^u, \\ k \in \{1, ..., K^u = 9\}\}. \end{aligned} \tag{2.16}$$

All points in $\mathcal{U}$ that have the same value for k form a single feature map. The 9 feature maps emerging from the above definition, $\{f_{k=1}^u\}$, $\{f_{k=2}^u\}$, $\{f_{k=3}^u\}$,$\{f_{k=4}^u\}$, $\{f_{k=5}^u\}$, $\{f_{k=6}^u\}$, $\{f_{k=7}^u\}$, $\{f_{k=8}^u\}$ and $\{f_{k=9}^u\}$, are chosen to represent intensity contrast, local orientations for 0°, 45°, 90°, 135° and color opponency for red-green, green-red, blue-yellow, yellow-blue, respectively. The distribution $\mathcal{D}^u$ is chosen to be a circular grid as shown in figure 2-12. Notice that as in $\mathcal{P}$, the density of points is higher in the fovea and decreases towards the periphery. Also notice that point coordinates in $\text{dom}(f_k^u)$ are expressed in the same units as coordinates in $\text{dom}(f_k^p)$, and they belong to the same space.

Each point f_k^u in $\mathcal{U}$ has its own receptive field in the receptor layer $\mathcal{P}$ spanning a set of coordinates in $\text{dom}(f_k^p)$. Each such RF is defined as follows:

$$\begin{aligned}\text{RF}_k^{pu}(f_k^u(r_o, \omega_o)) = \{(r, \omega) | (r, \omega) \in \text{dom}(f_{k'}^p), \\ (r_o, \omega_o) \in \text{dom}(f_k^u), \\ \text{and } \|(x, y), (x_o, y_o)\| \leq \rho(r_o, \omega_o)\}, \end{aligned} \tag{2.17}$$

where $\|.,.\|$ is the euclidean distance operator, $\rho(r_o, \omega_o)$ is the eccentricity-dependent radius of a circle centered at (r_o, ω_o), and is given by:

$$\rho(r_o, \omega_o) = \begin{cases} \alpha r_o & \text{if } r_o \geq \frac{\psi^u}{2}. \\ \alpha \frac{\psi^u}{2} & \text{otherwise,} \end{cases} \tag{2.18}$$

where α is the slope associated with the cortical magnification factor (CMF). Notice that (2.18) reflects the fact that receptive fields of cells within the fovea of a given layer tend to have roughly equal radii. However, these radii begin to increase linearly at the extremities of the fovea toward the periphery, which is behind the cortical magnification effect observed in the primate visual system (Gattass et al., 1981, 1988; Isik et al., 2011).

Notice that a radius $\rho(r_o, \omega_o)$ is measured in degrees of visual angles. Thus, a more precise way to compute the distance between (x, y) and (x_o, y_o) in (2.17) is to use the great circle distance according to a spherical geometry defined on layer $\mathcal{U}$. However, the spherical surface model of layers $\mathcal{P}$, $\mathcal{U}$ and $\mathcal{L}$ is supposed to be locally plane for simplicity, which allows for computing distances as being locally euclidean.

It is worth pointing out that the distribution $\mathcal{D}^u$ can only be determined if the number, sizes and positions of all receptive fields RF_k^{pu} are known. In other words, this distribution is chosen so that a certain overlap is respected between these RFs; the overlap along the radial line p_r, and the overlap p_c between RFs on the same circle. Figure 2-11 shows and example configuration of receptive fields $\mathrm{RF}_k^{pu}(f_k^u)$ with overlaps $p_r = p_c = 0.5$, and figure 2-12 depicts its corresponding distribution $\mathcal{D}^u$.

The input signals to points belonging to feature maps for intensity contrast and local orientations are given by:

$$\begin{aligned} s_k^{pu}(f_k^u(r_o, \omega_o))_{k\in\{1,\ldots,5\}} = \{ & f_{k'}^p(r, \omega) | \\ & (r, \omega) \in \mathrm{RF}_k^{pu}(f_k^u(r_o, \omega_o)), \\ & k' \in \{1, 2, K^p = 3\}, \\ & \text{and } f_{k'}^p(r, \omega) \in \mathcal{P}\}. \end{aligned} \tag{2.19}$$

Input signals to points within the feature map for red-green opponency are defined

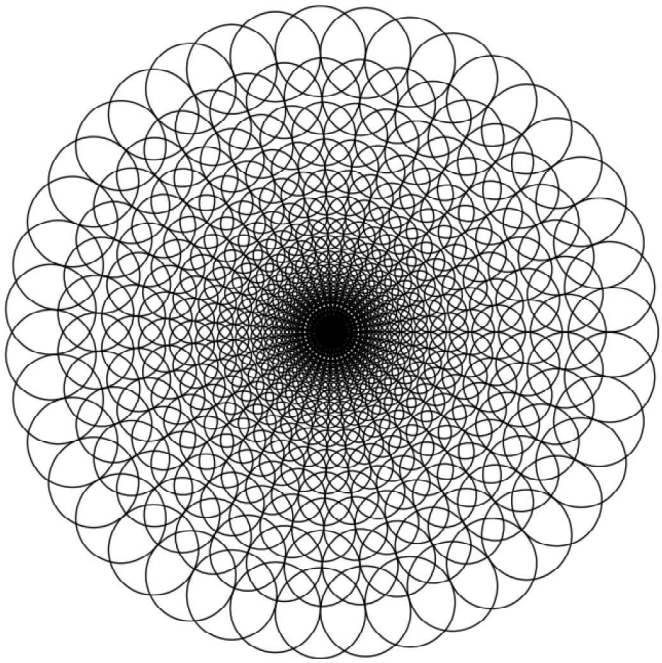


Figure 2-11: Receptive fields RF_k^{pu} associated with points in layer $\mathcal{U}$. Notice that these RFs are smaller and more dense in the fovea and grow bigger with decreasing density toward the periphery emulating the cortical magnification factor. This configuration corresponds to circular and radial overlap values $p_c = p_r = 0.5$, a visual angle span of $\Theta^u = 10^\circ$ and a slope $\alpha = 0.16$ for the cortical magnification factor.

as:

$$\begin{aligned} s_k^{pu}(f_k^u(r_o,\omega_o))_{k=6} = \{ & f_{k'}^p(r,\omega) | \\ & (r,\omega) \in \mathrm{RF}_k^{pu}(f_k^u(r_o,\omega_o)), \\ & (k'=1 \wedge \|(x,y),(x_o,y_o)\| \leq (\delta_c/2) \vee \\ & (k'=2 \wedge \|(x,y),(x_o,y_o)\| > (\delta_c/2)), \\ & \text{and } f_{k'}^p(r,\omega) \in \mathcal{P}\}, \end{aligned} \tag{2.20}$$

where δ_c is the diameter of the central zone of $\mathrm{RF}_k^{pu}(f_k^u(r'_o,\omega'_o))$ that has a center-surround configuration. We notice from (2.20) that red-green opponency is applied in the same way as in chromatic ganglion cells that get their input signals from L (red) cones in the central zone of their receptive fields, and from M (green) cones in

the surround.

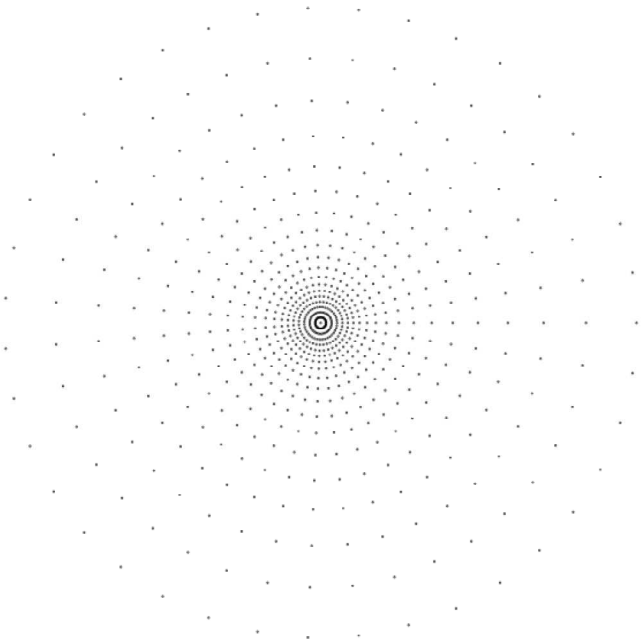


Figure 2-12: An example distribution $\mathcal{D}^u$ for points in layer $\mathcal{U}$ corresponding to circular and radial overlap values $p_c = p_r = 0.5$ between the receptive fields RF_k^{pu} associated with each point. This value is chosen for the clarity of display. A value of 0.8 is used for the experiments. The visual angle span of the layer's diameter is $\sigma(\mathrm{diam}(\mathrm{dom}(f_k^u))) = \Theta^u = 10°$.

Similarly, input signals for green-red, blue-yellow, yellow-blue feature maps are defined respectively as follows:

$$
\begin{aligned}
s_k^{pu}(f_k^u(r_o,\omega_o))_{k=7} = \{ & f_{k'}^p(r,\omega) | \\
& (r,\omega) \in \mathrm{RF}_k^{pu}(f_k^u(r_o,\omega_o)), \\
& (k' = 2 \wedge \|(x,y),(x_o,y_o)\| \leq (\delta_c/2)) \vee \\
& (k' = 1 \wedge \|(x,y),(x_o,y_o)\| > (\delta_c/2)), \\
& \text{and } f_{k'}^p(r,\omega) \in \mathcal{P}\},
\end{aligned}
\tag{2.21}
$$

$$
\begin{aligned}
s_k^{pu}(f_k^u(r_o,\omega_o))_{k=8} = \{f_{k'}^p(r,\omega)| \\
&(r,\omega) \in \mathrm{RF}_k^{pu}(f_k^u(r_o,\omega_o)), \\
&(k' = 3 \wedge \|(x,y),(x_o,y_o)\| \leq (\delta_c/2)) \vee \\
&(k' \in \{1,2\} \wedge \|(x,y),(x_o,y_o)\| > (\delta_c/2)), \\
&\text{and } f_{k'}^p(r,\omega) \in \mathcal{P}\},
\end{aligned} \tag{2.22}
$$

$$
\begin{aligned}
s_k^{pu}(f_k^u(r_o,\omega_o))_{k=9} = \{f_{k'}^p(r,\omega)| \\
&(r,\omega) \in \mathrm{RF}_k^{pu}(f_k^u(r_o,\omega_o)), \\
&(k' \in \{1,2\} \wedge \|(x,y),(x_o,y_o)\| \leq (\delta_c/2)) \vee \\
&(k' = 3 \wedge \|(x,y),(x_o,y_o)\| > (\delta_c/2)), \\
&\text{and } f_{k'}^p(r,\omega) \in \mathcal{P}\}.
\end{aligned} \tag{2.23}
$$

The value of each point f_k^u in the feature maps is then computed by applying a linear mapping ϕ_k^{pu}.

$$
f_k^u(r_o,\omega_o) = \phi_k^{pu}(s_k^{pu}(f_k^u(r_o,\omega_o))). \tag{2.24}
$$

This mapping consists in applying a DoG kernel on each input signal for the intensity contrast and color opponency feature maps, and a Gabor (GB) kernel in feature maps for local orientations. DoG kernels are classically used to model the center-surround configuration of RFs of parasol and midget ganglion cells involved in chromatic and achromatic vision, while GB kernels are typically used to model orientation selective responses of neurons in V1, as mentioned in section 2.3.

The DoG model proposed by Rodieck in (Rodieck, 1965) is used to compute the kernel coefficients associated with a point at a coordinate (r,ω):

$$\begin{aligned}\mathrm{DoG}(r_o, \omega_o, r, \omega) =& g_1 \frac{\pi}{\delta_1^2} . \exp(-\frac{\|(x,y),(x_o,y_o)\|^2}{\delta_1^2}) - \\ & g_2 \frac{\pi}{\delta_2^2} . \exp(-\frac{\|(x,y),(x_o,y_o)\|^2}{\delta_2^2}),\end{aligned} \tag{2.25}$$

where (r_o, ω_o) is the RF center to which (r, ω) belongs, δ_1 and δ_2 are the standard deviations of the central and the surround Gaussians of DoG kernels, g_1 and g_2 are two constants used to control the relative strengths of the two Gaussians.

Coefficients of Gabor kernels (Gabor, 1946) are similarly defined as follows:

$$\mathrm{GB}(r_o, \omega_o, r, \omega) = \exp(-\frac{X^2 + Y^2\gamma^2}{2\delta_3}) . \cos(\frac{2\pi}{\lambda} X), \text{ s.t.} \tag{2.26}$$

$$X = (x - x_o)\cos\theta + (y - y_o)\sin\theta \text{ and}$$

$$Y = -(x - x_o)\sin\theta + (y - y_o)\cos\theta. \tag{2.27}$$

Figure 2-13 depicts some examples of DoG and GB kernels we used. The mapping ϕ_k^{pu} is finally applied as the sum of elements of an input signal weighted by their corresponding kernel coefficients:

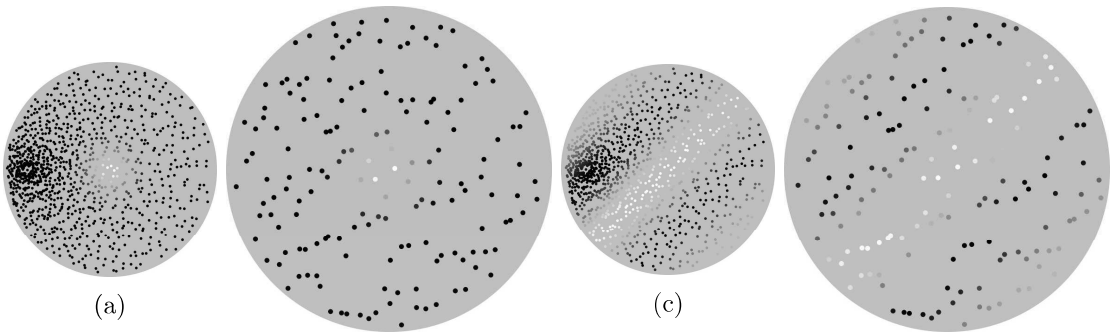


Figure 2-13: Some examples of the Difference of Gaussians (DoG) and Gabor (GB) kernels used for the mapping ϕ_k^{pu}. Notice that kernels whose RFs are closer to the fovea in (a) and (c) are smaller in size in and defined on more points that RFs in the periphery, (b) and (d), which is due to the cortical magnification factor. This difference in size and density is inspired by biological reality in the retina. The gray background and the size of points in these figures is adjusted for the clarity of display.

$$\phi_k^{pu}(s_k^{pu}(f_k^u(r_o,\omega_o)))_{k\in\{1,6,7,8,9\}} = \\ = \sum_{\substack{f^p(r,\omega)\subseteq \\ s_k^{pu}(f_k^u(r_o,\omega_o))}} \text{mean}(f^p(r,\omega)).\text{DoG}(r_o,\omega_o,r,\omega), \tag{2.28}$$

$$\phi_k^{pu}(s_k^{pu}(f_k^u(r_o,\omega_o)))_{k\in\{2,3,4,5\}} = \\ = \sum_{\substack{f^p(r,\omega)\subseteq \\ s_k^{pu}(f_k^u(r_o,\omega_o))}} \text{mean}(f^p(r,\omega)).\text{GB}(r_o,\omega_o,r,\omega), \tag{2.29}$$

where $f^p(r,\omega)$ here is the set of all points $f_{k'}^p(r,\omega)$ in $s_k^{pu}(f_k^u(r_o,\omega_o))$ defined on the same coordinate (r,ω). Figure 2-14 is an example of some feature maps we obtain by applying the above mappings.

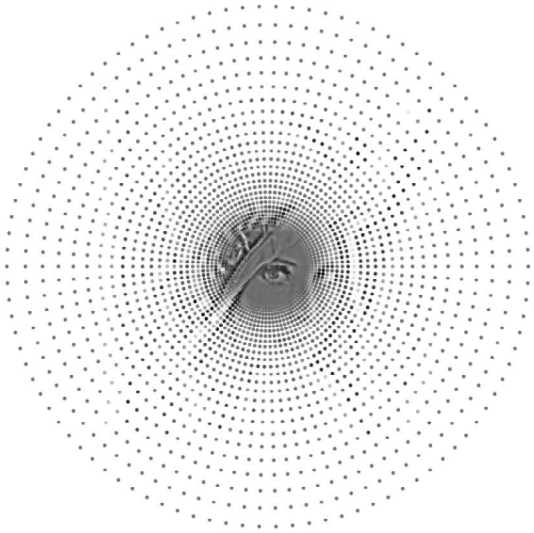
2.5.4 The saliency map layer $\mathcal{L}$

Finally, layer $\mathcal{L}$ is used to compute the saliency map:

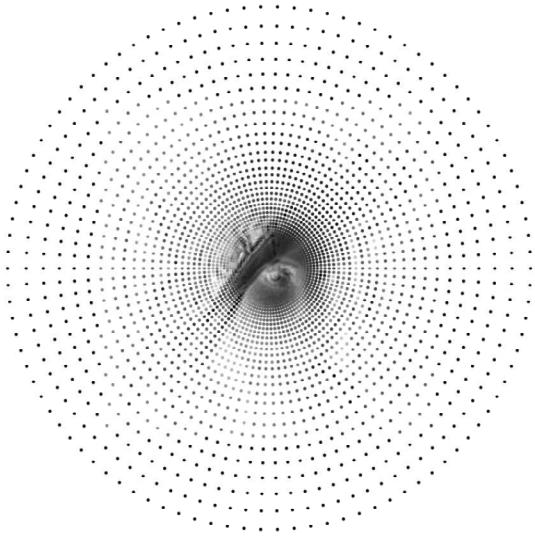
$$\begin{aligned}\mathcal{L}(\Theta^\ell,\psi^\ell,\mathcal{D}^\ell) = \{f_k^\ell | f_k^\ell : \mathbb{R}^2 \to \mathbb{R},\\ \sigma(\text{diam}(\text{dom}(f_k^\ell))) = \Theta^\ell,\\ \mathcal{D}^\ell = \mathcal{D}^u,\\ \text{and } k \in \{K^\ell = 1\}\},\end{aligned} \tag{2.30}$$

This saliency map has exactly the same distribution of point coordinates as that of feature maps. The RF of each point in $\mathcal{L}$ at a coordinate (r_o,ω_o) spans only the point at the same location in $\mathcal{U}$.

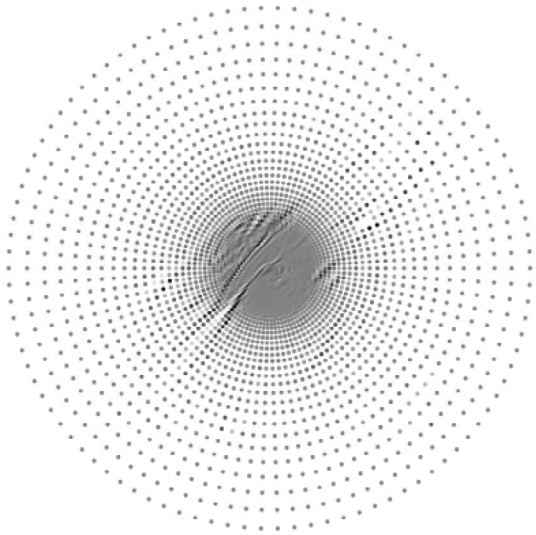
(a) Retinal image $\mathcal{P}$



(b) Feature map $\{f^u_{k=1}\} \subset \mathcal{U}$.



