Universidade Estadual de Campinas
Instituto de Computação

Ewerton Almeida Silva

A Study on Foveal Image Models in Computer Vision Applications

Um Estudo sobre Modelos de Foveação de Imagem em Aplicações de Visão Computacional

CAMPINAS
2020

Ewerton Almeida Silva

A Study on Foveal Image Models in Computer Vision Applications

Um Estudo sobre Modelos de Foveação de Imagem em Aplicações de Visão Computacional

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Ricardo da Silva Torres

Este exemplar corresponde à versão final da Tese defendida por Ewerton Almeida Silva e orientada pelo Prof. Dr. Ricardo da Silva Torres.

CAMPINAS
2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Si38s

Silva, Ewerton Almeida, 1988-
A study on foveal image models in computer vision applications / Ewerton Almeida Silva. – Campinas, SP : [s.n.], 2020.

Orientador: Ricardo da Silva Torres.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Visão por computador. 2. Fenologia. 3. Visualização de informação. 4. Biometria. 5. Computação - Consumo de energia. I. Torres, Ricardo da Silva, 1977-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Um estudo sobre modelos de foveação de imagem em aplicações de visão computacional
Palavras-chave em inglês:
Computer vision
Phenology
Information visualization
Biometry
Computer science - Energy consumption
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Ricardo da Silva Torres [Orientador]
Fábio Augusto Menocci Cappabianco
Keiller Nogueira
Hélio Pedrini
Marcos Vinicius Mussel Cirne
Data de defesa: 04-09-2020
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0003-4021-5906
- Currículo Lattes do autor: http://lattes.cnpq.br/5787113569249807

Universidade Estadual de Campinas
Instituto de Computação

Ewerton Almeida Silva

A Study on Foveal Image Models in Computer Vision Applications

Um Estudo sobre Modelos de Foveação de Imagem em Aplicações de Visão Computacional

Banca Examinadora:

- Prof. Dr. Ricardo da Silva Torres
Instituto de Computação - Unicamp
- Prof. Dr. Hélio Pedrini
Instituto de Computação - Unicamp
- Dr. Marcos Vinicius Mussel Cirne
Instituto de Computação - Unicamp
- Prof. Dr. Fábio Augusto Menocci Cappabianco
Instituto de Ciência e Tecnologia - Unifesp
- Dr. Keiller Nogueira
University of Stirling

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 04 de setembro de 2020

Dedicatória

À minha tia Layce (Lala).

Agradecimentos

E lá se foram sete anos e meio. Se eu soubesse com antecedência tudo o que aconteceria, talvez tivesse ficado com muito medo de começar. Porém, se eu pudesse sentir, no início, um pouquinho do afeto de todas as pessoas que conheceria adiante, certamente teria seguido em frente. A intensidade do amor, do carinho, da amizade, do companheirismo, da solidariedade e da humildade das pessoas que passaram (e continuam) na minha vida me moldaram de um jeito sem volta. Eu agradeço por isso.

Começo por três mulheres fortes: minha mãe (Ruthinha), minha irmã (Hivigne) e minha tia (Layce). Apesar da distância, vocês estiveram comigo o tempo todo. Obrigado por me ensinarem a não desistir fácil! Cada palavra que escrevo aqui só pôde existir porque vocês me apoiaram sempre, mesmo sem saber de todas as dificuldades. Amo vocês.

Ao meu orientador, Ricardo Torres: você é um professor especial! Você também não me deixou desistir, mas não é só por isso que lhe agradeço. Eu agradeço pela amizade e pela humanidade com que você me tratou em todos os momentos. Jamais esquecerei da nossa conversa sobre sonhos de vida. Você me contou o seu e eu lhe contei o meu, que era aprender a jogar voleibol. Essa e muitas outras conversas demonstraram o amor com o qual você conduz a sua profissão, que envolve lidar com pessoas. Carregarei comigo todo esse carinho e prometo recompensar o mundo com muito mais.

São vários os amigos que fizeram parte dessa trajetória. Talvez a primeira amiga de verdade, daquelas que a gente sabe que será pra toda vida, foi a Priscila Saboia. Essa é mais uma das mulheres fortes que deveria estar no segundo parágrafo destes agradecimentos. Nós dois passamos por situações difíceis juntos, e ninguém conseguia me entender tão bem como você. Obrigado demais pela amizade, pela compreensão, pelos cafés e pelos momentos inesquecíveis. Espero que no futuro tenhamos novas aventuras juntos.

Pela primeira vez na vida eu me senti incluído em um grupo com o qual me identifiquei totalmente: o grupo do pessoal do vôlei. Foi nesse grupo que conheci os amigos com quem passei a maior parte do período do doutorado. Eu nunca me senti tão feliz e livre, nem nunca havia rido tanto na minha vida! Por essa amizade sem igual, agradeço a vocês Gilmar Brito, Paulinho, Cadu, Vonei Cene, Thiago Rentz, Léo, Marquinhos, Will, Jair Mendes, Lucas Martins, Marcinho, Elias e Tiago Balbino. Não posso deixar de lembrar das amigas Vivian, Amanda, Nara e Camila, que também fazem parte dessa história do vôlei. Obrigado por tudo! Todos vocês têm uma grande parte do meu coração.

Dentre os amigos especiais e importantes nessa trajetória estão, ainda, o Hilário Seibel, a Danielle Dias, o Felipe Andrade, a Karina Bogdan, o Elias Ximenes, e o Daniel Moreira. Nossa união me deu muitas forças e a amizade de vocês sempre me traz muita felicidade! Agradeço também aos amigos Luiz Monteiro, Cássio Almeida, Alexandre Oliveira, Amanda Oliveira, Brunna Linhares e Jordy Lasso pelos bons momentos juntos e pela amizade leve.

Outro agradecimento especial é para uma pessoa que está na minha vida há pouco

tempo. Essa pessoa acompanhou os últimos anos da minha pesquisa e, mesmo que eu estivesse passando por um momento cheio de angústias e indecisões, acreditou em mim e continuou ao meu lado. Muito obrigado, Paulo Henrique de Sousa! Quero estar ao seu lado e para ver todas as suas vitórias, também!

Finalmente, agradeço ao CNPq, à FAPESP (#2014/12236-1, #2015/24494-8, #2016/50250-1, e #2017/20945-0), ao Instituto Virtual FAPESP-Microsoft (#2013/50155-0, #2013/50169-1, and #2014/50715-9) e à Norwegian University of Science and Technology (NTNU) pelo apoio. O presente trabalho foi realizado também com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Os enormes volumes de dados da era atual têm levado a muitos desafios científicos. Lidar com essa quantidade de dados requer que as aplicações de software considerem as restrições de armazenamento e energia de alguns ambientes computacionais. As aplicações de visão computacional (VC) exemplificam bem esse cenário, já que geralmente processam séries de imagens uniformes de alta resolução para produzir os resultados desejados. Nesse cenário, uma perspectiva não convencional a ser explorada é o uso de modelos foveais de imagem capazes de reamostrar imagens uniformes em imagens espaço-variantes. A inspiração para tal provém da retina humana, na qual a fóvea central possui resolução máxima, enquanto a periferia caracteriza-se por uma resolução que diminui gradualmente. Assim, para reduzir a quantidade de dados processados pelo cérebro, movemos nossos olhos para focar um ponto de interesse; nesse processo, também mantemos informações periféricas relevantes. Neste trabalho, exploramos o conceito de modelos foveais de imagem em aplicações de VC. A motivação do estudo decorre de desafios derivados (i) da alta disponibilidade de grandes volumes de dados de imagem, (ii) do aumento do uso de aplicativos de VC em diferentes plataformas de computação, (iii) da necessidade de lidar adequadamente com recursos limitados de armazenamento e energia de alguns ambientes e (iv) da necessidade de técnicas adequadas de visualização de informações que auxiliem a execução de inspeções visuais em séries temporais de imagens. Em nossa contribuição inicial, propomos um *framework* para a criação de modelos retinais de imagem orientados a aplicações. Validamos os modelos em uma aplicação de biometria, mostrando que boas taxas de acurácia podem ser mantidas juntamente às reduções de armazenamento e energia induzidas pelos modelos. Nossa segunda contribuição é uma abordagem de foveação de imagem orientada a mudanças para estudos de fenologia de plantas. Validamos a abordagem com uma base de dados de imagens de sensoriamento remoto. Os experimentos indicam que os modelos fornecem reduções de armazenamento e resultados interessantes de correlação com imagens uniformes, em uma configuração usualmente adotada em estudos no campo da fenologia. Finalmente, em nossa terceira contribuição, atacamos o desafio de lidar com séries temporais de imagens de altíssima resolução. Propomos uma abordagem de visualização de informação baseada em conceitos de foveação e detecção de regiões salientes. Validamos a abordagem por meio de uma base de dados usada em estudos de fenologia e mostramos que a abordagem pode facilitar inspeções visuais de grandes volumes de dados de imagem.

Abstract

The huge volumes of data from the current era have led to several scientific challenges. Dealing with such amount of data requires that software applications consider the storage and energy restrictions of some computing environments. Computer vision (CV) applications exemplify well this scenario, given that they usually process series of uniform high-resolution images to output the desired results. In this scenario, an unconventional perspective to explore is the use of foveal image models able to resample uniform images into space-variant ones. The inspiration comes from the human retina, in which the central fovea has maximum resolution, whereas the periphery possesses a gradually-decreasing one. Thus, to reduce the amount of data to be processed by the brain, we move our eyes to foveate a point of interest; we also keep relevant peripheral information in the process. In this work, we explore the concept of foveal image models towards CV applications. The motivation underlying our study arises from a combination of challenges derived from (i) the high availability of massive volumes of image data, (ii) the increasingly use of CV applications in different computing platforms, (iii) the need of properly handling the limited storage and energy resources of some computing environments, and (iv) the need of proper information visualization techniques to aid visual inspections and decisions regarding image time series. In our first contribution, we propose a framework for creating application-oriented retinal image models. We validate our models in a CV application in the biometry field, by showing that good accuracy rates can be retained alongside storage and energy reductions induced by the models. Our second contribution is a change-driven image foveation approach toward plant phenology investigations. We validate our approach with a database of remote sensing images. The experiments indicate that our models provide storage reductions and interesting correlation results with uniform images, considering a commonly-adopted setting of the phenology field. Finally, in our third contribution, we target the challenge of dealing with very-high-resolution image time series. We propose an information visualization approach based on concepts of foveation and saliency detection, and validate it with a database used for phenological investigations. We show that the proposed visualization may facilitate visual inspections of large volumes of image data.

List of Figures

1.1 The non-uniform distribution of photoreceptors across the human eye. (a) An illustration of the human eye showing the retina and the angle of view measured from the fovea. (b) The densities of rods and cones vary non-uniformly with the angle; a blind spot is a region lacking photoreceptors (redrawn and adapted from [7]). 22

1.2 Exemplary frames of a video from the dataset created in the context of the LGE-AOP project. The adopted protocol required that individuals crossed a door. In this procedure, faces would be nearly centralized at some point, thus allowing a more effective detection and recognition for biometry purposes. The dataset was only used in the early stages of our research. 24

1.3 A diagram representing the organization of this thesis. Strong arrows represent the chapter sequence, whereas dashed lines represent connections between ideas and chapters. The LGE-AOP project is directly related to the first contribution, while the concept of space-filling curves has been employed in the second and third contributions. 26

2.1 The log-polar transformation of a cartesian image and a model-based resampling (adapted from [89]). 32

2.2 Examples of retinal image models that may provide data reduction and a space-variant image representation through a retino-cortical mapping procedure. Each cell of a model is called a receptive field, gathering information from the region they circumvent in the uniform image. The central region of each model (i.e., the fovea, not shown here) is usually uniform, but the non-uniform representation provided by the model may be used as well. (**a**) Traditional log-polar grid, (**b**–**d**) models redrawn from [19], (**e**–**f**) models redrawn from [29] and [66], respectively. 33

2.3 The construction of a Hilbert curve in four different levels and the respective 1-d sequences. Figures (a), (b), (c), and (d) represent the first, second, third, and fourth levels, respectively, with each square having a correspondent 1-d subinterval. 38

2.4 The construction of a Peano SFC in three different levels. Figures (a), (b), and (c) represent the first, second, and third levels, respectively, with each square having a correspondent 1-d subinterval (shown for the first level, only). 39

3.1 The proposed framework to generate ARIMs. The workflow begins by defining the application's requirements regarding operation (e.g., objects' position, illumination) and efficiency (e.g., storage, accuracy). Then, a proper implicit function (e.g., l_2) and the spatial configuration of the retinal image model – comprising foveal and peripheral regions – are chosen. The next step is the generation of the model by means of an optimization procedure that considers the implicit function and the spatial configuration to resample points in the 2-d cartesian space. The final artifact is an ARIM comprised of uniformly- and non-uniformly-sampled foveal and peripheral regions, respectively. This model is used to resample uniform images, taking them to a space-variant domain and potentially contemplating the requirements determined beforehand. 42

3.2 Examples of implicit functions: (**a**) l_1, (**b**) l_2, and (**c**) l_∞. 43

3.3 Convergence analysis for ARIMs with a single fovea and based on the l_∞ implicit function. Examples of ARIMs containing (a) 96, (b) 192, and (c) 384 non-uniform points in the periphery. For each ARIM, the resulting global energy curve over 1000 iterations of the generation process is shown in the model's respective column. 45

3.4 Convergence analysis for ARIMs with no fovea and based on the l_2 implicit function. Examples of ARIMs containing (a) 96, (b) 192, and (c) 384 non-uniform points in the periphery. For each ARIM, the resulting global energy curve over 2000 iterations of the generation process is shown in the model's respective column. 46

3.5 The evolution of an example of ARIM with 256 foveal (uniform), and 192 peripheral (non-uniform) pixels. The l_∞ is the implicit function. 46

3.6 Example of a simulation using one of our ARIMs and a sample sequence from the employed dataset [105]. Images (**a**–**e**) and (**k**–**o**) are the original frames; images (**f**–**j**) and (**p**–**t**) are the reconstructions with a model that considers an optical flow peripheral representation. Green and yellow arrows indicate motion direction to the right and left sides, respectively, whereas the ON and OFF labels refer to the operational status of the foveal (face detection/recognition) and peripheral (optical flow) regions. Note that the motion analysis, besides triggering foveal analysis, is also able to restart conveniently, as long as faces are not detected in the fovea during a time interval of frames (left-most frame in the fourth row). 48

3.7 Implemented workflow for simulating the use of ARIMs in a specific Computer Vision (CV) application. In an ideal scenario, the ARIM, a captured image frame, and the chosen pixel representations for foveal and periphery areas are input to an hypothetical specific-purpose sensor that changes its configuration at run-time. Such a sensor would yield a stream (bytestream) of pixel data from each region of the captured image. The stream (not the 2D image) would be forwarded to the CV application. However, for simulation purposes, this architecture is fully implemented by software. 49

3.8 The pixel map of the evaluated ARIM and its configurations. The experimented foveal configurations comprised three uniform sampling setups: 100 × 100 (half density – Model_1), 150 × 150 (75% density – Model_2), and 200 × 200 (full density – Model_3) pixels. The pixel representations for the fovea and periphery were based on the grayscale and optical flow (magnitude and direction) values, respectively. 51

3.9 ROC curves regarding the face recognition task considering the image sequences (**a**) P1E_S2, (**b**) P1L_S2, (**c**) P1E_S3, (**d**) P1L_S3, (**e**) P1E_S4, and (**f**) P1L_S4 from all four cameras. The figures comprise a mean ROC (blue) curve from all 25 (light blue) class-specific curves, i.e., for each face class in the dataset. These class-specific ROC curves were calculated via a one-versus-all classification procedure. 53

3.10 Mean face recognition accuracy regarding each evaluated model, the images resized to 25% of their original sizes, and the benchmark images from the (**a**) P1E and (**b**) P1L datasets. 56

3.11 Mean face recognition accuracy regarding each evaluated model, the images resized to 25% of their original sizes, and the benchmark images from the (**a**) P2E and (**b**) P2L datasets. 57

3.12 Total energy consumption regarding each evaluated model, the images resized to 25% of their original sizes, and the original (full-size) images from the (**a**) P1E and (**b**) P1L datasets. 59

3.13 Total energy consumption regarding each evaluated model, the images resized to 25% of their original sizes, and the original (full-size) images from the (**a**) P2E and (**b**) P2L datasets. 60

4.1 Proposed workflow to generate foveal models and resample phenological images. 62

4.2 Example of the detection of the 010 and 0110 binary patterns using a series of typical phenological images. Both sides of the figure show the same image series and zoomed-in regions (i.e., the same location in each image). For illustration purposes, the analysis is done only for two pixels from the focused region. Nevertheless, the complete procedure requires the analysis of all image pixels. 64

4.3 Pipeline of the Binarized MHH generation. This example illustrates the entire process applied to detect a specific pattern throughout the image series. First, two consecutive images I_i and I_{i+1} from an image series $S = \{I_i, I_{i+1}, ..., I_n : 0 \leq i \leq n, n \geq 1\}$ are compared according to a phenological metric (in this example, the pixel *Increase*). The comparison yields a binary change map CM_i spotting out the pixels that conform to the metric. As comparisons proceed (i.e., as i increases) all CMs are inspected towards detecting pixels that change according to a behavior pattern (in this example, **010**). Finally, the MHH is gradually updated upon detections. Suppose a pixel at position (x_1, y_1) is found to follow the desired pattern (in this example, green border cells) at a certain point of the examinations. In such a case, MHH(x_1, y_1) would increase by one. After the entire time series has been processed, the MHH is binarized by means of a predefined threshold sigma (σ). 64

4.4 MHHs (**top**), KDE implicit functions (**middle**), and foveal models (**bottom**) generated for the (**a**) 010, (**b**) 0110, (**c**) 01110, and (**d**) 011110 patterns considering the 2012 images. 67

4.5 Example of image reconstruction. The original image (**left**) is reconstructed (**right**) using a Voronoi diagram (**middle**) obtained from a model. Each Voronoi cell is filled, in the right image, with the same RGB color of the cell's central pixel in the left image. 67

4.6 Examples of phenological images reconstructed with a log-polar mapping. Images (**a**, **c**, and **e**) are the original ones, and images (**b**, **d**, and **f**) are their reconstructions using an inverse log-polar mapping on the cartesian domain. We do not consider log-polar images in our quantitative tests. . . 69

4.7 Mean GCC time series and correlation results for *Aspidosperma tomentosum* and *Caryocar brasiliensis* individuals, considering original and foveal images with each model. In the 2013 dataset, the last available DOY from the evaluated period was 169. 70

4.8 Examples of phenological images from 2013. The DOYs 248, 255, 260, and 263 are column-wise organized. Each row depicts the following: (**a**–**d**) original images; masked images containing only the (**e**–**h**) *Aspidosperma tomentosum* and (**i**–**l**) *Caryocar brasiliensis* individuals; (**m**–**p**) images reconstructed with the model for the 0110 pattern; and masked-reconstructed images containing only the (**q**–**t**) *Aspidosperma tomentosum* and (**u**–**x**) *Caryocar brasiliensis* individuals. 71

4.9 Examples of sROIs (from the 010 model) in a quad-tree scheme at the 2nd (**left**), 6th (**middle**), and 8th (**right**) MRLs. Each level is represented in a distinct color. In this case, the lowest MRL is the 2nd, meaning that at least two recursive refinements were performed for any of the four points of the initial curve. The 2nd MRL therefore includes all points of the model. Nevertheless, the highest refinement level of the depicted model is the 8th one. Thus, the 6th MRL comprises the points at the 6th, 7th, and 8th refinement levels, whereas the 8th MRL includes only the points of its own level. 72

4.10 Mean GCC time series and correlation results for ROIs and sROIs at the 6th and 8th MRLs. Both *Aspidosperma tomentosum* and *Caryocar brasiliensis* plant species were considered. 73

4.11 Comparison of the storage sizes (in kilobytes, KB) of compressed images with our change-driven approach and with the usual JPEG compression process. The models evaluated were the ones generated by the 010, 0110, 01110, and 011110 binary patterns, whereas the JPEG quality factors considered were 10, 30, 50, 70, and 90. The graph shows the mean size of the compressed images. 74

4.12 Examples of phenological JPEG-compressed images with the quality factors (**a**) 90, (**b**) 70, (**c**) 50, (**d**) 30, and (**e**) 10. Image (**f**) is the original, uncompressed one. Although visually similar, images (**d**) and (**e**) present several compression artifacts (zoom in to check in detail). 75

5.1 Proposed framework to handle VRH image time series. First, each image in the time series is cropped according to a predefined grid of $M \times N$ size, thus generating several smaller images. Then, all cropped images at a specific grid position p in the time series are examined in the search for saliency points and regions. This procedure generates a binary map, which is used to build a saliency map spotting out the estimated salient regions (the darker the region, the more important it is). This has similarities with a top-down visual attention process, because we look for image points having plant individuals undergoing phenological changes over time. The next step consists of creating a foveal model - based on a Hilbert curve - that is able to represent the set of examined cropped images in the time series. The foveation process consists of applying the curve to sample points according to the degree of importance of the regions in the saliency map. In this case, for example, dark regions will have higher point density than white ones. Finally, each cropped image at position p will be represented as a sequence of points extracted by means of the model's curve. Such sequence can be further used as input to different information visualization strategies, such as visual rhythms, graphs, and charts. 79

5.2 Illustration of the normalization procedure for the creation of the normalized stacked bar chart. The chart is a representation of a sequence of collected points according to a model. In this example, we create a normalized stacked bar chart of eight positions based on a sequence comprising four points. We also consider a grayscale image where each pixel has a single color value in the [0, 255] range. First, we compute a color histogram that indicates the frequency of each value in the grayscale image. In this illustrative case, the frequencies for the 90, 150, and 220 values are 25%, 50%, and 25%, respectively. Then, we create the normalized stacked bar chart by arranging the grayscale values along the available eight positions of the chart. Such arrangement takes into account the frequencies of each value in the computed histogram. Thus, for instance, as the 90 value has 50% frequency, it will appear in four of the eight positions of the chart. . . 81

5.3 The proposed information visualization strategy for a specific cropped area from the VHR image time series. The top-most line of the visualization represents the normalized histogram of the desired pixel representation (GCC or BCC) colored according to the palette's configuration. The next charts (representing other images from the series) are vertically and chronologically stacked bellow the first one. 82

5.4 Example of the proposed information visualization scheme (**q**) encoding a Green Up phenomenon (**010** pattern) for a specific (cropped) area from the VHR image time series. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**–**m**) is a series of cropped images of a green area, (**n**) is the binary MHH calculated, (**o**) is the saliency map comprising regions of interest, and (**p**) is the foveal model generated. 84

5.5 Example of the proposed information visualization scheme (**d**) encoding a Green Up phenomenon (**0110** pattern) for a specific (cropped) area from the VHR image time series from Figure 5.4. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**) is the binary MHH calculated, (**b**) is the saliency map comprising regions of interest, and (**c**) is the foveal model generated. 85

5.6 Example of the proposed information visualization scheme (**d**) encoding a Green Up phenomenon (**01110** pattern) for a specific (cropped) area from the VHR image time series from Figure 5.4. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**) is the binary MHH calculated, (**b**) is the saliency map comprising regions of interest, and (**c**) is the foveal model generated. 85

5.7 Example of the proposed information visualization scheme (**q**) encoding a leaf flush phenomenon (**010** pattern) for a specific area from the VHR image time series. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**–**m**) is a series of cropped images of a green area, (**n**) is the binary MHH calculated, (**o**) is the saliency map comprising regions of interest, and (**p**) is the foveal model generated. 86

5.8 Example of the proposed information visualization scheme (**d**) encoding a leaf flush phenomenon (**0110** pattern) for a specific area from the VHR image time series from Figure 5.7. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**) is the binary MHH calculated, (**b**) is the saliency map comprising regions of interest, and (**c**) is the foveal model generated. . 87

5.9 Example of the proposed information visualization scheme (**d**) encoding a leaf flush phenomenon (**01110** pattern) for a specific area from the VHR image time series from Figure 5.7. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**) is the binary MHH calculated, (**b**) is the saliency map comprising regions of interest, and (**c**) is the foveal model generated. 87

5.10 Example of the proposed information visualization scheme (**q**) encoding a flowering phenomenon (**010** pattern) for a specific area from the VHR image time series. The phenomenon is caught by investigating and processing images in the BCC space. Time ranges from February 2016 to February 2017. (**a**–**m**) is a series of cropped images of an area containing blue flowers, (**n**) is the binary MHH calculated, (**o**) is the saliency map comprising regions of interest, and (**p**) is the foveal model generated. 88

5.11 Example of the proposed information visualization scheme (**q**) encoding a flowering phenomenon (**010** pattern) for a specific area from the VHR image time series. The phenomenon is caught by investigating and processing images in the BCC space. Time ranges from February 2016 to February 2017. (**a**–**m**) is a series of cropped images of an area containing blue flowers, (**n**) is the binary MHH calculated, (**o**) is the saliency map comprising regions of interest, and (**p**) is the foveal model generated. 89

6.1 Illustration of the remote sensing application scenario using ARIMs. The idea may be appropriated for plant phenology tasks. 96

6.2 Illustration of the industrial application scenario (**a**) using ARIMs (**b**). The idea may be appropriated for assembly line inspection tasks. 96

6.3 Illustration of the action recognition application scenario (**a**) using ARIMs (**b**). The idea may be appropriated for driver's fatigue and other facial expression recognition tasks. Image (**a**) is a frame from the Brain4Cars Dataset [44]. 97

6.4 Image reconstruction based on an ARIM that uses a grayscale pixel representation on the periphery. The original image (**a**) is reconstructed (**b**) with voronoi cells, each of which has a single grayscale color. 97

B.1 Examples of images reconstructed with a log-polar mapping. Left-column images are the original ones. Middle-column images are the cortical versions after a log-polar mapping procedure. Right-column images are the retinal images after an inverse log-polar mapping procedure has been performed on their cortical versions. The original images come from the ChokePoint Dataset [105]. 110

List of Tables

3.1 Number of pixels and data size reduction results for the evaluated models relative to the original and uniformly resized images (to 25% of their original sizes). 52

3.2 Minimum, mean, and maximum accuracy loss rates induced by our ARIMs compared to the provided benchmarks. For example, the minimum accuracy loss induced by Model 1 on the P1E dataset was 0.032 (i.e., 3.2%), whereas the mean and maximum accuracy loss in that same scenario were 0.123 (12.3%) and 0.264 (26.4%), respectively. 55

3.3 Minimum, mean, and maximum energy reduction rates induced by our ARIMs compared to the provided benchmarks. For example, the minimum energy reduction induced by Model 1 on the P1E dataset was 0.505 (i.e., 50.5%), whereas the mean and maximum energy reductions in that same scenario were 0.551 (55.1%) and 0.598 (59.8%), respectively. 58

4.1 Results for reconstructed images using the four models respective to the 010, 0110, 01110, and 011110 binary patterns. The considered metrics are root mean squared error (RMSE), mean absolute error (MAE), structural similarity index (SSIM), and peak signal-to-noise ratio (PSNR). 70

4.2 Storage results for full models and sROIs at the 6th-MRL, which includes regions at the 6th, 7th, and 8th refinement levels. 74

D.1 Technical information regarding the implemented CV application. Only non-default parameter values are shown. 113

Contents

1 Introduction 20

1.1 Problem Description 20

1.2 The Low-Power Always on Presence Sensors Project 23

1.3 Hypothesis and Research Questions 24

1.4 Thesis Outline 25

2 Background and Related Work 28

2.1 Image Compression 28

2.2 Space-Variant Computer Vision 29

2.2.1 Hardware-based foveation approaches 30

2.2.2 Software-based foveation approaches 30

2.2.3 Saliency estimation 32

2.2.4 Applications of saliency estimation and foveation 34

2.3 Space-Filling Curves 35

2.3.1 The construction of SFCs 36

2.4 Final Considerations 38

3 Application-Oriented Retinal Image Models for Computer Vision 40

3.1 Introduction 40

3.2 Proposed Approach 41

3.2.1 Definition of Application Requirements 42

3.2.2 Implicit Function Selection 43

3.2.3 Definition of Spatial Configuration 43

3.2.4 Model Generation 44

3.3 Materials and Methods 45

3.3.1 Target Application: Face Detection / Recognition 45

3.3.2 Dataset 50

3.3.3 Evaluated Models 51

3.3.4 Evaluation Criteria and Hardware Setup 51

3.4 Results and Discussion 52

3.4.1 Storage reduction 52

3.4.2 Face recognition accuracy 52

3.4.3 Energy consumption evaluation 55

3.4.4 Implications in real-time applications 55

3.5 Final Considerations 58

4 A Change-Driven Image Foveation Approach for Tracking Plant Phenology 61
4.1 Introduction . . . 61
4.2 Proposed Approach . . . 62
4.3 Materials and Methods . . . 65
4.3.1 Dataset . . . 66
4.3.2 Foveal Models . . . 66
4.3.3 Image Reconstruction . . . 67
4.4 Results and Discussion . . . 68
4.4.1 Evaluation for Reconstructed Images and ROIs . . . 68
4.4.2 Evaluation for sROIs . . . 72
4.4.3 Storage Usage . . . 73
4.5 Final Considerations . . . 76

5 A Foveation-based Framework to Support the Visualization of Very-High-Resolution Image Time Series 77
5.1 Introduction . . . 77
5.2 Proposed Approach . . . 78
5.2.1 Visual attention for estimating salient regions . . . 78
5.2.2 Foveation for generating foveal models . . . 80
5.2.3 Proposed information visualization approach . . . 80
5.3 Usage Scenarios . . . 82
5.4 Final Considerations . . . 90

6 Conclusions 91
6.1 Scientific Contributions . . . 91
6.2 Future Work . . . 93
6.2.1 On the first contribution . . . 93
6.2.2 On the second contribution . . . 95
6.2.3 On the third contribution . . . 95

Bibliography 98

A Algorithm for Generating ARIMs 107

B Log-polar Images for Comparison 109

C Motion Analysis Algorithm 111

D Technical Information and Parameters 113

E Algorithm for Generating Foveal Models 114

Chapter 1

Introduction

In this chapter, we introduce the subject of the thesis by first describing the problem and its underlying challenges. Next, we expose information on a research project that represents our initial investigation on the subject. Then, we show the formulated hypothesis and some research questions encompassing relevant topics of investigation. Finally, we present the thesis's organization.