(c) Feature map $\{f^u_{k=6}\} \subset \mathcal{U}$.



(d) Feature map $\{f^u_{k=5}\} \subset \mathcal{U}$.

Figure 2-14: The retinal image represented by the receptors layer $\mathcal{P}$ (a) and some corresponding feature maps held by layer $\mathcal{U}$: intensity contrast feature map (b), red-green opponency feature map (c) and 135°-orientation feature map (d). These feature maps correspond to circular and radial overlap values $p_c = p_r = 0.8$, a visual angle span of $\Theta^I = 10°$ and a slope $\alpha = 0.16$ for the cortical magnification factor.

$$\begin{aligned} \mathrm{RF}^{u\ell}_k(f^\ell_k(r_o, \omega_o)) = \{(r,\omega) | (r,\omega) \in \mathrm{dom}(f^u_k), \\ (r_o, \omega_o) \in \mathrm{dom}(f^\ell_k), \\ \text{and } (r,\omega) = (r_o, \omega_o)\}. \end{aligned} \tag{2.31}$$

Before point values in $\mathcal{L}$ could be computed, a modulation function m^u as defined in (2.2) is applied to $\mathcal{U}$. This modulation is used to increase the contrast of the most salient regions in each feature map in a similar way the map normalization operator $\mathcal{N}(.)$ is applied in (Itti et al., 1998).

$$\begin{aligned}\tilde{\mathcal{U}} = m^u(\mathcal{U}) = \{f_k^{\tilde{u}} | f_k^{\tilde{u}} : \mathbb{R}^2 \rightarrow [0,1], \\ \mathrm{dom}(f_k^{\tilde{u}}) = \mathrm{dom}(f_k^u), \\ k \in \{1, ..., k^u = 9\}\}.\end{aligned} \tag{2.32}$$

The steps for computing the value of the modulated points $f_k^{\tilde{u}}$ are the following:

1. A half-wave rectification is first applied to feature maps to remove negative values.

$$f_k^{\tilde{u}} = \max(0, f_k^u). \tag{2.33}$$

2. The values within each feature map are scaled to the interval $[0, 1]$.

$$f_k^{\tilde{u}} \leftarrow \frac{f_k^{\tilde{u}} - \min\limits_k(f_k^{\tilde{u}})}{\max\limits_k(f_k^{\tilde{u}}) - \min\limits_k(f_k^{\tilde{u}})}. \tag{2.34}$$

3. A multiplicative factor β_k is computed.

$$\beta_k = \left(\max_k(f_k^{\tilde{u}}) - \mathrm{mean}_k(f_k^{\tilde{u}})\right)^2. \tag{2.35}$$

4. The multiplicative factor β_k is then applied to each point of the feature maps.

$$f_k^{\tilde{u}} \leftarrow \beta_k f_k^{\tilde{u}} \tag{2.36}$$

The input signal to each point in the saliency map can now be defined on the modulated feature maps:

$$s_k^{u\ell}(f_k^\ell(r_o,\omega_o)) = \{f_{k'}^{\tilde{u}}(r,\omega)| \\ (r,\omega) \in \mathrm{RF}_k^{u\ell}(f_k^\ell(r_o,\omega_o)), \\ f_{k'}^{\tilde{u}} \in \tilde{\mathcal{U}}.\}. \tag{2.37}$$

Finally, the saliency map is computed using the mapping $\phi_k^{u\ell}$, which is the mean of all modulated feature maps in $\tilde{\mathcal{U}}$.

$$\begin{aligned} f_k^\ell(r_o,\omega_o) &= \phi_k^{u\ell}(s_k^{u\ell}(f_k^\ell(r_o,\omega_o))) \\ &= \mathrm{mean}(s_k^{u\ell}(f_k^\ell(r_o,\omega_o))). \end{aligned} \tag{2.38}$$

2.5.5 Creating fixation maps

Fixation maps are created by an iterative processes consisting of a Winner-Take-All (WTA) step, which extracts the coordinates of the most salient point in the saliency map followed by an Inhibition-of-Return (IOR) step, which guarantees that previously fixated locations should no longer be visited in subsequent iterations. Here are the details of these two steps:

1. A fixation location (r_o,ω_o) is extracted from the saliency map $\mathcal{L}$.

$$(r_o,\omega_o) = \underset{(r,w)}{\mathrm{argmax}} f_k^\ell. \tag{2.39}$$

2. IOR is applied using a modulation function m^ℓ.

$$m_k^\ell(\mathcal{L}):$$
$$f_k^\ell(r,\omega) \leftarrow \begin{cases} f_k^\ell(r,\omega) & \text{if } \|(x,y),(x_o,y_o)\| > h \\ 0 & \text{otherwise,} \end{cases} \tag{2.40}$$

where h is the radius of the inhibited zone in visual angles.

3. the pixel indexes (i,j) in the image $\mathcal{I}$ corresponding to the fixation location (r_o,ω_o) are then computed using (2.11).

Figure 2-15 depicts an example of a saliency map in layer $\mathcal{L}$ and the corresponding smoothed saliency map. A smoothed saliency map is one consisting in convoluting a gaussian kernel on the extracted fixation locations, in order to produce a continuous gray-scale saliency map of the same size as the input image $\mathcal{I}$.

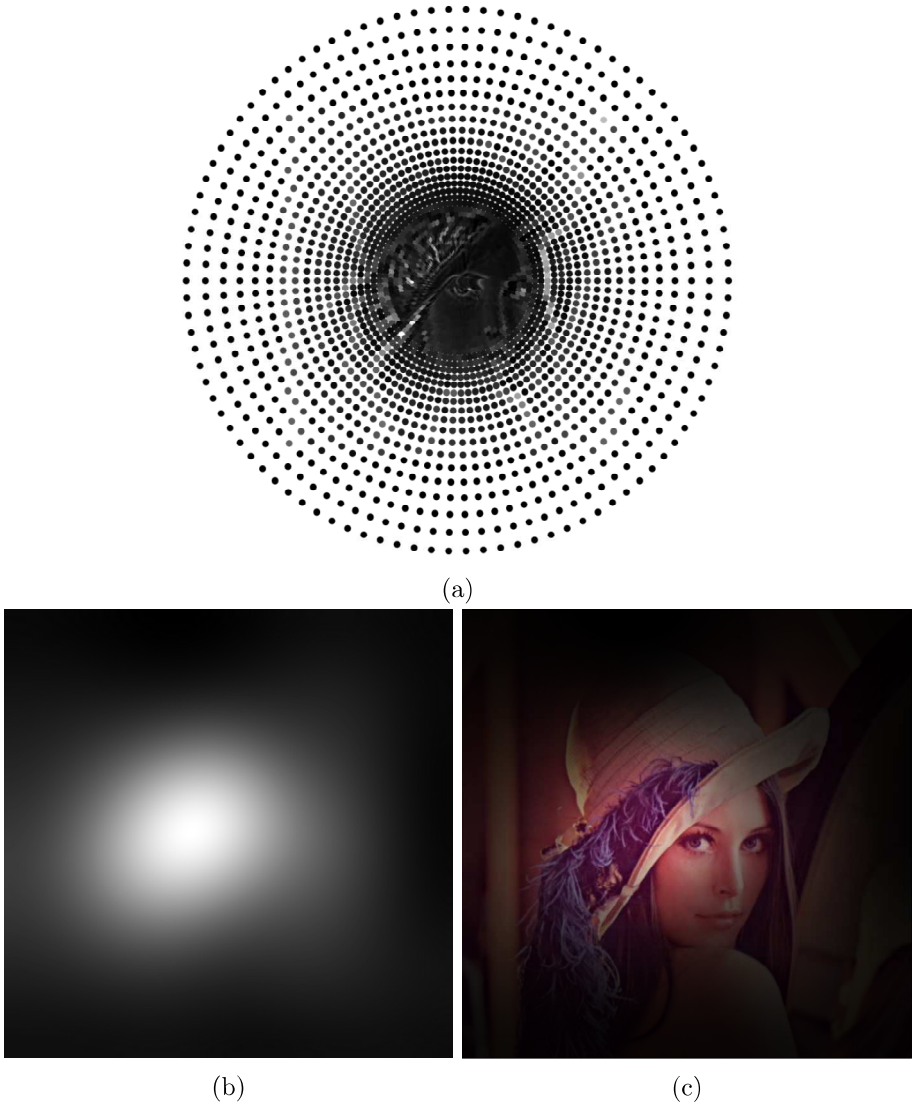
In the next section, we provide a performance evaluation of the proposed attention model along with a comparison with some of the state-of-the-art models, and a discussion of the results.

2.6 Results and discussion

In order to validate the performance of the proposed model on estimating bottom-up visual saliency, we ran the algorithm on the CAT2000 test dataset provided by the MIT saliency benchmark. This dataset contains 2000 images from 20 different categories with a fixed size of 1920×1080 pixels (Borji et al., 2013a).

Before beginning the test on the above dataset, we performed a minor optimization of the model parameters on the CAT2000 train dataset containing 2000 images of the same 20 categories as in the CAT2000 test set (Borji et al., 2013a).

Hence, we set the model parameters as follows: The visual angles $\Theta^I = \Theta^p = \Theta^u = \Theta^\ell = 10°$, which represent the visual field available to the system. The diameter of the fovea of all layers $\psi^I = \psi^p = \psi^u = \psi^\ell = 1°$. The total number of receptors in $\mathcal{P}$ is set



(a) (b) (c)

Figure 2-15: An example of a saliency map carried by layer $\mathcal{L}$ (a) where the size of single points is adapted for a better clarity of display. The corresponding smoothed saliency map is shown in (b) made by convoluting a Gaussian kernel on the first 250 fixation locations. In (c), the smoothed saliency map is superimposed on the original image $\mathcal{I}$.

to 123853 of which 30000 are within the fovea. This is equivalent to 41284 total RGB pixels of which 10000 are within the fovea, as shown in figures 2-7 and 2-10. This means that retinal images used by the algorithm has more than 50 times less pixels than the original images, which represents a significant reduction of information. The overlap parameters p_r and p_c between RFs of points in $\mathcal{U}$ are both set to 0.8. The

slope α in (2.18) associated with the cortical magnification factor in layer $\mathcal{U}$ is set to 0.16 which is close to its value in layer $V1$ of the ventral stream found by Gattass in (Gattass et al., 1988). For each image, the 250 most salient fixations locations are extracted by an iterative WTA and IOR process. The radius h of the inhibited zone at each IOR iteration is set to 0.05° of visual angles.

Parameters of DoG kernels were adapted from (Rodieck, 1965). We set g_2/g_1 to 0.8, δ_2/δ_1 to 3 and ρ/δ_1 to 11.8, where g_2/g_1 is a measure of the ratio of strength of the surround to the center Gaussians of the DoG kernel, and δ_1 and δ_2 are the effective widths of the center and surround Gaussians, respectively.

For Gabor kernels, we adapted parameter values used for designing simple cells in the Hmax model (Serre et al., 2007); The aspect ratio is set as $\gamma = 0.3$. We also set $\delta_3/\lambda = 0.8$ and $\rho/\delta_3 = 2.5$, where δ_3 is the effective width of the Gaussian component of the filter, λ is the wavelength of the cosine component, and ρ in both DoG and Gabor kernels denotes the eccentricity-dependent radius of a given kernel computed from (2.18) and expressed in visual angles.

Figure 2-16 depicts some examples of images taken from the CAT2000 test dataset and their corresponding smoothed saliency maps.

Table 2.1 shows the scores of our models according to several metrics used by the benchmark and how they compare to other models. This table and more detailed comparisons are also available on the MIT Saliency Benchmark website `http://saliency.mit.edu/results_cat2000.html`.

As shown in table 2.1, the proposed model shows good performance scores relative to other models. These scores are computed according to 7 metrics: the Similarity (Sim), the Correlation-Coefficient (CC), the Normalized Scanpath Saliency (NSS) and Earth Mover's Distance (EMD) and the Area Under the ROC Curve metrics.

It is worth pointing out that the IttiKoch2, GBVS, Judd's and several other models are optimized for smoothing parameters and center-bias. The proposed model, has only a minor optimization for the width of the Gaussian kernel used for smoothing fixation maps while no explicit center-bias is applied. However, such bias arises naturally in the model due to retinal sampling and cortical magnification factors, which

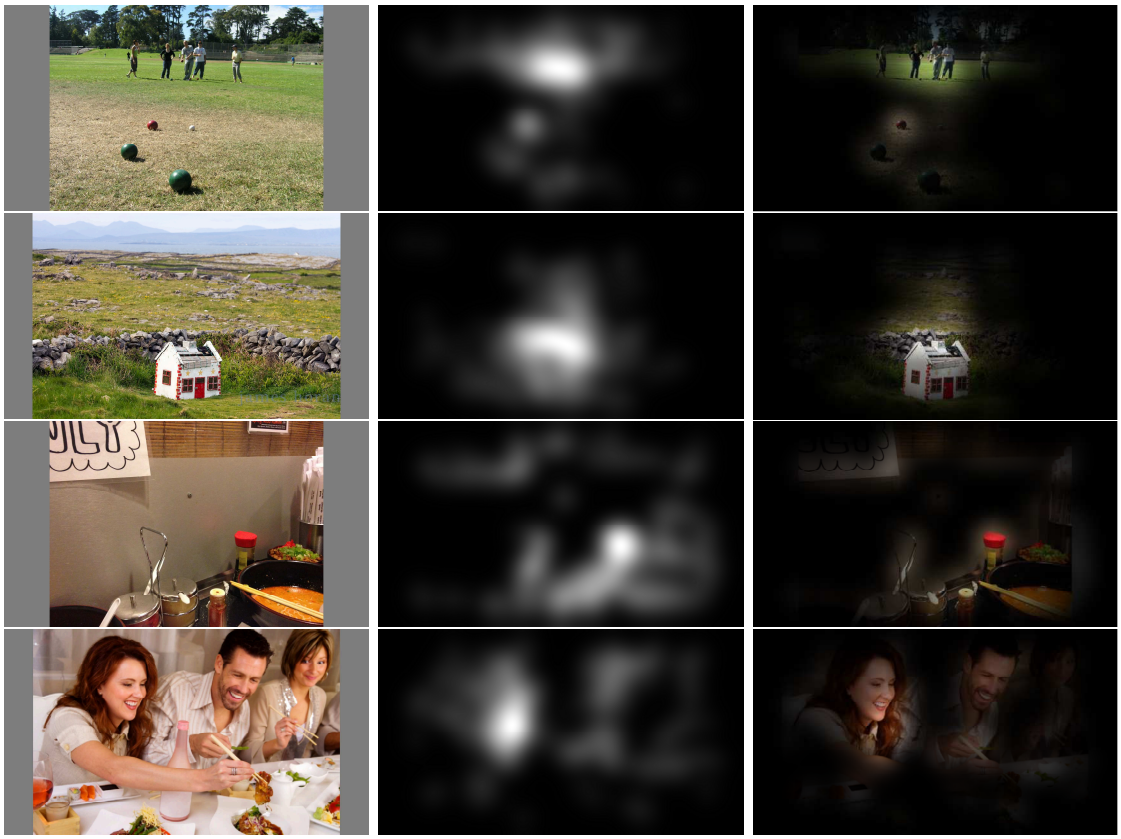


Figure 2-16: Some example images from the CAT2000 test dataset (left column), the corresponding smoothed saliency maps (middle column) and with saliency maps superimposed (right column).

allocate more resources to processing central zones of the image than to peripheral ones. It would be interesting to explore the role of retinal sampling and cortical magnification in influencing center-bias that human subjects manifest when free viewing images.

An important point to discuss is the fact the attentional behavior manifested by eye fixations differs as a function of the viewing distance, (or equivalently the visual angle) between the subject and an image (Borji et al., 2013b). However, attention models in table 2.1 do not have a direct way for measuring their performance as a function of the visual angle. This makes interpreting the performance of such models against a given saliency dataset more ambiguous and less straightforward. For example, suppose having two datasets D_1 and D_2, associated with two visual angles θ_1 and θ_2, respectively. If a given attention model performs better on D_1 than

Table 2.1: A performance comparison between the proposed model and other models on the CAT2000 test dataset of the MIT Saliency Benchmark. These results can be found on the MIT saliency benchmark Web page `http://saliency.mit.edu/results_cat2000.html`.

Model		Sim	AUC-Judd	EMD (Lower is better)	AUC-Borji	CC	NSS	sAUC
Proposed model		**0.58**	**0.80**	**2.10**	**0.77**	**0.64**	**1.57**	**0.55**
BMS	(Zhang and Sclaroff, 2013)	0.61	0.85	1.95	0.84	0.67	1.67	0.59
GBVS	(Harel et al., 2006)	0.51	0.80	2.99	0.79	0.50	1.23	0.58
Context-Aware saliency	(Goferman et al., 2012)	0.50	0.77	3.09	0.76	0.42	1.07	0.60
AWS	(Garcia-Diaz et al., 2012)	0.49	0.76	3.36	0.75	0.42	1.09	0.62
IttiKock2		0.48	0.77	3.44	0.76	0.42	1.06	0.59
WMAP	(López-García et al., 2011)	0.47	0.75	3.28	0.69	0.38	1.01	0.60
Judd model	(Judd et al., 2009)	0.46	0.84	3.61	0.84	0.54	1.30	0.56
Torralba saliency	(Torralba et al., 2006)	0.45	0.72	3.44	0.71	0.33	0.85	0.58
Murray model	(Murray et al., 2011)	0.43	0.70	3.79	0.70	0.30	0.77	0.59
SUN saliency	(Zhang et al., 2008)	0.43	0.70	3.42	0.69	0.30	0.77	0.57
IttiKock	(Itti et al., 1998)	0.34	0.56	4.66	0.53	0.09	0.25	0.52
Achanta	(Achanta et al., 2009)	0.33	0.57	4.45	0.55	0.11	0.29	0.52

on D_2, there would be no clear way for determining whether this is due to the fact that it is intrinsically more adapted to the angle θ_1 that to θ_2, or due to other factors.

The model we propose provides the possibility to fix all other parameters while varying the image's visual angle Θ^I. Figures 2-17 and 2-18 depict how the performance of the proposed attention algorithm varies according to different evaluation metrics as a function of Θ^I. This provides a mechanism to check whether the model matches ground truth attentional behavior when measured at different visual angles. We think that this is a useful factor to consider for models that seek biological plausibility. However, as to our knowledge, no available benchmarks provide fixation data measured at different visual angles yet. Creating such a benchmark would provide the possibility to analyze and validate our performance curves in figures 2-17 and 2-18, as well as those of future models that might choose to integrate a visual angle

parameter.

Finally, while the proposed model does not always give the best saliency prediction according to table 2.1, it provides some advantages over other models from a biological point of view:

- Eye movements and fixations can be emulated more faithfully using the proposed vision framework. As in the real retina, moving the fovea over the image will change the resolution perceived at each region of the image due to retinal sampling and cortical magnification factors.
- A more straight-forward way to compare to ground truth on vision tasks. Effects of fundamental vision parameters absent from most saliency models, such as viewer distance, visual field, cortical magnification and retinal sampling could potentially be studied more closely using the proposed framework.

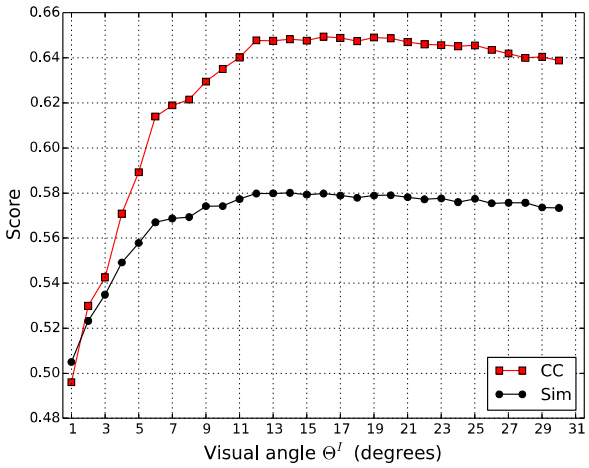


Figure 2-17: The influence of changing the images' visual angle Θ^I on the models performance according to the Similarity (Sim) and the Cross Correlation (CC) metrics.

2.7 Conclusion and future work

In this chapter, treated the visual acquisition problem which is the lowest stage of the visual processing pipeline. We proposed a new framework for building visual

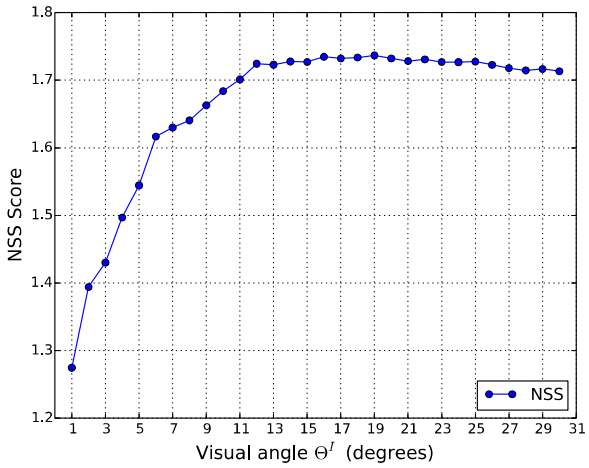


Figure 2-18: The influence of changing the images' visual angle Θ^I on the models performance according to the Normalized Scanpath Saliency (NSS) metric.

information processing models. This model is more closely inspired by the architecture of the primate visual system, and is motivated by the recent trend in the computer vision community toward a closer modeling of the visual system in the hope of going beyond some limitations in current vision systems.

We have seen that the architecture of the proposed framework offers some interesting properties found in the visual system. For example, the presence of a receptor layer makes the acquired image signal totally independent from the input image's resolution and size. It also motivates the use of such a framework for applications such like saccadic eye movements since the receptor layer is not constrained by the image borders and can be used to receive its signal from any part of the input scene. Moreover, the proposed framework has a very clear notion of a visual angle emulating the ubiquitous presence of this parameter in biological vision. This provides the possibility to better understand the influence of a viewer's distance from an image on vision tasks. Another important property the framework offers is the information reduction by means of retinal sampling and cortical magnification which are two important and omnipresent factors of primate visual systems. We have seen that these two mechanisms can be implemented seamlessly, while avoiding classical problems like spatial deformations and the dependency on the input image size.

In section 2.5, we proposed a saliency-driven model of attention built on top of the proposed vision framework. We showed that this model attains state-of-the-art performance. More particularly, we showed that it has a better performance than Itti and Koch's model on which it is based, while using lower resolution and a fewer number of feature maps. This application motivated the use of the proposed vision framework and raises some important questions such as the role of the visual angle in attention modeling and its importance for a better understanding of attentional behavior and benchmarking results. One possible method we propose to start such exploration, would be to design an attention benchmark that provide eye-fixation data on a given dataset for a range of visual angles. It would be then interesting to study how models' performances should be analyzed and understood given this variability of fixation data associated with different visual angles.

In future work, we will also consider the question of how common architectures for visual processing, especially Convolutional Neural Networks (CNNs) might be adapted for being implemented using the proposed framework. The challenge would be in modifying its learning algorithm so that it can take the cortical magnification factor into account and the associated variability in kernel sizes in each layer.

Another research perspective would be to use the proposed framework for implementing attention-based object recognition processors to account for the retinal transformation stage in models such like (Zheng et al., 2015).

The proposed framework and the associated attention model are already implemented and are publicly available as a Git repository on the Web[1]. However, future work will include further development and improvement of the proposed framework along with its code implementation. We hope that through collaboration, this framework could evolve as an alternative, full-fledged toolbox for neuro-inspired visual processing, in the same way as current programming libraries offer optimized implementations of traditional image processing algorithms.

[1] https://bitbucket.org/ala_aboudib/see

Chapter 3

A new retrieval algorithm for Sparse Clustered Networks

In this chapter, we are interested in the highest stage of the information processing pipeline which is memory. More precisely, we focus on associative memories which are data structures addressed using part of the content rather than an index. They offer good fault reliability and a good model of biological memories which are associative in nature. Among different families of associative memories, sparse ones are known to offer the best efficiency (ratio of the amount of bits stored to that of bits used by the network itself). Their retrieval process performance has been shown to benefit from the use of iterations. In this chapter, we focus our attention on a recently proposed model of sparse associative memories called the Sparse Clustered Network (SCN). We review the different rules used for the retrieval process of data from these networks. We then suggest a new policy that provide a better retrieval performance than existing techniques. Extensive performance evaluations among the different retrieval algorithms are finally provided along with a discussion of the different aspects that govern their behaviour.

3.1 Introduction

Associative memories are alternatives to classical index-based memories where content is retrieved using a part of it rather than an explicit address. Consider for example accessing a website using a search engine instead of a uniform resource locator (URL). This mechanism is analogous to human memory (Anderson and Bower, 1973) and has inspired many neural-networks-based solutions as in (Willshaw et al., 1969; Hopfield, 1982).

A new artificial neural network model was proposed recently by Gripon and Berrou (Gripon and Berrou, 2011). It employs principles from information theory and error correcting codes and aims at explaining the long-term associative memory functionality of the neocortex. This model was proved to outperform Hopfield neural networks (Hopfield, 1982) in terms of diversity (the number of messages the network can store), and efficiency (the ratio of the amount of useful bits stored to that of bits used to represent the network itself) (Gripon and Rabbat, 2013). It was later extended in (Aliabadi et al., 2014) to a sparser version which can be viewed as a generalization of the Willshaw-Palm associative memory model (Willshaw et al., 1969; Palm, 2013).

The key difference between the models proposed in (Aliabadi et al., 2014) and (Willshaw et al., 1969) is the use of specific structures in the network. This is done by grouping neurons into clusters within which connections are not allowed (multi-partite graph). These clusters are considered analogous to cortical columns of mammalian brains in (Gripon and Berrou, 2011) in which nodes are likened to micro-columns. This is supported by Mountcastle (Mountcastle, 1997), who suggests that a micro-column is the computational building block of the cerebral neo-cortex. In addition, here are some reasons to motivate the use of clusters:

- It is believed that micro-columns in each cortical column react to similar inputs. The concept of clustering is meant to imitate this stimulus-similarity-based grouping. A consequence is the possibility to use this network for retrieving messages from inaccurate observations. This type of retrieval is addressed in (Gripon and Jiang, 2013).

- Clusters allow for simple and natural mapping between nonsparse input messages and sparse patterns representing them in the associative memory. In the case where each cluster contains only one unit, a model equivalent to the classical Willshaw-Palm networks is obtained, where input messages have to be sparse.

- It was observed that micro-columns usually have many short inhibitory connections with their neighbors (Buxhoeveden and Casanova, 2002; Mountcastle, 1997), which means that the activation of one micro-column causes all of its near neighbors to be deactivated. This is due to the locally limited energy supply of the brain. This mechanism is represented by the local winner-takes-all (WTA) rule introduced in (Gripon and Berrou, 2011), in which a neural mechanism for implementing the WTA process has been proposed.

- Using clusters allows for introducing guided data recovery in which a prior knowledge of the location of clusters containing the desired data can significantly enhance performance. A detailed study of this type of data retrieval is available in (Aliabadi et al., 2014).

Our main contribution is to provide a generic formulation of the several retrieval rules previously proposed for SCNs. We also propose a new rule that is shown to provide a better retrieval performance.

This chapter is organized as follows: in section 3.2, we describe the general architecture of the network model we use. Section 3.3 introduces a generic formulation of the different retrieval algorithms that were proposed previously. Then, the following few sections are devoted to explaining each step of this algorithm. For each step, previous rules are reviewed, and new rules are proposed. In section 3.7, performance comparisons of several combinations of retrieval rules are presented. Section 3.9 is the chapter conclusion.

3.2 Network topology and storing messages

This section focuses on the neural-network-based auto-associative memory introduced in (Gripon and Berrou, 2011). It is dedicated to defining this network and describing how it can be extended to store variable-length messages.

3.2.1 Architecture

The network can be viewed as a graph consisting of n vertices or units initially not connected (zero adjacency matrix) organized in χ parts called clusters with each vertex belonging only to one cluster. Clusters are not necessarily equal in size but for simplicity, they will be all considered of size ℓ throughout this chapter. Each cluster is given a unique integer label between 1 and χ, and within each cluster, every vertex is given a unique label between 1 and ℓ. Following from this, each vertex in the network can be referred to by a pair (i, j), where i is its cluster label, and j is the vertex label within cluster i. As argued in (Aliabadi et al., 2014), a unit in this model is chosen to represent a cortical micro-column instead of a single neuron. This is based on the arguement that microcolumns might actually be considered as the computational building blocks within the cerebral cortex (Cruz et al., 2005; Jones, 2000).

At any given moment, a binary state v_{ij} is associated with each unit (i, j) in the network. It is given the value 1 if (i, j) is active or 0 otherwise. Initially, all units are supposed to be inactive. The adjacency matrix for this graph W is a binary symmetric square matrix whose elements take values in $\{0, 1\}$. In this representation, a zero means an absence of a connection while a one indicates that an undirected (or a symmetric) connection is present. Note that despite the fact that biological neural networks are known to be asymmetric, we argue that units in the proposed model represent populations of tens of neurons, and therefore can be mutually connected.

Row and column indexes of the weight matrix are pairs (i, j). So in order to indicated that two units (i, j) and (i', j') are connected, we write $W_{ij;i'j'} = 1$. All connection combinations are allowed except those among units belonging to the same

cluster, resulting in a χ-partite undirected graph. When the memory is empty, W is a zero matrix.

3.2.2 Message storing procedure

We now describe how to store sparse messages using this network. This methodology has been first introduced in (Aliabadi et al., 2014). Suppose that each message consists of χ submessages or segments. Some of these segments are empty, i.e., they contain no value that need to be stored, while the rest has integer values in $\{1, ..., \ell\}$. For the sake of simplicity, let us consider that all messages contain the same number of submessages c. Only those nonempty submessages are to be stored while empty ones are ignored. For example, in a network with $\chi = 6$ and $\ell = 12$, a message $m = \{, 10, 7, , 12, 11\}$ with $c = 4$ has two empty segments (the first and the fouth ones), while the remaining ones have values that need to be stored. In order to store m, the position i of each nonempty segment within this message is interpreted as a cluster label, and the segment value j is interpreted as a unit label within the cluster i. Thus, each nonempty segment is associated with a unique unit (i, j). So the message m maps to the 10^{th} unit of the 2^{nd} cluster, the 7^{th} unit of the 3^{rd} cluster, the 12^{th} unit of the 5^{th} cluster and the 11^{th} unit of the 6^{th} cluster.

Then, given these elected units in distinct clusters, the adjacency matrix of the network is updated according to (3.1) so that a fully connected subgraph (clique) is formed of these selected units.

$$w_{ij;i'j'} = \begin{cases} 1 & \text{if } (i,j) \text{ and } (i',j') \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where $w_{ij;i'j'}$ refers to the undirected connection between (i, j) and (i', j') which are two units associated to message segments m_i and $m_{i'}$, respectively. Note that i and i' also denote cluster indices, while j and j' denote unit indices. A single message is not allowed to use more than one unit within the same cluster because, by definition, connections are not allowed within a cluster.

The value of the parameter c can be identical for all stored messages, or it can be variable. A discussion of how to choose an optimal value of c is provided in (Aliabadi et al., 2014) in which all messages are considered of the same size.

It is important to note that if one wishes to store a message m' that overlaps with m, i.e., the clique corresponding to m' shares one or more connections with that of m, the value of these connections, which is 1, should not be modified. As a direct consequence, the network's connection map is the union of all cliques corresponding to stored messages. It is worth noting that when $\ell = 1$, this network becomes equivalent to the Willshaw-Palm model (Schwenker et al., 1996).

3.3 The retrieval process

The goal of the retrieval process is to recover an already stored message (by finding its corresponding clique) from an input message that has undergone partial erasure. A message is erased partially by eliminating some of its nonempty segments. For example, if $m = \{, 1, 8, , 10, 12\}$ is a stored message, a possible input for the network might be $m = \{, , , , 10, 12\}$.

We propose a generic formulation of this retrieval process as an iterative twofold procedure composed of a dynamic rule and an activation rule as depicted in algorithm 1.

An input message should be fed to the network in order to trigger the retrieval process. For example, suppose that we have a stored message $m = \{7, 1, 5, 11, , \}$, and that we wish to retrieve m from a query message $\overline{m} = \{, , 5, 11, , \}$ that is a partially erased version of m. In order to do that, all units corresponding to nonempty segments should be activated. That is, each unit (i, j) associated with segment $\overline{m}_i$ is activated by setting $v_{ij} = 1$. So, $\overline{m}$ would activate two units: $(3, 5)$ and $(4, 11)$. Having a number of active units, a dynamic rule should then be applied.

Algorithm 1: The proposed generic formulation for the retrieval process.

input : Query message $\overline{m}$.

Apply a dynamic rule.

Phase 1
Apply an activation rule.
Apply a dynamic rule.

Phase 2
while *stopping criterion is not attained* **do**
  Apply an activation rule.
  Apply a dynamic rule.
end

output: Message corresponding to active units.

3.4 Dynamic rules

A dynamic rule is defined as the rule according to which unit scores are calculated. We will denote the score of a unit (i, j) by λ_{ij} . Calculating units' scores is crucial to deciding which ones are to be activated. A score is a way of estimating the chance that a unit belongs to a bigger clique within the set of active units and thus the chance that it belongs to the message we are trying to recover. In principle, the higher the score the higher this chance is. Two dynamic rules have been already introduced, namely, the Sum-of-Sum (Gripon and Berrou, 2011) and the Sum-of-Max (Gripon and Berrou, 2012) rules. We propose a new rule that we shall refer to as the Normalization rule.

3.4.1 The Sum-of-Sum (SoS) rule

The SoS rule states that the score of a unit score λ_{ij} is computed as the number of active units connected to (i, j) plus a predefined memory effect γ which is only added if (i, j) is active. Scores should be calculated for all of the units in the network. This

rule is described by the following equation:

$$\forall i,j : 1 \leq i \leq \chi, 1 \leq j \leq \ell :$$
$$\lambda_{ij} = \gamma v_{ij} + \sum_{i'=1}^{\chi} \sum_{j'=1}^{\ell} w_{ij;i'j'} v_{i'j'}. \tag{3.2}$$

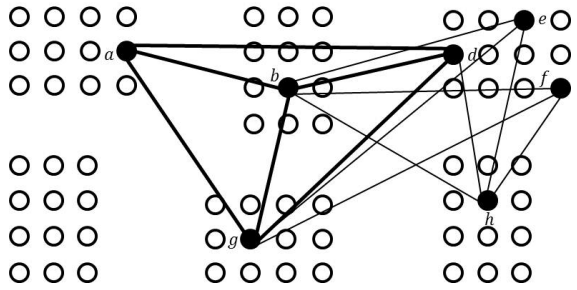


Figure 3-1: An example configuration of stored messages inside the network during the retrieval process. Only black units are active.

This rule has a major drawback; in some cases, the scores give a false estimate of the chance that a given unit belongs to a bigger clique within the set of active units. To clarify this point we consider the example of figure 3-1 where black circles represent active units at an iteration $t > 1$. The clique we wish to retreive is *abdg*. Now, we will see what happens when we calculate the scores of units a and h given a memory effect $\gamma = 1$ where a is part of the searched message while h is not. According to the Sum-of-Sum rule, unit a has a score of 4 while unit h has a score of 5. This indicates that the latter unit is more likely to belong to a bigger clique than the former because it has a higher score. This observation is not true since most of the active units connected to h belong to the same cluster and by conception, a message can only contain at most one unit per cluster. In order to solve this problem, the Sum-of-Max and the Normalization dynamic rules can be applied.

3.4.2 The Normalization (Norm) rule

In the Normalization rule that we introduce here, units' scores are calculated using the following equation:

$$\forall i,j : 1 \leq i \leq \chi, 1 \leq j \leq \ell :$$
$$\lambda_{ij} = \gamma v_{ij} + \sum_{i'=1}^{\chi} \frac{1}{|v_{i'}|} \sum_{j'=1}^{\ell} w_{ij:i'j'} v_{i'j'}. \tag{3.3}$$

where $|v_{i'}|$ is the number of active units in cluster k. Equation (3.3) states that the contribution of a unit (i', j') to the score of another unit connected to it is normalized by the number of active units in cluster k. That is, if the cluster i' contains x active units, then the contribution of the unit (i', j') is equal to $1/x$. So, by applying this rule to the network of figure 3-1, unit h gets a score of 3 and unit a gets a score of $3\frac{1}{3}$, which privileges the activation of the latter unit and thus solves the problem we encountered when using the Sum-of-Sum rule.

3.4.3 The Sum-of-Max (SoM) rule

According to the Sum-of-Max rule, the score of a unit (i, j) is the number of clusters in which there is at least one active unit (i', j') connected to (i, j) plus the memory effect γ if (i, j) is active:

$$\forall i,j : 1 \leq i \leq \chi, 1 \leq j \leq \ell :$$
$$\lambda_{ij} = \gamma v_{ij} + \sum_{i'=1}^{\chi} \max_{1 \leq j' \leq \ell} (w_{ij:i'j'} v_{i'j'}). \tag{3.4}$$

So referring back to figure 3-1, and according to (3.4), unit a has a score of 4 whereas unit h has a score of 3. This is a more accurate result than the one obtained by the Sum-of-Sum rule in the sense that it indicates that the latter unit, although connected to more active units, is less likely to belong to a bigger clique within the set of active units than unit a.