1.1 Problem Description

Biological systems are often a source of inspiration and ideas for technological fields. In particular, the human visual system, given its complexity and extensive possibilities, has motivated many studies to understand how we perceive the structures or objects from the real world. Examples of studies permeate the field of computer vision (CV), in which the goal is to model and interpret the visual world by capturing, processing, and extracting useful information from digital images. Typical CV tasks range from object detection and recognition (e.g., vegetation, human faces) to activity recognition (e.g., accidents, human actions) to 3-d scene reconstruction from multiple images [86].

In several CV applications, a relatively small amount of abstract information is extracted from digital images. However, the inspected images usually contain much more data than the necessary for the application to deliver its results. This is a consequence of the widespread of general-purpose sensors – such as the ones used for digital photography – in different computing devices and platforms. These sensors capture high-resolution images and send them for processing. However, in some circumstances, not all captured information is relevant for the application considered. In this case, the amount of data to be processed will be huge. As an interesting comparison, while other domains have their own specifically-designed sensors (e.g., temperature, humidity), CV applications often need to rely on conventional uniform image sensors. Although these sensors may allow a broad range of applications to be exploited, some concerns arise with respect to the use computing resources.

When looking for accuracy and speed, CV applications may employ sophisticated methods and hardware technologies. However, even in these cases, its necessary to take into account possible limitations in computing resources. For instance, according to Bornholt et al. [22], battery life is a critical factor in mobile devices, since it can prevent users from having a satisfactory experience. When dealing with uniform high-resolution images, the amount of data stored is usually very high, and so is the amount of energy spent to process these data. Thus, careful investigations of the storage and energy issues, aimed at optimization purposes, may be required. In fact, these are paramount actions when designing CV applications for restricted computing environments.

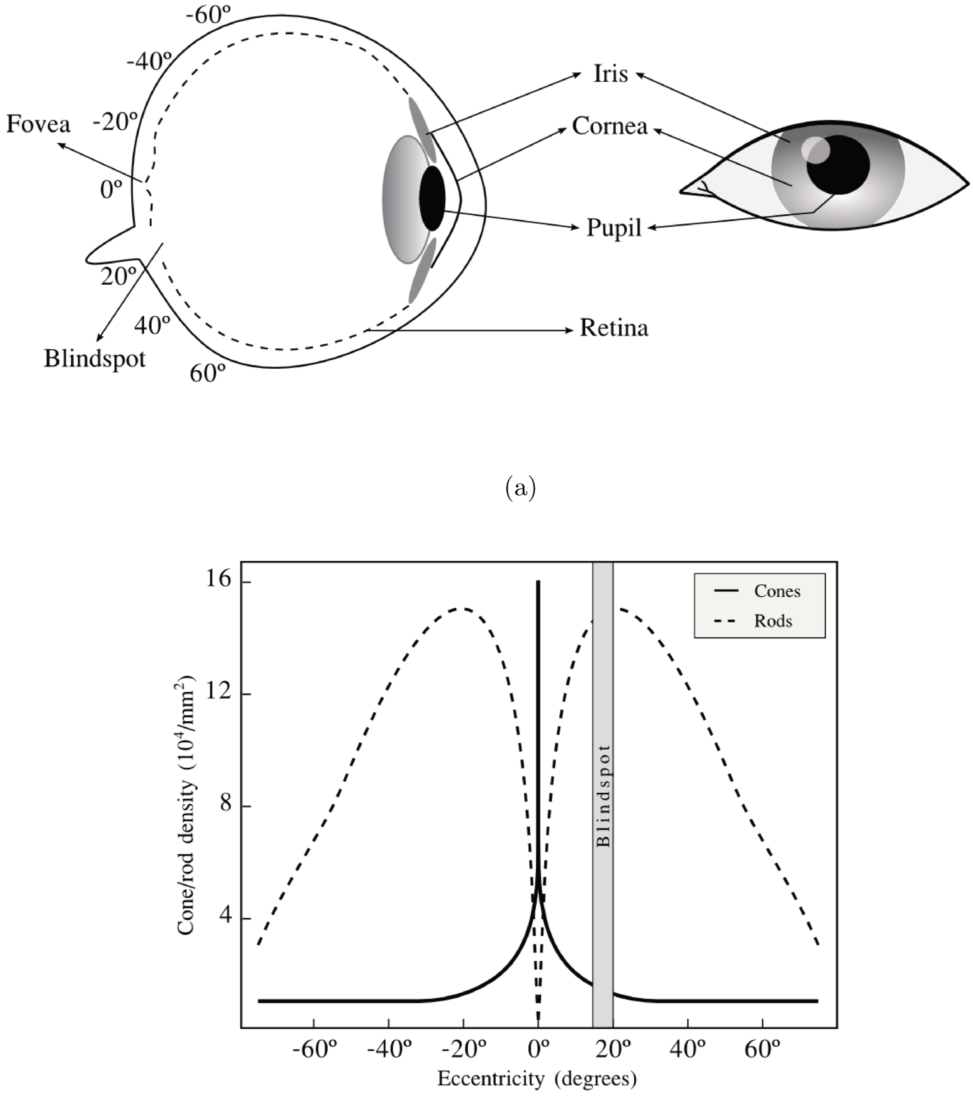
Classical approaches for saving energy and storage resources often rely on decreasing the camera's resolution and frame rates. Space-variant computer vision approaches, however, take an unconventional avenue by resampling images in a biologically-oriented manner. This unconventional perspective is the one we explore in this work.

Sampling is a common procedure performed inside a camera to generate a digital image. The idea is to sample the 3-d real world into a discrete 2-d image that can be handled by our computer apparatus. Nevertheless, the amount of information generated might still be huge for the application on hand. Our insight is that a uniform sampling of images into matrices of regular-sized pixels is neither necessary nor an efficient way to perform CV. Adapting that process according to the requirements of specific applications could favour the storage and energy issues at the cost of some acceptable performance loss.

The conventional uniform image sampling procedure poses a contrast to biological visual systems. The retina – an area of the eye responsible for capture light information – has a non-uniform spreading of two types of photoreceptors: *rods* and *cones*. Rods exist in a larger amount and are densely distributed in the peripheral areas of the retina. Rods are also very sensitive to low levels of illumination, thus being highly efficient at detecting the presence and direction of motion. In opposition, an overall smaller population of cones is almost fully concentrated on a central area of the retina known as fovea. Cones are also very sensitive to color, being divided into three types distinguished by their responsiveness to short, medium, and long light wavelengths [98].

Figure 1.1 shows the distribution of rods and cones on the human eye versus the angle of view. Both types of photoreceptors are non-uniformly distributed: two density peaks of rods on the periphery and a downhill to zero (the closer to the fovea) oppose to a single density peak of cones (in the fovea). Interestingly, there is also a "blindspot" area where no photoreceptor is found [98].

In this study, we investigate the creation and use of foveal image models that define an image resampling strategy similar to the one performed in the retina. The proposed approaches have inspiration on the human visual system, which is able to efficiently handle large volumes of data through a foveation process that is executed in the early steps of the visual processing chain. This includes inspecting foveal and peripheral regions in different ways. We use a regular image sensor and perform the sampling procedures by means of a software layer, thereby simulating the operation of an ideal specific-purpose sensor. With this, we were able to easily adapt the simulated sensor for different image settings and CV applications.



(a)

(b)

Figure 1.1: The non-uniform distribution of photoreceptors across the human eye. (a) An illustration of the human eye showing the retina and the angle of view measured from the fovea. (b) The densities of rods and cones vary non-uniformly with the angle; a blind spot is a region lacking photoreceptors (redrawn and adapted from [7]).

1.2 The Low-Power Always on Presence Sensors Project

The Low-Power Visual Always-on-Presence Sensors (LGE-AOP) was a scientific project funded by LG Electronics from 2015 to 2016. The collaboration involved three professors from the Institute of Computing, University of Campinas (IC – Unicamp), two doctoral students, and a postdoctoral fellow. The project can be considered as the starting point of this Ph.D research. The main objective was to explore novel approaches to deal with CV applications in terms of AOP imaging sensors, which are sensors that are constantly capturing and sending image data for processing. Examples of this scenario comprise the use of photography-oriented sensors in surveillance applications regarding biometry and traffic analysis. In these, CV algorithms are often applied to extract a relatively small amount of abstract information (e.g., the ID of an individual, the position of a vehicle) from large sequences images, which exposes a contrast between the volumes of data captured/processed, and the amount of information extracted/produced. Since the processed information is several orders of magnitude larger than the final desired information, energy and storage consumption become a paramount issue that CV applications must deal with.

We tackled the challenge of handling the limitations in computing resources from the image sampling perspective. We hypothesized that a uniform sampling of images into matrices of regular-sized pixels is neither necessary nor an efficient way to perform CV. When looking into biological systems, for example, we see that the human retina presents a non-uniform configuration of photoreceptors. Thus, investigating non-uniform image sampling procedures was seen as a promising approach. In particular, the energy consumption investigation was split into two fronts: the sensor operation itself and the data processing/analysis. The research on both fronts required a search for industrial patents and literature works proposing methods similar to the one we were proposing.

The hardware front has been investigated by means of an intermediate layer that collects only a sparse subset of the available pixels. Such a layer could be built on top of a re-configurable sparse sensor that would be adaptable for different situations. We believed this could reduce the amount of memory required to store the image inside the sensor, as well as the overall energy consumption of the embedded application. This research front is still being examined by some project members.

In turn, the software front, which has been evolved into this thesis, relied on simulating the use of a specific-purpose sensor toward a CV biometry application. We employed a conventional imaging sensor as the underlying data capture mechanism, and performed the non-uniform sampling of pixels by means of a software layer. We have developed (i) a method to devise image models to resample images according to the characteristics of the biometry scenario considered, (ii) the biometry application itself, and (iii) a proper dataset for evaluating the models. Figure 1.2 shows some frames of the dataset created, which was only used in the early investigations of the LGE-AOP project. The adopted protocol required that individuals crossed a door, thus nearly centralizing their faces at some point. This could allow a more effective face detection and recognition for biometry purposes. However, the dataset was only used to confirm our initial expectations that the use of a non-uniform resampling procedure (according to some predefined image models) could induce memory storage and energy savings in a CV application.

Figure 1.2: Exemplary frames of a video from the dataset created in the context of the LGE-AOP project. The adopted protocol required that individuals crossed a door. In this procedure, faces would be nearly centralized at some point, thus allowing a more effective detection and recognition for biometry purposes. The dataset was only used in the early stages of our research.

The project's findings suggested that the use of non-uniform image models is a viable strategy. We obtained good reductions in memory storage and energy, thus improving the efficiency of the biometry application. However, we also faced very low face recognition accuracy results, which ended up compromising the effectiveness of the application. With these findings and difficulties on hand, we decided to "reshape" our research by using a more consolidated dataset, a modern face recognition algorithm, and by providing a more detailed analysis of the complete approach. Such actions have been crucial for improving our comprehension of the literature on image foveation, and to the gradual evolution of the conducted study.

1.3 Hypothesis and Research Questions

In this work, we aim at proposing approaches to develop foveal image models and investigating their use in computer vision applications. The motivation underlying our study arises from a combination of challenges derived from (i) the high availability of massive volumes of image data, (ii) the increasing use of computer vision applications in different computing platforms, (iii) the need of properly handling the limited storage and energy resources of some computing environments, and (iv) the need of proper information visualization techniques to aid visual inspections and decisions regarding image time series. In this light, our main hypothesis can be stated as follows:

The use of foveal image models might induce a reduction in storage usage and energy consumption of computer vision applications, while also keeping satisfactory accuracy rates, and possibly allowing easier visual inspections of series of image data.

Given this hypothesis, we work on the following research questions (RQ). We provide initial thoughts right after each RQ.

- **RQ1:** Would the use of foveal image models lead to savings in computational resources, such as memory storage and energy?

  By this RQ, we aim at exploring foveal models to verify whether their adoption in CV applications might favour savings in energy and memory storage. For such, we work at the convergence point of (i) the insights obtained from the LGE-AOP project's execution, and (ii) the idea of space-filling curves (in the sampling process) to potentially reduce memory usage.

- **RQ2:** Does the use of foveal image models heavily impact in the accuracy rates of the application?

  By this RQ, we aim at examining in detail the efficiency of the CV application after the use of foveal models. The analysis may be in terms of 2-d reconstructed images – which can be seen as compressed images –, and 1-d sequences of points coming from a space-filling curve employed to resample the image space.

- **RQ3:** How to adapt the proposed framework to different application scenarios?

  By this RQ, we aim at discussing and exploring the flexibility of the proposed frameworks (to devise foveal image models) toward different CV applications. This is of much relevance, because it allows the results of study to be extended and applied in several ways. Initially, we aim at investigating the biometry and plant phenology application domains. The biometry scenario is well-established in the CV literature, comprising a plethora of studies, techniques, and available datasets. In turn, plant phenology is a field that has received a growing attention from researchers, because it may take advantage of modern CV techniques, mobile platforms, and large-scale storage devices. This field is concerned with the study of the life-cycle events of plant species in a certain area over time. In this light, the amount of data (images and videos) to be processed and analyzed calls for novel approaches of managing these data. We believe foveal models may be a promising answer to these needs.

- **RQ4:** How to devise foveal image models by taking into account relevant image regions found automatically?

  By this RQ, we aim at creating image models via algorithmic processes that take fewer and less-specific input parameters from the considered application's domain. This might represent a big step from the previous LGE-AOP study, since foveal and peripheral areas would be found automatically by taking into account the dynamics of the image's content, instead of being fixed and defined beforehand.

- **RQ5:** How to combine foveal image models with information visualization techniques?

  By this RQ, we aim at investigating other benefits of foveal models in the information visualization field. In this light, combining foveal models with techniques from that field may lead to interesting strategies to reduce the amount of information to be visually inspected during some technical and scientific tasks.

1.4 Thesis Outline

This thesis is organized into six chapters, as the workflow of Figure 1.3 shows. In Chapter 2, we review the literature on space-variant computer vision, with a focus on works

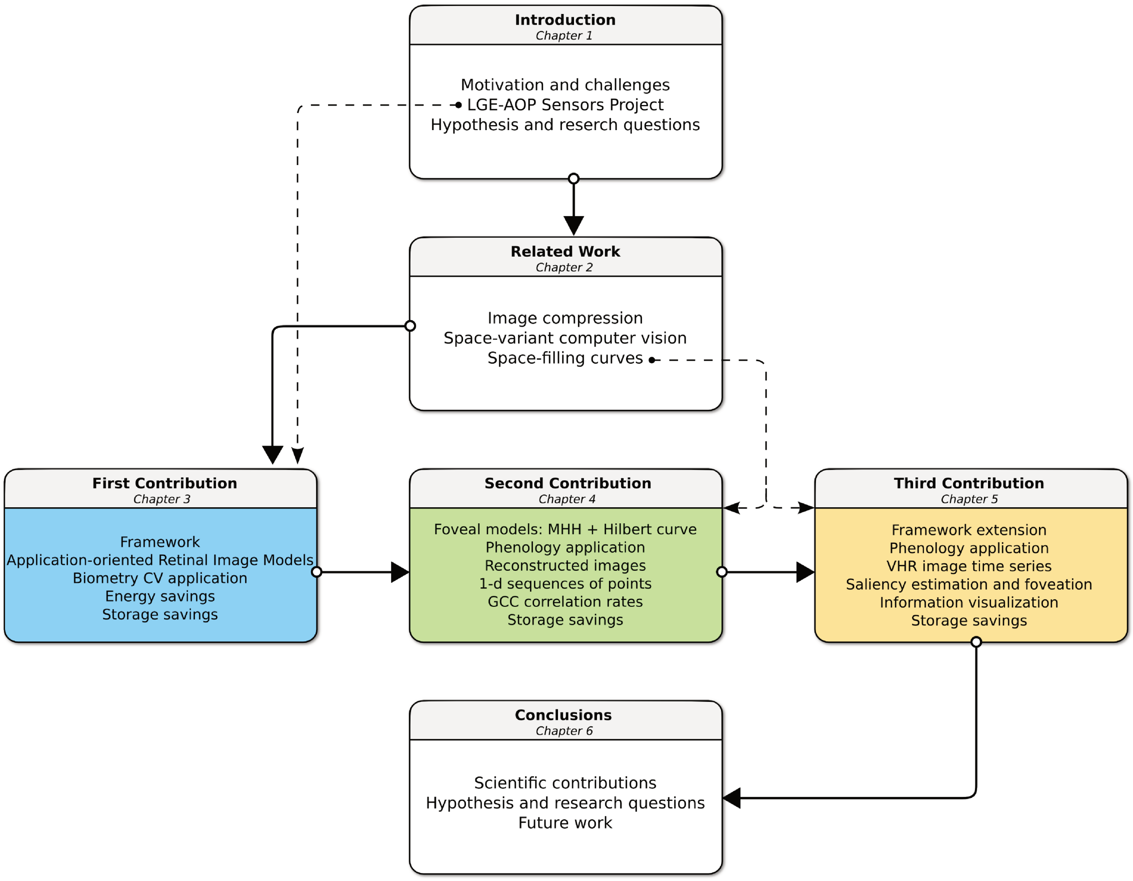


Figure 1.3: A diagram representing the organization of this thesis. Strong arrows represent the chapter sequence, whereas dashed lines represent connections between ideas and chapters. The LGE-AOP project is directly related to the first contribution, while the concept of space-filling curves has been employed in the second and third contributions.

proposing different foveal image models and applications that exploit the benefits of such models. We also present a background on space-filling curves, which are employed in one of our scientific contributions as an alternative scheme to generate foveal models.

In Chapter 3, we present our first contribution regarding the generation and use of foveal image models in the context of a face authentication application. We propose a framework for generating application-oriented retinal image models. The foveal models are validated in terms of memory storage and energy consumption, and their impact on the accuracy of the application.

In Chapter 4, we present our second contribution regarding a change-driven image foveation approach for dealing with plant phenology investigations. We devise foveal models for a target application of the phenology field by considering the phenological behaviors of plant species depicted in time series of near-remote remote sensing images. In contrast to our first contribution, the foveal and peripheral regions of the models are found via an automatic process that takes as input a phenological metric and a binary behavior pattern representing a phenological change of interest. We validate these foveal models in terms of the correlation rates and memory storage consumption they induce.

In Chapter 5, we present our third contribution regarding the use of foveal image models. We extend our second contribution to the scenario where investigations are performed over very-high-resolution image time series. Such huge volumes of image data may hamper detailed scientific investigations. Again, we target the plant phenology field, which is the one we have previously examined. In this sense, we propose a new information visualization approach based on two concepts of the human visual system: foveation and saliency estimation. We validate our approach qualitatively by means of visual inspections related to the plant phenology context.

In Chapter 6, we conclude our work by summarizing our main contributions, reviewing the hypothesis and research questions, and pointing out future work possibilities.

We provide further information regarding first contribution in Appendices A, B, C, and D, in which we show the algorithm for creating retinal models, some log-polar images for visual comparison with the proposed models, the motion analysis algorithm, and technical information of the CV application, respectively. Furthermore, in Appendix E, we detail the algorithm for generating our foveal models from our second contribution.

Chapter 2

Background and Related Work

In this chapter, we provide a theoretical background and a literature review. First, we overview image compression techniques in Section 2.1. Next, we provide a literature review on space-variant computer vision in Section 2.2. After that, we present a background on space-filling curves in Section 2.3. We expose our final considerations in Section 2.4.

2.1 Image Compression

To store or transmit images in and through digital media, it is sometimes necessary to reduce the images' sizes. The need for compact image representations led to the study of compression algorithms able to exploit data redundancy in 2-d intensity arrays. According to Gonzalez and Woods [36], three main types of data redundancy can be exploited: (i) coding redundancy, when the number of bits to represent an information is greater than the necessary; (ii) spatial and temporal redundancy, understood as the similarity or dependence between neighboring pixels from spatial (and temporal, in videos) correlation present in 2-d arrays; and (iii) irrelevant information, when information is not essential for the application or when it is ignored by the human visual system.

There are two types of compression systems: lossless systems, in which information is preserved such that the decoded image is a replica of the original encoded one; and lossy systems, in which the reconstructed image is "distorted", i.e., it presents distinct pixel information compared to the original image. One of the most used compression standards was proposed by the *The Joint Photographic Experts Group* (JPEG). This standard is a lossy compression method that works on 8×8 image blocks for coding and subsequent quantization. The underlying process consists in traversing the image collecting blocks of pixels in a raster-scan fashion and further processing these with the block transform, run-length, and/or Huffman [43] coding strategies. Other coding approaches encompass the Golomb [35], arithmetic [2], LZW [101], symbol-based [12], bit-plane, predictive [14, 51], and wavelet-based [45] ones. For a detailed explanation on these approaches, refer to [36].

2.2 Space-Variant Computer Vision

Over the years, properties and mechanisms of many biological systems have been replicated in different domains of science. A good example is in CV, where the paradigm of capturing and processing uniform images shows an interesting contrast to our visual system. In the human retina, cones and rods (the photo-receptors) present a non-uniform spatial configuration that induce variable visual acuity levels across the retina [98]. Cones and rods are responsible for detecting color and luminance, respectively. The area with the highest density of cones is the fovea, the central area of the retina. When moving to the periphery (i.e., apart from the fovea), the density of cones and the visual acuity decrease, whereas the density of rods increases. This configuration provides a wide peripheral field of view, along with a high resolution region that is used to "foveate" (i.e., focus on) a point in a real scene, which implies data processing reduction [19,98]. Foveation is the biological action of moving the eyes to focus objects/regions on the fovea, so as to have a detailed view of these. This action might be the result of a decision taken after obtaining and processing peripheral information. For instance, when crossing a street, a person might notice an object moving towards her by means of the peripheral vision. After detecting the object's presence, she foveates and recognizes it as a car, thereby deciding on stop walking.

One of the first studies investigating the space-variant mechanisms of biological vision dates back to late 70's, when Schwartz [78] reported that the transmission of information from the retina (where sensing occurs) to the visual cortex (where processing takes place in the brain) of primates could be approximated by a log-polar mapping. In such a mapping, radial lines from the center of the uniform (cartesian) image are converted to straight horizontal lines in the *cortical* (log-polar) image, whereas circular lines are mapped to straight vertical lines. Figure 2.1 illustrates the log-polar transformation on a cartesian image.