Moreover, it has been shown in (Gripon and Berrou, 2012) that for the particular

case, when $c = \chi$, the Sum-of-Max rule guarantees that the retrieved massage is always either correct or ambiguous but not wrong. An ambiguous output message means that in some clusters more than one unit might sometimes be activated, among which one is the correct unit.

3.5 Activation rules

The activation rule is applied for selecting the units to be activated based on their scores after the application of a dynamic rule. So basically, a unit (i, j) is activated if its score λ_{ij} satisfies two conditions:

- λ_{ij} is greater or equal than a global threshold that may be chosen differently for each activation rule.

- $\lambda_{ij} \geq \sigma_{ij}$ where σ_{ij} is the activation threshold of unit (i, j).

The difference between the two thresholds defined above is that σ_{ij} could be set differently for each unit, so it can be used to control a unit's sensitivity to activation. For a very large value of σ_{ij}, a unit (i, j) is inhibited. This is helpful for excluding a group of units from the search of a certain message in order to save time. The global threshold has a unique value independent of any individual unit. So it is used to elect units to be activated in a competitive activation process. For example, in a winner-take-all competitive process, this threshold could be dynamically set to the value of the highest score in the network in order to activate only units with the highest score.

The activation rule should be able to find two unknowns: The subset of clusters to which the message we are trying to recover belongs, and the exact units within these clusters representing the submessages. We propose two new activation rules in this chapter: the Global Winners Take All rule (GWsTA) which is a generalization of the Global Winner Take All (GWTA) rule, and an enhanced version of the Global Losers Kicked Out (GLsKO) rule initially presented in (JIANG et al., 2012).

3.5.1 The GWsTA rule

In the Global Winner Takes All (GWTA) rule introduced in (Aliabadi et al., 2014), only units that have the maximal score across the network are activated. The problem with this rule is that it supposes that units belonging to the message we are looking for have equal scores. It also supposes that this unified score should be the maximal network score which is not necessarily the case. It has been shown in (Aliabadi et al., 2014) that spurious cliques, i.e., cliques that share one or more edges with the clique we are searching, might appear and render the scores of the shared units of the searched clique higher than others'.

For example, in the network of figure 3-1, if the clique we are seeking is $abdg$, then bdh is an example of a spurious one. Now, by applying the SoM rule on the black units which are supposed to be active, and considering $\gamma = 1$, we get the scores: $\lambda_a = 4$, $\lambda_b = 5$, $\lambda_d = 5$, $\lambda_e = 4$, $\lambda_f = 4$, $\lambda_q = 4$ and $\lambda_h = 3$. Thus, according to the GWTA rule, only units b and d will be kept active and the clique $abdg$ is lost. This is caused by the spurious clique bdh which increases the scores of b and d. The generalization of the GWTA rule we propose is meant to account for this problem.

The behavior of the GWsTA rule is the same in both phases of the retrieval process we described in algorithm 1. It selects a subset of units with maximal and near-maximal scores to be activated. In other words, it defines a global threshold θ at each iteration, and only units whose scores are greater or equal to this threshold are activated.

In order to compute this threshold, we first fix an integer parameter α . Then we make a list of the α highest scores in the network including scores that appear more than once. For example, if units' scores in a network with a total number of units, $n = 10$, are $\{25, 18, 25, 23, 23, 19, 18, 19, 18, 17\}$ and $\alpha = 7$, then the list becomes $\{25, 25, 23, 23, 19, 19, 18\}$. The minimum score in this list which is 18 is assigned to

the threshold θ. Then we apply the following formula:

$$\forall i,j : 1 \leq i \leq \chi, 1 \leq j \leq \ell :$$
$$v_{ij} = \begin{cases} 1 & \lambda_{ij} \geq \theta \text{ and } \theta \geq \sigma_{ij}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

It is worth pointing out that this activation rule is equivalent to the retrieval rule proposed in (Sommer and Palm, 1999) in that units are activated by comparing their scores to a fixed threshold θ. One problem with this rule is that the choice of an optimal α for a certain message size would not be adapted for other message sizes. This limits the possibility of using this rule for retrieving messages of variable sizes. However, this problem is solved by using the GLsKO rule we present in the next subsection.

3.5.2 The GLsKO rule

As we have seen, The GWsTA rule needs a prior knowledge of the value of the message size c. This means that if c is not available, the rule may not be able to correctly retrieve information. The Global Losers Kicked Out (GLsKO) rule is designed to address this problem by being independent of any prior information about c which should also enable it to retrieve variable-sized messages more efficiently than the GWsTA rule. In order to achieve this, the GLsKO rule has a behavior in *phase 1* of the retrieval process that differs from that of *phase 2* as follows:

- ***phase 1***: Apply the GWTA rule.
- ***phase 2***: Kick losers out.

In *phase 1*, the GWTA rule is applied which results in the activation of a subset of units to which the searched message is guaranteed to belong. After this, the activation thresholds of inactive units are set to infinity because we are no more interested in searching outside the set of activated units.

In *phase 2*, the rule changes behavior. At each subsequent iteration, we make a list containing the β lowest nonzero scores of the active units only. For example, if the set $\{25, 18, 25, 23, 23, 19, 18, 19, 17, 17\}$ represents the scores of active units in a network with a total number of units $n = 10$ and we fix $\beta = 3$, then the list of lowest scores becomes $\{18, 19, 18, 19, 17, 17\}$. After that, a threshold θ equal to the maximum score in the latter list is set, and only units with scores greater than θ are kept active. This can be described by the following equation:

$$\forall i, j : 1 \leq i \leq \chi, 1 \leq j \leq \ell :$$
$$v_{ij} = \begin{cases} 1 & \lambda_{ij} \geq \theta \text{ and } \theta \geq \sigma_{ij}, \\ 0 \text{ and } \sigma_{ij} \to \infty & \text{otherwise.} \end{cases} \tag{3.6}$$

The reason why σ_{ij} is set to an infinitely large value is that after the first phase of the algorithm, a subset of units is activated. The clique corresponding to the message we are looking for is guaranteed to exist in this subset given that we are dealing with partially erased messages. So, setting σ_{ij} this way ensures that units that have failed to be active upon the first phase would be out of the search scope throughout the retrieval process.

We propose to enhance the performance of the GLsKO rule by controlling the number of units μ to be deactivated. This is only interesting when $\beta = 1$. For example, if we set $\beta = 1$ in the network example of the previous paragraph, we get the following list of scores $\{17, 17\}$. If μ is not specified, all losers are deactivated. But by setting $\mu = 1$, only one of these two units is randomly chosen to be deactivated. This may be useful if we wish to exclude losers one at a time and thus reduce incautious quick decisions.

3.6 Stopping criteria

Since the retrieval process is iterative, a stopping criterion should be used in order to put this process to an end. In the following subsections we review the criteria used

typically for this goal and we propose new ones.

3.6.1 A fixed number of iterations (Iter)

A stopping criterion can be defined as a fixed number of iterations for the retrieval process. So dynamic and activation rules are applied iteratively, and when a counter attains the desired number of iterations, the retrieval process terminates and the units that stay active are taken as the retrieved message. The problem with this approach is that a stopping criterion as a simple iteration counter is independent of the nature of retrieved messages. That is, activated units after the last iteration are not guaranteed to form a clique corresponding to an stored message. This use of this stopping criterion is only interesting with the GWsTA rule.

3.6.2 The convergence criterion (Conv)

This criterion states that if the set of active units at iteration $t + 1$ is the same as that of iteration t, the retrieval process terminates and the result is output. The convergence criterion is only compatible with the GWsTA rule. In the case of the GLsKO rule, one or more active units are deactivated in each iteration. So it is not possible to have the same set of active units across two subsequent iterations.

3.6.3 The equal scores criterion (EqSc)

According to the EqSc criterion, when scores of active units are all equal, the retrieval process terminates and the result is output.

3.6.4 The clique criterion (Clq)

The Clq criterion we propose depends on the relationship between the number of activated units and their scores. If activate units form a clique the retrieval process terminates. Thus, the retrieved message is more likely to correspond to a stored message although it would not necessarily be the correct result. In order to check

if activated units form a clique, we define the set of active units as $A = \{a_i | i = 1, 2, ..., |A|\}$, $\lambda(a_i)$ as the score of the active unit a_i and ρ as an integer, then we apply the procedure depicted in algorithm 2.

Algorithm 2: The clique stopping criterion (Clq).

$\forall i, j \in \{1, 2, ..., |A|\}$

if $\lambda(a_i) = \lambda(a_j) = \rho$ ***and*** $|A| = \rho - (\gamma - 1)$ **then**
| Output the result.
end

Terminate the retrieval process.

To make sense of algorithm 2, we take an intuitive situation when $\gamma = 1$. In this case, the stopping criterion is that when all active units have an equal score which is equal to the number of these units, a clique is recognized, so the process terminates and the result is output.

It is worth noting that when using the GWsTA rule, it is always preferable to combine any stopping criterion with the Iter criterion so that when any one of them is satisfied the process terminates avoiding infinite looping.

3.7 Results

We have seen that there are many possible combinations of dynamic, activation rules and stopping criteria in order to construct a retrieval algorithm. In this section we will demonstrate the performance of some of these combinations. All messages used for the following tests are randomly generated from a uniform distribution over all possible message values. Reported retrieval error rates for a given number of stored messages are averaged over 100 trials. However, no significant difference was found between average error rates and error rates resulting from single trials.

3.7.1 Comparing dynamic rules

Figure 3-2 shows that both the SoM and the SoS dynamic rules give a similar performance. The Norm rule was found to give the same results also, but it is not

shown in the figure for clarity. This is not the case with the original network introduced in (Gripon and Berrou, 2011) where the SoM rule was proved to give better results (Gripon and Berrou, 2012). This is an interesting phenomenon that is worth studying. It may indicate that the major source of retrieval errors in this sparse version of the network is not related to the different methods for computing units' scores. This renders the differences in performance due to the use of different dynamic rules insignificant.

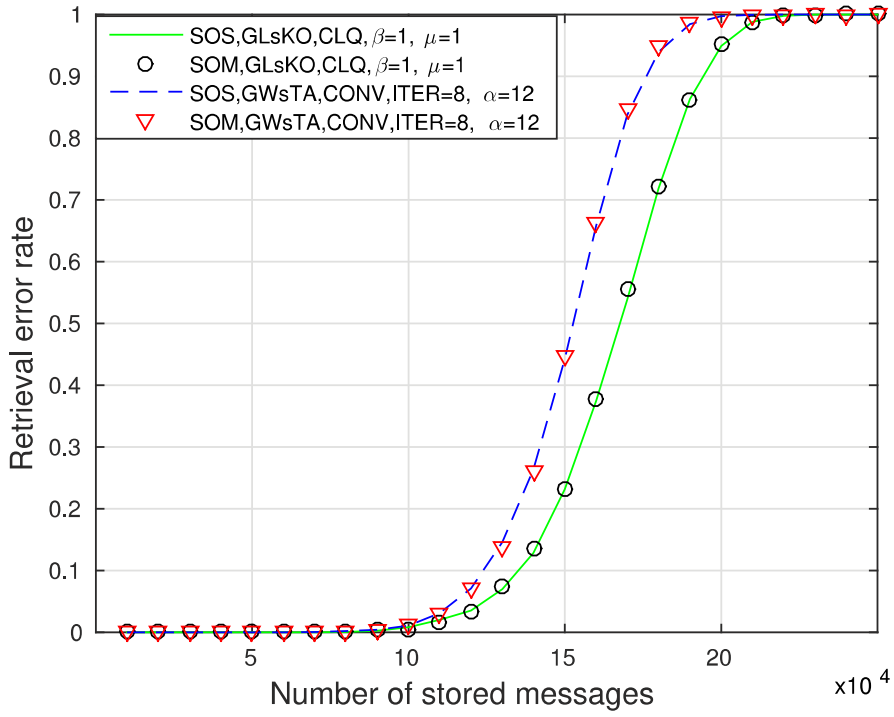


Figure 3-2: Influence of dynamic rules on retrieval error rates in a network with $\chi = 100$, $\ell = 64$, $c = 12$, $\gamma = 1$, $\sigma_{ij} = 0$ initially, with 3 segments of partial erasure in input messages.

3.7.2 Comparing retrieval strategies

We notice in figure 3-3 that the GWsTA ($\alpha = 12$) rule gives a better performance than the GWTA (equivalent to GWsTA with $\alpha = 1$) rule used with the Conv stopping criterion with 30 iterations allowed at most. This is due to the fact that the former rule has a better immunity to the phenomenon of spurious cliques described in section 3.5.

We also notice that the GWsTA ($\alpha = 12$) rule gives even a lower error rate when the memory effect γ is set to a large value such as 1000. This is because setting γ to that value restrains the search to only a limited subset of units in the network where the target message is thought to be found. This is due to the fact that a large value of γ guarantees that active units always get higher scores than other ones. Therefore, in subsequent iterations, the set of active units would most often be the same or a subset of the previous active set. In all cases, the GLsKO ($\alpha = 1$, $\mu = 1$) rule using the EqSc or the Clq (not shown on the figure) stopping criterion has the lowest error rate which almost achieves the performance of the brute force Maximum Likelihood retrieval algorithm (ML) (which is a simple exhaustive search for a maximum clique) for 3 erased input submessages out of 12. This is because the GLsKO rule configured with such parameter values searches for the output in a limited subset of units resulting from *phase 1* and excludes only one unit at a time before testing for the stopping criterion. This is proved by the degraded performance shown in figure 3-3 of this same rule but without specifying a value of μ which results in the exclusion of more than one unit at a time rendering the retrieval process less prudent and more susceptible to bad exclusions.

We also notice that when a Willshaw-Palm network with $n = 6400$ units is used with the GWsTA ($\alpha = 12, \gamma = 1$) rule, the same performance as in a clustered network is obtained.

3.8 The number of iterations

Figure 3-4 shows that the average number of iterations required to retrieve a message is relatively constant for all rules up to 140000 stored messages. Beyond this, the number of iterations required for the GLsKO and the GWsTA rules with $\gamma = 1$ begins to increase rapidly. It is worth emphasizing that the maximum number of iterations we allowed for the GWsTA rule is 30 so the constant level reached by the curve representing this rule with $\gamma = 1$ is just a result of that constraint. However, the number of iterations for the GWsTA rule with $\gamma = 1000$ increases only slightly

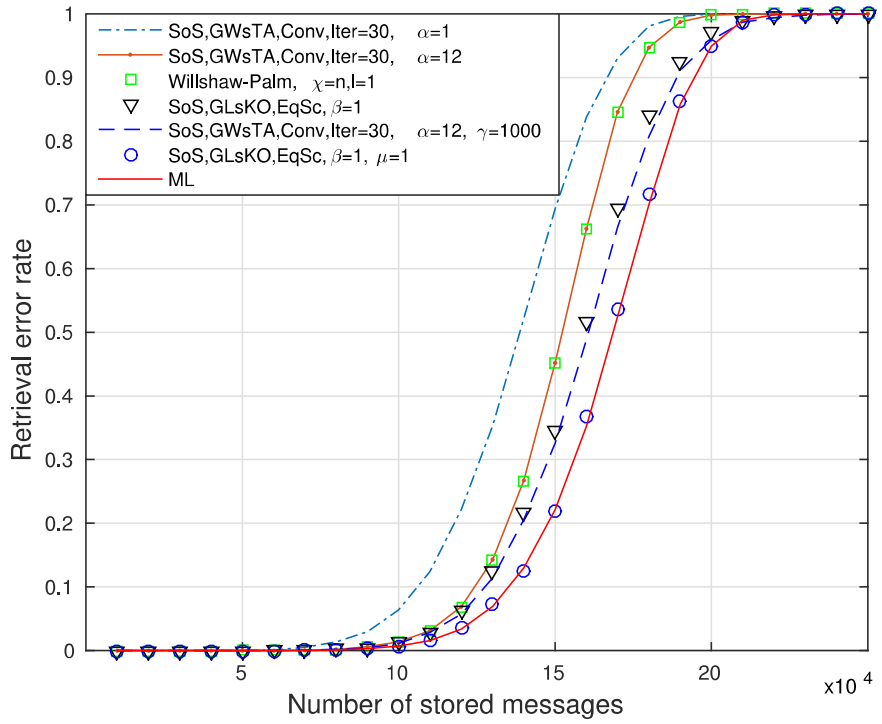


Figure 3-3: Influence of activation rules on retrieval error rates in a network with $\chi = 100$, $\ell = 64$, $c = 12$, $\gamma = 1$ if not stated otherwise, $\sigma_{ij} = 0$ initially with 3 segments of partial erasure in input messages.

approaching an average of 3.3 for 250000 stored messages.

The reason for this explosion of the number of iterations in the case of the GLsKO rule is that the number of units activated after the first phase increases with the number of stored messages. So more iterations would then be needed in order to exclude losers and thus shrink the set of active units.

In the case of the GWsTA rule with $\gamma = 1$, all units in the network are concerned with the search for a message in each iteration. So when the number of stored messages increases, the connection density in the network gets higher and it would be more likely that new winners appear at each iteration violating the Conv criterion. Setting γ to 1000 limits the possibility of the apparition of new winners at each iteration and decreases the number of iterations needed before convergence.

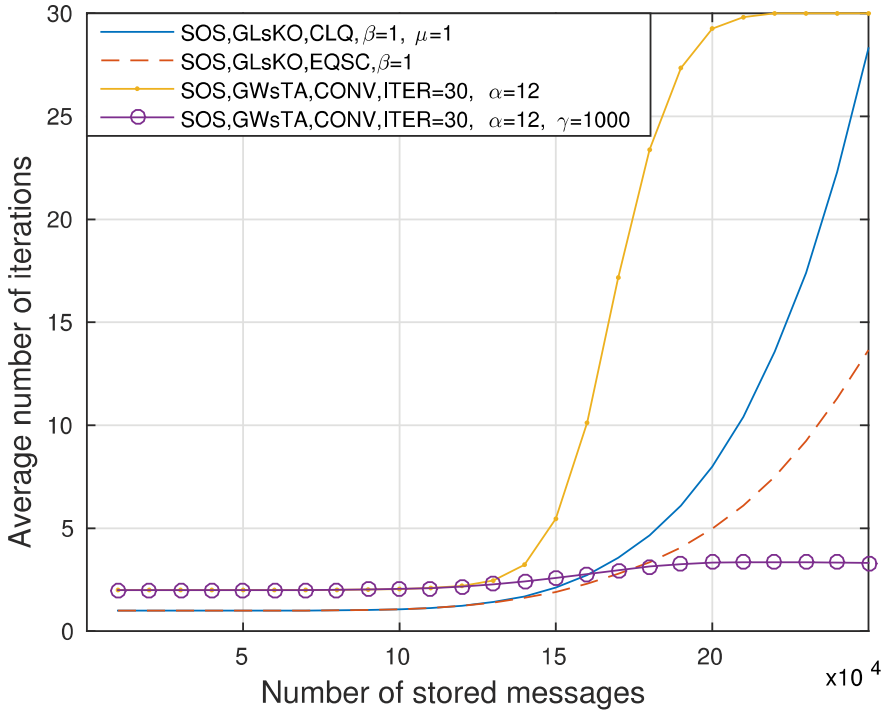


Figure 3-4: Average number of iterations for different scenarios in a network with $\chi = 100$, $\ell = 64$, $c = 12$, $\gamma = 1$ if not stated otherwise, $\sigma_{ij} = 0$ initially with 3 segments of partial erasure in input messages.A stopping criterion Iter with a maximum of 30 iterations is imposed on the GWsTA rule with $\gamma = 1$.