Bio-inspired computing has become a hotspot in many areas of computer science and engineering [26]. The human visual system (HVS) stands as a proper biology-related topic from which many computing approaches have borrowed concepts. Although the HVS is vast in terms insights it may provide, two of its concepts have been deeply explored in the computer science literature. The first one, related to visual attention, is in charge of selecting/estimating salient (relevant) regions from a scene to trigger saccadic eye movements towards these regions. The motivation revolves around the limited capacity of our brain to process the entire scene in detail [106]. The second concept refers to the foveation process. When performing saccadic movements, our eyes seek to foveate a small region of the field of view to process this region at high resolution in the brain. Conversely, a wider peripheral region keeps being processed at coarser resolutions, lowering the "computing cost" in the brain and saving body resources [3]. Thus, saliency estimation and foveation combine forces to provide our brains less relevant information to be quickly understood.

Foveation has also been explored from the hardware and software perspectives. The first one contemplates hardware implementations of retina-like sensors able to capture images following specific space-variant schemes. The second avenue refers to retinal image models (RIMs) able to map uniform images to space-variant domains. The latter approach is suitable for simulations and studies, since it works with conventional uniform image sensors and different RIMs can be experimented, but it might decrease the applications' efficiency in on-line scenarios due to the additional mapping procedure. A more complete review on space-variant imaging from the hardware and software perspectives using log-polar mappings is detailed in [88].

2.2.1 Hardware-based foveation approaches

The hardware side comprises two types of approaches. The first one deals with the manufacturing of image sensors with specific non-uniform spatial configurations [18], which allow the capture of topology-fixed foveated images at sensing time. The second type is related to the use of an intermediary hardware layer to resample uniform images, taking these to a space-variant space. This latter type of approach provides more flexibility to change the mapping without relying on software routines. In this context, Bailey and Bouganis [15] exploited the versatility of Field Programmable Gate Arrays (FPGA) to implement, at logical level, different space-variant image mappings. The authors also considered a moving fovea that is dynamically adjusted according to the application's requirements.

Gonzalez et al. [37] integrated attention and segmentation mechanisms into a foveal vision system. The architecture of the solution comprised a hardware layer (responsible for mapping uniform cartesian images to space-variant ones) and software layer (responsible for performing segmentation and saliency estimation). The overall idea is that salient regions from a frame might trigger a foveal shift. Such shift is performed by hardware when the next frame arrives, thereby adjusting the resolution of certain image regions accordingly.

Niu et al. [67] proposed a novel foveal imaging system. The authors combined a set of lenses and a scanning microlens to produce images having a local magnification effect in some (foveal) regions, while other regions (periphery) remain unchanged. The authors claim that the system is suitable for tasks such as object tracking and monitoring, due to the dynamic nature of implemented foveation mechanism. However, peripheral resolution cannot be degraded, and the proposed system's architecture requires the use of different microlenses to vary the structural foveal arrangement.

2.2.2 Software-based foveation approaches

Software-based approaches offer more flexibility to simulations, albeit with higher computational costs, in comparison to their hardware counterparts. Smeraldi and Bigun [84] present a saccadic search strategy based on foveation for facial landmark detection and authentication. First, the authors apply a log-polar mapping to some image points. Then, they extract Gabor filter responses at the locations of these points, a process that imitates the characteristics of the human retina. For training, the authors use Support Vector Machine (SVM) classifiers to discriminate between positive and negative classes of facial landmarks (eyes and mouth) represented by the Gabor responses. During the test stages, the saccadic search procedure evaluates several image points to find candidate landmarks that are used to authenticate the depicted individual.

Ekbas and Eckstein [3] proposed a foveated object detector that operates on variable-resolution images obtained by a resampling procedure with a simplified model of the human visual cortex. According to the authors, the detector was capable of approximating the accuracy of a uniform-resolution-oriented one. The results of the study could also provide a satisfactory insight and understanding about evolutionary biology processes.

Phillips et al. [71] exploited the concept of image foveation in a single-pixel camera architecture to promote a compromise between resolution and frame rate. The input images were resampled by a dynamic space-variant model. Such model is constantly reshaped to match the regions of interest detected by a motion tracking procedure. This process simulates a moving fovea that increasingly gathers high-resolution data across frames and finds relevant regions to be foveated.

To facilitate comparisons among different sensor arrangements, Wen et al. [102] describe a method able to provide a common space for creating lattices of any kind. The authors demonstrate the viability of the idea by resampling images according to the rectangular and hexagonal lattices, and further compare the images on these different arrangements.

To exploit the benefits of a log-polar representation, such as data reduction and the possibility to deal with different resolutions across a single image, the common task is to perform a RIM-based image re-sampling (also called as retino-cortical mappping) to simulate the retina's space-varying property in the cartesian domain (see Figure 2.1). A RIM specifies how pixels are arranged into receptive field structures [9, 19], which are analogous to super-pixels. Different image models have been proposed in the literature [19, 66, 76, 103] to re-sample the uniform image (or reconstruct, for the cortical image) into a new one where the processing takes place. Data reduction is accomplished with a trade-off between a high resolution in the central region and a coarse wide-field resolution in the periphery. Bolduc and Levine [19] describe representative image models for re-sampling images following an inverse log-polar mapping. Figure 2.2 presents several RIMs that could be applied to a cartesian uniform image to reduce data, and to provide a space-variant image representation. Each cell of a models is called a receptive field, gathering information from the region they circumvent in the uniform image (e.g., the color of the central pixel, and the average color from all pixels inside the region could be used).

A challenge of using RIMs is how to properly handle space-variant images. This is due to the distinct pixel sizes and shapes, and the non-uniform lattice topologies (i.e., the pixels do not relate to each other by a fully 4- or 8-connected neighborhood, as usual). In this scenario, image processing algorithms for different spatial configurations of pixels must be created. Wallace et al. [97] proposed to represent images as connectivity graphs (CG). The graphs generalize the lattices' topologies, thus reducing the image processing problem to a graph-theoretical one. The authors illustrate their idea by providing graph-based algorithms for local image operators, pyramid operations, geometric transformations, connected components, and template matching. Although it was claimed that the CG idea is independent of RIMs or sensor geometries, only log-polar mappings were contemplated.

Grady [38] also investigated image processing in odd topology domains defined by

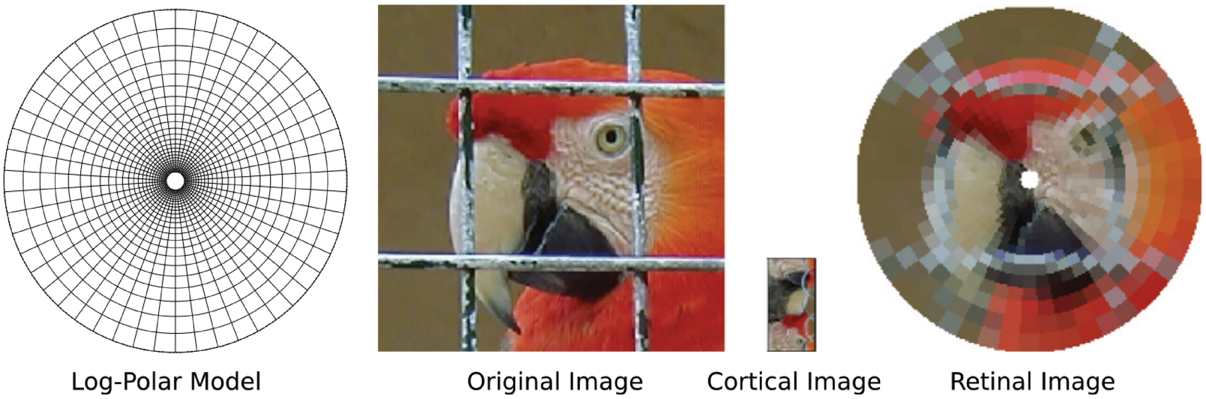


Figure 2.1: The log-polar transformation of a cartesian image and a model-based re-sampling (adapted from [89]).

arbitrary graphs. Besides a human retina-based model, the author shows visual models from several animals (e.g., cat, rabbit, and bird), representing those as graphs with non-uniform densities useful for re-sampling images in a variety of spatial configurations. The fact that convolution (or correlation) algorithms presume shift-invariant topology schemes, which is not the case in space-variant images, led the author to formulate graph-based versions of image processing algorithms like interpolation, filtering, pyramid-based segmentation, and edge finding. The re-sampled images are displayed by colouring Voronoi cells or by interpolating across a Delaunay triangulation. Still, no implementation of a CV algorithm was reported.

2.2.3 Saliency estimation

Several approaches for estimating salient regions in an image have been reported. Top-down schemes seek task-dependent cognitive cues (e.g., faces, objects, and text) that vary according to individual's experience in the task. In contrast, bottom-up schemes rely on basic perceptual cues (e.g., color, shape, and size) that vary little between individuals and induce involuntary attentional shifts [3, 106, 109]. The final artifact is a saliency map spotting out regions our eyes are likely to fixate in an image. Attentional models for still images may also incorporate the mechanism of inhibition of return, which prevents recently-visited locations from being revisited in a short time window [3, 100]. However, such mechanism is not effective at handling dynamic settings where objects or the visual system itself are in motion, because the regions to be avoided may change their location over time. To overcome this limitation, object positions might be tracked and updated at each frame to allow an effective inhibition [58]. Borji and Itti [20] review attentional models applied to computer vision tasks and other computing domains.

More recent literature on this subject has focused on calculating saliency maps visually highlighting the features, objects, and regions encoded by deep convolutional neural networks in each of their deep layers. In two relevant works, Wang et al. [99] present an in-depth overview of saliency estimation approaches for object detection using deep neural networks, whereas Borji [21] provides a survey on deep visual saliency models. In this thesis, we take a different perspective regarding saliency estimation, because our

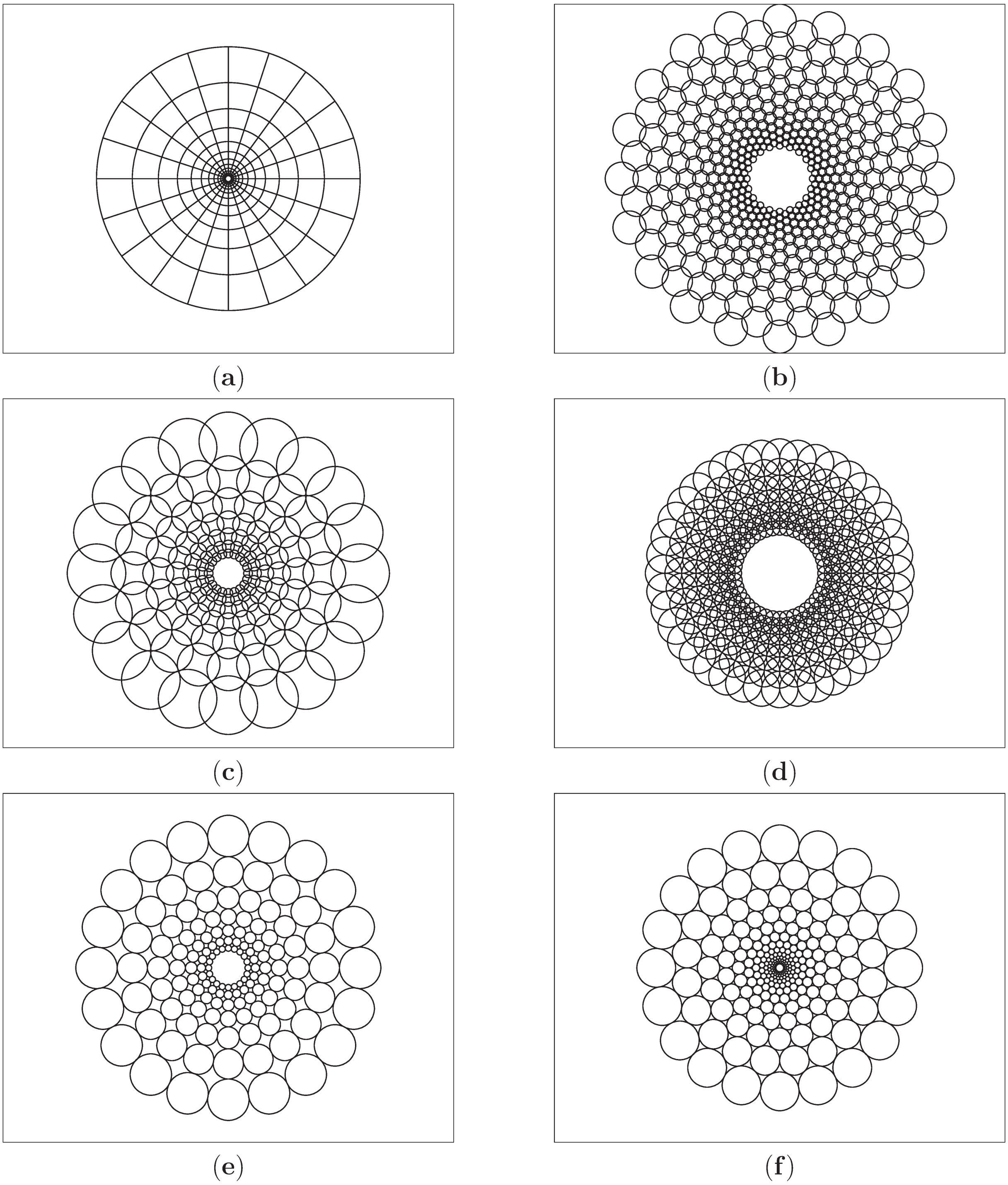


Figure 2.2: Examples of retinal image models that may provide data reduction and a space-variant image representation through a retino-cortical mapping procedure. Each cell of a model is called a receptive field, gathering information from the region they circumvent in the uniform image. The central region of each model (i.e., the fovea, not shown here) is usually uniform, but the non-uniform representation provided by the model may be used as well. (**a**) Traditional log-polar grid, (**b**–**d**) models redrawn from [19], (**e**–**f**) models redrawn from [29] and [66], respectively.

maps are not the result of any machine learning method, but are rather created from the spatial analysis of sequences of images and according to some criteria related to the application's domain. Finally, there is also a modern approach for dealing with the attentional vision challenge by means of event-based cameras. Whereas traditional cameras stream data according to a fixed frame-rate, event-based ones are oriented to asynchronously processing and streaming events taking place in the scene. Thus, instead of capturing entire image frames, these cameras work by sending data – such as time, spatial location, and sign – from salient regions measured in terms of image-pixels brightness changes. Gallego et al. [31] show a detailed survey on event-based vision, an emergent field where event cameras are investigated in applications, such as computer vision and robotics.

2.2.4 Applications of saliency estimation and foveation

The need to jointly optimize resources' usage and performance requirements is a strong motivation for exploiting the idea of space-variant images. In the literature, applications of foveation comprise image transmission/communication [16,32,74] and CV tasks [47,87,89] integrated into robotic systems [69, 77]. This alliance is able to simulate the foveation mechanism well by coupling cameras in mobile robots' arms. Such systems, known as active computer vision systems (ACVS), oppose to their passive (PCVS) counterpart that do not rely on camera movements and are application-dependent (e.g., subjects must foveate themselves).

In ACVS, the peripheral and fovea regions usually work in a collaborative fashion:

1. The periphery provides general information about the wide-field scene, such as presence of motion or the position of potential objects of interest;

2. The system moves its camera (foveates) to the informed positions;

3. The fovea investigates the suspect regions in high resolution.

A good example of an ACVS is presented in [47], where Jurie applies a log-polar mapping to face detection and tracking applications. The system comprises a camera integrated into a mobile robot arm with foveation purposes. To detect faces, the camera first has to foveate the image at some fixation points selected by a histogram comparison algorithm. These points can be situated in the periphery. Then, foveated regions are examined with the aid of an eigenfaces algorithm. If a face is found, tracking proceeds by restarting these same steps. Although periphery information is useful for determining areas where a face might be located, the detection itself requires foveation, and so it might not be suitable for a PCVS.

Traver et al. [87] emphasize the usefulness of peripheral information at detecting objects or events. According to the authors, most works rely on foveal information to do such tasks, but to exploit the data reduction at its best, both regions should be used properly. A multi-model approach for detecting objects in the periphery of log-polar re-sampled images is then proposed. In the periphery, depicted objects get distorted non-uniformly by the varying spatial resolution of retinal images. Hence, it is plausible to consider multiple spatial models to contemplate the distortions at different image positions. A face detection application was chosen to validate the multi-model approach. The authors extracted features from faces and non-faces at different positions of the retinal images and used a linear regression-based technique for classification. The reported average face detection rates were greater than 80%, indicating that just like in the human peripheral vision, it is possible to discriminate face features even at coarse resolutions. However, further analysis would be necessary to evaluate the classification in specific peripheral regions, and to verify the application's efficiency in off- and on-line scenarios. The idea can also be adapted to a PCVS.

The interplay between saliency and foveation has also shown an interesting application in image and video compression. Shi et al. [81] proposed a HSV-based adaptive compression scheme that is suitable for remote sensing images. The process depends on calculating (i) a weighting mask for modeling visual sensitivity, and (ii) the energy of each sub-band from a wavelet decomposition to determine the scanning order among and within all weighted sub-bands. The aim was to encode higher-energy sub-bands and high-valued coefficients first, thereby improving the quality of the reconstructed image. The authors showed promising qualitative and quantitative results, but did not explore the effect of different fixation points to compress the images. Lu and Zhang. [55] combined top-down and bottom-up saliency models to find relevant regions within a scene. Such regions are used to adjust the image's fidelity before performing its compression with standard algorithms. Essentially, the authors foveate video frames by means of a bilateral filtering operator that smooths peripheral visual content while preserving salient (foveal) regions. As a result, further compression actions were more effective due to the prior resolution degradation of perceptually irrelevant regions.

Bektas et al. [17] described a method – based on HSV concepts – to compress geographical images in a perceptually lossless manner, thereby reducing the amount of visual information presented to the eye, and favoring user experience in tasks involving visual interpretation of images. In the experimental evaluation, real users – which had their eyes tracked – were required to perform searching tasks in a sequence of images. The results showed that few participants noticed the degradation artifacts caused by processing the images with the models. Thus, the removal of perceptually irrelevant 2-d data did not considerably impact the efficiency of the visualization task. However, since the evaluated task comprised looking for a circular map symbol in each image, other specific activities, such as identifying and recognizing places, structures, and other contextual elements, were not investigated.

2.3 Space-Filling Curves

Computing 1-d sequential orders for multidimensional data is a recurrent task in scientific computing. This is evidenced, for instance, in the field of computer graphics, where models to describe the geometry of 3-d objects are generated by computing their vertices, edges, and faces. One approach for such generation works by first fulfilling the objects with spatial cells (e.g., small cubes) and then gradually refining (detailing) some areas by inserting more cells. This process can be performed by means of quadtrees (in 2-d) or octrees (in 3-d). Next, finding a sequential order of cells is paramount to further process and update the model. The efficiency of these procedures can be influenced by the computed order. For example, a sequential order that preserves spatial relationships could benefit a search procedure, as neighboring positions corresponding to similar data would remain close to each other in the sequence [13].

Space-filling curves (SFC) map a 1-d parameter space to a higher-dimensional space. The simplest SFCs visit each point in a square at least once. These curves differ, in general, in two ways: (i) in the number of subdivisions required for their gradual expansion and (ii) in the visiting order determined by their elementary patterns, both of which lead to different data sequences. Examples of SFCs comprise the Hilbert, Peano, Sierpinsky, and Gosper curves.

Some SFCs are particularly appealing in data transformations, as the sequential orders they induce favor the preservation of locality properties inherent in multidimensional data. A satisfactory example is in traversing digital images to generate more spatially-coherent sequences of pixels than the ones produced by the traditional raster-scan approach (left to right, line by line). The coherency property implies that every pixel of a neighboring region will be traversed before moving to others in more distant positions. This property is a consequence of the recursive nature of SFCs. Futhermore, different curves provide variable degrees of autocorrelation between traversed pixels [25].

Several domains of scientific computing are contemplated by SFCs, including dimensionality reduction [25, 91], digital halftoning [10, 46, 93–95, 104, 107, 108], visual criptography [54, 80], image compression [40, 41, 53, 75], and data structures and databases [1, 6, 11, 23, 63]. In the following, we present a theoretical basis on the construction of SFCs.

2.3.1 The construction of SFCs

A sequential order procedure generates a mapping from a 1-d array of elements of the form $\{1, .., n^d\}$ into a d-dimensional array of indices of the form $\{1, .., n\}^d$, where n stands for the number entries of the latter. In most problems, however, continuous data need to be dealt with, and a continuous mapping, say from a higher-dimensional unit space $[0, 1]^n$ to the 1-d unit interval $[0, 1]$ should be established [13].

A continuous mapping should be, preferably, *bijective* and *continuous*. The former property defines a one-to-one correspondence between elements of a 1-d and a higher-dimensional array, whereas the latter prevents "holes" to appear in the final sequence by avoiding "jumps" in the parameter (1-d) space. Still, one could ask how such mapping is possible, given that both intervals supposedly have different lengths. In 1878, Georg Cantor showed that, if such a mapping exists, both intervals (he used the general concept of *sets*) must contain the same number of elements. Later, he proved that the 1-D unit interval has as many points as the 2-d unit square, and that there is a bijective mapping function between these sets [13].

In 1879, a step forward was given by Eugen Netto, who proved that a continuous bijective mapping could not exist for smooth manifolds codomains (e.g., the target square). The proof motivated mathematicians to search for *surjective continuous* mappings, as the property of one-to-one correspondences towards ideal continuous mappings had been dropped. The pioneer discoveries in this field comprise the Peano curve, by Giuseppe Peano in 1890, and the Hilbert curve, by David Hilbert in 1891. Later, other curves were found, such as the Moore (1900), Lebesgue (1904), Sierpinsky (1912), and Polya (1913) curves [13, 70].

The Hilbert Curve

Hilbert was the pioneer to provide a geometrical principle for generating a SFC. The Hilbert curve is now considered a prominent one, in that its generation method derived a class of SFCs, and also due to its real-world applications [70]. According to Valgaerts [92], Hilbert's method to create a 2-d SFC can be summarized by three general rules based on a recursive point-of-view:

1. A continuous mapping from the unit interval I to the unit square Ω is assumed. From that, we first partition I into four congruent subintervals of equal size, and Ω into four congruent sub-squares. Next, we continuously map each subinterval to a sub-square. This process is then recursively repeated for each sub-interval and their correspondent sub-squares.

2. We need to preserve the continuity of the mapping along the repetitions by assuring that adjacent sub-squares from Ω will be mapped to adjacent subintervals from I at each stage of the recursive process.

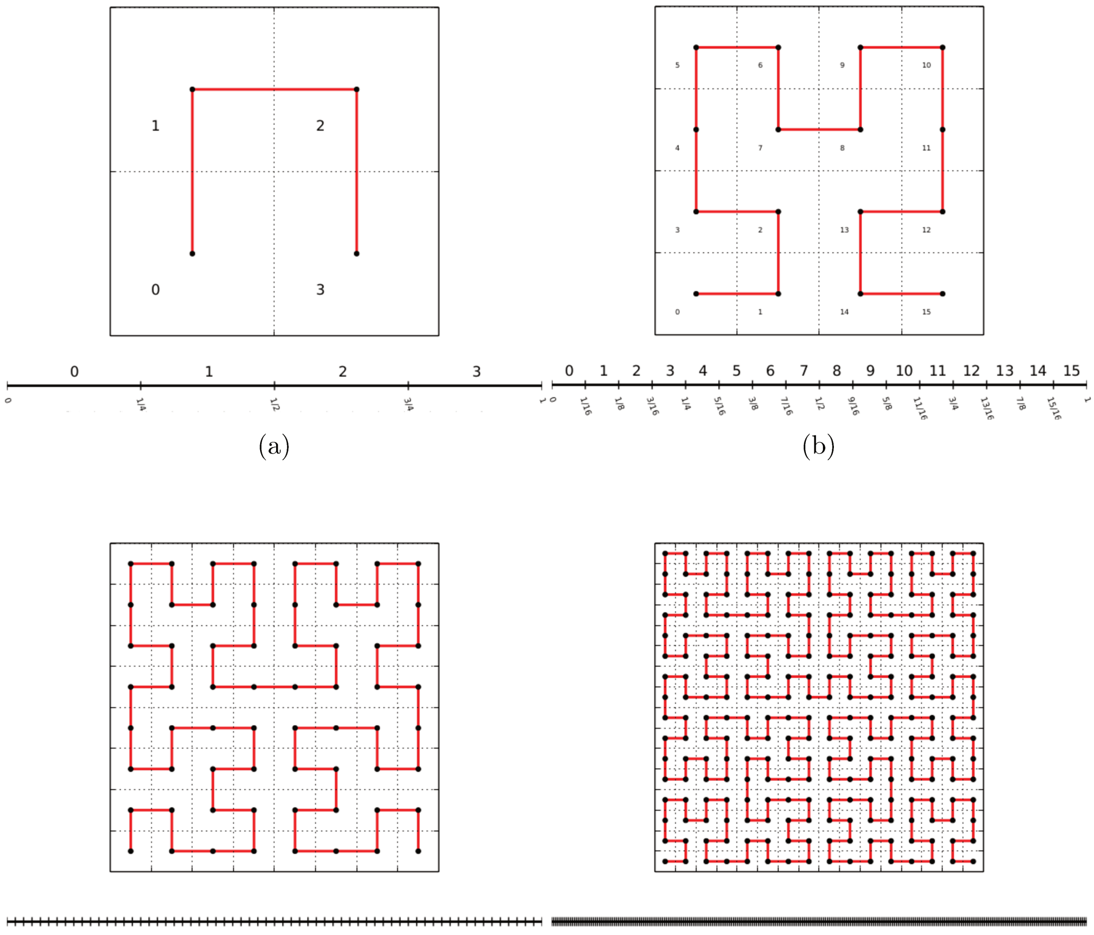
3. Consider an interval I_i that is mapped to a square Ω_i at some *iteration* of the process. A new iteration begins after all squares of the same size as Ω_i have been broken. At a certain point of the subsequent iteration, I_i will be divided into four congruent subintervals, each of which will necessarily correspond to a sub-square from the likewise partitioned Ω_i. By doing so, the mapping from the previous iteration gets preserved.

This procedure can be visualized by connecting the midpoints of each sub-square in the order defined by the sub-interval mapping sequence, as depicted in Figure 2.3. The red path presented in Figure 2.3-a can be understood as the elementary (initial) pattern used to refine the path of the next iterations.

In the n-th iteration, the pattern from the $(n-1)$-th iteration is first replicated 2^n times, the replicas are downscaled, and some of them undergo a 90^o clock- or anticlockwise rotation throughout the image. The replicas are then placed such that the new path preserves the overall direction of the path from the $(n-1)$-th iteration. Each $t \in I$ is taken as the limit of a unique sequence of nested closed intervals. Analogously, every point in Ω belongs to a unique sequence of nested closed squares. As the number of iterations increases, the convergence of the Hilbert curve improves [70, 92].

The Peano Curve

The Peano curve is created by a process analogous to that of the Hilbert curve. In summary, recursive partitions of both the unit interval I and the unit square Ω are done. In



(a)

(b)

(c)

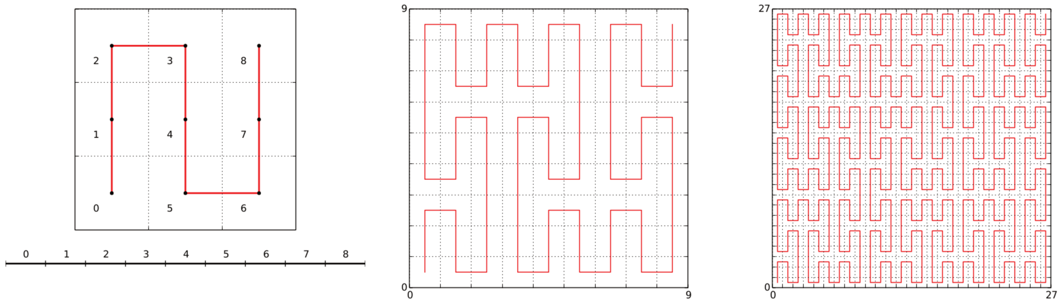
(d)

Figure 2.3: The construction of a Hilbert curve in four different levels and the respective 1-d sequences. Figures (a), (b), (c), and (d) represent the first, second, third, and fourth levels, respectively, with each square having a correspondent 1-d subinterval.

the n-th iteration, 3^n subintervals and 3^{2n} congruent sub-squares of equal sizes are created. It is mandatory that adjacent sub-squares always map to adjacent sub-intervals, thus preserving the continuity of the mapping. Figure 2.4 illustrates this process. Similarly to the Hilbert curve procedure, the generation of the Peano curve can also be seen as a process in which the pattern from the previous iteration is replicated, resized, and fit into the unit square. However, such patterns undergo horizontal and vertical reflections instead of rotations.

2.4 Final Considerations

Space-variant computer vision represents a change of paradigm where image data is severely reduced by a non-uniform re-sampling scheme. Increasingly smaller, but powerful mobile computing devices may take advantage of this paradigm, as they operate with limited energy resources. The field has been explored in the past, mainly in conjunction with



(a) (b) (c)

Figure 2.4: The construction of a Peano SFC in three different levels. Figures (a), (b), and (c) represent the first, second, and third levels, respectively, with each square having a correspondent 1-d subinterval (shown for the first level, only).

Active Systems in which cameras can move along their axes to exploit the foveation idea. Hardware (sensors) and software (retina image models [19]) implementations have been reported and explored, and we emphasized the latter ones due to the flexibility provided.

Log-polar models to re-sample cartesian images play a big role in simulating how the information is sensed in the human retina. Some works tackling graph-based representations [38, 97] for space-variant images were described. Such works turn out to be very important for our research, as they propose image processing algorithms adapted for space-variant image domains with variable grid topology, and pixel sizes and shapes. However, many additional investigations are latent, such as those related to CV algorithms.

In this Ph.D. research, we investigated different approaches for creating image models for different CV applications. We also aimed at combining HVS concepts, such as foveation and visual attention (saliency estimation), and the idea of SFCs, such as the Hilbert curve. The motivation of our exploration comes from the possibility of saving critical computing resources like energy and memory storage, while also keeping satisfactory accuracy rates for the examined CV applications.

Chapter 3

Application-Oriented Retinal Image Models for Computer Vision

This chapter refers to the paper [82] published in the *MDPI Sensors* journal under the title *Application-Oriented Retinal Image Models for Computer Vision*. The chapter contains only a few additional content and other minor differences that do not conflict in any means with the results and discussions shown in the original paper.[1]

In this chapter, we present our first contribution regarding the use of foveal image models in CV applications. The chapter is organized as follows. Section 3.1 introduces the subject. Section 3.2 describes our approach for creating application-oriented retinal image models in detail. Section 3.3 describes the target application we chose to validate our approach and the adopted dataset. Section 3.4 presents experimental details regarding the validation steps, shows the results, and discusses them. Section 3.5 presents the final considerations. The study presented in this chapter is derived from a collaborative project entitled Low-Power Always on Presence Sensors, which has been the initial research point of this thesis.

3.1 Introduction

By means of a conventional sensor, one can easily capture uniform high-resolution images and describe what is depicted. However, for computers, interpreting images is not trivial, demanding complex CV algorithms along with a proper management of the available resources, to allow the software applications to run efficiently in different hardware platforms. As a matter of fact, a computational burden might come into play due to real-time restrictions often imposed by the available hardware to process these high-resolution data [15]. In the mobile environment, for example, managing energy (i.e., battery life) is mandatory, as its negligence might prevent users from enjoying a satisfactory experience [22]. Instead of decreasing uniform-images' resolution or cameras' frame-rate, foveal image models can be used to mimic the space-variant configuration of the human eye. Ac-

[1] Reprinted from Sensors, 20, Ewerton Silva, Ricardo da S. Torres, Allan Pinto, Lin Tzy Li, José Eduardo S. Vianna, Rodolfo Azevedo, and Siome Goldenstein, "Application-Oriented Retinal Image Models for Computer Vision," 3746, 2020. The work was published under the Creative Commons Attribution License (`https://creativecommons.org/licenses/by/4.0/` – As of Oct. 2020).

cording to [15], since some tasks, such as tracking and pattern recognition, do not demand high resolution data across the whole image, it is reasonable to work with space-variant images.

Despite the progress in CV research fields in exploiting space-varying models, there is a lack of a single generic framework for handling seamlessly images generated by heterogeneous pixel sampling strategies. We address this issue by proposing a framework for designing Application-Oriented Retinal Image Models (ARIM) that establish a non-uniform sampling configuration of uniform images. We propose to define the appropriate model for an application on-demand, taking into account specific requirements of the target application. By exploiting such models, we hypothesize it might be possible to decrease the energy spent in computer vision tasks. We show how to create the models and validate their use in a face detection/recognition application, considering the compromise among storage rates, energy, and accuracy. We use a regular image sensor and perform the sampling procedures by means of a software layer, thus simulating the operation of a specific-purpose space-variant sensor and providing some flexibility. Our main contributions are the following:

- we provide a framework for designing ARIMs towards CV applications;
- we evaluate the use of ARIMs in a CV application of the biometry field in terms of memory storage and energy reductions;
- we discuss the trade-offs between the application's accuracy and the reductions in the computing resources induced by the ARIMs;
- we compare our results to other common setups (original and downsized uniform-resolution images) and show that the obtained storage and energy savings are relevant; and
- we briefly discuss the use of ARIMs in real-life application scenarios and the nuances of having an ideal hardware layer that resamples images according to ARIMs.

3.2 Proposed Approach

In this section, we describe our methodology to generate ARIMs. Figure 3.1 shows the steps of the proposed framework, which are detailed ahead. The models represent the expected configuration of space-variant images, as defined by a prior analysis of the CV application's domain. An image resampled by using an ARIM presents variable resolutions across its space. These areas may be exploited in different ways in order to save computing resources and still allow for satisfactory accuracy rates for the desired application. The components of the proposed methodology will be presented in the context of a biometric application.

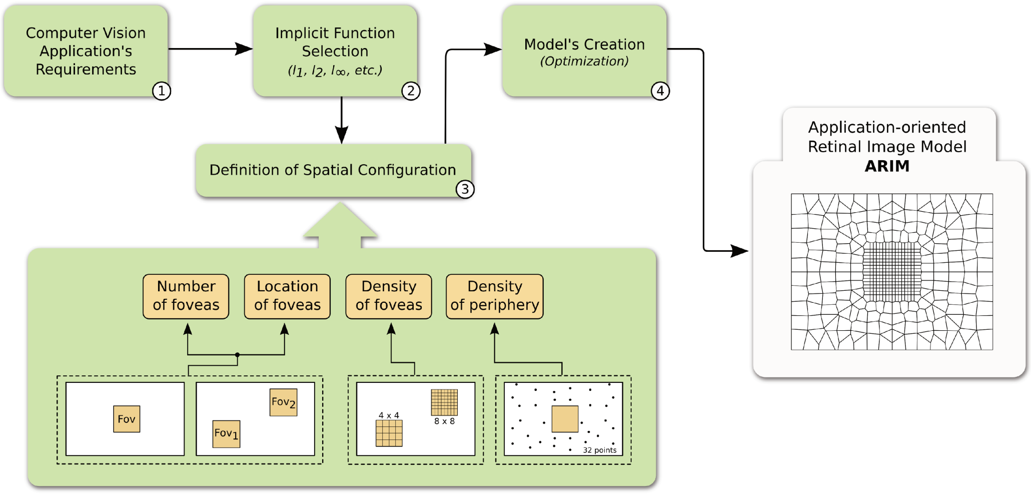

Computer Vision Application's Requirements
1
Implicit Function Selection
(l_1, l_2, l_∞, etc.)
2
Model's Creation
(Optimization)
4
Definition of Spatial Configuration
3
Application-oriented Retinal Image Model
ARIM
Number of foveas
Location of foveas
Density of foveas
Density of periphery
Fov
Fov_2
Fov_1
4 x 4
8 x 8
32 points

Figure 3.1: The proposed framework to generate ARIMs. The workflow begins by defining the application's requirements regarding operation (e.g., objects' position, illumination) and efficiency (e.g., storage, accuracy). Then, a proper implicit function (e.g., l_2) and the spatial configuration of the retinal image model – comprising foveal and peripheral regions – are chosen. The next step is the generation of the model by means of an optimization procedure that considers the implicit function and the spatial configuration to resample points in the 2-d cartesian space. The final artifact is an ARIM comprised of uniformly- and non-uniformly-sampled foveal and peripheral regions, respectively. This model is used to resample uniform images, taking them to a space-variant domain and potentially contemplating the requirements determined beforehand.

3.2.1 Definition of Application Requirements

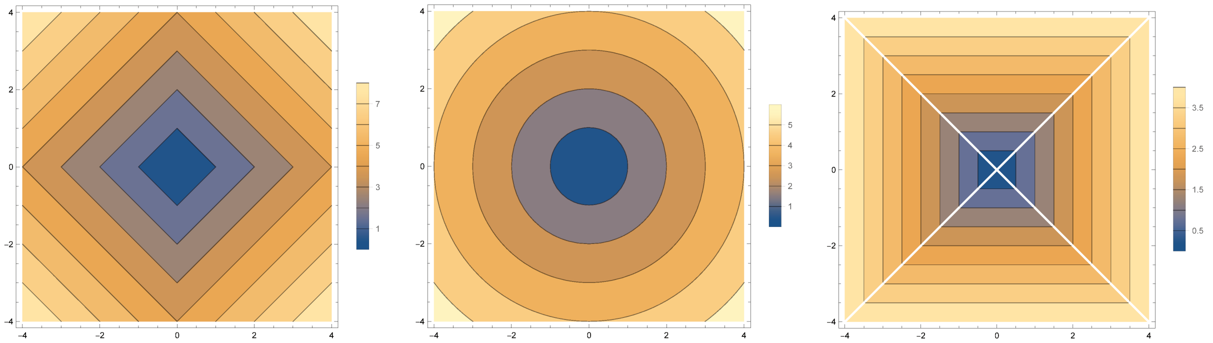
Instead of using a traditional image, coming from a general uniform sensor, we argue that the best approach is to examine the target application and investigate its requirements and demands. CV applications can comprise a very diverse set of requirements, ranging from efficiency-related ones, such as storage, speed, energy, and accuracy, to other very application-specific ones, such as the need for objects to move slowly or be positioned in specific locations in the scene, be situated in a minimum/maximum distance from the camera, be illuminated by a close light source, and so further. The application considered in this thesis is concerned with user authentication based on his/her face: the individual enters and leaves the scene by any sides, placing himself in front of a camera that captures the scene in a wide field of view.

Although the authentication across a wide field of view is a good idea, since more faces are collected throughout the video, the central part of the image is usually the protagonist of the process. In this vein, it is recommended that the individual stands or walks near the center of the image to properly position his/her face (e.g., to avoid severe rotations and perspective changes) for a more accurate authentication process. Thus, if one intends to reduce energy consumption, collecting faces only in a bounded central region (e.g., a square window) might be enough. On the other hand, restricting the image to its central part, albeit effective, might be seen as a very extreme decision, since other image areas may contribute with useful information for the authentication. In this sense, retaining some pixel data in such areas, even in a sparse manner is also appropriate. Finally, another suitable strategy towards energy reduction is downsampling the image before performing face detection/recognition. This might reduce the energy spent in the whole authentication process, but at the cost of a drop in accuracy.

The issues discussed above illustrate examples of requirements to be defined by the analysis of an application's domain. In this thesis, they were essential to guide the definition of a model for the biometric application.

3.2.2 Implicit Function Selection

The design of the model starts with selecting a proper implicit function. The idea is that the function will act as a control mechanism to spread out the non-uniform sampled points over a desired image region. Figure 3.2 depicts examples of implicit functions we explored (l_1, l_2, and l_∞).



(**a**) (**b**) (**c**)

Figure 3.2: Examples of implicit functions: (**a**) l_1, (**b**) l_2, and (**c**) l_∞.

3.2.3 Definition of Spatial Configuration

This step is concerned with the spatial characteristics the model must obey. We developed hybrid space-variant models inspired by the human retina. In general, the models comprise two very distinct regions: the fovea and the periphery. The fovea is a fixed-size region of uniformly sampled pixels according to a predefined grid. For instance, a region of size $2^6 \times 2^6$ pixels can be uniformly sampled by a grid of size $2^5 \times 2^5$ pixels. Given these characteristics, we can apply conventional CV algorithms in the fovea. In opposition, the periphery is a fovea-surrounding region with a non-uniform pixel density that decreases with the distance from the fovea.

The following four parameters should be informed prior to the creation of the hybrid model:

- **Number of foveas**: Surely a human eye has only one fovea, but it is perfectly fine for a model to comprise more than one region of uniform sampling, depending on the application on hand. In our biometric application, we took into account only one fovea.

- **Location of foveas**: The foveas should be spatially organized adhering to the specific requirements of the application. In ours, the fovea is centralized in the image.

- **Density of foveas**: The foveas can be downsampled to simulate a uniform image resolution reduction. We tested different densities (grids) for our fovea.

- **Density of periphery**: The periphery is an important region that encompasses few sparse data in a non-uniform sampling configuration. As discussed previously, by retaining and wisely handling sparse peripheral information (e.g., detecting motion and coarse objects in such an area), the application's resource usage might be optimized.

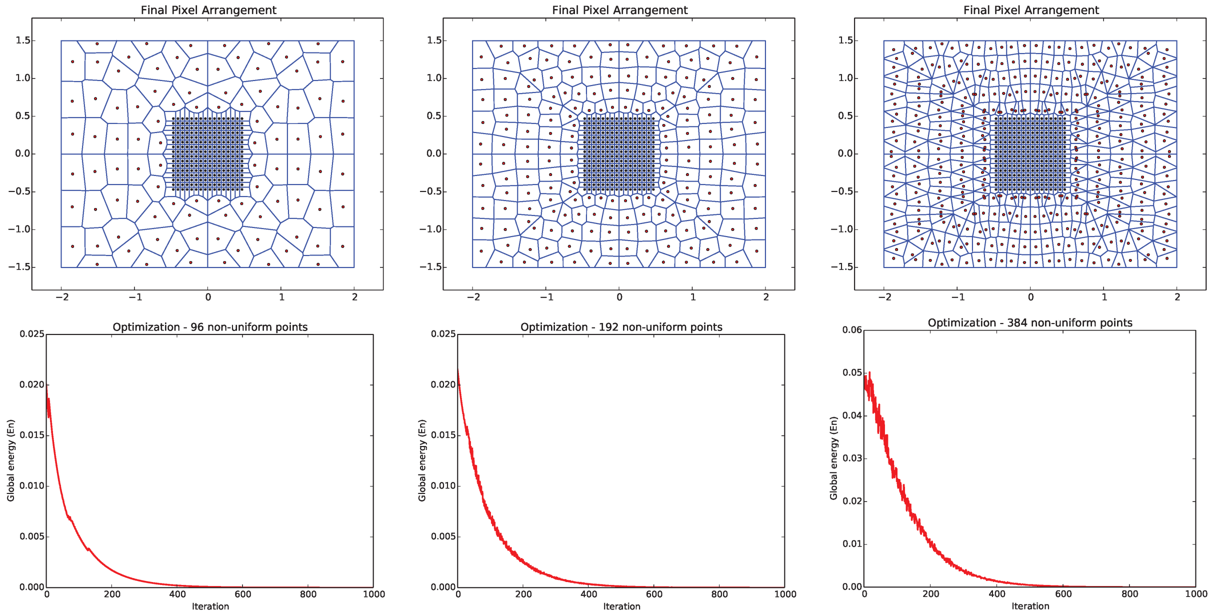
3.2.4 Model Generation

There are several ways to achieve a non-uniform point distribution. Our approach is inspired by the computer graphics literature and previous works [27, 34]. Besides the implicit function, the number of peripheral (non-uniform) points and the aspect ratio of the sensor must be provided. We generate a point distribution via a local non-linear optimization procedure that, from an initial distribution, tries to minimize a global energy function defined in Equation 3.1, where $\vec{x}$ is a point in image space.

$$En\left(\{\vec{x}_i\}\right) = \sum_i \sum_{\vec{x}_j \leftrightarrow \vec{x}_i} \left(||\vec{x}_i - \vec{x}_j|| - (f(\vec{x}_i) + f(\vec{x}_j))\right) \tag{3.1}$$

The optimum solution for Equation 3.1, i.e., when $En = 0$, would be a placement of every $\vec{x}_i$ such that the distance to its "neighbors" is the sum of the values of the implicit function at their locations. However, there is neither a closed-solution for this problem (the implicit function can be anything), nor any guarantees of a perfect solution for a scenario with an arbitrary number of points and implicit functions. Thus, we propose an approximation by means of a non-linear optimization procedure based on *Mass-Spring Models*. When doing so, each pair of points tries to attract each other if they are too far, and tries to repel each other when they are too close. We do not use Newton's physical model of forces from springs. Instead, we have a mass-free system, so springs generate "velocity forces." Figures 3.3 and 3.4 show the behavior of the global energy optimization for models with different configurations, implicit functions, and number of points. In Appendix A, we present the algorithmic steps for the generation ARIMs.

The optimization process is very sensitive to its initial conditions. A uniform distribution of the initial positions over the valid domain coupled with a careful choice of the implicit function allows the system to converge under 2000 iterations. Figure 3.5 illustrates the generation of an ARIM where the optimization of uniform point distribution is carried out using the l_∞ implicit function. Upon convergence, we obtain the full neighborhood map (Voronoi diagram) of the model.



(a) (b) (c)

Figure 3.3: Convergence analysis for ARIMs with a single fovea and based on the l_∞ implicit function. Examples of ARIMs containing (a) 96, (b) 192, and (c) 384 non-uniform points in the periphery. For each ARIM, the resulting global energy curve over 1000 iterations of the generation process is shown in the model's respective column.

3.3 Materials and Methods

In this section, we present the experimental setup that is necessary for simulating the usage of the proposed models. The chosen dataset closely resembles one of a biometric application.

3.3.1 Target Application: Face Detection / Recognition

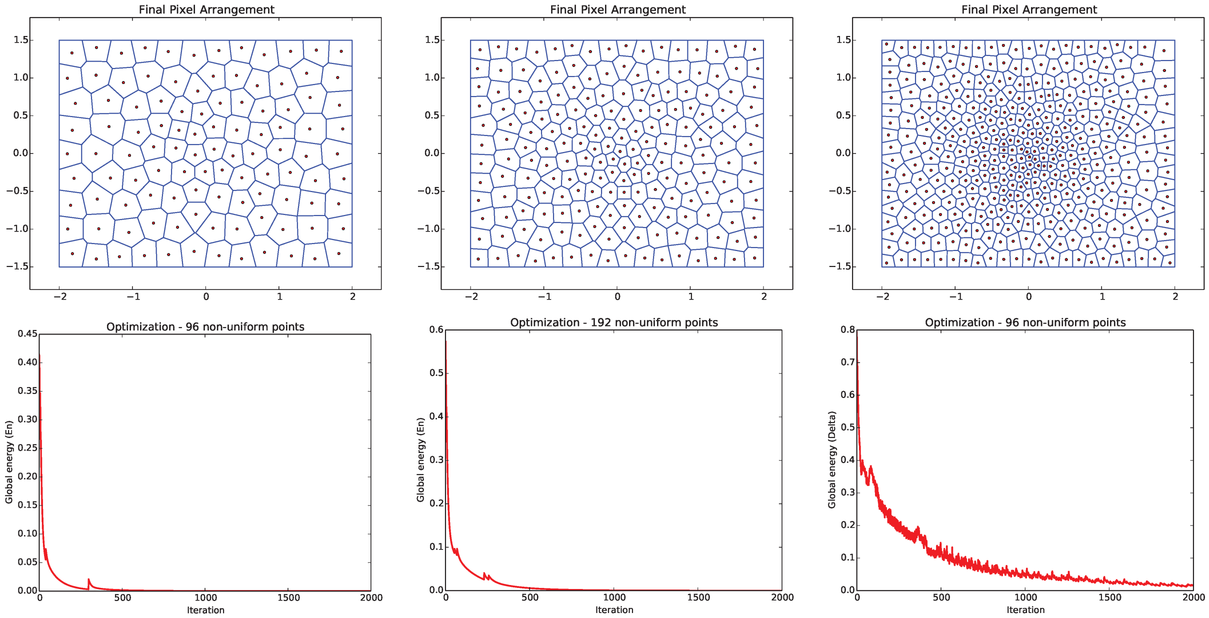
In this thesis, we selected a CV application from the biometry domain to evaluate the proposed framework.

On the Application Selection

Although the framework may be appropriated to many CV applications, such as those that are related to surveillance and remote sensing, the biometry domain is characterized by well-consolidated techniques and datasets, due to the several studies in the area over the years. Moreover, the considered application could fit the complete process that is described in Section 3.2. In this case, the application's characteristics and requirements could be examined by reasoning about each step of the framework (Figure 3.1), as follows:

- Step 1: We analyzed the CV application's demands and characteristics. In the considered biometry application, we observed aspects regarding:

  1. the use of computational resources, which should be preferably low when running in environments of strict energy and storage limitations;



(a) (b) (c)

Figure 3.4: Convergence analysis for ARIMs with no fovea and based on the l_2 implicit function. Examples of ARIMs containing (a) 96, (b) 192, and (c) 384 non-uniform points in the periphery. For each ARIM, the resulting global energy curve over 2000 iterations of the generation process is shown in the model's respective column.

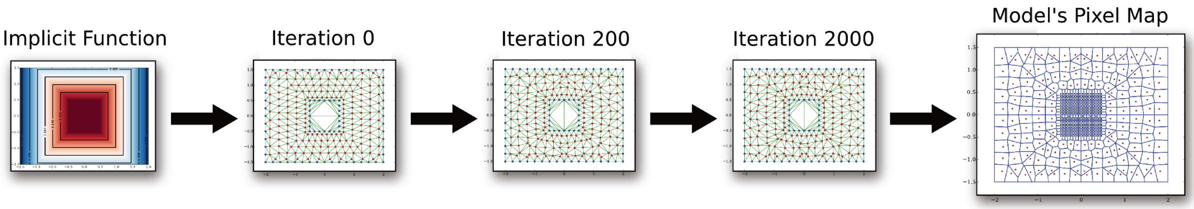


Figure 3.5: The evolution of an example of ARIM with 256 foveal (uniform), and 192 peripheral (non-uniform) pixels. The l_∞ is the implicit function.

2. the intrinsic characteristics of the application's domain, such as the task to be executed (face authentication), the expected "behavior" of the input data (person movement and positioning in the images), camera angles, the most relevant part of the image to process, etc.;

3. the possibility of balancing the pixel density of different image regions. In this sense, given the application on hand, we decided that the processed image would have different resolutions across its space. This will induce a compromise between energy, storage, and accuracy; and

4. the possibility of adopting distinct pixel representations across the image in order to save computational resources. In the current case, an additional motion analysis is performed by taking advantage of an optical flow pixel representation in some image regions.

- Step 2: subsequently, by the previous analysis, we selected an appropriate implicit function to represent the pixel distribution of the image;

- Step 3: next, we defined the spatial configuration of foveal and peripheral regions by knowing, for instance, that individuals often move to the central part of the image to allow a better authentication. In this case, we defined a single central fovea;
- Step 4: finally, we created ARIMs encompassing and consolidating the expected properties of the images defined in the previous steps.