3.9 Conclusion and future work

In this chapter, we focused on the memory stage of the information processing and representation pipeline. We presented the Sparse Clustered Network originally introduced in(Gripon and Berrou, 2011) and (Aliabadi et al., 2014), and we proposed a generic formulation of its several retrieval algorithms which is meant to faciliate the process of designing new algorithms. We also proposed an improvement of the retrieval performance of SCNs by enhancing the GLsKO activation rule.

We found that our modified version of the GLsKO activation rule combined with the equal scores or the clique stopping criteria gives the best results in terms of the retrieval error rate but with a rapidly increasing number of iterations. Actually, the second phase of the GLsKO rule along with the clique criterion can be viewed as an operation equivalent to searching the maximum clique among active units. This is a

famous NP-complete problem. However, many suboptimal solutions were suggested for this problem (or equivalently, the minimum vertex cover problem) such as (Xu and Ma, 2006; Geng et al., 2007) and many more. We believe that such suboptimal solutions are adaptable to our problem and can be integrated in our retrieval algorithm in the future in order to give a better performance with a more reasonable number of iterations.

Chapter 4

Sparse clustered networks for solving the feature correspondence problem

In this chapter, we address the intermediate stage of the visual processing pipeline. As we argued in section 1, this conceptual stage is situated between the lowest (visual acquisition stage) and the highest (memory representation) stage. More precisely, we address the feature correspondence problem. Finding correspondences between image features is a fundamental question in computer vision. Many models in literature have proposed to view this as a graph matching problem whose solution can be approximated using optimization principles. In this chapter, we propose a different treatment of this problem from a neural network perspective. We present a new model for matching features inspired by the architecture of a recently introduced neural network. We show that by using common neural network principles like max-pooling, k-winners-take-all and iterative processing, we obtain a better performance at matching features in cluttered environments. The proposed solution is accompanied by experimental evaluations on a synthetic dataset as well as on natural images. It is also compared to state-of-the-art matching models.

4.1 Introduction

The problem of finding correspondences between features of two images is fundamental to computer vision. Solving this problem would be of particular importance to a variety of vision tasks. This includes object tracking (Jiang et al., 2011), object recognition (Grauman and Darrell, 2005), stereo matching (Tuytelaars and Gool, 2000), object discovery (Leordeanu and Collins, 2005), structure from motion (Rothganger et al., 2007), and a variety of other tasks.

The basic idea is simple: given two images m and m', where m contains only one object b (the query object), we are interested in finding a possibly deformed instance b' of b in the image m', knowing that m' might contain other objects than the one in question. In order to achieve that, we take two sets of local image features $\mathcal{V}$ and $\mathcal{V}'$ representing m and m', respectively. Then, we search a mapping from $\mathcal{V}$ to $\mathcal{V}'$ that is injective.

An early class of algorithms consisted in matching features based on the similarity between their descriptor vectors. Such similarity can be obtained using simple metrics such as euclidean or hamming distances (Szeliski, 2010). While such methods are still widely popular, their ability to find correct matches is limited in more complex situations such as in the presence of multiple instances of the object b' in the destination image m', or in the case of matching two different objects that belong to the same class, e.g., matching faces of two different persons, or in the presence of clutter.

Considering geometrical consistency between features in addition to their descriptor similarity was suggested as a better way to achieve correct matching. For instance, in early methods such as RANSAC (Fischler and Bolles, 1981) and ICP (Besl and McKay, 1992), a solution is accepted only if the matched features in $\mathcal{V}'$ are constrained to some parametric transformation (e.g. epipolar or affine) of their counterparts in $\mathcal{V}$. However, given that non-rigid transformations are very common in natural images, applying these parametric constraints becomes a limitation in such cases.

In order to take both feature similarity and their geometrical proximity into account, including the case of non-rigid deformations, a class of methods were proposed

in the last two decades that formulated feature matching as a graph matching (GM) problem (Zhou and la Torre, 2012),(Leordeanu and Hebert, 2005),(Cho et al., 2010). Two graphs $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$ are constructed on the sets of features $\mathcal{V}$ and $\mathcal{V}'$ representing the graph nodes. Graph edges in $\mathcal{E}$ and $\mathcal{E}'$ are assigned values of some measure of geometrical proximity between pairs of nodes in $\mathcal{V}$ and $\mathcal{V}'$, respectively. Then we search the sub-graph of $\mathcal{G}'$ that best matches $\mathcal{G}$ in terms of unary feature similarity and pairwise geometric consistency.

This graph matching problem constrained to an injective mapping from $\mathcal{V}$ to $\mathcal{V}'$ is known to be NP-hard. A whole class of methods proposed to approach it as a Quadratic Assignment Problem (QAP)(Zass and Shashua, 2008; Duchenne et al., 2011; Zhou and la Torre, 2012), where an approximate solution can be obtained by optimizing a well-defined objective function. Some of these methods suggested an iterative approach to optimizing this objective function such as the max-pooling matching (MPM) (Cho et al., 2014), spectral matching (SM) (Leordeanu and Hebert, 2005), re-weighted random walks (RRWM) (Cho et al., 2010) or balanced graph matching (BGM) (Cour et al., 2007).

Little work, however, was devoted to seeking a potential neural network model for solving the graph matching problem. We think that this is an interesting question from an algorithmic point of view, as well as for researchers interested in Marr's third level of analysis that seeks possible neural mechanisms for implementing vision algorithms (Marr, 1982a). While the present chapter addresses this level of analysis, we do not pretend providing a real bio-mimetic solution.

The main contribution we present in this chapter is to introduce an artificial neural network (ANN) model for addressing the correspondence problem. This model is adapted from the SCN model recently introduced by Gripon and Berrou in (Gripon and Berrou, 2011), which is a generalization of the Palm-Willshaw neural network (Schwenker et al., 1996). The model we present implements a cooperative algorithm, meaning that each neuron needs only to know about the activity of a few neighboring neurons, which allows for the algorithm to be run in parallel.

This model implements an iterative process and provides a better matching accu-

racy of features in cluttered images. It enforces the injective mapping constraint at each iteration. Actually, the injective mapping constraint from $\mathcal{V}$ to $\mathcal{V}'$ implies two different constraints: (1) A feature $v_i \in \mathcal{V}$ is allowed to match at most one feature in $\mathcal{V}'$ (by the definition of a mapping). (2) A feature $v'_a \in \mathcal{V}'$ is allowed to match at most one feature in $\mathcal{V}$ (injectivity constraint). Unlike conventional algorithms, we neither relax these constraints nor we enforce them both at the same time. Each iteration of the algorithm we propose enforces one of these constraints at a time. It alternates between them at each iteration until a good match is obtained.

The rest of this chapter is organized in five sections. In section 4.2, a brief overview of state-of-the-art matching algorithms is presented. In section 4.3, a formal definition of the correspondence problem is provided. Then, the architecture of the neural network along with the algorithm we propose are presented in section 4.4. The performance of the proposed model is evaluated in section 4.5 and compared to some other algorithms. Section 4.6 is the chapter conclusion.

4.2 Related work

As mentioned in section 4.1, feature correspondence can be viewed as a GM problem, which is traditionally formulated as a quadratic assignment problem (QAP) known to be NP-hard. Its solution is usually approximated by optimizing an objective function with relaxed constraints (Zhou and la Torre, 2012),(Zass and Shashua, 2008),(Leordeanu and Hebert, 2005). There were also some attempts to approximate this optimization procedure by applying an iterative process without defining an explicit objective to optimize (Cho et al., 2014),(Cho et al., 2010),(Cour et al., 2007),(Gold and Rangarajan, 1996). Iterative approaches to matching problems date back to as early as Marr's cooperative algorithm for stereo matching (Marr, 1982a). It provided an insight on how iterative algorithms can be used to tackle difficult vision tasks using only local image information.

Max-pooling matching (MPM) introduced by Cho *et al.* in (Cho et al., 2014) is one recent example of such iterative algorithms. It applies max-pooling to preserve

important information while discarding irrelevant details making it more robust in the presence of outliers.

Some other methods that use a similar iterative approach include re-weighted random walk matching (RRWM) (Cho et al., 2010). This model uses the principle of random walks on the associative graph, where the matching contraints are enforced at each step. Other examples include the Integer Projected Fixed Point (IPFP) method (Leordeanu et al., 2009), Spectral Matching (SM) (Leordeanu and Hebert, 2005), balanced graph matching (Cour et al., 2007) and more (Gold and Rangarajan, 1996).

Our approach is similar to MPM in that it applies max-pooling to discard irrelative details. Unlike MPM, pooling is not only applied among features of one image but also in the second one. Another major difference is that the final discretization step is replaced by a non-linear activation function applied at each iteration and a winner-take-all (WTA) applied at the end, which is akin to local inhibition observed among neural assemblies (Mountcastle, 1997).

In the following sections, we provide the formal definition of the feature correspondence problem as a (GM) problem. Then, we describe our ANN model and specify the details of the matching algorithm it implements. We use a similar terminology as in (Cho et al., 2014) in order to highlight the similarities and differences between the two algorithms, and to show where the proposed model is positioned relative to the state-of-the-art.

4.3 Problem statement

In this section we present the classical formalism of the correspondence problem as a GM problem, then we relate it to the error-correcting codes theory. Establishing this relation will be useful in the next section, where we use turbo decoding principles introduced in (Berrou and Glavieux, 1996) to explain the details of the proposed model.

4.3.1 Formalism

We follow the graph matching approach (GM) to the correspondence problem. The objective is to match a query graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, to a sub-graph of $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$. We define an assignment matrix $\mathbf{X} \in \{0,1\}^{nn'}$ as in (Leordeanu and Hebert, 2005), where $n = |\mathcal{V}|$ and $n' = |\mathcal{V}'|$. Elements of $\mathbf{X}$ are set as follows:

$$\mathbf{X}_{ia} = \begin{cases} 1 & \text{if feature } v_i \text{ matches } v'_a, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

We also use an assignment vector $\mathbf{x}$, which is a column-wise vectorized copy of $\mathbf{X}$. We define a unary affinity function $S_V(v_i, v'_a)$ to measure the similarity between two feature descriptors, and a pairwise affinity function $S_E(e_{ij}, e'_{ab})$ that measures similarity between two edges $e_{ij} \in \mathcal{E}$ and $e'_{ab} \in \mathcal{E}'$. We use these functions to populate a unary affinity vector as $\mathbf{y}_{ia} = S_V(v_i, v'_a)$, and a pairwise affinity matrix $\mathbf{A} \in \mathbb{R}^{nn' \times nn'}$:

$$\mathbf{A}_{ia;jb} = \begin{cases} S_E(e_{ij}, e'_{ab}) & \text{if } i \neq j \text{ and } a \neq b, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

An objective function is defined using the above affinity functions:

$$f(\mathbf{x}) = \sum_{\substack{\mathbf{x}_{ia}=1 \\ \mathbf{x}_{jb}=1}} S_E(e_{ij}, e'_{ab}) + \sum_{\mathbf{x}_{ia}=1} S_V(v_i, v'_a). \tag{4.3}$$

This is a known quadratic assignment problem (QAP) that can be written in matrix form as:

$$f(\mathbf{x}) = \mathbf{x}^\intercal(\mathbf{A} + diag(\mathbf{y}))\mathbf{x}, \tag{4.4}$$

where $diag(\mathbf{y})$ is a square matrix that contains zeros everywhere except on its main diagonal where it holds the vector $\mathbf{y}$. The solution to this problem can be expressed

as the assignment vector $\mathbf{x}^*$ that maximizes the objective function $f(\mathbf{x})$:

$$\widetilde{\mathbf{x}}^* = {}_{\widetilde{\mathbf{x}}}\widetilde{\mathbf{x}}^{\intercal}(\mathbf{A} + diag(\mathbf{y}))\widetilde{\mathbf{x}}, \tag{4.5}$$

$$\mathbf{x}^* = z(\widetilde{\mathbf{x}}^*), \tag{4.6}$$

$$s.t.\ \mathbf{x}^* \in \{0,1\}^{nn'}, \widetilde{\mathbf{x}} \in \mathbb{R}^{nn'},$$

and $\mathbf{x}^*$ represents an injective mapping

from $\mathcal{V}$ to $\mathcal{V}'$.

Notice that the constraint on $\mathbf{x}$ being discrete is relaxed during the optimization process. This relaxed version of the assignment vector is denoted $\widetilde{\mathbf{x}}$. Notice also that the objective function does not enforce the injective mapping constraint from $\mathcal{V}$ to $\mathcal{V}'$ we are seeking. This constraint is usually relaxed during the optimization procedure to reduce the complexity of the problem.

The final continuous assignment vector $\widetilde{\mathbf{x}}^*$ obtained is then discretized in (4.6) using the function $z(.)$ that usually applies a greedy or a Hungarian algorithm enforcing injective mapping and the discrete-value constraints (Leordeanu and Hebert, 2005; Leordeanu et al., 2009; Cho et al., 2014).

The algorithm we propose follows a different procedure; while $\mathbf{x}$ is allowed to be continuous during the process, the injective mapping constraint is not totally relaxed during optimization; they are enforced at each iteration, alternating between the mapping constraint and the injectivity one, until a satisfying solution is obtained.

4.3.2 Relation to coding theory

Our proposed solution is inspired by the functionning of turbo codes, a state-of-the-art class of error correcting codes. In this subsection we elaborate on this analogy by explaining how GM can be likened to an error correcting problem.

One way to relate the correspondence problem to a coding/decoding procedure is illustrated in figure 4-1. In this configuration, the query graph $\mathcal{G}$ is treated as the transmitted codeword, and the destination graph $\mathcal{G}'$ as the observation, which is

viewed as a corrupted version of $\mathcal{G}$ due to a noisy transmission channel.

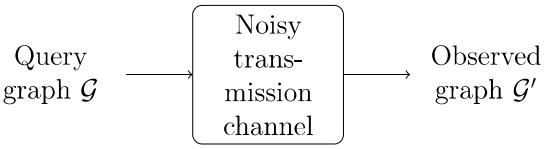


Figure 4-1: Feature matching viewed as transmission problem.

The noise in the transmission channel is due to three different factors:

- Spatial deformation of feature locations in $\mathcal{V}'$ compared to their counterparts in $\mathcal{V}$ due to all kinds of rigid and non-rigid object transformations.
- The intrinsic ambiguity of the problem in some cases, where more than one matching solution might be possible. One good example is in the case of matching features having an equilateral triangular configuration in each image, with a pairwise affinity function $S_E(.)$ that only considers relative positions of features. In this case, each feature in $\mathcal{V}$ can match any feature in $\mathcal{V}'$.
- The presence of outliers (clutter) which are features that do not belong to the objects we are trying to match.

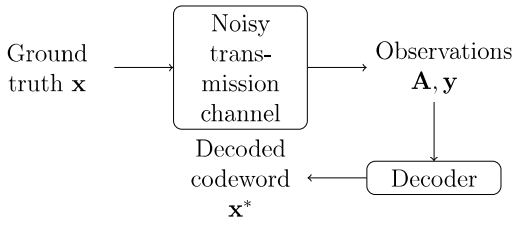


Figure 4-2: The matching problem viewed as an error correcting problem of a codeword received through a noisy transmission channel.

However, since we are seeking to find a match among graph nodes rather than to recover the graph $\mathcal{G}$ from the observation, a better way to build the transmission network is to take a ground truth assignment vector $\mathbf{x}$ as the transmitted codeword. The pairwise affinity matrix $\mathbf{A}$ and the unary affinity vector $\mathbf{y}$ are the observed variables as depicted in figure 4-2. Our objective is then to decode our observations in order to get the vector $\mathbf{x}^*$ belonging to the constrained domain $\{0,1\}^{nn'}$:

We are particularly interested in what happens inside the decoder in figure 4-2. In the next section, we will present our proposed matching model as a neural network that will play the decoder role. The most common method in literature is to apply an optimization procedure to find $\widetilde{\mathbf{x}}^*$ as in (4.5). Then discretization is applied on that vector as in (4.6). We show how our solution differs from this classical approach, and how it can be viewed as a process inspired by the turbo decoding concept.

4.4 Methodology

In this section we introduce the matching model we propose, which is based on the architecture of a neural network (ANN) recently proposed in (Gripon and Berrou, 2011). We call this ANN the sparse clustered network (SCN). We also make a parallel between the turbo-decoding principle and the proposed model in order to illustrate its function in greater detail.

Here is a reminder of the notations we shall be using throughout this section to refer to signals manipulated and produced by the matching process:

- The pairwise affinity matrix $\mathbf{A}$.
- The unary affinity vector $\mathbf{y}$.
- Relaxed assignment vectors $\widetilde{\mathbf{x}} \in \mathbb{R}^{nn'}$. These vectors are called relaxed because they do not respect the injective mapping constraint.
- Semi-relaxed assignment vectors $\bar{\mathbf{x}} \in [0,1]^{nn'}$. They are semi-relaxed because they partially enforce one of the two constraints; injectivity or mapping at each time. These vectors are sparse; most of their elements are set to zero.
- The final assignment vector $\mathbf{x}^*$. The assignment described by this vector respects both the mapping and the injectivity constraints.

4.4.1 The neural network model

The neural network we propose for solving the correspondence problem is constructed on the graph captured by the affinity matrix **A**, as in the example of Fig. 4-4. The architecture of this network is adapted from the SCN (Gripon and Berrou, 2011) which was proposed by Gripon and Berrou as a generalization of Palm-Willshaw networks (Schwenker et al., 1996) using error correcting principles.

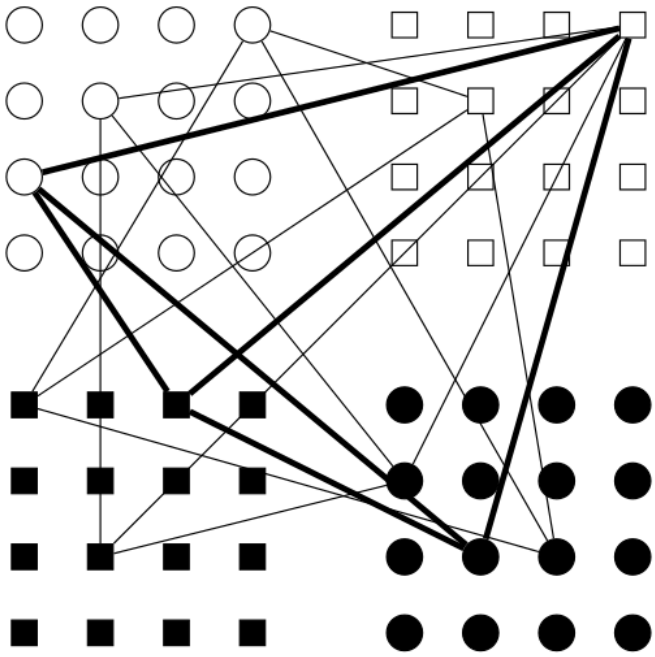


Figure 4-3: Architecture of the sparse clustered network (SCN) as originally proposed in (Gripon and Berrou, 2011). Units are grouped into clusters. Only one unit per cluster can be used to store a message.

As we saw in chapter 3, an SCN is an ANN consisting of a set of discrete units. These units are organized into groups called clusters. Within each cluster, a WTA constraint is imposed onto units during the network activity. SCNs were originally used as associative memories that can store and retrieve patterns called messages. In order to store a message, only one unit is selected within each cluster. A fully connected graph or a 'clique' is then created on the selected units. This clique represents one stored message. Several messages can be stored in the network following the same procedure as depicted in figure 4-3.

The network grid structure depicted in Fig. 4-4 corresponds to the 2D configuration of the assignment matrix $\mathbf{X}$. As in SCNs, we impose a grouping configuration on the network neurons in the form of clusters; neurons of the same row are grouped into one cluster, and the same holds for neurons of the same column. Thus, each neuron belongs to two clusters as shown in Fig. 4-4. Within each cluster, a WTA activation constraint is imposed; at most one neuron per cluster can be active at the end of the matching process with a binary activation level (0 or 1) captured by $\mathbf{X}$ as in (Gripon and Berrou, 2011). However, during the network activity, and before neurons reach their final state, this constraint is relaxed into a k-winners-take-all (kWTA) constraint, and we allow neurons to temporarily have continuous activation values. The connections between neurons are captured by the pairwise affinity matrix $\mathbf{A}$, and as we notice from (4.2), no connections exists between neurons of the same cluster ($\mathbf{A}_{ia;jb} = 0$) as in SCNs.

The WTA and kWTA constraints we impose within clusters are meant to encourage the one-to-one matching constraint between features in $\mathcal{V}$ and $\mathcal{V}'$. From a biological perspective, this is akin to the local competition among neural assemblies enforced by short inhibitory synaptic connections (Mountcastle, 1997).

The network activity starts by assigning to each neuron its unary affinity value ($\bar{\mathbf{x}}_{ia} \leftarrow \mathbf{y}_{ia}$). Then, within each row cluster, every neuron receives the max-pooled propagated activity of all other neurons in the network to which it connects as in (Cho et al., 2014) and (Aboudib et al., 2014):

$$\widetilde{\mathbf{x}}_{ia} \leftarrow \bar{\mathbf{x}}_{ia} \sum_{j \in \mathcal{V}} \max_{b \in \mathcal{V}'} \bar{\mathbf{x}}_{jb} \mathbf{A}_{ia;jb}. \tag{4.7}$$

The activity values within this cluster are then normalized to their maximum, and a kWTA operation is applied in which only a few neurons per row cluster are kept active. At this point, the matching process proceeds to the next iteration which is similar to the first one except that max-pooling and kWTA are applied on column clusters. We alternate between row-wise and column-wise iterations until the convergence of $\bar{\mathbf{x}}$ or until a fixed maximum number of iterations is attained. This is

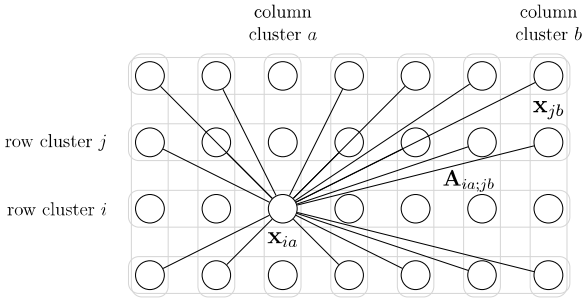


Figure 4-4: The architecture of the proposed neural network for graph matching.

akin to the process of alternating codewords between two decoding units in a turbo decoder (Berrou and Glavieux, 1996). Notice that for row clusters, max-pooling and kWTA are applied row-wise, while they are applied column-wise for column clusters.

Finally, after the last iteration, only neurons with a maximal activation value ($\mathbf{x}^*_{ia} = 1$) are kept active while others are deactivated ($\mathbf{x}^*_{ia} \leftarrow 0$). A WTA operation is then applied within every row and column cluster; if more than one neuron is active in a given cluster, they are all deactivated and no winner is declared. This is equivalent to imposing an 'at most' one-to-one matching constraint from $\mathcal{V}$ to $\mathcal{V}'$.

To sum up, the network behavior consists in each neuron adding up its input signals, which are the max-pooled weighted activities of other neurons. Then, a non-linear activation function is applied to this neuron, taking into account the activity level of other members of its cluster. This is akin to the classic accumulate-and-fire neuron model of McCulloch-Pitts (McCulloch and Pitts, 1943). In the next subsection, the proposed matching algorithm will be presented as a decoding process in order to help explain each step in more detail.

4.4.2 Matching as a decoding process

The architecture of the decoding process we propose is depicted in figure 4-5. This process is implemented by the neural network architecture we introduced in the previous subsection. As shown in the figure, there are four main units that operate in an iterative fashion, and one unit, the WTA unit, applied only one time at the end.

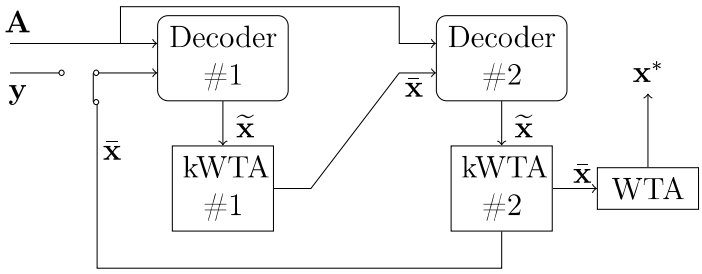


Figure 4-5: The architecture of the proposed decoder.

Decoder units

Each decoder unit takes two inputs: the observation $\mathbf{A}$ and either the unary affinity vector $\mathbf{y}$, or a semi-relaxed assignment vector $\bar{\mathbf{x}}$. The vector $\mathbf{y}$ is only taken by the first decoder in the first iteration. In all subsequent iterations, the vector $\bar{\mathbf{x}}$ is used instead. The output of each decoder unit is a relaxed assignment vector $\widetilde{\mathbf{x}}$. This vector is computed as a max-pooled weighted sum of elements in $\mathbf{A}$ as in (4.7). This equation is applied by the first decoder. Notice that pooling is applied on elements in $\mathcal{V}'$ (row-wise pooling) as in (Cho et al., 2014). The second decoder applies max-pooling on elements in $\mathcal{V}$ (column-wise pooling):

$$\widetilde{\mathbf{x}}_{ia} \leftarrow \bar{\mathbf{x}}_{ia} \sum_{b \in \mathcal{V}'} \max_{j \in \mathcal{V}} \bar{\mathbf{x}}_{jb} \mathbf{A}_{ia;jb}. \tag{4.8}$$

The operation applied by each of these decoders is akin to the power method used in spectral matching (SM) (Leordeanu and Hebert, 2005) to find the first eigen vector of matrix $\mathbf{A}$. Max-pooling is added to discard irrelevant details while preserving necessary information as in (Cho et al., 2014).

kWTA units

Each k-winner take all (kWTA) unit takes a relaxed assignment vector $\widetilde{\mathbf{x}}$ as its input, and produces a semi-relaxed assignment vector $\bar{\mathbf{x}}$ as an output. The first kWTA unit is only concerned about the mapping constraint. It 'encourages' the vector $\widetilde{\mathbf{x}}$ to respect that constraint without strictly enforcing it. In other words, it reduces the number of matches in $v_a \in \mathcal{V}'$ that a single feature $v_i \in \mathcal{V}$ can take. This is done by

applying a kWTA operation as follows:

$$\widetilde{\mathbf{x}}_{ia} \longleftarrow \frac{\widetilde{\mathbf{x}}_{ia}}{\max_{a \in \mathcal{V}'} \widetilde{\mathbf{x}}_{ia}}, \tag{4.9}$$

$$\bar{\mathbf{x}}_{ia} \leftarrow \widetilde{\mathbf{x}}_{ia} h(\widetilde{\mathbf{x}}_{ia} - \tau), \tag{4.10}$$

$$\forall i \in \mathcal{V}, a \in \mathcal{V}',$$

where $h(.)$ is the unit step function and τ is the kWTA activation threshold.

The second kWTA units applies a similar operation for the injectivity constraint to reduce the number of features in $\mathcal{V}$ mapped to a single feature in $\mathcal{V}'$:

$$\widetilde{\mathbf{x}}_{ia} \longleftarrow \frac{\widetilde{\mathbf{x}}_{ia}}{\max_{i \in \mathcal{V}} \widetilde{\mathbf{x}}_{ia}}, \tag{4.11}$$

$$\bar{\mathbf{x}}_{ia} \leftarrow \widetilde{\mathbf{x}}_{ia} h(\widetilde{\mathbf{x}}_{ia} - \tau), \tag{4.12}$$

$$\forall i \in \mathcal{V}, a \in \mathcal{V}'.$$

Notice that the max function in (4.11) is applied across elements of $\mathcal{V}$ (column clusters), while in (4.9), it is applied across elements of $\mathcal{V}'$ (row clusters). The output $\bar{\mathbf{x}}$ of the second kWTA unit is then used either as an input to the first decoder unit, or as an input to the WTA unit after the last iteration. This iterative process stops when the vector $\bar{\mathbf{x}}$ converges. However, since a theoretical guarantee of convergence is yet to be proved, we typically fix a maximum number of allowed iterations beyond which the process terminates.

WTA unit

The winner-takes-all (WTA) unit takes a semi-relaxed assignment vector $\bar{\mathbf{x}}$ as an input and produces the final assignment vector $\mathbf{x}^* \in \{0,1\}^{nn'}$, which respects the injectivity mapping constraint. The first step is to zero all values in $\bar{\mathbf{x}}$ that do not

equal one, which is the maximal values in $\bar{\mathbf{x}}$:

$$\mathbf{x}^*_{ia} \leftarrow \delta_1^{\bar{\mathbf{x}}_{ia}}, \quad \forall i \in \mathcal{V}, a \in \mathcal{V}', \tag{4.13}$$

where δ is the Kronecker delta. After that, each non-zero value $\bar{\mathbf{x}}_{ia}$ is set to zero if there exists at least one non-zero value of the form $\bar{\mathbf{x}}_{ik}$ or $\bar{\mathbf{x}}_{ka}$ different from $\bar{\mathbf{x}}_{ia}$. By applying this procedure, the resulting assignment vector $\bar{\mathbf{x}}$ is guaranteed to respect the injective mapping constraint. The complete matching process we propose is described in algorithm (3).

Algorithm 3: Proposed matching algorithm.

input : Pairwise affinity matrix $\mathbf{A}$, Unary similarity vector $\mathbf{y}$
output: Assignment vector $\mathbf{x}$

$\bar{\mathbf{x}} \longleftarrow \mathbf{y}$
repeat
  foreach $i \in \mathcal{V}$ **do**
    foreach $a \in \mathcal{V}'$ **do**
      $\widetilde{\mathbf{x}}_{ia} \leftarrow \bar{\mathbf{x}}_{ia} \sum_{j \in \mathcal{V}} \max_{b \in \mathcal{V}'} \bar{\mathbf{x}}_{jb} \mathbf{A}_{ia;jb}$
    end
    $\widetilde{\mathbf{x}}_{ia} \longleftarrow \frac{\widetilde{\mathbf{x}}_{ia}}{\max_{a \in \mathcal{V}'} \widetilde{\mathbf{x}}_{ia}} : a \in \mathcal{V}'$
    $\bar{\mathbf{x}}_{ia} \longleftarrow \widetilde{\mathbf{x}}_{ia} h(\widetilde{\mathbf{x}}_{ia} - \tau) : a \in \mathcal{V}'$
  end

  foreach $a \in \mathcal{V}'$ **do**
    foreach $i \in \mathcal{V}$ **do**
      $\widetilde{\mathbf{x}}_{ia} \leftarrow \bar{\mathbf{x}}_{ia} \sum_{b \in \mathcal{V}'} \max_{j \in \mathcal{V}} \bar{\mathbf{x}}_{jb} \mathbf{A}_{ia;jb}$
    end
    $\widetilde{\mathbf{x}}_{ia} \longleftarrow \frac{\widetilde{\mathbf{x}}_{ia}}{\max_{i \in \mathcal{V}} \widetilde{\mathbf{x}}_{ia}} : i \in \mathcal{V}$
    $\bar{\mathbf{x}}_{ia} \longleftarrow \widetilde{\mathbf{x}}_{ia} h(\widetilde{\mathbf{x}}_{ia} - \tau) : i \in \mathcal{V}$
  end
until $\bar{\mathbf{x}}$ *converges OR last iteration attained*

$\mathbf{x}^*_{ia} \leftarrow \delta_1^{\bar{\mathbf{x}}_{ia}} : i \in \mathcal{V}$ and $a \in \mathcal{V}'$ $\quad$ #δ is the Kronecker delta.

WTA: Zero all rows and columns in $\mathbf{x}^*$ with more than one non-zero element.

$\mathbf{x} \leftarrow \mathbf{x}^*$

4.5 Experimental evaluation

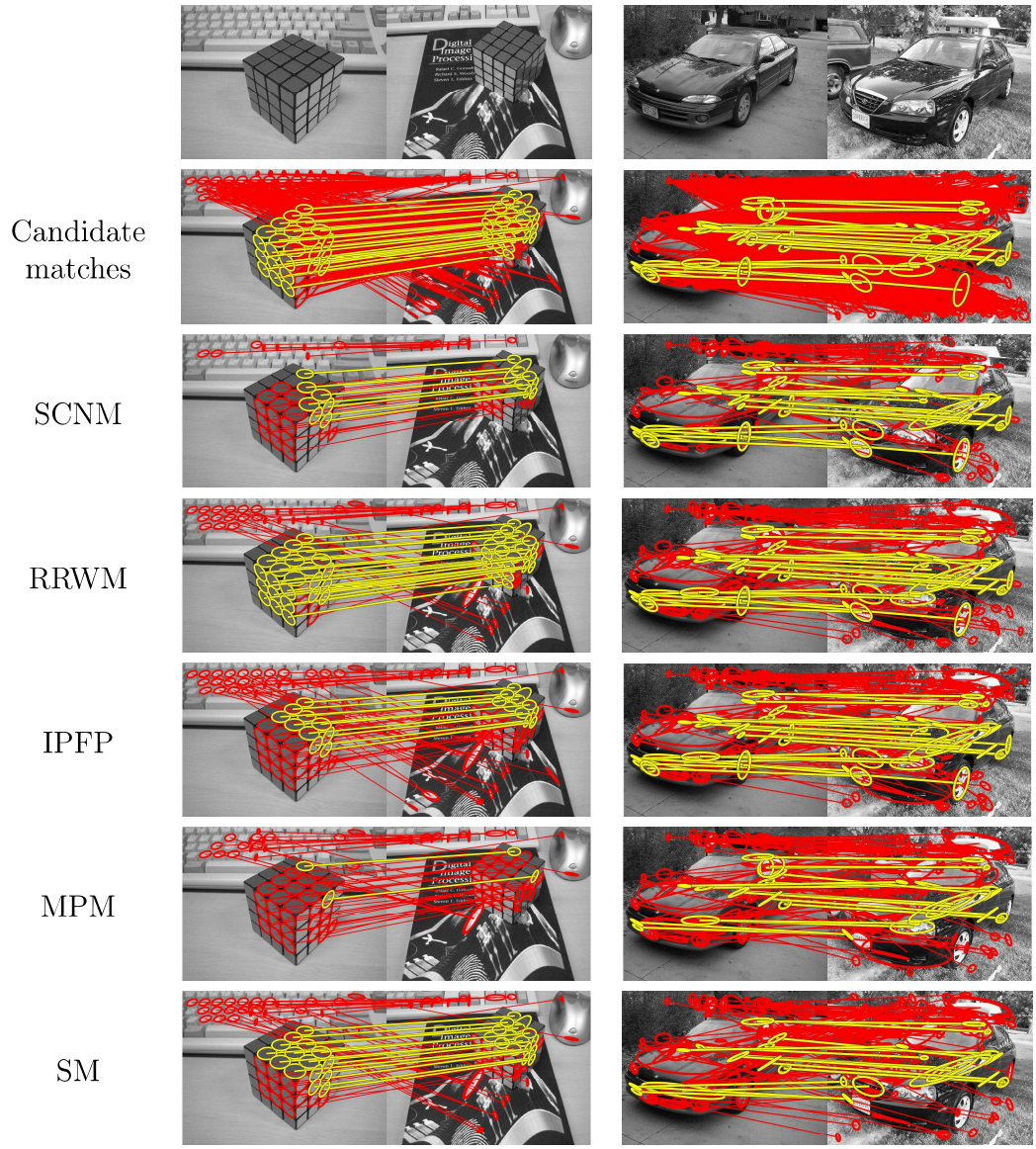


Figure 4-6: Some examples of matching features between image pairs obtained by different algorithms. True matches are shown in yellow, while false ones in red.

In this section, we present a performance comparison between the proposed matching model and several other state-of-the-art models. We perform this evaluation on two types of benchmarks. The first one is synthetic, where matching is applied on data points generated manually, while the other experiment is carried out in a more realistic setting in which natural images are used.

4.5.1 Synthetic point matching

A typical evaluation method used in feature matching literature is accomplished using synthetic datasets. We create two sets P and P' containing points in $\mathbb{R}^2$. Graphs $\mathcal{G}$ and $\mathcal{G}'$ are created using P and P', respectively. Each set contains two types of points: inliers and outliers. Inliers are points representing features that we are seeking to match. Outliers, on the other hand, are points that represent features that describe clutter or noise that we wish to ignore during the matching process.

We randomly generate n_{in} inliers with coordinates sampled uniformly from the interval $[-1, +1]$, and we add them to P. We then add a Gaussian noise $\mathcal{N}(0, \sigma^2)$ to each of these inliers before adding them to the set P'. After that, we add n_{out} outliers generated from the same distribution as the inliers to each of P and P'.

The unary affinity function is considered to be always constant $S_V(p_i, p'_a) = 1$, while the pairwise affinity function is defined as follows:

$$S_E(e_{ij}, e'_{ab}) = \exp(-\big|\|p_i - p_j\| - \|p'_a - p'_b\|\big|). \tag{4.14}$$

Using a constant $S_V(.)$ represents a difficult case where matching depends only on the geometrical consistency of features. We set the kWTA threshold $\tau = 0.98$, which we found to give the best matching performance. Convergence of the algorithm is attained after $4 - 6$ iterations in most cases. Therefore, the maximum number of allowed iterations is set to 6.

We first evaluate the performance of the proposed model in the presence of outliers. We refer to the proposed model by the term SCNM standing for Sparse Clustered Network Matching. In all of our experiments, performance is measured in terms of the average *Recall* score, which is the percentage of the number of correct matches to the total number of inliers.