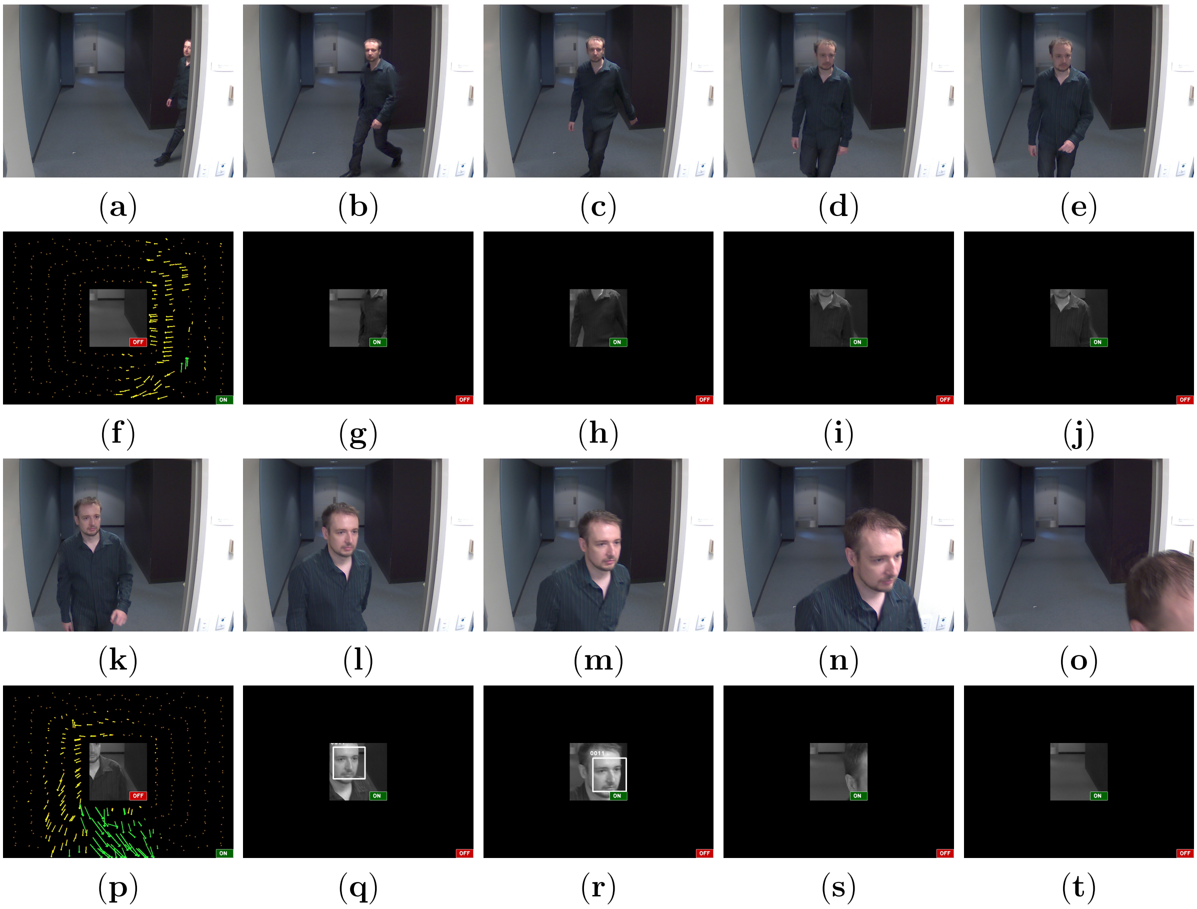
On the Application Implementation

The face recognition process is based on the classical nearest neighbor strategy. First, we trained our classifier with some face samples from a training dataset containing different individuals. This process required a face recognition model to extract feature vectors from all samples. Then, in the test stage, we extracted the feature vector from an unknown detected face and compared such vector to all the others from the training dataset (previously extracted) using the Euclidean distance metric. The label of the closest face was considered as the label for the unknown face, thus resembling a 1-Nearest Neighbor (1-NN) strategy. Section 3.3.1 presents more details on the face recognition model used.

Simulation Details

We simulated the operation of a specific-purpose sensor by re-sampling images according to our ARIMs. The idea was to generate images containing two regions: (i) the fovea, encompassing a small area where resolution is uniform, and (ii) the periphery, where pixels are arranged non-uniformly over a wider area. With such a configuration, we were able to perform experiments when considering different foveal resolutions, while also taking advantage of the periphery according to the specific requirements of the application. In this sense, we adopted an optical flow representation (orientation and magnitude) for peripheral pixels. The motivation around that representation is that the detection/recognition in the fovea could only be triggered when there is movement towards it coming from the periphery. Additionally, both the detection and recognition procedures turn off when no face is found under a predefined time interval. Therefore, in this scenario, more energy can be saved.

Figure 3.6 shows an example of a simulation using one of our ARIMs and a sample sequence from the employed dataset [105]. The first and third rows show the original frames, while the second and fourth rows show images reconstructed with a model that considers an optical flow peripheral representation. Green and yellow arrows indicate motion direction to the right and left sides, respectively, whereas the ON and OFF labels refer to the operational status of the foveal (face detection/recognition) and peripheral (optical flow) regions. Besides triggering foveal analysis, the motion analysis is also able to restart conveniently, as long as faces are not detected in the fovea during a time interval of frames (see Figure 3.6(p)). Additionally, for a visual comparison with log-polar images (which are not considered in our experimental evaluation), see Appendix B.


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

Figure 3.6: Example of a simulation using one of our ARIMs and a sample sequence from the employed dataset [105]. Images (**a**–**e**) and (**k**–**o**) are the original frames; images (**f**–**j**) and (**p**–**t**) are the reconstructions with a model that considers an optical flow peripheral representation. Green and yellow arrows indicate motion direction to the right and left sides, respectively, whereas the ON and OFF labels refer to the operational status of the foveal (face detection/recognition) and peripheral (optical flow) regions. Note that the motion analysis, besides triggering foveal analysis, is also able to restart conveniently, as long as faces are not detected in the fovea during a time interval of frames (left-most frame in the fourth row).

Ideally, an ARIM should be first computed by software in an offline step. Subsequently, at the application's run-time, the computed model should be an input to a reconfigurable hardware layer that will extract the necessary pixel information from a full-size image captured by a conventional uniform image sensor. Additionally, optical flow should be computed only for the peripheral points, thereby further discarding more image data. As foveal and peripheral pixel configurations do not change often for a specific application, the computational cost to control this discarding procedure by hardware should be low. All the steps of this procedure are summarized in Algorithm 3, in Appendix C.

The workflow of the simulation process is depicted in Figure 3.7, where we distinguish between the software and hardware layers to illustrate a hypothetical case where a specific-purpose (space-variant) sensor was available. In an ideal scenario, the ARIM, a captured image frame, and the chosen pixel representations for foveal and periphery areas are input to an hypothetical specific-purpose sensor that changes its configuration at run-time. Both layers are connected by a 1-d vector (named as bytestream) that stores the foveal and peripheral pixel values captured by the sensor (i.e., the sampled image), and are input to the application. For simulation purposes, however, this architecture is fully implemented by software. We adopted bytestreams instead of a two-dimensional (2D) image representation in the software simulation to bring the process closer to the ideal conceived scenario. The simulator was implemented in C++ using the OpenCV 3.0.0 library.

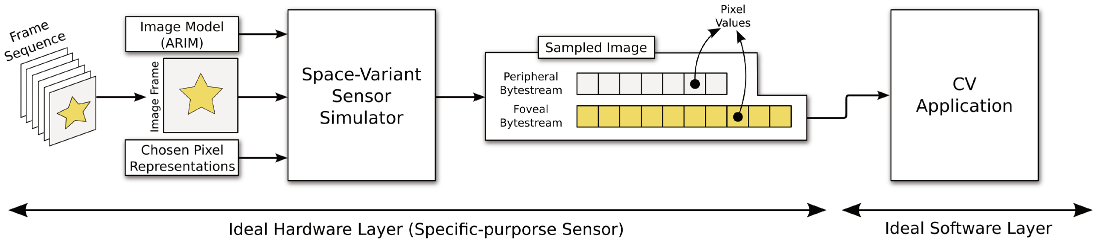


Figure 3.7: Implemented workflow for simulating the use of ARIMs in a specific Computer Vision (CV) application. In an ideal scenario, the ARIM, a captured image frame, and the chosen pixel representations for foveal and periphery areas are input to an hypothetical specific-purpose sensor that changes its configuration at run-time. Such a sensor would yield a stream (bytestream) of pixel data from each region of the captured image. The stream (not the 2D image) would be forwarded to the CV application. However, for simulation purposes, this architecture is fully implemented by software.

Technical Information

The biometric application uses the Viola–Jones [96] algorithm, which is a well-consolidated and widely used face detection method in the literature. As for recognizing faces, we used a descriptor based on a pretrained Deep Neural Network (DNN) model, which is essentially a ResNet network with 29 convolutional layers (based on [39]) trained on a dataset containing approximately 3 million faces. The model is publicly available and integrates the Dlib C++ Library [49]. These and other technical information, such as the methods and parameters used, are displayed in Table D.1, in Appendix D.

3.3.2 Dataset

In our evaluations, we employed the ChokePoint dataset [105] aimed at person identification/verification. The dataset comprises 48 sequences of images of 800 × 600 pixels resolution and is publicly available. Each sequence depicts several individuals entering or leaving a portal, one at a time. There are 25 and 29 individuals walking through portals 1 and 2, respectively. Moreover, each sequence is registered by three cameras that are placed above the portals to provide diverse sets of faces in different illumination and pose conditions. Due to the adopted settings, one of the cameras is able to capture image sequences of near-frontal faces. Figure 3.6 shows a sample sequence from the employed dataset in the first and third rows.

Justification for the Selected Dataset

The dataset was chosen because it represents a satisfactory scenario where a CV biometry application may take place via a detailed investigation of its characteristics and requirements. When an individual is about to cross the portal, his/her face gets well centered in the image, providing the application with the necessary data to perform the biometry procedures. Furthermore, individuals do not suddenly appear in the center of the images; they slowly move towards the portal. This peripheral movement data could be exploited to activate the authentication in the central region of the image (where a face is supposed to be).

Dataset Organization

The dataset is partitioned into the following four subsets:

- P1E and P1L: the subsets of frame sequences of people entering and leaving portal 1, respectively;

- P2E and P2L: the subsets of frame sequences of people entering and leaving portal 2, respectively.

A subset is comprised of four (4) frame sequences (S1, S2, S3, and S4), each of which is registered by three cameras (C1, C2, and C3). For instance, the frame sequence P1E_S2_C3 refers to the second sequence (S2) of people entering portal 1 (P1E) and captured by camera 3 (C3).

We used 34 image sequences (out of 48) from the dataset during our evaluations due to the following reasons:

1. One (1) of the sequences of individuals entering a portal (P1E_S1_C1) was used to train the face recognizer. Such sequence comes from camera 1, which obtains near frontal-face images. That sequence is also captured by cameras 2 and 3 at different angles, hence, to avoid biased evaluations, we ignored such sequences (P1E_S1_C2 and P1E_S1_C3), as both of these contain, essentially, the same faces of the former up to slight angle variations.

2. Eleven (11) sequences where no face is found in the fovea were ignored. This decision was taken because no face recognition accuracy evaluations (using our models) would apply to these sequences.

3.3.3 Evaluated Models

We evaluated three different ARIMs. Each model comprises 384 non-uniform peripheral points and a central foveal region of size 200 × 200 pixels. The models diverge from each other in the uniform-sampling configuration sizes adopted for their foveas, which are 100 × 100 (half density – Model_1), 150 × 150 (75% density – Model_2), and 200 × 200 (full density – Model_3). These settings allow for us to simulate different foveal resolutions. For all models, optical flow peripheral information is used to trigger the face detection/recognition in the fovea. Figure 3.8 shows an illustration of the pixel map of these models and their configurations.

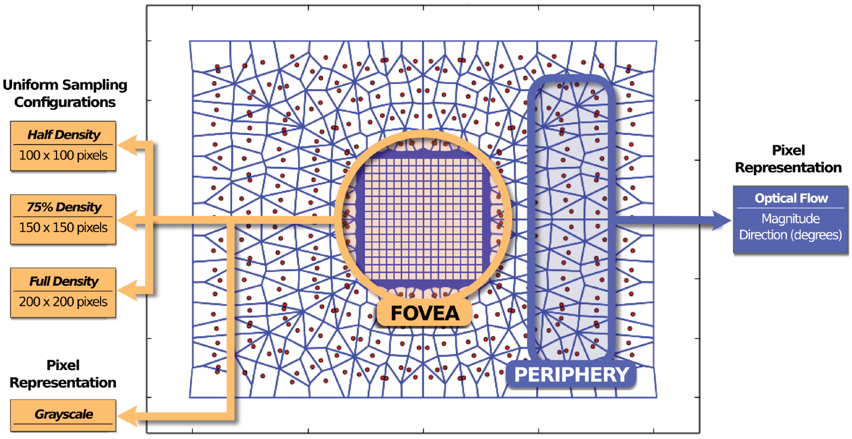


Figure 3.8: The pixel map of the evaluated ARIM and its configurations. The experimented foveal configurations comprised three uniform sampling setups: 100 × 100 (half density – Model_1), 150 × 150 (75% density – Model_2), and 200 × 200 (full density – Model_3) pixels. The pixel representations for the fovea and periphery were based on the grayscale and optical flow (magnitude and direction) values, respectively.

3.3.4 Evaluation Criteria and Hardware Setup

We compared the storage usage by computing the amount of bytes for storing the video, measured the energy spent (in Joules) in the biometric application for each evaluated model, and computed the mean recognition accuracy of each evaluated model when considering all video frames. To measure energy, we used the Intel RAPL (Running Average Power Limit) interface [48], which is a set of internal registers from Intel processors, called model specific registers (MSR). At the code level, we read these registers before and after a block of instructions, and calculate the difference between these values. More specifically, we read the MSR_RAPL_POWER_UNIT register to measure the energy spent in image readings, face detection/recognition procedures, and optical flow analysis (when using ARIMs). The hardware setup to perform the experiments comprised an Intel Core i7-5500U, with 2.04 GHz clock, 4 MB cache, and 16 GB RAM.

3.4 Results and Discussion

In this section, we present the experimental results regarding storage allocated, face recognition accuracy, and energy consumption induced by different ARIMs. We also discuss these results and their implications for real-time applications.

3.4.1 Storage reduction

Quantifying reductions in numbers of pixels and image data sizes are essential for assessing the benefits of using different ARIMs in practical situations. Table 3.1 shows these measurements for the original (full-size) images, images uniformly resized, and the three evaluated models. Compared to original images, ARIMs showed a reduction of more than 91% in the number of pixels and bytes, whereas a uniform resize of the images to 25% of their original sizes provided a reduction of 75% in both quantities.

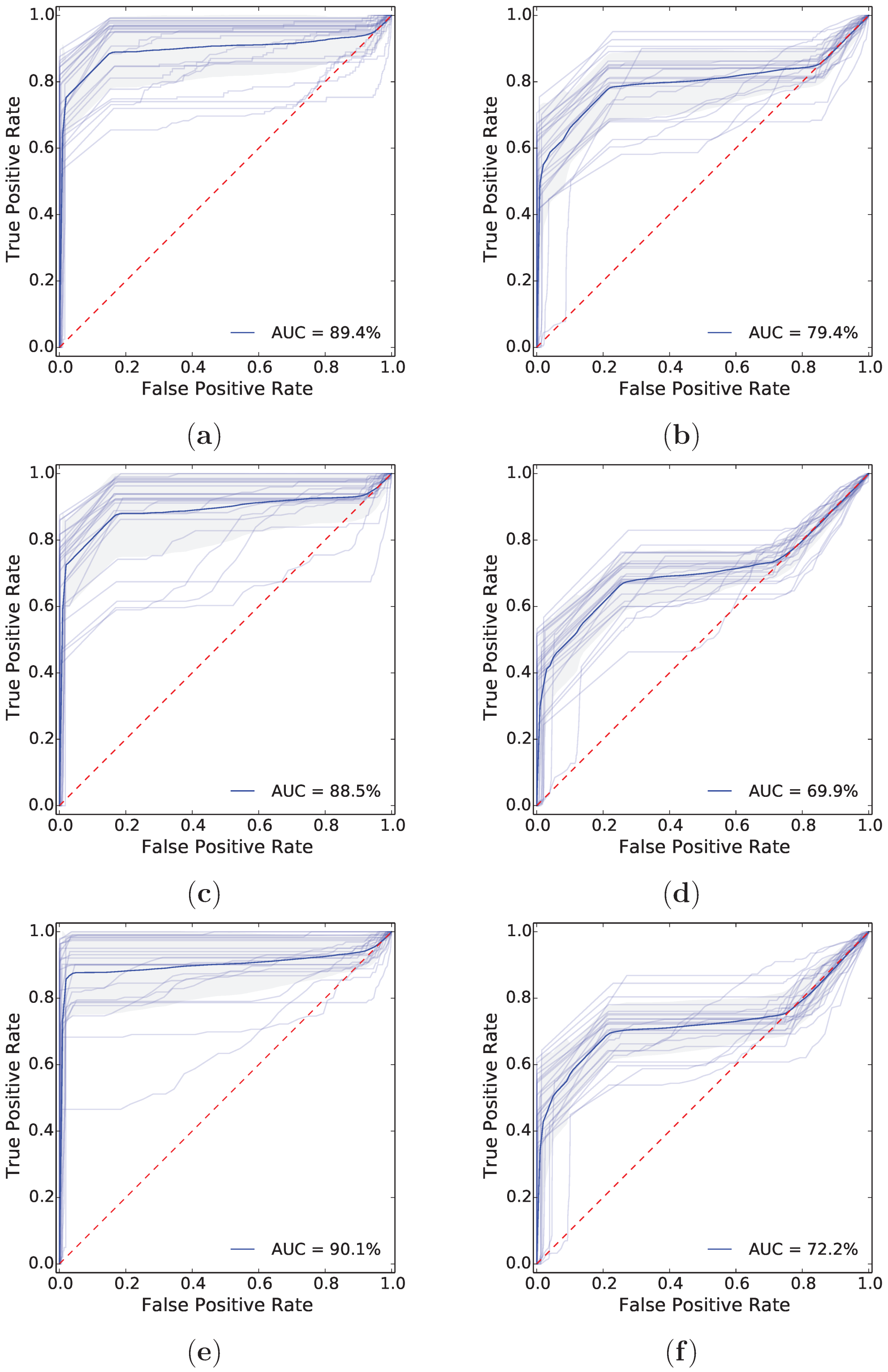
Table 3.1: Number of pixels and data size reduction results for the evaluated models relative to the original and uniformly resized images (to 25% of their original sizes).

	Num. of pixels	**Num. of pixels reduction**	**Bytes per region**		**Total bytes**	**Data size reduction**
			FOV	**PER**		
Original	480000	-	-	-	1440000	-
Resized (25%)	120000	75.00%	-	-	360000	75.00%
Model_1	10384	97.83%	30000	768	30768	97.86%
Model_2	22884	95.23%	67500	768	68268	95.25%
Model_3	40384	91.58%	120000	768	120768	91.61%

3.4.2 Face recognition accuracy

We defined accuracy as the number of true positives (i.e., correctly labeled faces) in the foveal region of a frame sequence, each of which has a benchmark for comparison. The employed dataset informs all faces appearing in each image frame. However, for a fair accuracy comparison among the uniform images and the ones re-sampled by our models, we take into account only available information regarding the foveal region, meaning that faces in the periphery are not considered. Thus, for the present analysis, the reference benchmark of each frame sequence can be understood as the accuracy of the recognizer considering all of those full-resolution images in which a face is placed on the fovea.

First, our face recognizer alone has satisfactory accuracy. Figure 3.9 shows receiving operating characteristic (ROC) curves regarding the face recognition task considering six image sequences from all cameras. Each figure comprises a mean ROC (blue) curve from all 25 (light blue) class-specific curves, i.e., for each face class in the dataset. These class-specific ROC curves were calculated via a one-versus-all classification procedure and evaluate the accuracy of our classifier in the experimented dataset (i.e., full-size images). We did not consider any models to generate the curves. For example, the area under the curve (AUC) is greater for the P1E_S2 dataset (89.4%) in comparison to the P1L_S2 dataset (79.4%). This is possibly because P1E_S2 sequences share similar traits to the P1E_S1_C1 sequence used to train the face recognizer.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.9: ROC curves regarding the face recognition task considering the image sequences (**a**) P1E_S2, (**b**) P1L_S2, (**c**) P1E_S3, (**d**) P1L_S3, (**e**) P1E_S4, and (**f**) P1L_S4 from all four cameras. The figures comprise a mean ROC (blue) curve from all 25 (light blue) class-specific curves, i.e., for each face class in the dataset. These class-specific ROC curves were calculated via a one-versus-all classification procedure.

Figures 3.10 and 3.11 show, for each dataset, an expected face recognition accuracy decreasing of our ARIM-resampled frame sequences compared to their correspondent benchmarks and to images uniformly resized to 25% of their original sizes. ARIMs rely on movement analysis to authenticate users, which creates a dependency between peripheral and the analysis of foveal information, some faces can be lost. Another variable influencing the accuracy rates is the foveal resolution of each tested ARIM. In fact, the accuracy rates increase with foveal resolution, and are not too low even under the 50% sampling degradation induced by Model_1, for example. In the case of Model_3, where foveal resolution matches that of the benchmark, the small loss in accuracy is justified by the quality of optical flow analysis, which seems to be acceptable for the tested application. Table 3.2 presents the minimum, mean, and maximum accuracy loss rates induced by each model in comparison to the benchmarks. Whereas the maximum obtained loss was 50% for Model_1 and the P2E dataset, very small loss rates (close to 0%) were registered in more than one scenario. Another interesting phenomenon is the high loss rates observed for the P2E and P2L datasets, possibly due to slight divergent lighting and pose conditions relative to the P1E and P1L datasets.

The accuracy results on resized images are often lower than those of models, showing a constant behavior on the P1L, P2E, and P2L datasets. In the P1E dataset, however, the accuracy for resized images were unexpectedly high in some cases. We believe this behavior may be justified by the fact that the evaluated sequences and the face training sequence share similar conditions (e.g., lighting). Additionally, the superior accuracy results of the P1E_S2_C2 and P1E_S3_C3, compared to the benchmark, may be due to a removal of noise as a consequence of the huge resizing operation. Nevertheless, we believe these rare cases do not conflict with our general results and conclusions.

The accuracy results for the P1L, P2E, and P2L datasets follow an expected tendency that the values increase with the resolution of the foveal region. In addition, the results for the downsized images were also very low compared to those of the tested models (being zero in some cases), possibibly due to (i) difficulties in finding faces, and to (ii) wrong classifications of the face recognizer, as in the P2E_S4_C1, and P2L_S3_C2 sequences. Another interesting observation is that Model_1 shows much higher accuracy results than the downsized images, thereby indicating that even a model with 50% foveal resolution can be effective in the considered scenario.

Table 3.2: Minimum, mean, and maximum accuracy loss rates induced by our ARIMs compared to the provided benchmarks. For example, the minimum accuracy loss induced by Model 1 on the P1E dataset was 0.032 (i.e., 3.2%), whereas the mean and maximum accuracy loss in that same scenario were 0.123 (12.3%) and 0.264 (26.4%), respectively.

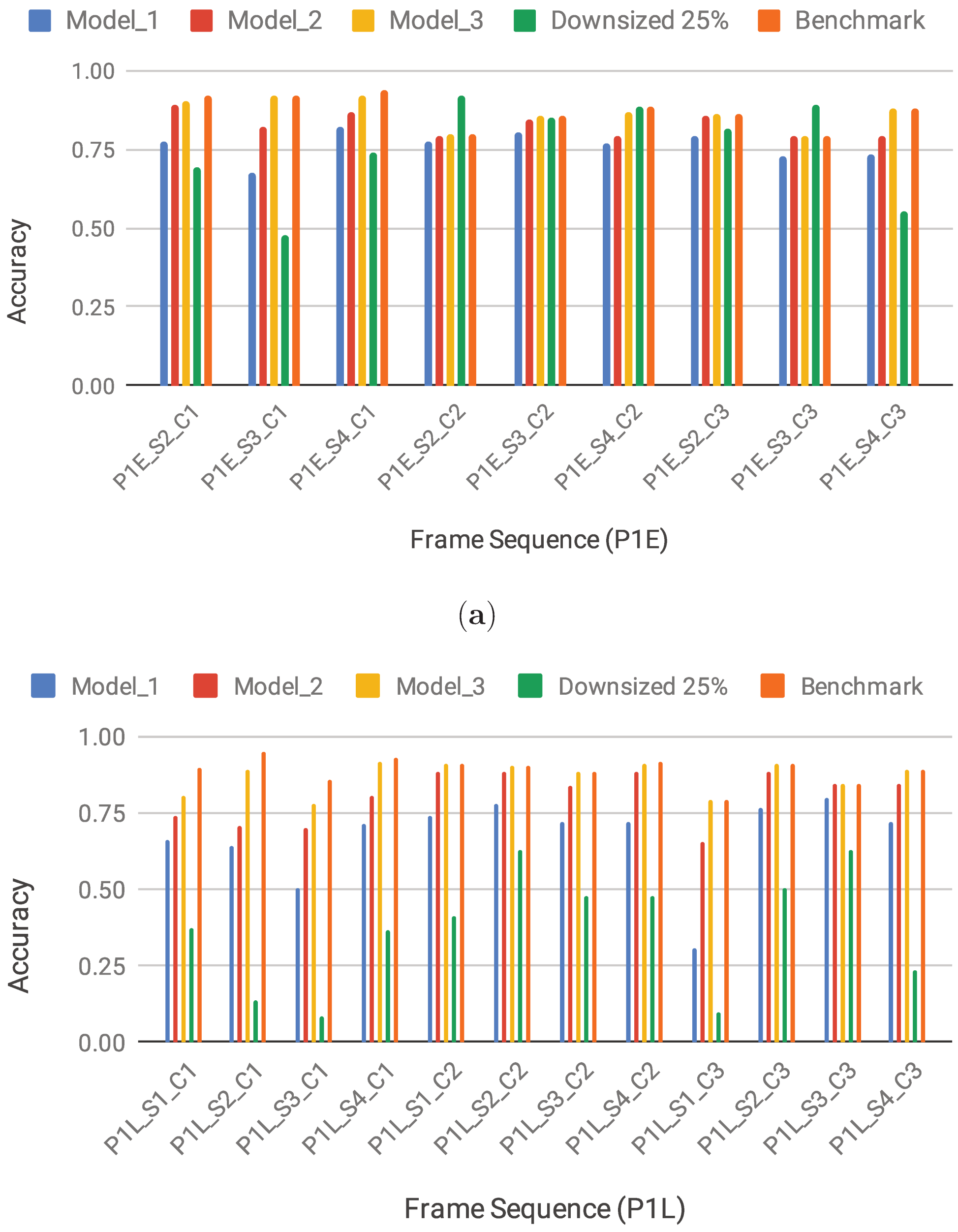
Dataset	**Accuracy Loss**								
	Model 1			**Model 2**			**Model 3**		
	Min.	**Mean**	**Max.**	**Min.**	**Mean**	**Max.**	**Min.**	**Mean**	**Max.**
P1E	0.032	0.123	0.264	0	0.050	0.108	0	0.006	0.021
P1L	0.060	0.248	0.613	0	0.094	0.255	0	0.023	0.103
P2E	0.174	0.353	0.500	0.032	0.172	0.318	0	0.006	0.037
P2L	0.143	0.300	0.529	0.033	0.086	0.265	0	0.063	0.206

3.4.3 Energy consumption evaluation

Figures 3.12 and 3.13 present, for each dataset, a comparison among the total energy spent by the processing of original images (benchmark), images uniformly resized to 25% of their original sizes, and the three tested models. Here, the benchmark can be understood as the full-resolution image frames. Thus, in the energy comparison benchmark, the biometry procedures are executed over the whole full-resolution images. The experiments show lower energy consumption values for scenarios involving our models. The difference in energy values among our models and the benchmark comes directly from the data amount reduction caused by the combination of peripheral optical flow and the sampled foveal face detection/recognition. The robust and timely activation/deactivation of these latter algorithms, therefore, reduce the total energy spent in the whole authentication process, while keeping accuracy rates acceptable, as previously discussed. Table 3.3 presents the minimum, mean, and maximum energy reduction rates induced by each model relative to the benchmark, i.e., the obtained energy savings. As expected, the reduction rates decrease with the increase in foveal resolution, because there is more data to process. This is verifiable by a quick comparison between the mean rates of Model_1 (half density) and Model_3 (full density), for example. For resized images, the energy values are often higher than the ones produced by ARIMs, the exception being the P2L dataset where similar values were found. This result strengthens our hypothesis that the use of ARIMs might lead to energy savings.