In the first experiment, we fix the number of inliers to $n_{in} = 15$ in both sets $\mathcal{V}$ and $\mathcal{V}'$, the standard deviation of the Gaussian noise to $\sigma = 0.04$, and we vary the number of outliers in both sets as shown in figure 4-7. The *Recall* score of the proposed model is then compared to some state-of-the-art matching algorithms

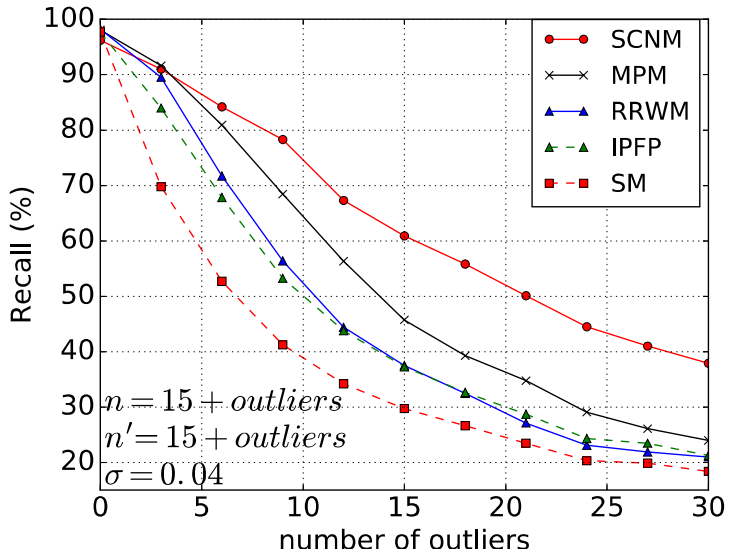
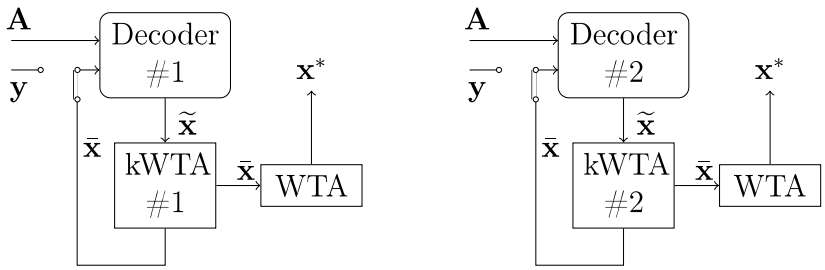


Figure 4-7: A comparison among models' performance in the presence of outliers. The number of outliers is varied for a fixed value of σ. The same number of outliers shown on the horizontal axis is added to both sets $\mathcal{V}$ and $\mathcal{V}'$.

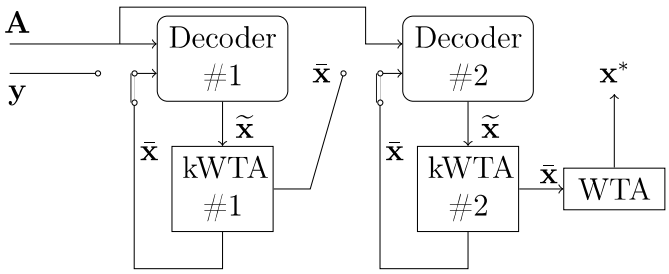
including MPM (Cho et al., 2014), RRWM (Cho et al., 2010), IPFP (Leordeanu et al., 2009) and SM (Leordeanu and Hebert, 2005). We notice that the score of the SCNM model surpasses state-of-the-art by a significant margin, even when the number of outliers is twice the number of inliers. This robustness to outliers is a very interesting property since outliers in the form of clutter and noise are omnipresent in natural images.

In a second experiment, we evaluate the performance gain obtained by the alternating double-decoder scheme of figure 4-5. In other words, we try to answer the question of whether alternating between decoders is behind the performance gain we observe, or there exists other configurations that give a comparable performance. In order to do that, we compare the turbo matcher with three alternate configurations illustrated in figure 4-8: (1) in the absence of decoder#1 and kWTA#1 or the SCNM-c configuration (2) in the absence of decoder#2 and kWTA#2 or SCNM-r (3) without alternation between the two constraints or SCNM-sep. In the latter test case, two separate iterative phases are run consecutively. The first one includes decoder#1 and kWTA#1 which 'encourages' the mapping constraint. The second phase includes only



(a) SCNM-r

(b) SCNM-c



(c) SCNM-sep

Figure 4-8: Other possible configurations of decoder and kWTA units. In (a) and (b), only one decoder and one kWTA units are used. In (c), both decoder units and kWTA units are used, but no alternation between constraints is involved.

decoder#2 and kWTA#2, and 'encourages' the injectivity constraint.

Figure 4-9 shows that enforcing both the mapping and the injectivity constraints, whether in an alternating or a non-alternating fashion gives a better score than using only one decoder unit with its associated kWTA unit. However, turbo-style alternating between decoder units gives a better score than enforcing constraints separately without alternation.

The final experiment consists in fixing the number of inliers to 30 with no outliers. The parameter σ is then varied. The *Recall* score of the SCNM model is evaluated, and compared to state-of-the-art for each value of σ. We notice in figure 4-11 that the proposed matcher gives a rather modest score in this case outperformed by both IPFP and RRWM. However, as stated in (Cho et al., 2014): while a better performance in

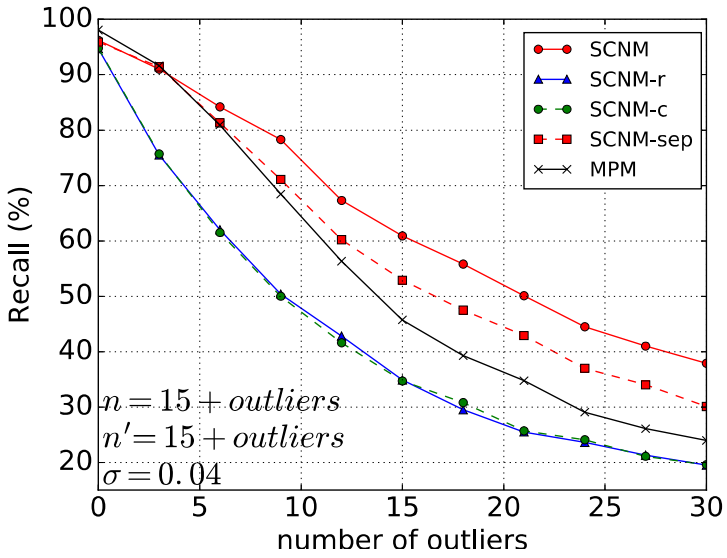


Figure 4-9: Performance gain obtained by using turbo-style decoding. We show how turbo-style alternation between decoders gives a better score than using only one decoder or using both decoders consecutively rather than in an alternating fashion.

the absence of outliers might be interesting in some situations, it is not a sufficient property from a practical point of view, since outliers are always present in natural images. In such cases, robustness to outliers is an indispensable property for matching algorithms to be equipped with.

4.5.2 Matching in natural images

In this experiment, we compare the performance of the proposed model against some of the state-of-the-art algorithms. The experiment we apply follows the one presented in (Cho et al., 2010). A set of 30 image pairs are collected from Caltech-101 and MSRC datasets. Then, the MSER detector (Matas et al., 2004) and the SIFT descriptor with 128 dimensions (Lowe, 1999) are used to generate candidate correspondences between images of each pair. These candidate correspondences are chosen based on the Euclidean distance between SIFT descriptors; a distance threshold $\delta = 0.6$ is first chosen, then a feature pair is only kept if the distance between its two corresponding descriptor vectors is inferior or equal to δ. Notice that this filtering process does not guarantee an injective mapping between features of an image pair, because it

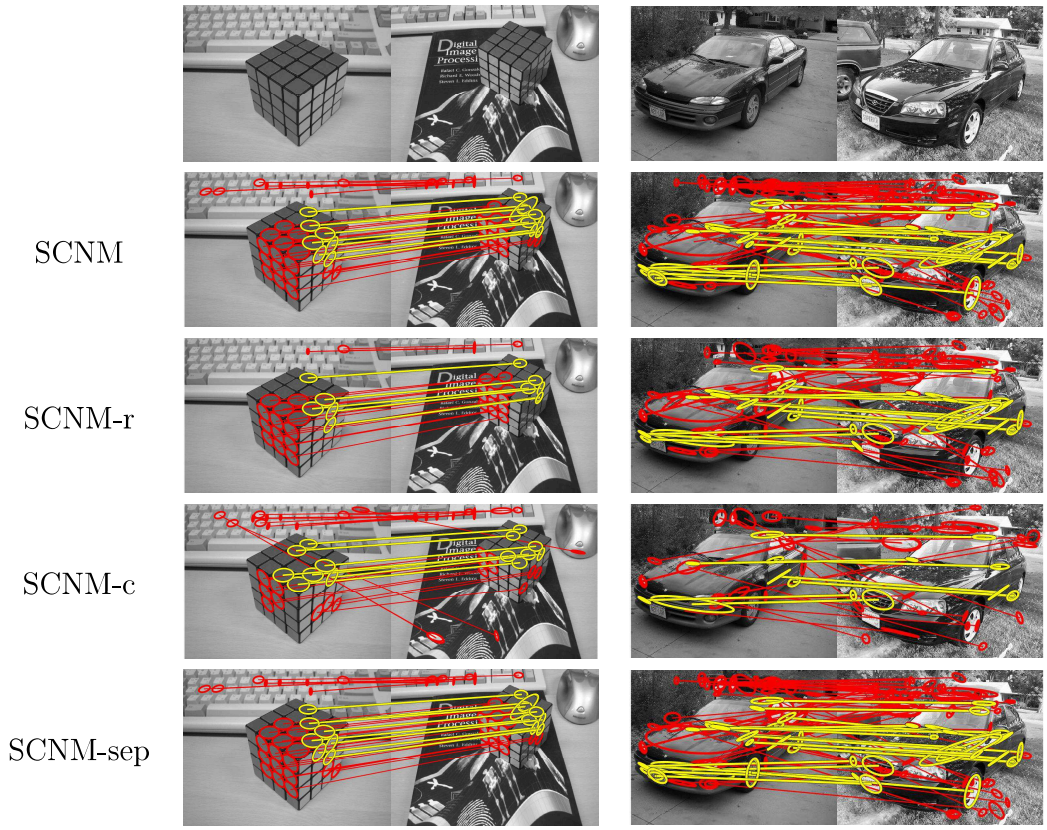


Figure 4-10: Some examples of matching features in natural images obtained by the proposed algorithm and some of its variants. True matches are shown in yellow, while false ones in red.

allows for more than one candidate match for a given feature. The pairwise similarity function is computed as follows:

$$S_E(e_{ij}, e'_{ab}) = \max(50 - d_{ia;jb}, 0), \tag{4.15}$$

where $d_{ia;jb}$ is the same mutual projection error function used in (Cho et al., 2010) and (Cho et al., 2009). We use the same correspondence ground truth as in (Cho et al., 2010), which was manually labeled.

We used three different criteria to evaluate matching algorithms. The first one is the *Recall* score, computed as the average ratio of the number of true matches obtained by an algorithm to the total number of matches in the ground truth. This is the same accuracy criterion used by most models including (Cho et al., 2010, 2014).

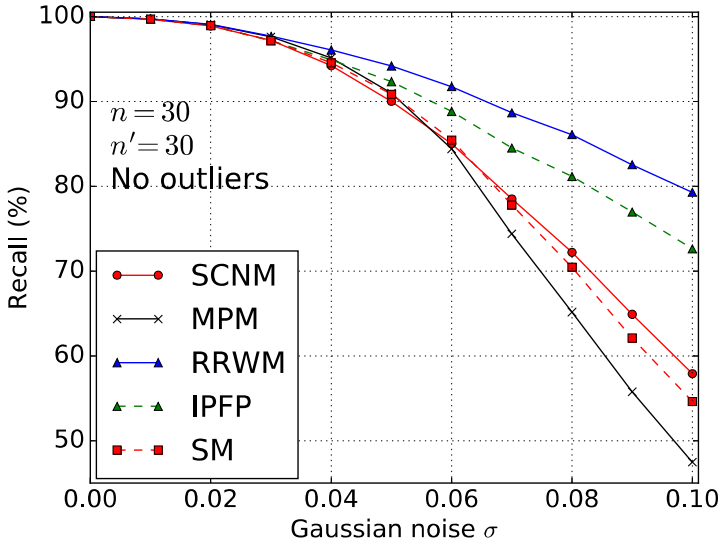


Figure 4-11: A performance comparison in the absence of outliers. The standard deviation σ of the Gaussian noise is varied, and *Recall* score is evaluated at each step.

However, we think that in the case where outliers are present in both the query image m and the destination one m', matching pairs of outliers becomes undesirable. In this case, the matching process can be viewed as two processes. The first is finding the correct matches, and the second is excluding matches between pairs of outliers.

Table 4.1: An evaluation of matching performance according to three different criteria on 30 pairs of real images taken from Caltech-101 and MSRC datasets.

Model	F_1 (%)	*Accuracy* (%)	*Recall* (%)
SCNM	**43.27***	89.09	58.20
SCNM-r	29.14	**90.02***	29.76
SCNM-c	29.11	89.92	29.88
SCNM-sep	41.95	89.11	55.96
RRWM	41.65	85.24	**73.61***
IPFP	37.93	83.89	69.77
MPM	36.89	84.44	64.09
SM	35.28	83.48	64.45

In order to evaluate the capacity of matching models in obtaining true matches

while excluding false ones, we use the F_1 score and the accuracy criteria as follows:

$$F_1 = \frac{2 \times Pr \times Re}{Pr + Re}, \tag{4.16}$$

$$Accuracy = \frac{tp + tn}{\#\text{candidate matches}}, \tag{4.17}$$

where Pr is the *Precision* score, Re is the *Recall*, tp is the number of true positives and tn is the number of true negatives.

The evaluation scores are shown in table 4.1. Notice that RRWM has the best score according to the *Recall* criterion. However, our proposed model gives the best performance according to the F_1 and the accuracy criteria. The reason for this is that the proposed model is better at excluding false matches and thus it gives less false positives than other models. This can be noticed in figures 4-6 and 4-10, where true and false matches are shown in different colors.

4.6 Conclusion and future work

In this chapter, we proposed a new approach for treating the feature correspondence problem using artificial neural networks. We compared our model to state-of-the-art algorithms, and showed that it enjoys a higher robustness to outliers thanks to the application of max-pooling and kWTA operations, and to alternating rows and columns during iterations. This robustness to outliers is an essential property for matching objects in cluttered scenes. We also evaluated the performance of the proposed model on matching features of natural images, we showed that our algorithm is better at excluding false matches between outliers, which is an interesting property for matching objects in natural scenes where clutter is omnipresent. Further development of our model will include searching for a better way of choosing final matches than zeroing rows and columns of the assignment matrix containing more than one winner. We think that it is a simple but a brutal procedure that might be excluding some good matches.

Chapter 5

Conclusion and openings

5.1 Conclusion

In this thesis, we have dealt with the information processing problem in the visual system. We proposed neuro-inspired architectures and algorithms to model some of the main functions achieved by the visual processing pipeline. These functions are located along three different stages: visual acquisition on the lowest stage, feature correspondence on the intermediate one and memory on the highest stage.

On the lowest stage, we proposed a neural network model for the acquisition and early processing of visual information in chapter 2. The architecture of this network is closely inspired by the retina and early layers of the ventral stream. Our main contribution was to equip the proposed model with the flexibility to incorporate ubiquitous properties of the visual system at that stage. This includes modeling the distribution of cone photoreceptors in the retina, which is responsible for retinal sampling of the visual scene, as well as modeling the spatial distribution of receptive fields and their eccentricity-dependent sizes, which give rise to the cortical magnification phenomenon. Retinal sampling and cortical magnification produce an input signal with a variable spatial resolution. This reduces the amount of information entering the visual stream. In order to get a better resolution at a given point in the scene, eye movement is employed to direct the fovea toward that point. We have shown that the model we proposed is also adapted to model eye movements to get the desired

spatial resolution at any part of its input. We suggested that this model can serve as a generic framework for implementing tasks that need to take advantage of the properties mentioned above. Visual attention and saliency prediction models are key applications that we suggested to build using our framework. To demonstrate this, we proposed to build a network for bottom-up saliency prediction based on the model of (Itti et al., 1998) and implemented using the proposed framework, and showed that it attains state-of-the-art performance in predicting salient regions on popular benchmarks.

On the memory stage in chapter 3, we extended Sparse Clustered Networks introduced in (Gripon and Berrou, 2011) and (Aliabadi et al., 2014) by mainly enhancing the data retrieval performance of the Losers Kicked Out activation rule. Moreover, we introduced a generic formalism through which existing retrieval algorithms can be understood, and to guide the process of designing new ones. We think that SCNs are interesting associative memories because they offer a large storage capacity compared to other models. Grouping units into clusters and constraining only one unit per cluster to be used to store a message, which is inspired by the observed short-range inhibitory connections in the visual cortex, is an interesting property that was at the heart of the feature matching model we introduced in chapter 4.

The feature matching algorithm we proposed was implemented using a neural network that we adapted from SCNs. We showed that the clustering constraint in an SCN can be a useful concept in applications that go beyond associative memories. In our case, clustering was essential to enforcing the injectivity mapping or the one-to-one matching constraint between two sets of features. The competition between candidate matches within each cluster proved essential to obtaining a matching solution that attains state-of-the-art performance on synthetic datasets as well as on real world images. It also proved robust in realistic situations in which clutter is often present in both images. We have also demonstrated an interesting property of the proposed matching network that was inspired by the turbo-decoding concept in coding theory, in which an alternate enforcement between the injectivity constraint and the mapping one was essential to reaching the obtained performance. We have also shown that

considering only one constraint and dropping the other, or considering both but in a sequential non-alternating fashion gives a significantly lower performance.

5.2 Openings

In this section, we will try to connect the dots by drawing the big picture that guided our research during the period of my PhD thesis.

5.2.1 Visual attention for less supervision

Machine learning with convolutional neural networks (CNNs) and other architectures deemed today as *deep learning* networks (DLNs) have achieved an unprecedented success in object recognition, and a wide variety of other difficult tasks that were not possible before.

However, state-of-the-art performance in object recognition can be currently achieved only when very large datasets are available for training in a supervised fashion, along with a precise label for each single training image. Humans, on the other hand, can achieve an equivalent or a better recognition performance with much fewer training examples and less supervision or even in a completely unsupervised way. This suggests that the learning processing humans use is fundamentally different from that implemented in DLNs.

One idea we wish to explore is the role of bottom-up saliency prediction and feature correspondence in establishing a learning paradigm that can be achieved with less supervision and a fewer number of training examples.

Actually, the question on the role of bottom-up saliency in learning has been already explored directly and indirectly in several areas including research on developmental learning in human infants and children recovering from blindness (Ostrovsky et al., 2006, 2010), as well as in computational models of object category discovery (Kwak et al., 2015; Wang and Gupta, 2015). As suggested by these studies, we think that learning from video rather than from static images might be the key ingredient for getting around the need for strong supervision. Bottom-up visual attention

and especially movement saliency could play an important role in achieving this. The basic idea is that tracking a moving object across different frames could provide the necessary supervision for learning. This is because the movement of an object is often smooth, and while its appearance and position change slightly across frames, the tracking signal would guarantee that the learner recognizes all of these different postures as the same object, and that it assigns them the same label. We believe that the vision acquisition framework we proposed in chapter 2 can be useful for implementing any model that needs to experiment with such ideas where emulating visual attention and eye movement is necessary.