3.4.4 Implications in real-time applications

Real-life imaging applications have a broad spectrum of possibilities, associated technologies, and challenges. These applications often need to capture and process high volumes of data – materialized in uniform images – in real-time to provide their users with the desired outputs. However, the users' experience may be affected by these high processing demands in some degree, and should be considered during the application's conception. On one hand, in visual entertainment applications, users want the most possible amount



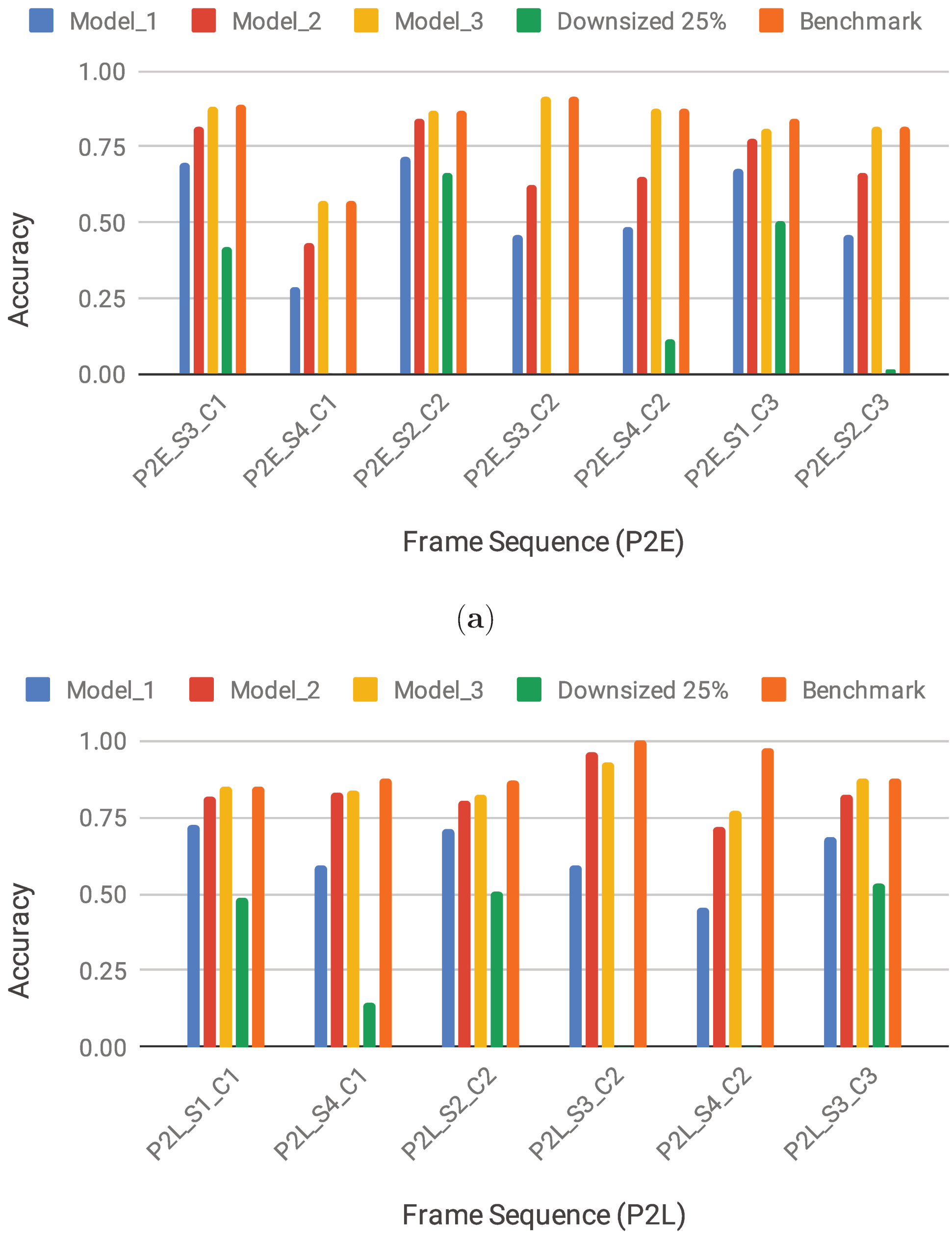
(**a**)

(**b**)

Figure 3.10: Mean face recognition accuracy regarding each evaluated model, the images resized to 25% of their original sizes, and the benchmark images from the (**a**) P1E and (**b**) P1L datasets.

of details in pictures and videos; such a requirement can be accomplished by using uniform images. On the other hand, when the final objective is related to real-time authentication, movement analysis, and action recognition, for example, non-uniform images might come into play to provide an interesting balance among accuracy, storage resources, and energy, thus possibly favoring the users' experience.

In our simulations, the combination of uniform and non-uniform areas induced by our ARIMs allowed significantly fewer data to be processed. Besides, we noticed that



(**a**)

(**b**)

Figure 3.11: Mean face recognition accuracy regarding each evaluated model, the images resized to 25% of their original sizes, and the benchmark images from the (**a**) P2E and (**b**) P2L datasets.

the use of different pixel representations may be a good strategy in certain situations, because peripheral data may be inherently related to the application's domain. These data may be exploited to decide when to process denser data volumes. This modeling itself already provided a significant reduction in data processing, thereby contributing to real-time performance.

Furthermore, if a hardware layer dedicated to extract and resample conventional images is available (e.g., implemented into an FPGA), real-time pre-processing constraints

Table 3.3: Minimum, mean, and maximum energy reduction rates induced by our ARIMs compared to the provided benchmarks. For example, the minimum energy reduction induced by Model 1 on the P1E dataset was 0.505 (i.e., 50.5%), whereas the mean and maximum energy reductions in that same scenario were 0.551 (55.1%) and 0.598 (59.8%), respectively.

Dataset	**Energy Reduction**								
	Model 1			**Model 2**			**Model 3**		
	Min.	**Mean**	**Max.**	**Min.**	**Mean**	**Max.**	**Min.**	**Mean**	**Max.**
P1E	0.505	0.551	0.598	0.463	0.508	0.550	0.414	0.456	0.489
P1L	0.612	0.667	0.711	0.582	0.619	0.710	0.490	0.548	0.657
P2E	0.536	0.610	0.672	0.439	0.549	0.619	0.381	0.454	0.551
P2L	0.533	0.571	0.618	0.406	0.516	0.620	0.332	0.464	0.603

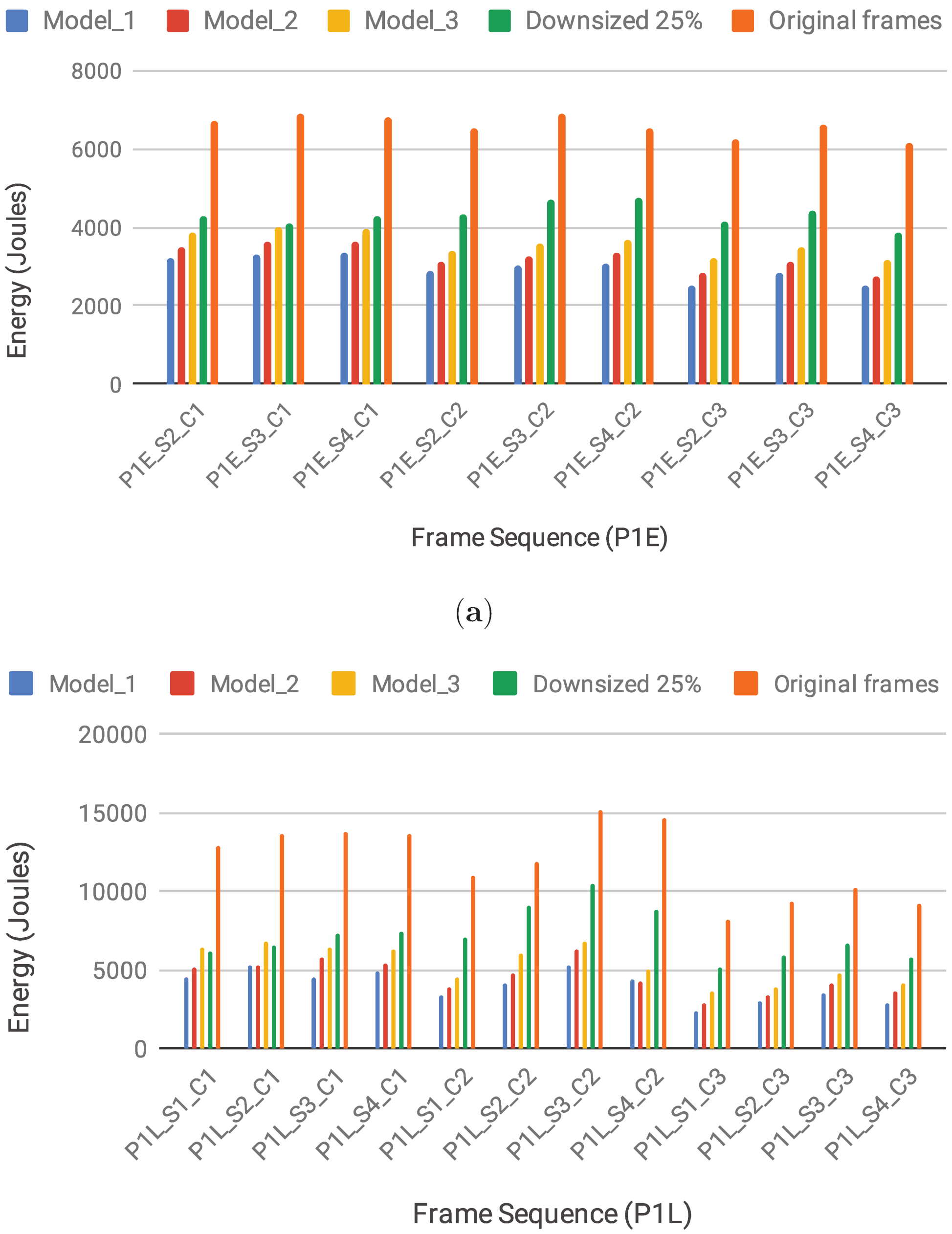
may be balanced, maintained, or even reduced, if necessary. In this scenario, the time to process frames, data volumes, energy, and computational processing power could be relieved, because many operations would be performed previously by hardware routines.

3.5 Final Considerations

In this chapter, we presented our first research avenue towards the use of foveal image models in CV applications. A crucial observation is that image data captured by uniform sensors are often dense and redundant, leading to computationally expensive solutions in terms of storage, processing, and energy consumption. We addressed this issue by exploiting a space-variant scheme that was inspired by mechanisms of biological vision related to the way that humans sense through the retina. We introduced a generic framework for designing application-oriented retinal image models (ARIM), which is how we represent the foveal image models created for specific CV applications. The models should be used to re-sample the input images prior to executing a specific CV task. We selected a biometric application to illustrate the conception and usefulness of appropriate models.

The experiments with three ARIMs having different point configurations demonstrate the flexibility of the proposed framework in devising models with different properties regarding storage requirements, energy consumption, and accuracy performance. We could observe, for example, that the use of different space-variant strategies may lead to a big reduction in terms of storage resources and energy consumption, whereas the accuracy loss rates were low in most cases. Such a trade-off evidences the viability of the proposed models and the conformity to our initial expectations regarding saving computational resources.

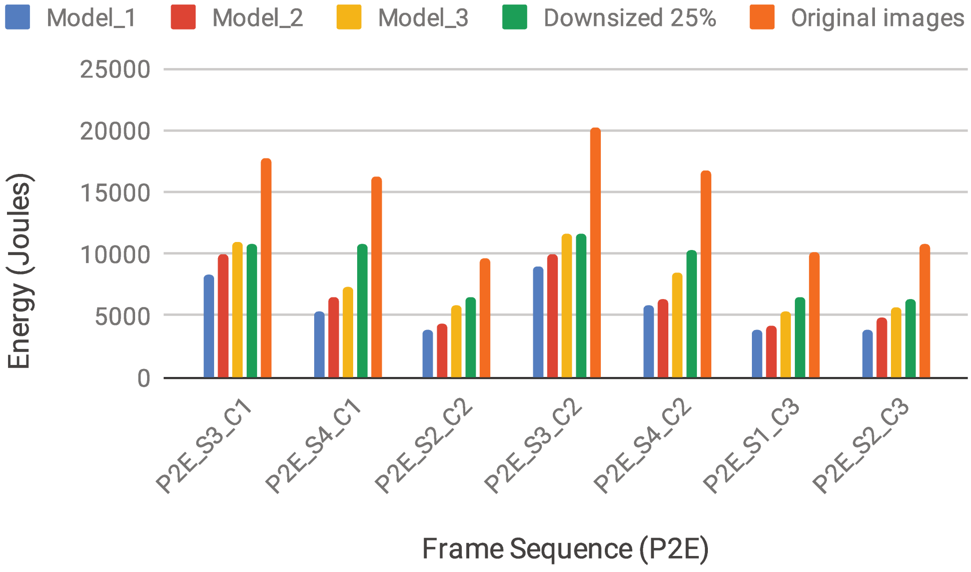
In summary, we have shown that the concept of foveal image models may be employed in CV applications aiming at decreasing the consumption of storage and energy. For such, we have proposed a framework to devise ARIMs. The results of this primary study have led us to explore novel ideas on the creation of foveal models to other application contexts.



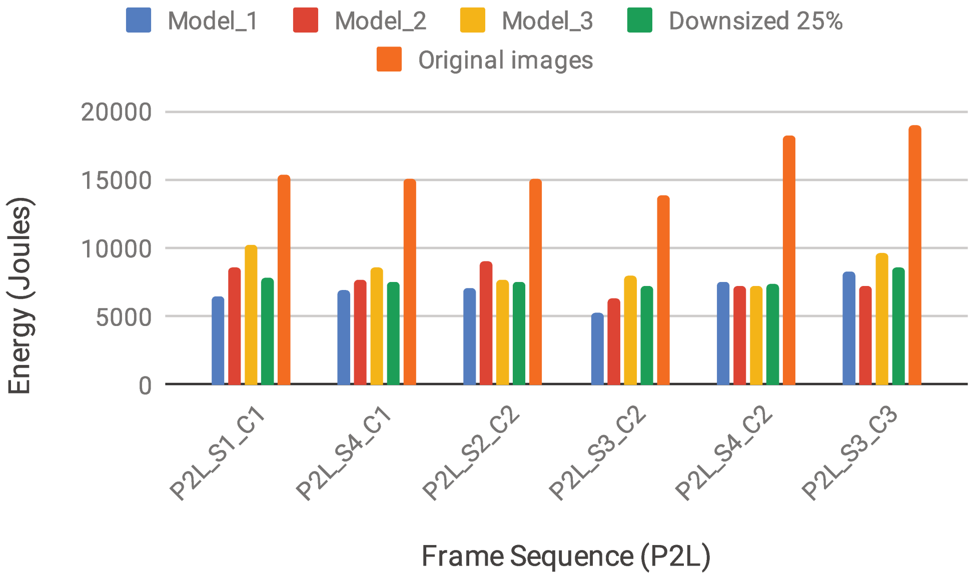
(**a**)

(**b**)

Figure 3.12: Total energy consumption regarding each evaluated model, the images resized to 25% of their original sizes, and the original (full-size) images from the (**a**) P1E and (**b**) P1L datasets.



(**a**)



(**b**)

Figure 3.13: Total energy consumption regarding each evaluated model, the images resized to 25% of their original sizes, and the original (full-size) images from the (**a**) P2E and (**b**) P2L datasets.

Chapter 4

A Change-Driven Image Foveation Approach for Tracking Plant Phenology

This chapter refers to the paper [83] published in the *MDPI Remote Sensing* journal under the title *A Change-Driven Image Foveation Approach for Tracking Plant Phenology.* The chapter contains only a few additional content and other minor differences that do not conflict in any means with the results and discussions shown in the original paper.[1]

In this chapter, we present our second contribution regarding the use of foveal image models in CV applications. The chapter is organized as follows. Section 4.1 introduces the subject. Section 4.2 describes our approach for devising image models based on a change-driven strategy and on foveation concepts. Section 4.3 shows the adopted dataset, as well as the steps needed to create the models and to reconstruct images on top of these models. Section 4.4 presents experimental details regarding the validation steps, shows the results, and discusses them. Section 4.5 presents the final considerations.

4.1 Introduction

As a rich source of data, digital images have triggered scientific advances in several areas. In plant phenology, where life cycle events are investigated along with their relationships to climate [64, 79], knowledge of plant and vegetation dynamics over time is essential to understand ecosystem processes such as carbon and water exchange [73]. Manual on-the-ground data sampling and investigation can be coupled with hardware/software apparatuses to better capture, manage, and process multimedia data (e.g., images and videos), thus supporting phenological studies in several ways [5]. For the past decades, image-based phenological studies have been successfully developed using satellite-based imaging [72], near-surface cameras [5, 73], and, more recently, unmanned aerial vehicles [50].

A technical challenge imposed by phenological imaging is related to the memory space occupied, given the need for frequent and extensive imaging. Although hardware technology is continuously advancing in terms of storage capacity, some computing environments

[1] Reprinted from Remote Sensing, 12, Ewerton Silva, Ricardo da S. Torres, Bruna Alberton, Leonor Patricia C. Morellato, and Thiago S. F. Silva, “A Change-Driven Image Foveation Approach for Tracking Plant Phenology,” 1409, 2020. The work was published under the Creative Commons Attribution License (`https://creativecommons.org/licenses/by/4.0/` – As of Oct. 2020).

still face restrictions on data volumes, such as near real-time data processing applications and bandwidth-limited remote data transfer. For uses, such as phenological monitoring, redundancy in image sequences due to invariant or slow-changing pixels contributes largely to the storage issue. Another significant feature of uniform high-resolution images is that they provide detailed data about relevant and non-relevant regions, without distinction, as a result of the underlying acquisition process in traditional cameras. In foveated vision systems, such issue relates to the task of finding fixation points (i.e., foveal centers) across one or more images, which enables a space-variant image representation to be devised. When applying models of human vision, however, one must be aware of the difficulty in predicting human fixations, as these vary strongly between different subjects and tasks. Thus, as an extra challenge, algorithms to find appropriate fixation points should be application-dependent, as there is no optimal routine that covers all possible vision tasks [30].

The interesting balance between efficiency and resource allocation provided by foveal images has led us to further explore this idea. Here, we demonstrate an image representation approach that resembles the human retina in plant phenology tracking. Our method combines segmentation and image foveation to create foveal models and images with varying degrees of spatial resolution. Such variation depends on a contextual relevance that, in our validation scope, is determined by a phenological metric and a behavior pattern of choice. We show that our models yield storage savings, while also retaining efficiency in commonly adopted settings of the phenology research field.

4.2 Proposed Approach

Figure 4.1 shows the complete workflow of our approach. First, a set of images captured over a time period is used for training. Then, the following steps are performed:

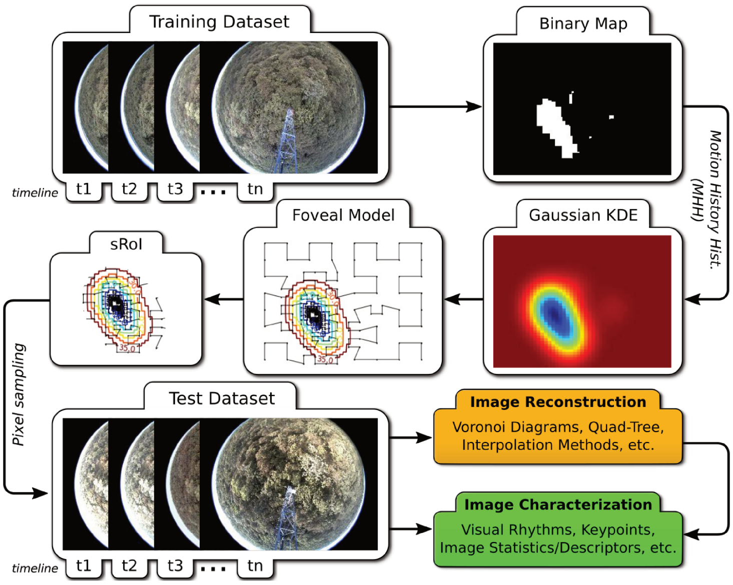


Figure 4.1: Proposed workflow to generate foveal models and resample phenological images.

1. **Binary Map (MHH)**: This step creates a binary map encoding a phenological event, and segments the image space into foveal and peripheral regions. For such, we employ motion history histograms (MHH) [62] to create a frequency map representing the spatial occurrence of binary behavior patterns over a sequence of binary change maps (CM). A CM encodes a desired change, which is determined by a phenological metric, between two RGB images. For instance, the metric of *Increase* (Equation (4.1)) can be used to encode a *Green Up* event, which we define here as the increase in the amount of leaves of plant individuals. Such event is of high interest for phenological studies. It may be captured by analyzing increases in the green-channel pixel values of a certain image regions over time.

   $$CM_{Inc}(x,y,t) = \begin{cases} 1, & \text{if} \quad I_{t+1}(x,y) > I_t(x,y), \\ 0, & \qquad \text{otherwise} \end{cases} \tag{4.1}$$

   In Equation (4.1), $CM_{Inc}(x,y,t)$ stands for the change map of two consecutive images from the sequence (the first one at timestamp t), and $I_t(x,y)$ refers to the (x,y)-pixel value in the t-th image. Note that a sequence of N images yields $N-1$ CMs.

   The successful use of CMs to depict changes on plant phenology was developed and tested using data derived from ground-based direct visual observation, and relying on MHHs to detect and represent temporal changes in arbitrary temporal multivariate numerical data [59]. Here, we applied the proposed binary encoding approach to characterize phenological events. The MHH stores the frequency each pixel featured a specific behavior in the CM sequence. For instance, a "short" *Increase* behavior can be modeled as **0110**, which refers to an increase of the values of a specific pixel in two consecutive images. Figure 4.2 exemplifies the idea of detecting binary patterns in a series of typical phenological images. However, we use a binary version of MHH as we are only interested in pixel positions where a pattern was detected with a frequency greater than the threshold σ. Figure 4.3 illustrates the complete process.

2. **Gaussian KDE (Kernel Density Estimation)**: After delineating foveal and peripheral regions, the next step estimates a 2-d non-parametric probability density function from the binary map using a KDE. The final artifact of this process is a 2-d real-valued matrix representing the corresponding function and matching the size of the map. However, as an inverse analogy to the distribution of cones in the retina, we adjusted the function such that its values increase with the distance to the fovea(s), thus resembling an upside down 2-d mixture of Gaussians. In Figure 4.1, the red and blue regions indicate high and low values, respectively.

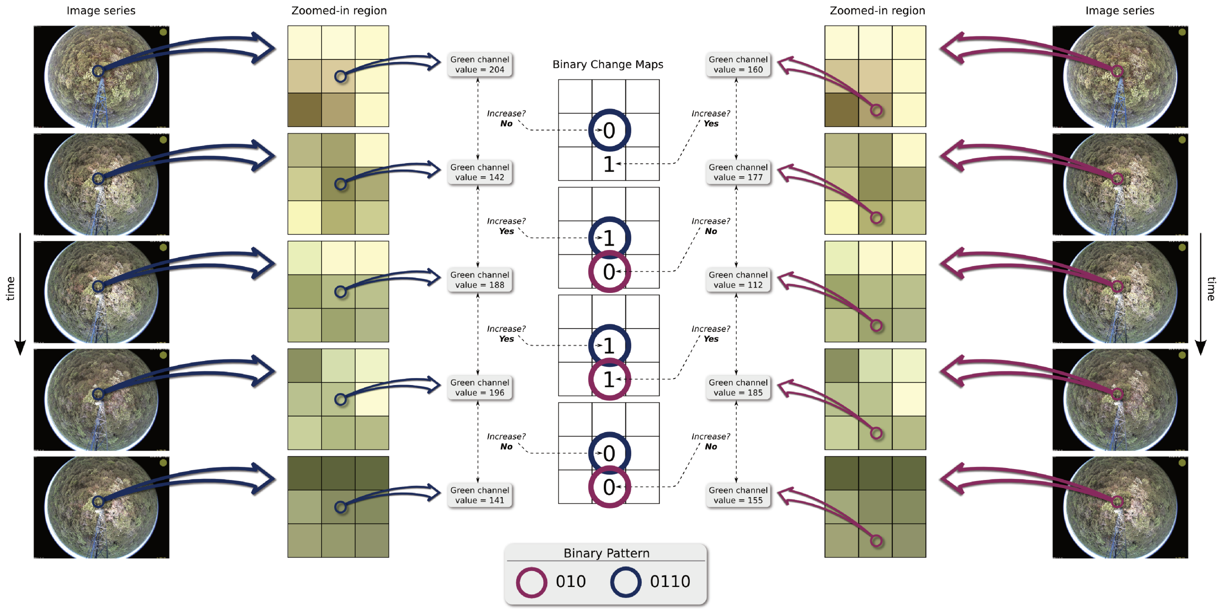


Image series
Zoomed-in region
Binary Change Maps
Zoomed-in region
Image series
time
Green channel value = 204
Increase? No
Green channel value = 142
Increase? Yes
Green channel value = 188
Increase? Yes
Green channel value = 196
Increase? No
Green channel value = 141
Green channel value = 160
Increase? Yes
Green channel value = 177
Increase? No
Green channel value = 112
Increase? Yes
Green channel value = 185
Increase? No
Green channel value = 155
0
1
1
0
1
1
0
0
time
Binary Pattern
010
0110

Figure 4.2: Example of the detection of the 010 and 0110 binary patterns using a series of typical phenological images. Both sides of the figure show the same image series and zoomed-in regions (i.e., the same location in each image). For illustration purposes, the analysis is done only for two pixels from the focused region. Nevertheless, the complete procedure requires the analysis of all image pixels.