In addition to establishing a correspondence between frames in the same video by means of tracking, correspondence should also be established among frames in different videos containing the same object as in (Wang and Gupta, 2015) based on the method proposed in (Cho et al., 2015). This is essential to creating a training set that is not limited to one object instance per category. In future work, we hope to use the feature matching model we developed in chapter 4 to accomplish such a task.

5.2.2 Better representation, less training examples

After an object has been tracked throughout frames, and a label has been assigned to all of its instances, a supervised learning algorithm should be applied on the extracted training examples. For instance, a CNN is used in (Wang and Gupta, 2015) to create a Siamese-triplet network for learning the training set extracted from hundreds of thousands of videos. Obviously, while this model proposes to get around the strong supervision problem, it still needs a tremendous number of training examples. We think that in order to come up with a model that is able to generalize from a fewer number of examples, a better representation in feature space should be figured out. In other words, a fewer examples should be sufficient to learn a representation that is generic enough to recognize new instances of the same category. In future work, we would like to explore how SCNs presented in chapter 3 can be used to store an object representation based on more generic properties in an attempt to reduce the size of the training set.

As a summary, visual attention, feature matching and SCNs are three elements that might be helpful for designing new learning algorithms that need less supervision and less training examples than current model. In future work, we shall explore how to build a model that combines the proposed vision framework, the feature matching model and SCNs in order to build a learner enjoying these properties.

Bibliography

Aboudib, A., Gripon, V., and Coppin, G. (2015). A model of bottom-up visual attention using cortical magnification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2015*, pages 1493–1497. IEEE.

Aboudib, A., Gripon, V., and Jiang, X. (2014). A study of retrieval algorithms of sparse messages in networks of neural cliques. In *COGNITIVE 2014 : the 6th International Conference on Advanced Cognitive Technologies and Applications*, pages 140–146, Venise, Italy.

Achanta, R., Hemami, S., Estrada, F., and Susstrunk, S. (2009). Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009*, pages 1597–1604. IEEE.

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169.

Aliabadi, B. K., Berrou, C., Gripon, V., and Jiang, X. (2014). Storing sparse messages in networks of neural cliques. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):980–989.

Anderson, J. R. and Bower, G. H. (1973). *Human associative memory*. Winston and Sons.

Anselmi, F., Rosasco, L., and Poggio, T. (2015). On invariance and selectivity in representation learning. *arXiv preprint arXiv:1503.05938*.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Berrou, C. and Glavieux, A. (1996). Near optimum error correcting coding and decoding: turbo-codes. *IEEE Transactions on Communications*, 44(10):1261–1271.

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics.

Bonaiuto, J. and Itti, L. (2005). Combining attention and recognition for rapid scene analysis. In *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 90–90. IEEE.

Borji, A. and Itti, L. (2013). State-of-the-art in visual attention modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):185–207.

Borji, A., Sihite, D. N., and Itti, L. (2013a). Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *Image Processing, IEEE Transactions on*, 22(1):55–69.

Borji, A., Sihite, D. N., and Itti, L. (2014). What/where to look next? modeling top-down visual attention in complex interactive environments. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 44(5):523–538.

Borji, A., Tavakoli, H. R., Sihite, D. N., and Itti, L. (2013b). Analysis of scores, datasets, and models in visual saliency prediction. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 921–928. IEEE.

Buxhoeveden, D. P. and Casanova, M. F. (2002). The minicolumn hypothesis in neuroscience. *Brain*, 125(5):935–951.

Cho, M., Kwak, S., Schmid, C., and Ponce, J. (2015). Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region

proposals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*, pages 1201–1210.

Cho, M., Lee, J., and Lee, K. M. (2009). Feature correspondence and deformable object matching via agglomerative correspondence clustering. In *IEEE 12th International Conference on Computer Vision (ICCV), 2005.*, pages 1280–1287.

Cho, M., Lee, J., and Lee, K. M. (2010). *11th European Conference on Computer Vision (ECCV), 2010*, chapter Reweighted Random Walks for Graph Matching, pages 492–505. Springer Berlin Heidelberg, Berlin, Heidelberg.

Cho, M., Sun, J., Duchenne, O., and Ponce, J. (2014). Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014*, pages 2091–2098.

Cour, T., Srinivasan, P., and Shi, J. (2007). Balanced graph matching. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 313–320. MIT Press.

Cruz, L., Buldyrev, S. V., Peng, S., Roe, D. L., Urbanc, B., Stanley, H., and Rosene, D. L. (2005). A statistically based density map method for identification and quantification of regional differences in microcolumnarity in the monkey brain. *Journal of Neuroscience Methods*, 141(2):321–332.

Dowling, J. E. (1987). *The retina: an approachable part of the brain.* Harvard University Press.

Duchenne, O., Bach, F., Kweon, I. S., and Ponce, J. (2011). A tensor-based algorithm for high-order graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2383–2395.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Freeman, J. and Simoncelli, E. P. (2011). Metamers of the ventral stream. *Nature neuroscience*, 14(9):1195–1201.

Gabor, D. (1946). Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93(26):429–441.

Gao, F., Zhang, Y., Wang, J., Sun, J., Yang, E., and Hussain, A. (2015). Visual attention model based vehicle target detection in synthetic aperture radar images: A novel approach. *Cognitive Computation*, 7(4):434–444.

Garcia-Diaz, A., Leboran, V., Fdez-Vidal, X. R., and Pardo, X. M. (2012). On the relationship between optical variability, visual saliency, and eye fixations: A computational approach. *Journal of vision*, 12(6):17–17.

Gattass, R., Gross, C., and Sandell, J. (1981). Visual topography of v2 in the macaque. *Journal of Comparative Neurology*, 201(4):519–539.

Gattass, R., Sousa, A., and Gross, C. (1988). Visuotopic organization and extent of v3 and v4 of the macaque. *The Journal of neuroscience*, 8(6):1831–1845.

Geng, X., Xu, J., Xiao, J., and Pan, L. (2007). A simple simulated annealing algorithm for the maximum clique problem. *Information Sciences*, 177(22):5064–5071.

Goferman, S., Zelnik-Manor, L., and Tal, A. (2012). Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926.

Gold, S. and Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(4):377–388.

Gonzalez, R. C. and Woods, R. E. (2002). Digital image processing.

Goodale, M. A. and Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1):20–25.

Grauman, K. and Darrell, T. (2005). The pyramid match kernel: discriminative classification with sets of image features. In *Tenth IEEE International Conference on Computer Vision (ICCV), 2005.*, volume 2, pages 1458–1465 Vol. 2.

Gripon, V. and Berrou, C. (2011). Sparse neural networks with large learning diversity. *IEEE Transactions on Neural Networks*, 22(7):1087–1096.

Gripon, V. and Berrou, C. (2012). Nearly-optimal associative memories based on distributed constant weight codes. In *IEEE Information Theory and Applications Workshop (ITA), 2012*, pages 269–273. IEEE.

Gripon, V. and Jiang, X. (2013). Mémoires associatives pour observations floues. In *Proceedings of XXIV-th Gretsi seminar*.

Gripon, V. and Rabbat, M. (2013). Maximum likelihood associative memories. In *IEEE Information Theory Workshop (ITW), 2013*, pages 1–5. IEEE.

Harel, J., Koch, C., and Perona, P. (2006). Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*. New York, Wiley.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.

Isik, L., Leibo, J. Z., Mutch, J., Lee, S. W., and Poggio, T. (2011). A hierarchical model of peripheral vision. Technical report, MIT's Computer Science and Artificial Intelligence Laboratory.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259.

Jiang, H., Yu, S. X., and Martin, D. R. (2011). Linear scale and rotation invariant matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1339–1355.

JIANG, X., GRIPON, V., and Berrou, C. (2012). Learning long sequences in binary neural networks. In *Cognitive 2012 : 4th International Conference on Advanced Cognitive Technologies and Applications*, pages 165 – 170, Nice, France.

Jones, E. G. (2000). Microcolumns in the cerebral cortex. *Proceedings of the National Academy of Sciences*, 97(10):5019–5021.

Judd, T., Ehinger, K., Durand, F., and Torralba, A. (2009). Learning to predict where humans look. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009*, pages 2106–2113. IEEE.

Koch, C. and Ullman, S. (1987). Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kruthiventi, S. S., Ayush, K., and Babu, R. V. (2015). Deepfix: A fully convolutional neural network for predicting human eye fixations. *CoRR*, abs/1510.02927.

Kwak, S., Cho, M., Laptev, I., Ponce, J., and Schmid, C. (2015). Unsupervised object discovery and tracking in video collections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3173–3181.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Larochelle, H. and Hinton, G. E. (2010). Learning to combine foveal glimpses with a third-order boltzmann machine. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 1243–1251. Curran Associates, Inc.

Le Meur, O., Le Callet, P., Barba, D., and Thoreau, D. (2006). A coherent computational approach to model the bottom-up visual attention. *IEEE transactions on pattern analysis and machine intelligence*, 28:802–817.

Le Meur, O. and Liu, Z. (2015). Saccadic model of eye movements for free-viewing condition. *Vision research*, 116:152–164.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Leordeanu, M. and Collins, R. (2005). Unsupervised learning of object features from video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.*, volume 1, pages 1142–1149 vol. 1.

Leordeanu, M. and Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV), 2005.*, volume 2, pages 1482–1489 Vol. 2.

Leordeanu, M., Hebert, M., and Sukthankar, R. (2009). An integer projected fixed point method for graph matching and map inference. In *Advances in neural information processing systems*, pages 1114–1122.

López-García, F., Dosil, R., Pardo, X. M., and Fdez-Vidal, X. R. (2011). *Scene recognition through visual attention and image features: A comparison between sift and surf approaches*. INTECH Open Access Publisher.

Lorach, H., Benosman, R., Marre, O., Ieng, S.-H., Sahel, J. A., and Picaud, S. (2012). Artificial retina: the multichannel processing of the mammalian retina achieved with a neuromorphic asynchronous light acquisition device. *Journal of neural engineering*, 9(6):066004.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on Computer vision, 1999.*, volume 2, pages 1150–1157. IEEE.

Marčelja, S. (1980). Mathematical description of the responses of simple cortical cells*. *JOSA*, 70(11):1297–1300.

Marr, D. (1982a). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA.

Marr, D. (1982b). Vision, a computational investigation into the human representation and processing of visual information. *WH San Francisco: Freeman and Company*.

Marr, D. and Poggio, T. (1976). Cooperative computation of stereo disparity. In *From the Retina to the Neocortex*, pages 239–243. Springer.

Martínez, J. and Robles, L. A. (2006). A new foveal cartesian geometry approach used for object tracking. *SPPRA*, 6:133–139.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Milner, A. D. and Goodale, M. A. (2008). Two visual systems re-viewed. *Neuropsychologia*, 46(3):774–785.

Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain*, 120(4):701–722.

Murray, N., Vanrell, M., Otazu, X., and Parraga, C. A. (2011). Saliency estimation using a non-parametric low-level vision model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011*, pages 433–440. IEEE.

Navalpakkam, V. and Itti, L. (2006). An integrated model of top-down and bottom-up attention for optimizing detection speed. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006*, volume 2, pages 2049–2056. IEEE.

Nilsson, N. J. (2009). *The quest for artificial intelligence*. Cambridge University Press.

Ostrovsky, Y., Andalman, A., and Sinha, P. (2006). Vision following extended congenital blindness. *Psychological Science*, 17(12):1009–1014.

Ostrovsky, Y. et al. (2010). *Learning to see: The early stages of perceptual organization*. PhD thesis, Massachusetts Institute of Technology.

Palm, G. (2013). Neural associative memories and sparse coding. *Neural Networks*, 37:165–171.

Pan, J., Li, X., Li, X., and Pang, Y. (2016). Incrementally detecting moving objects in video with sparsity and connectivity. *Cognitive Computation*, 8(3):420–428.

Poggio, T., Mutch, J., and Isik, L. (2014). Computational role of eccentricity dependent cortical magnification. *arXiv preprint arXiv:1406.1770*.

Ranzato, M., Hinton, G., and LeCun, Y. (2015). Guest editorial: Deep learning. *International Journal of Computer Vision*, 113(1):1–2.

Ray, S., Scott, S., and Blockeel, H. (2010). *Encyclopedia of Machine Learning*, chapter Multi-Instance Learning, pages 701–710. Springer US, Boston, MA.

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025.

Rodieck, R. W. (1965). Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision research*, 5(12):583–601.

Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. (2007). Segmenting, modeling, and matching video clips containing multiple moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):477–491.

Rybak, I. A., Gusakova, V., Golovan, A., Podladchikova, L., and Shevtsova, N. (1998). A model of attention-guided visual perception and recognition. *Vision research*, 38(15):2387–2400.

Salin, P.-A. and Bullier, J. (1995). Corticocortical connections in the visual system: structure and function. *Physiological reviews*, 75(1):107–155.

Schwartz, E. L. (1984). Anatomical and physiological correlates of visual computation from striate to infero-temporal cortex. *Systems, Man and Cybernetics, IEEE Transactions on*, (2):257–271.

Schwenker, F., Sommer, F., and Palm, G. (1996). Iterative retrieval of sparsely coded associative memory patterns. *Neural Networks*, 9(3):445 – 455.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3):411–426.

Sommer, F. T. and Palm, G. (1999). Improved bidirectional retrieval of sparse patterns stored by hebbian learning. *Neural Networks*, 12(2):281–297.

Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.

Torralba, A., Oliva, A., Castelhano, M. S., and Henderson, J. M. (2006). Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766.

Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136.

Tu, Z., Abel, A., Zhang, L., Luo, B., and Hussain, A. (2016). A new spatio-temporal saliency-based video object segmentation. *Cognitive Computation*, pages 1–19.

Tuytelaars, T. and Gool, L. V. (2000). Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference (BMVC), 2000.*, pages 412–425.

Walther, D., Rutishauser, U., Koch, C., and Perona, P. (2004). On the usefulness of attention for object recognition. In *Workshop on Attention and Performance in Computational Vision at ECCV*, pages 96–103. Citeseer.

Walther, D. B. and Koch, C. (2007). Attention in hierarchical models of object recognition. *Progress in brain research*, 165:57–78.

Wang, X. and Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802.

Willshaw, D. J., Buneman, O. P., and Longuet-Higgins, H. C. (1969). Non-holographic associative memory. *Nature*.

Wohrer, A. and Kornprobst, P. (2009). Virtual retina: a biological retina model and simulator, with contrast gain control. *Journal of computational neuroscience*, 26(2):219–249.

Xu, X. and Ma, J. (2006). An efficient simulated annealing algorithm for the minimum vertex cover problem. *Neurocomputing*, 69(7):913–916.

Zass, R. and Shashua, A. (2008). Probabilistic graph and hypergraph matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.*, pages 1–8.

Zhang, J. and Sclaroff, S. (2013). Saliency detection: A boolean map approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 153–160.

Zhang, L., Tong, M. H., Marks, T. K., Shan, H., and Cottrell, G. W. (2008). Sun: A bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32.

Zhao, J., Sun, S., Liu, X., Sun, J., and Yang, A. (2014). A novel biologically inspired visual saliency model. *Cognitive Computation*, 6(4):841–848.

Zheng, Y., Zemel, R., Zhang, Y.-J., and Larochelle, H. (2015). A neural autoregressive approach to attention-based recognition. *International Journal of Computer Vision*, 113(1):67–79.

Zhou, F. and la Torre, F. D. (2012). Factorized graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012*, pages 127–134.

Zhu, J.-Y., Wu, J., Xu, Y., Chang, E., and Tu, Z. (2015). Unsupervised object class discovery via saliency-guided multiple class learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(4):862–875.

Résumé

L'apprentissage automatique et la vision par ordinateur sont deux sujets de recherche d'actualité. Des contributions clés à ces domaines ont été les fruits de longues années d'études du cortex visuel et de la fonction des réseaux cérébraux. Dans cette thèse, nous nous intéressons à la conception des architectures neuro-inspirées pour le traitement de l'information sur trois niveaux différents du cortex visuel. Au niveau le plus bas, nous proposons un réseau de neurones pour l'acquisition des signaux visuels. Ce modèle est étroitement inspiré par le fonctionnement et l'architecture de la retine et les premières couches du cortex visuel chez l'humain. Il est également adapté à l'émulation des mouvements oculaires qui jouent un rôle important dans notre vision. Au niveau le plus haut, nous nous intéressons à la mémoire. Nous traitons un modèle de mémoire associative basée sur une architecture neuro-inspirée dite 'Sparse Clustered Network (SCN)'. Notre contribution principale à ce niveau est de proposer une amélioration d'un algorithme utilisé pour la récupération des messages partiellement effacés du SCN. Nous suggérons également une formulation générique pour faciliter l'évaluation des algoritmes de récupération, et pour aider au dévéloppment des nouveaux algorithmes. Au niveau intermédiaire, nous étendons l'architecture du SCN pour l'adapter au problème de la mise en correspondence des caratéristiques d'images, un problème fondamental en vision par ordinateur. Nous démontrons que la performance de notre réseau atteint l'état de l'art, et offre de nombreuses perspectives sur la façon dont les architectures neuro-inspirées peuvent servir de substrat pour la mise en œuvre de diverses tâches de vision.

Mots-clés : Vision par ordinateur, Réseaux de neurones artificiels, Architectures neuro-inspirées, Intelligence artificielle

Abstract

Computer vision and machine learning are two hot research topics that have witnessed major breakthroughs in recent years. Much of the advances in these domains have been the fruits of many years of research on the visual cortex and brain function. In this thesis, we focus on designing neuro-inspired architectures for processing information along three different stages of the visual cortex. At the lowest stage, we propose a neural model for the acquisition of visual signals. This model is adapted to emulating eye movements and is closely inspired by the function and the architecture of the retina and early layers of the ventral stream. On the highest stage, we address the memory problem. We focus on an existing neuro-inspired associative memory model called the Sparse Clustered Network. We propose a new information retrieval algorithm that offers more flexibility and a better performance over existing ones. Furthermore, we suggest a generic formulation within which all existing retrieval algorithms can fit. It can also be used to guide the design of new retrieval approaches in a modular fashion. On the intermediate stage, we propose a new way for dealing with the image feature correspondence problem using a neural network model. This model deploys the structure of Sparse Clustered Networks, and offers a gain in matching performance over state-of-the-art, and provides a useful insight on how neuro-inspired architectures can serve as a substrate for implementing various vision tasks.

Keywords : Computer vision, Artificial neural networks, Neuro-inspired architectures, Artificial intelligence

TELECOM Bretagne

n° d'ordre : 2016telb0419
Télécom Bretagne
Technopôle Brest-Iroise - CS 83818 - 29238 Brest Cedex 3
Tél : + 33(0) 29 00 11 11 - Fax : + 33(0) 29 00 10 00