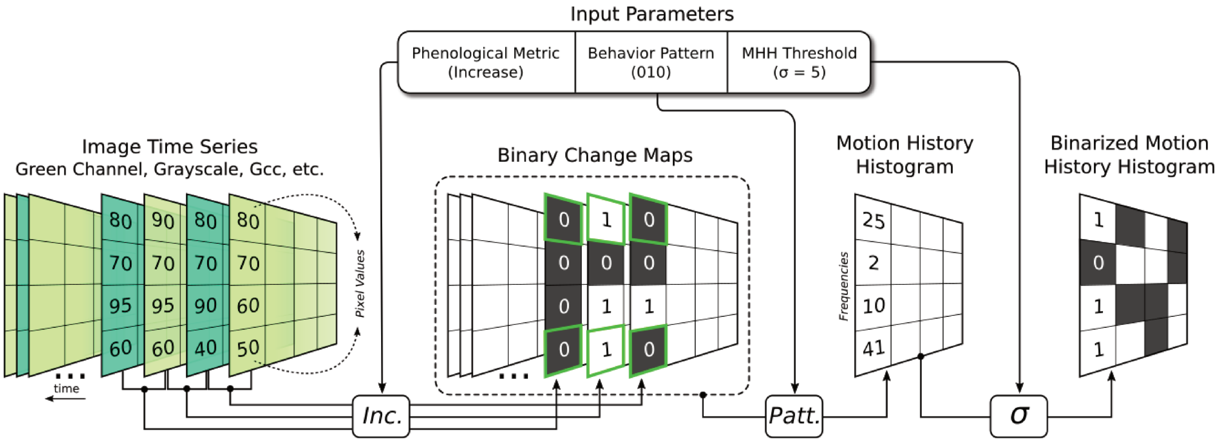


Figure 4.3: Pipeline of the Binarized MHH generation. This example illustrates the entire process applied to detect a specific pattern throughout the image series. First, two consecutive images I_i and I_{i+1} from an image series $S = \{I_i, I_{i+1}, ..., I_n : 0 \leq i \leq n, n \geq 1\}$ are compared according to a phenological metric (in this example, the pixel *Increase*). The comparison yields a binary change map CM_i spotting out the pixels that conform to the metric. As comparisons proceed (i.e., as i increases) all CMs are inspected towards detecting pixels that change according to a behavior pattern (in this example, **010**). Finally, the MHH is gradually updated upon detections. Suppose a pixel at position (x_1, y_1) is found to follow the desired pattern (in this example, green border cells) at a certain point of the examinations. In such a case, MHH(x_1, y_1) would increase by one. After the entire time series has been processed, the MHH is binarized by means of a predefined threshold sigma (σ).

3. **Foveal Model**: We used the non-parametric function and a Hilbert curve in a pixel-sampling procedure to create a foveal model. The Hilbert curve maps a 1-d parameter space to a higher-dimensional space (i.e., 2-d), thus creating a sequential order by visiting each midpoint of a square once. Unlike a usual raster-scan approach, the Hilbert curve favors the preservation of locality properties inherent in multidimensional data, as it traverses neighboring regions prior to visiting distant ones [13].

   The proposed method initially divides the image space into four squares around its center. Given that the Euclidean distances of the square's midpoint to its two closest neighbors are greater than the value of the non-parametric function at midpoint, each square is then recursively divided into four equal-sized squares. This can be envisioned as a gradual refinement process in which more pixels are sampled the closer these are to a fovea. The stop condition occurs when trying to reach subpixel positions, when a square is not refined any further. The final curve is not homogeneous (see Figure 4.1), with its vertices constituting the non-uniform sampling scheme. We present a more detailed explanation of this procedure in Appendix E (see Algorithm 4), where we describe this idea.

4. **Space-Variant Region of Interest (sROI)**: Although the foveal model may contain regions of variable interest, one might choose only a subset of them. Having a gradually-decreasing resolution from the foveal centers towards peripheral regions could be helpful in specific circumstances. Instead of using a delimiting rectangular window or a binary mask over a uniform image, a space-variant model allows us to deal with a non-uniformly constrained region of interest, which we refer hereafter as sROI, that may represent a phenological behavior of a given plant individual over time.

After a "training" step, in which a set of images is processed to obtain one or more foveal models, a new set of images from the same area and different time periods can be used to test (i.e., evaluate) these models and their extracted sROIs. Images can be represented as a set of sparse points, or be reconstructed by several methods (e.g., Voronoi diagrams, quad-trees) that provide a 2-d space-variant representation. From the chosen representation, data can be extracted and processed (e.g., phenological visual rhythms and interest-point/statistical descriptors). By using space-variant images, memory storage may be reduced given that some images are represented at different resolutions and contain less data.

4.3 Materials and Methods

In this section, we detail the experimental procedure adopted, including the dataset used, how foveal models are generated, and the image reconstruction (or resampling) step.

4.3.1 Dataset

The dataset employed contains daily sequences of RGB images (in JPEG format) with 1280 × 960 pixels. The sequence covers the years from 2012 to 2015: where, everyday, an average of six images were captured for every hour (between 6am to 6pm). Binary masks of field-identified individual plants are also available. The images were produced with a digital hemispherical lens camera (Mobotix Q 24, Mobotix AG – Germany), which was placed on a monitoring tower far above the canopy and recorded the phenology of a Cerrado (neotropical savanna) area at Itirapina, São Paulo, Brazil [4, 5]. Our study occurred during the transition between the dry and the wet season, a period when most of the plant species are producing new leaves. According to [4], in which a dataset from the same area was used, leaf flush periods occur from the end of August to the beginning of October. Thus, for most experiments, we considered the day of the year (DOY) ranging between 241 and 273, when available. The sunrise, sunset, and solar elevation angle were 6:23 h, 18:01 h, 47.2° (DOY 241), and 05:52 h, 18:10 h, 59° (DOY 273), respectively.

4.3.2 Foveal Models

To generate the models, we used the 2012 image dataset, which was preprocessed as follows. First, we removed unwanted elements, such as the camera tower and the information inserted by the camera's software. Then, images were down-scaled to 25% of their original size (320 × 240 pixels) to favor the creation of well-structured models (i.e., with a few, smooth foveal regions). Using scale-space inspections, allowed by gaussian pyramid decompositions, we noticed that images at lower scale levels (i.e., with reduced sizes) provide more satisfactory models, because noise-like data and other irrelevant details are removed. Each image was then converted into its green chromatic coordinate (GCC) representation. The GCC is a common index used for near-surface phenology that reflects a measurement of the proportion of green color signal on an RGB image pixel or region [4, 5, 85]. Finally, we calculated the 90th percentile of each day to encapsulate relevant daily data into a single image and, possibly, minimize the impact of lighting changes (e.g., intensity, angles) on the time series [85].

We selected four binary patterns typifying different behaviors of a phenological change of *Increase* (see Section 4.2) encoded in MHHs. The **010**, **0110**, **01110**, and **011110** patterns (ordered from the shortest to the longest) indicate how pixels behave in terms of frequencies of continuous changes (sequences of "1"s) bounded by steady states (bordering "0"s) throughout the image series. Although other temporal patterns could be contemplated, we have chosen these because, along with the *Increase* metric, they are capable of encoding the *Green Up* change. For each pattern, we used different values of sigma (σ; found empirically) to threshold MHHs into their binary version. Finally, the models generated (i.e., their points) were re-scaled back to the original, high-resolution size. Figure 4.4 shows the MHHs, KDE implicit functions, and foveal models generated for each pattern. The "spreading" aspect of MHHs also allows us to identify all regions that respond to a particular event.

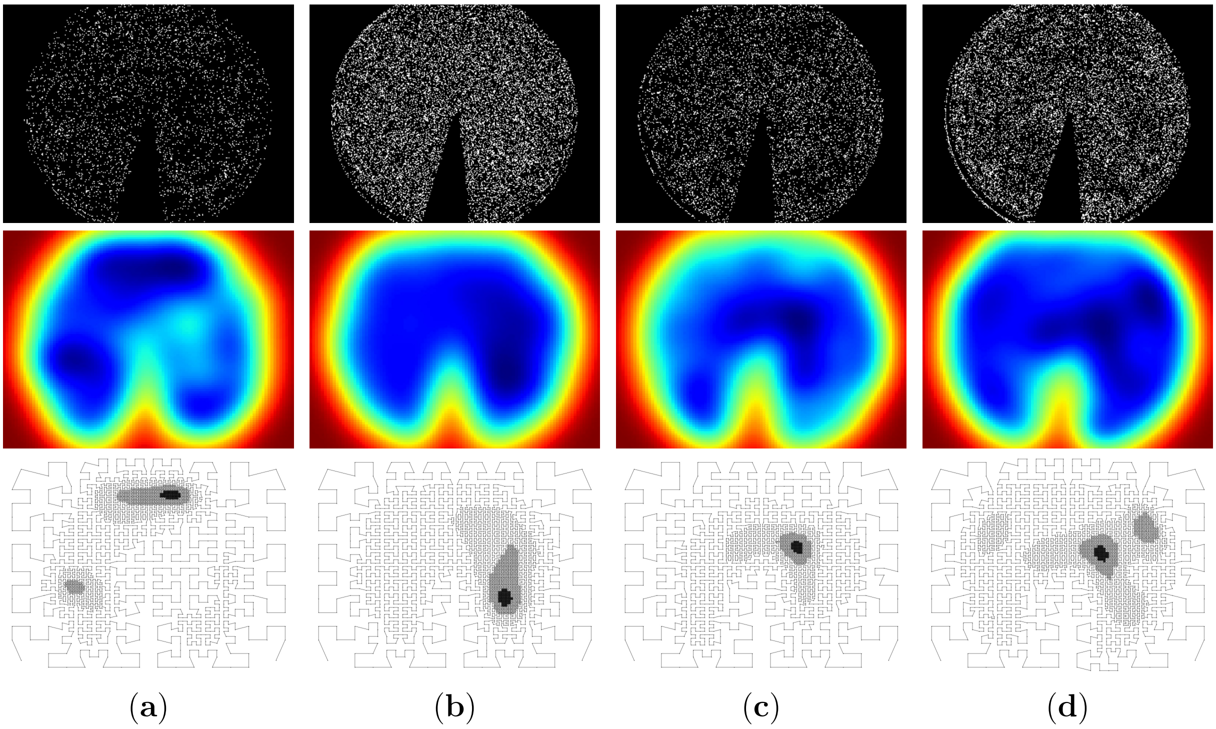


Figure 4.4: MHHs (**top**), KDE implicit functions (**middle**), and foveal models (**bottom**) generated for the (**a**) 010, (**b**) 0110, (**c**) 01110, and (**d**) 011110 patterns considering the 2012 images.

4.3.3 Image Reconstruction

A plausible approach consists in returning to the 2-d space by means of a reconstruction step. When doing this, the image is essentially uniform, but its contents get represented in a space-variant domain. Figure 4.5 illustrates the procedure we adopted in this thesis, which relies on calculating a Voronoi diagram for the set of points comprising each model, then reconstructing the image by drawing Voronoi cells filled with the same RGB color of the cell's central pixel in the captured scene. This creates a *foveal image* that has greater content heterogeneity in foveal regions, as these carry higher resolutions compared to peripheral ones.

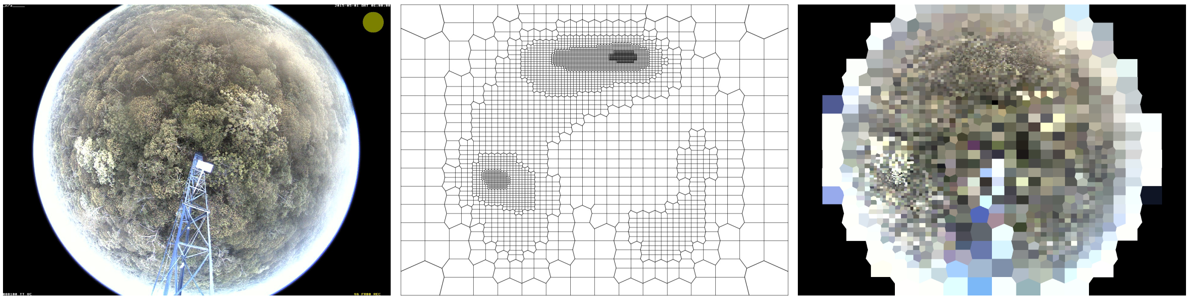


Figure 4.5: Example of image reconstruction. The original image (**left**) is reconstructed (**right**) using a Voronoi diagram (**middle**) obtained from a model. Each Voronoi cell is filled, in the right image, with the same RGB color of the cell's central pixel in the left image.

4.4 Results and Discussion

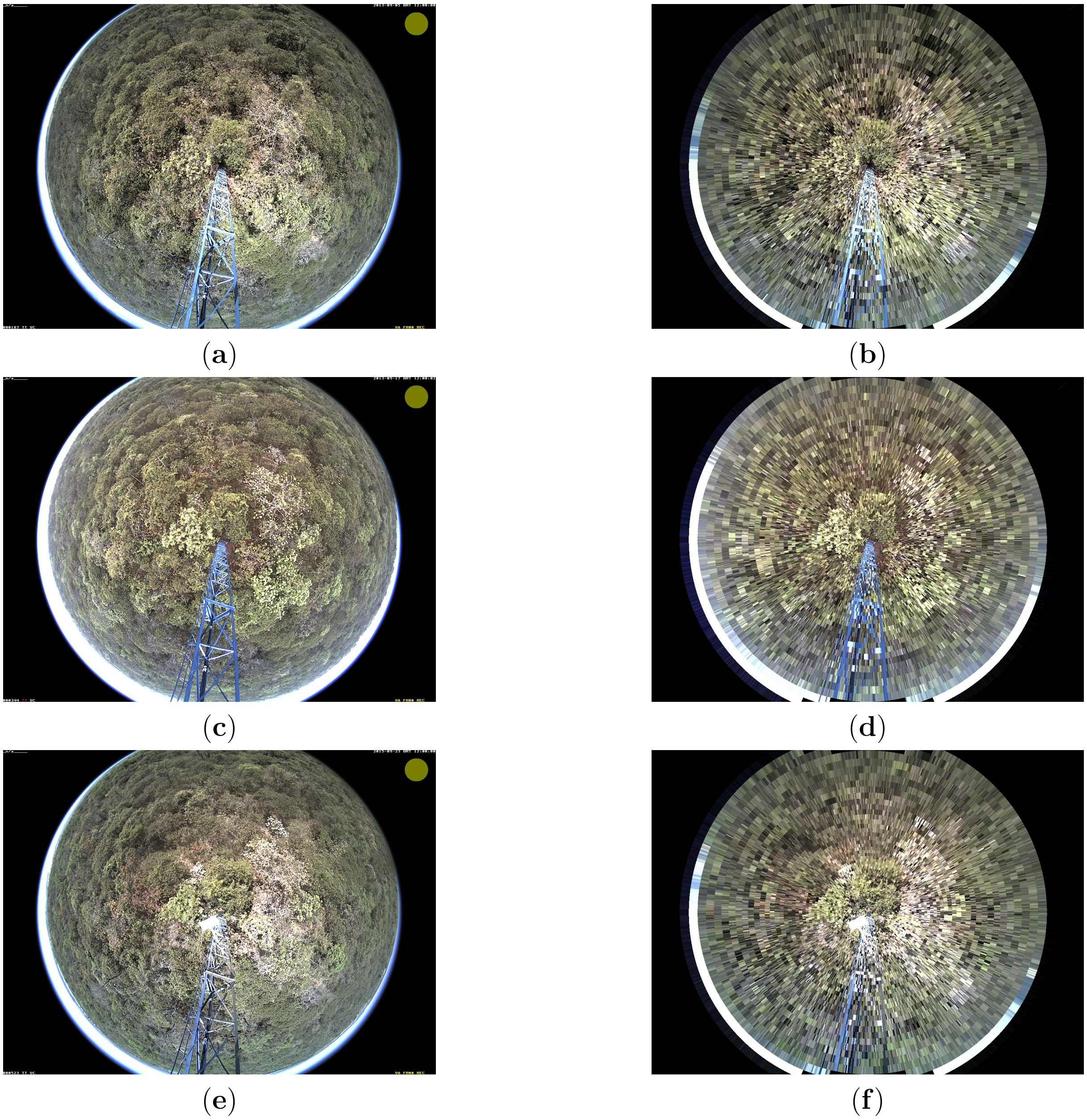
In this section, we present a validation for the proposed workflow. We assess the effectiveness of foveal models in terms of time series' correlation rates of an appropriate vegetation index and memory storage usage induced by models.

4.4.1 Evaluation for Reconstructed Images and ROIs

In plant phenology, monitoring variations in plant individual features, such as those related to color and shape, are paramount to understand the phenophases that these individuals undergo, and, consequently, the associated ecosystem processes [5]. A suitable measure to determine these variations in digital images over time is the mean GCC of image pixels or regions. Thus, we validated the foveal models from Section 4.3.2 by comparing regions from original and reconstructed images having plant individuals known to undergo the same phenological change (i.e., *Increase*) encoded in the models. We used individuals of the *Aspidosperma tomentosum* and *Caryocar brasiliensis* species (for details see [4]).

We tested the models with high-resolution images from the years of 2013, 2014, and 2015. The top panels of Figure 4.7-top show the mean GCC time series regarding the tested years for original and foveal images with each model. To measure the similarity between the original and foveal time series, we calculated the Pearson-Correlation value between series (Figure 4.7-bottom panels). The high positive correlation results suggest that the reconstructed visual information is still significant even under varying resolution and degradation levels caused by the space-variant representation. Additionally, models encoding the **010** and **0110** patterns seem more effective at incorporating visual information from *Aspidosperma tomentosum* individuals, whereas the **011110** pattern is more effective for *Caryocar brasiliensis*. Figure 4.8 presents examples of the phenological images used in the experiments, and their masked, reconstructed, and mask-reconstructed versions for a visual inspection. Figure 4.6 illustrates image reconstructions with an inverse log-polar mapping for a visual comparison with the ones obtained with our foveal models. We do not consider log-polar reconstructions in our quantitative tests.

We also conducted image quality evaluations. Table 4.1 shows root mean squared error (RMSE), mean absolute error (MAE), structural similarity index (SSIM), and peak signal-to-noise ratio (PSNR) results for reconstructed images using each model (binary pattern). PSNR and SSIM are two image quality metrics commonly used in the literature to compare different image compression/reconstruction schemes. The PSNR is calculated as the signal peak divided by the strength of the noise, whereas the SSIM quantifies (in the $[0, 1]$ range) the similarity between two images by considering perceptual differences, which include the structural divergences between the depicted objects (e.g., along their edges) caused by image degradation [110]. As expected, SSIM and PSNR values are low, due to the large Voronoi cells in peripheral areas. However, RMSE and MAE values were acceptable, indicating that the degradation, although severe in some regions, does not shift the error rates too much, thereby suggesting that foveal images may still be accurate for some analyses, such as plant phenology tracking (as shown in the present study).



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 4.6: Examples of phenological images reconstructed with a log-polar mapping. Images (**a**, **c**, and **e**) are the original ones, and images (**b**, **d**, and **f**) are their reconstructions using an inverse log-polar mapping on the cartesian domain. We do not consider log-polar images in our quantitative tests.

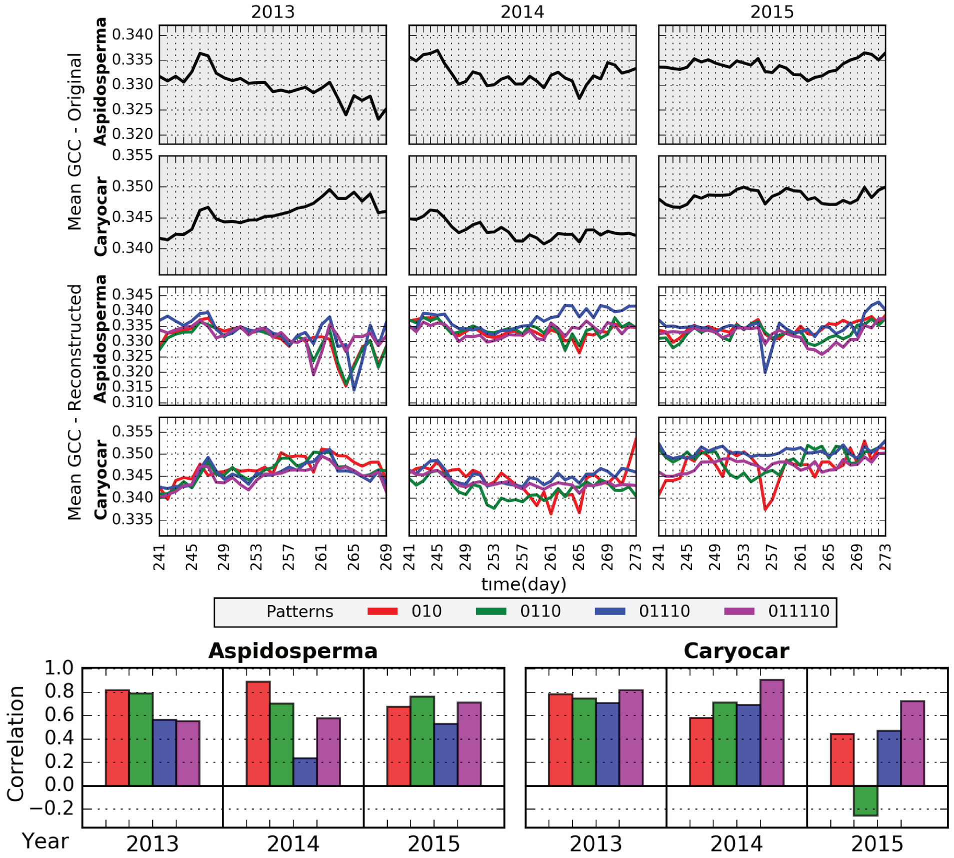


Figure 4.7: Mean GCC time series and correlation results for *Aspidosperma tomentosum* and *Caryocar brasiliensis* individuals, considering original and foveal images with each model. In the 2013 dataset, the last available DOY from the evaluated period was 169.

Table 4.1: Results for reconstructed images using the four models respective to the 010, 0110, 01110, and 011110 binary patterns. The considered metrics are root mean squared error (RMSE), mean absolute error (MAE), structural similarity index (SSIM), and peak signal-to-noise ratio (PSNR).

Metric/Pattern	**010**	**0110**	**01110**	**011110**
RMSE	0.2532	0.2493	0.2509	0.2414
MAE	0.1580	0.1555	0.1570	0.1508
SSIM	0.3837	0.3862	0.3836	0.3833
PSNR	11.9327	12.0676	12.0135	12.3479

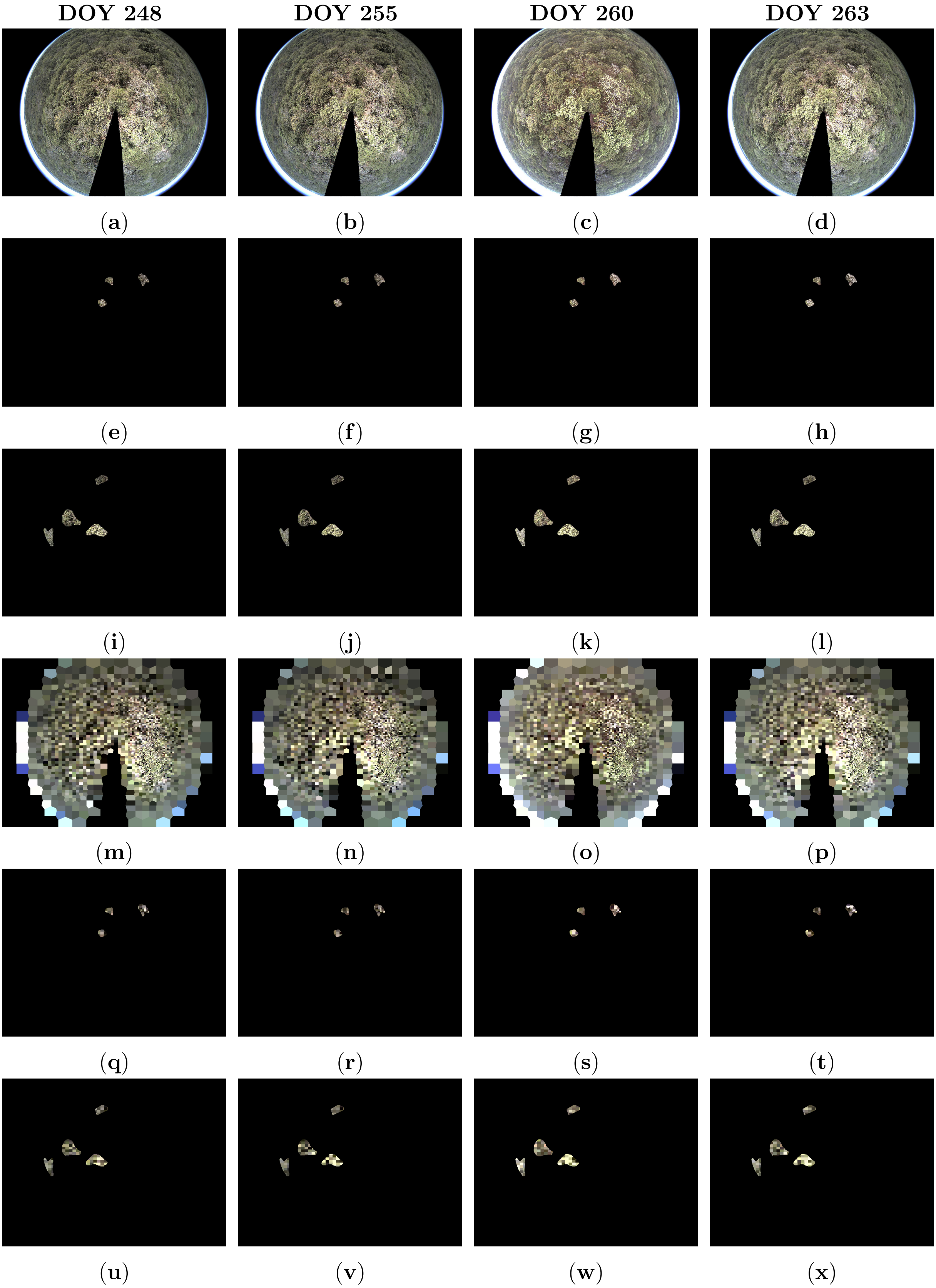


Figure 4.8: Examples of phenological images from 2013. The DOYs 248, 255, 260, and 263 are column-wise organized. Each row depicts the following: (**a**–**d**) original images; masked images containing only the (**e**–**h**) *Aspidosperma tomentosum* and (**i**–**l**) *Caryocar brasiliensis* individuals; (**m**–**p**) images reconstructed with the model for the 0110 pattern; and masked-reconstructed images containing only the (**q**–**t**) *Aspidosperma tomentosum* and (**u**–**x**) *Caryocar brasiliensis* individuals.

4.4.2 Evaluation for sROIs

Instead of image reconstructions, a subset of the 1-d points from a Hilbert curve can be used. The curve embedded in each foveal model is non-homogeneous and presents variable point densities across the 2-d space it fills. By considering the curve as a sequence of points, such variability can also be verified, but it is necessary to know the refinement level each point in the sequence belongs to. In essence, a new refinement level is obtained every time a recursive call is performed during the model generation (see the algorithmic description in Section 4.2 -3), thereby dividing a square area (at the n-th level) into four equal-sized squares, each of which will then belong to the $(n+1)$-th level. Points at lower levels are associated with poorly refined regions and coarser resolutions, while points at higher levels are associated with deeper refinements and finer resolutions. In this vein, we introduce the concept of minimum refinement levels (MRLs) to constrain a sequence to a subset comprising all points **at and above** a certain desired level. We also refer to this MRL-constrained sequence as sROI. In Figure 4.9, we use (i) different colors to represent distinct refinement levels and (ii) a quad-tree layout we believe is convenient for visualizing the refinement steps leading to the non-homogeneous aspect of the curve. For instance, the 6th MRL includes points at the 6th, 7th, and 8th refinement levels.

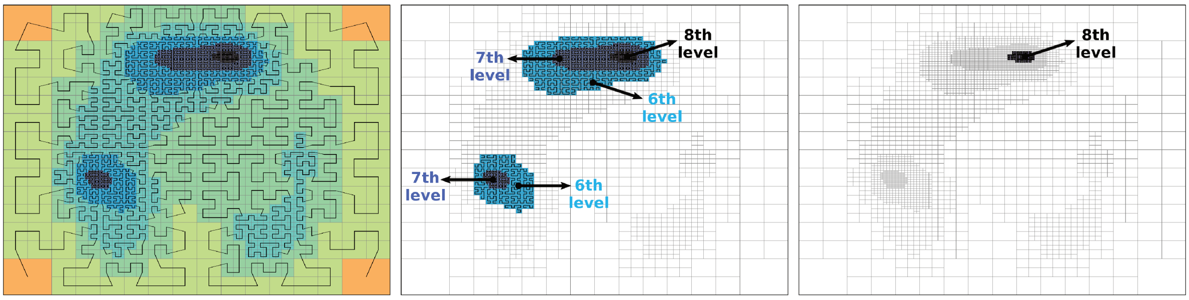


Figure 4.9: Examples of sROIs (from the 010 model) in a quad-tree scheme at the 2nd (**left**), 6th (**middle**), and 8th (**right**) MRLs. Each level is represented in a distinct color. In this case, the lowest MRL is the 2nd, meaning that at least two recursive refinements were performed for any of the four points of the initial curve. The 2nd MRL therefore includes all points of the model. Nevertheless, the highest refinement level of the depicted model is the 8th one. Thus, the 6th MRL comprises the points at the 6th, 7th, and 8th refinement levels, whereas the 8th MRL includes only the points of its own level.

The neighborhood preservation property of the Hilbert curve allows us to compare mean GCC correlations from ROIs and sROIs directly. The ROI, in this case, refers to the binary masks of each individual provided by the dataset. Figure 4.10 shows time series and correlation values for sROIs at the 6th and 8th MRLs of each model. Since sROIs enclose many regions conforming to the event encoded in their models, low correlation values are expected (Figure 4.10—bottom panels). Moreover, the correlation analyses showed a reverse behavior of some models across distinct species. For example, in the 2013 data, the **010** pattern is positively correlated with *Aspidosperma tomentosum*, but negatively correlated with *Caryocar brasiliensis*.

By providing different spatial resolutions according to a phenological metric and a binary pattern, foveal models are able to encode important spatial information that is useful for tracking plant phenology. This statement is in accordance with the obtained results for reconstructed images and sROIs. However, as in our validations we considered a single year to create the models, the variable correlation results of the 2014, 2015, and 2016 years are expected, because phenological changes may be influenced by uncertainties over time, such as anthropogenic factors and climate changes. Thus, one strategy we believe may improve the results is using data from many years of observation. By doing so, the created models could capture more information about the phenological behavior of plants, thereby better generalizing for new years.

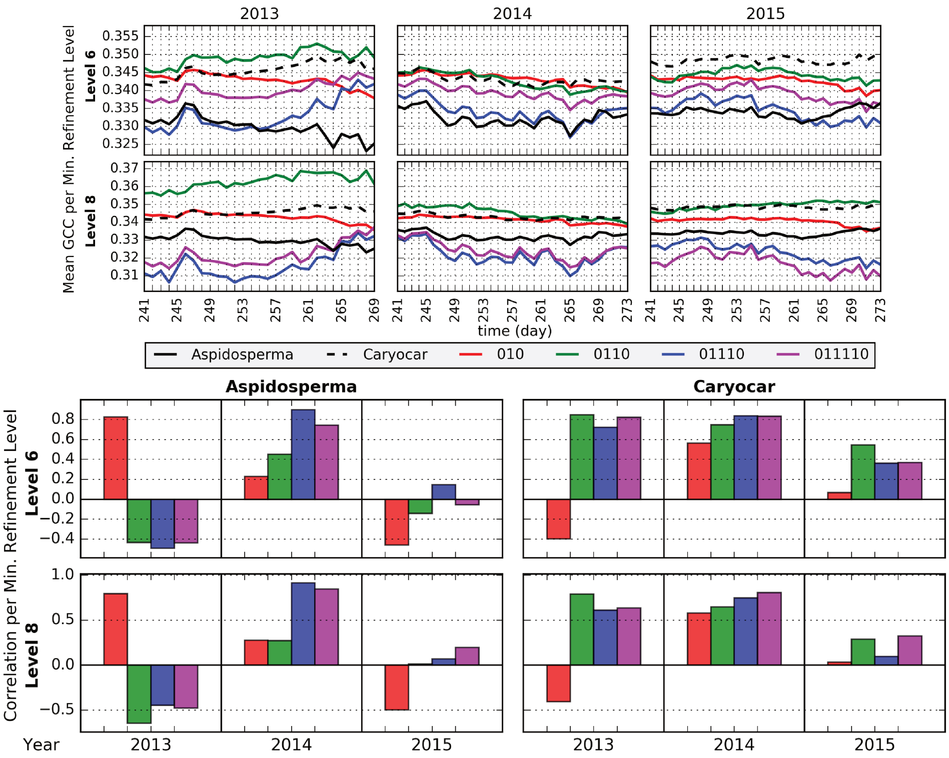


Figure 4.10: Mean GCC time series and correlation results for ROIs and sROIs at the 6th and 8th MRLs. Both *Aspidosperma tomentosum* and *Caryocar brasiliensis* plant species were considered.

4.4.3 Storage Usage

A benefit of adopting space-variant image representations is the reduced data volumes. First, our image reconstruction process essentially compresses the images in a space-variant manner. Thus, it is enough to compare the size of these images with the ones obtained with other compression methods. We compared the original image, JPEG images with five quality factors varying from 90 (best) to 10 (worst), and the four foveal models from Section 4.3. Figure 4.11 shows the quantitative comparison performed, whereas Figure 4.12 shows examples of the compressed images with JPEG quality factors varying from 90 to 10. Our foveal models generated resampled-image sizes that are similar to those obtained via more aggressive JPEG compression actions, i.e., by applying qualities 30 and 10. Additionally, there is significant difference between the image sizes obtained with our foveal models and with JPEG quality factors above 30, which produce images with less degradation at the cost of higher storage sizes. In contrast, we observed a reduction of around 99% in the number of points (pixels) for the images generated from the foveal models in comparison to the original image (Table 4.2). The same holds for bytes.

Compared to ROIs and sROIs, the Voronoi reconstruction leads to a higher impact on memory storage rates, since more data is required to represent a foveal image in a 2-d uniform cartesian grid. However, as foveal images show greater visual homogeneity in peripheral regions—due to large super-pixel-like artifacts (i.e., Voronoi cells)—additional standard compression techniques can also be applied to decrease the storage sizes even further. Moreover, there is no need to save Voronoi cells' vertices, as these can be calculated by Voronoi algorithms, but this may require extra and repetitive computational processing. Finally, the combined results of Table 4.2 and Section 4.4.2 suggest that the usage of full or MRL-constrained sROIs in phenology tracking (as substitutes for 2-d images) is a valid and viable approach that aims to reduce the manipulated data sizes.

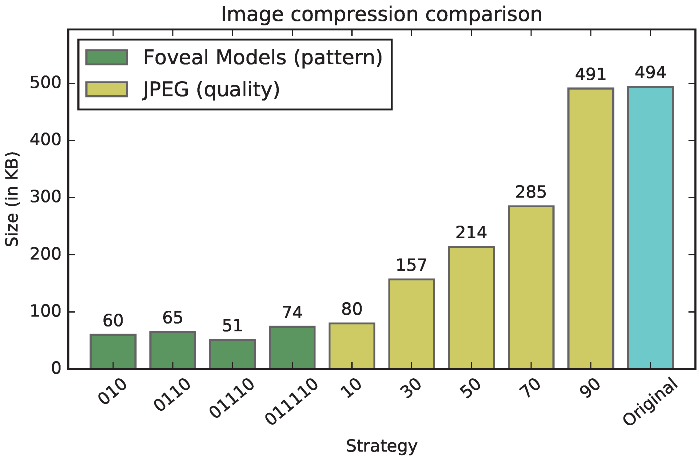


Figure 4.11: Comparison of the storage sizes (in kilobytes, KB) of compressed images with our change-driven approach and with the usual JPEG compression process. The models evaluated were the ones generated by the 010, 0110, 01110, and 011110 binary patterns, whereas the JPEG quality factors considered were 10, 30, 50, 70, and 90. The graph shows the mean size of the compressed images.

Table 4.2: Storage results for full models and sROIs at the 6th-MRL, which includes regions at the 6th, 7th, and 8th refinement levels.

Storage Results	**010**		**0110**		**01110**		**011110**	
	Full	**sROI**	**Full**	**sROI**	**Full**	**sROI**	**Full**	**sROI**
Number of points (pixels)	4123	2954	4954	3606	3286	2069	5629	4187
Approx. size (in kilobytes)	12.07	8.65	14.51	10.56	9.62	6.06	16.49	12.26
Original image size	(1280 × 960) pixels × 3 bytes = 3600 kilobytes							
Approx. reduction in size (%)	99.66	99.75	99.59	99.70	99.73	99.83	99.54	99.65

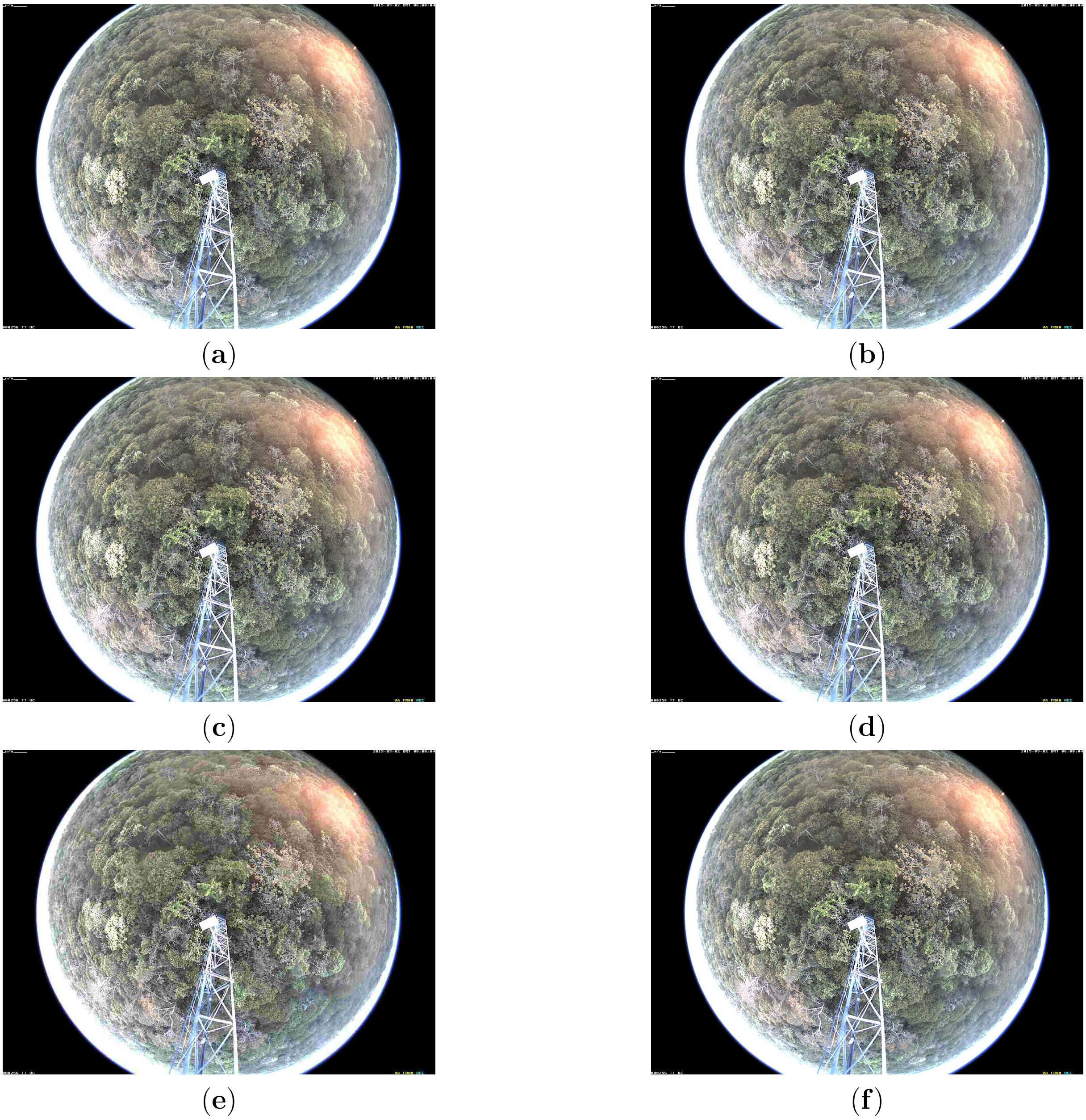


Figure 4.12: Examples of phenological JPEG-compressed images with the quality factors (**a**) 90, (**b**) 70, (**c**) 50, (**d**) 30, and (**e**) 10. Image (**f**) is the original, uncompressed one. Although visually similar, images (**d**) and (**e**) present several compression artifacts (zoom in to check in detail).

4.5 Final Considerations

In this chapter, we described our second contribution regarding the use of foveal image models in CV applications. We introduced a change-driven image foveation approach to deal with large volumes of data from phenological images. Several phenology applications must be aware of storage limitations, such as real-time expert processing systems, very-high resolution imaging sensors, and low-bandwidth remote data transmission. As global long-term and widespread databases of phenological imaging become available, efficient storage with minimal loss will be necessary. To solve these problems, we proposed the creation of foveal models that are able to encode phenological metrics and a behavior pattern. MHHs and a Hilbert curve provided the fixation points/regions and the varying-resolution aspect of the models, respectively. We then proposed that these models be applied to create foveal images having less, but also most of the relevant data.

We evaluated model correlation rates for mean GCC time series (2-d and 1-d scenarios), visual quality, and memory storage. Our results show a reduction in the amount of stored data and a viable new image representation, both in terms of quality and relevant-data preservation. In the image compression spectrum, our approach is also valuable as it reaches similar image storage sizes to those obtained with a simple JPEG compression technique using low quality factors. Although the compressed images show visual artifacts, our compression proceeds in a semantics-wise manner as determined and encoded by the foveal models.

Data variability may represent a challenge to any remote sensing approach targeting vegetation tracking, and we have employed foveation precisely to account for such variation. Although our foveal models may be static and built on top of the behavior seen in a specific year, their resolution-degrading configuration still correlates well with those from subsequent years. For very long time series, however, our approach might have some natural drawbacks, as climate and anthropogenic issues contribute to amplify uncertainties over time.

In summary, our solution may be considered as more flexible alternative, when compared to our first one (from Chapter 3), to handle huge volumes of image data. As a matter of fact, the process of devising foveal models does not require the prior definition of spatial properties. Instead, we took into account more abstract and general information from the application's domain. In our study, such information has been related to the phenological behaviour of plant species over time. Finally, we analyzed two types of representation for the foveal images: a 2-d representation, which resembles a compressed scheme, and a 1-d representation based on a Hilbert curve.

Chapter 5

A Foveation-based Framework to Support the Visualization of Very-High-Resolution Image Time Series

In this chapter, we present our third contribution regarding the use of foveal image models in CV applications. The chapter is organized as follows. Section 5.1 introduces the subject. Section 5.2 describes a framework to support the visualization of very-high-resolution (VHR) image time series and the information visualization approach; the framework relies on mechanisms of the human visual system (HVS). Section 5.3 presents usage scenarios and discussions on them. Section 5.4 presents the final considerations.

5.1 Introduction

If one had to characterize the current era in a sentence, the word "data" would appear somewhere. Along with proper technology, this keyword has been opening doors for scientific discoveries and new research opportunities. Large collections of digital images are no exception to this fact, undergoing continuous expansion and exploration by experts and researchers in their technical or academic investigations. In face of such big volumes of data, appropriate hardware and software are required to store, process, and extract useful information [24, 42, 57]. In this context, saving resources is an important issue to be concerned with [57], be them computing or human ones.

High-resolution imagery is used in a variety of remote sensing tasks, such as urban planning [28], crop monitoring [42], geographical damage assessment [90], and plant phenology [5]. Different types of sensors and infrastructure setups can be employed in the data acquisition process. For instance, image-based phenological studies have used near-surface cameras [5, 73], satellite-based imaging [72], and unmanned aerial vehicles (UAV) [50] to monitor life cycle events over time. Human inspections and spatial analyzes of such images are time- and energy-demanding, often being supported by data visualization techniques – such as those from geographical information systems (GIS) software – able to provide meaningful, less complex, and reduced data representations [33]. Furthermore, these studies may also rely on the inspection of VHR images obtained by means of modern technological apparatuses, such as drones. Nevertheless, given the ascending curve of spatial and temporal scales of data, image time series usually come into play, thus magnifying the need for more sophisticated and effective information visualization techniques [24].

The plant phenology field, in particular, is open to different information visualization ideas that may facilitate the analysis of long series of high-resolution vegetation images. These images usually contain several plant individuals (from diverse species) undergoing phenological changes that may be hard to spot out visually, due to the nature of the remote-sensing images, climate conditions, shadows, lighting, and other factors of uncertainty. Morellato et al. [65], in the context of the e-Phenology Project, highlighted the relevance of developing novel models, algorithms, and data analytic tools to support the analysis, integration, and visual interpretation of phenological images. The challenge of conducting phenological studies by means of long-term image series is also evidenced in terms of the many types of variables that may be considered. To deal with this challenge, Mariano et al. [61] proposed an information visualization approach that combines visual rhythms and a radial structure. The approach, which was validated with experts on the phenology field, demonstrated not only the suitability of the visualization tool, but also that the use of different perspectives and visual schemes could benefit phenological studies. Thus, providing meaningful and concise visualization tools to handle phenological image data is paramount, and several other studies have contributed to this scientific demand [8, 52, 59, 60].

In some circumstances, coping with VHR images may turn into a very demanding and challenging task. To deal with these issues, in this chapter we combine insights gleaned from previous works on foveation, visual attention, and information visualization. We extend upon our previous work (Chapter 4) by proposing a framework extension that allows the processing of VHR image time series based on HVS properties. We also develop a perceptual and less complex information visualization approach to support more palpable human inspections of such huge volumes of data. The framework relies on a mechanism of foveation guided by top-down "cues," which are spatio-temporal regions potentially relevant (salient) in a task-dependent context. To our knowledge, this is one of the first initiatives to apply foveation and visual attention strategies to VHR image time series.

5.2 Proposed Approach

Figure 5.1 illustrates the main steps of our framework to handle VHR image time series based on concepts of the HVS, namely, visual attention and foveation. In the following, we dissect all steps of the proposed framework.

5.2.1 Visual attention for estimating salient regions

The visual attention mechanism requires the analysis of images in the search for salient regions, i.e., regions that might be of interest for a task. We proceed similarly to what was

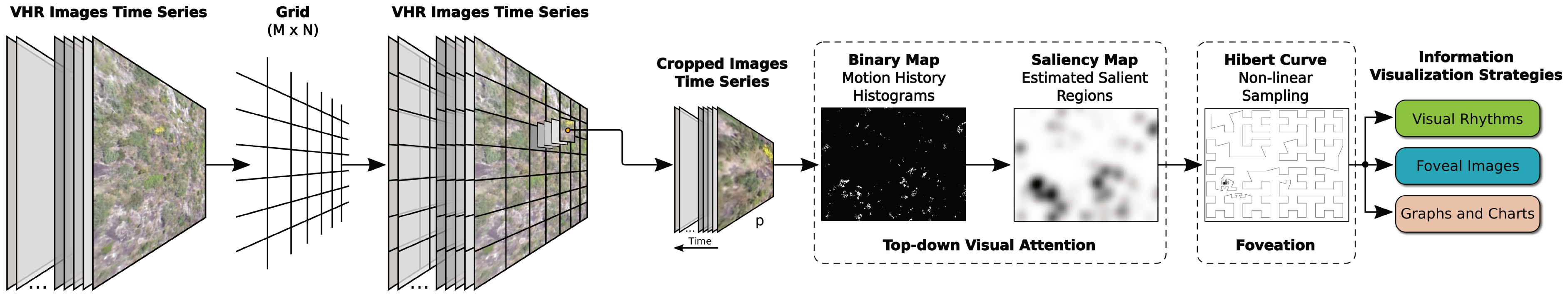


Figure 5.1: Proposed framework to handle VRH image time series. First, each image in the time series is cropped according to a predefined grid of $M \times N$ size, thus generating several smaller images. Then, all cropped images at a specific grid position p in the time series are examined in the search for saliency points and regions. This procedure generates a binary map, which is used to build a saliency map spotting out the estimated salient regions (the darker the region, the more important it is). This has similarities with a top-down visual attention process, because we look for image points having plant individuals undergoing phenological changes over time. The next step consists of creating a foveal model - based on a Hilbert curve - that is able to represent the set of examined cropped images in the time series. The foveation process consists of applying the curve to sample points according to the degree of importance of the regions in the saliency map. In this case, for example, dark regions will have higher point density than white ones. Finally, each cropped image at position p will be represented as a sequence of points extracted by means of the model's curve. Such sequence can be further used as input to different information visualization strategies, such as visual rhythms, graphs, and charts.

detailed in Section 4.2. First, to make the process more feasible, each image in the time series is cropped according to a predefined grid of $M \times N$ size, thus generating several smaller images. Then, all cropped images at a specific grid position p in the time series are examined in the search for saliency points and regions. This procedure, which requires the definition of a phenological metric and a behaviour pattern, generates a binary map that is used to build a saliency map spotting out the estimated salient regions (the darker the region, the more important it is). Thus, instead of obtaining a single saliency map for the whole VHR image, we generate several $M \times N$ saliency maps. This has similarities with a top-down visual attention process, because we look for image points having plant individuals undergoing phenological changes over time. The next step consists of creating a foveal model - based on a Hilbert curve - that is able to represent the set of examined cropped images in the time series. The foveation process consists of applying the curve to sample points according to the degree of importance of the regions in the saliency map. In this case, for example, dark regions will have higher point density than white ones. Finally, each cropped image at position p will be represented as a sequence of points extracted by means of the model's curve. Such sequence can be further used as input to different information visualization strategies, such as visual rhythms, graphs, and charts.

5.2.2 Foveation for generating foveal models

This step proceeds in the same way as described in Section 4.2. We employ saliency maps (previously referred to as Gaussian KDEs) in the creation of foveal models based on the Hilbert curve. Essentially, the model can be understood as a sequence of 2-d points resampled according to the distribution of a specific saliency map, and following the properties of the Hilbert curve. The variable relevance of regions in the map induces different refinement degrees across the 2-d space and, consequently, a non-linear aspect on the resulting curve. Thus, a foveal model is able to represent areas in variable resolutions. While foveal regions have high resolutions due to the dense sampling procedures, peripheral ones have low resolutions, because they comprise few sampled points. Appendix E provides more details on this sampling process by means of Algorithm 4.

5.2.3 Proposed information visualization approach

After the saliency estimation and the construction of foveal models, the next step consists of using such models to summarize the data into a more palpable information visualization strategy. By "palpable" we mean a visual representation (i) that contains an amount of information that is considerably lower than the original data, and (ii) that retains relevant information to provide a proper visual analysis of the image time series.

Our visualization approach is inspired on the work of Leite et al. [52], in which several color palettes are introduced to encode different phenological changes in remote sensing images. Here, we use one of the proposed palettes along with a strategy relying on histograms and stacked bar charts. First, for a cropped image, we collect GCC or BCC (blue chromatic coordinate) pixel values of the points defined in the respective model's curve. These vegetation indexes are found by calculating the percentage of the green and blue for each pixel of an image $I(x, y)$, according to Equations 5.1, and 5.2, respectively. The percentage values are calculated up to the second decimal place.

$$GCC(x, y) = \frac{I_G(x, y)}{I_R(x, y) + I_G(x, y) + I_B(x, y)} \tag{5.1}$$

$$BCC(x, y) = \frac{I_B(x, y)}{I_R(x, y) + I_G(x, y) + I_B(x, y)} \tag{5.2}$$

After computing the desired vegetation index for each pixel of an image, we create a normalized stacked bar chart from the sequence of points collected according the model's curve. The chart can be seen as a representation of that sequence in a normalized array of 100 positions, which takes into account the distribution of values, as determined by a histogram. This procedure essentially removes bins that contribute with less than 1% for the total distribution, and also reduces the initial, usually large sequence, to a more compact one. For illustration purposes, Figure 5.2 describes this procedure by showing the creation of a normalized stacked bar chart containing eight positions. In short, the normalized stacked bar chart is created by arranging sequence points (pixel values) along the available positions of the chart according to their frequencies (in the sequence).

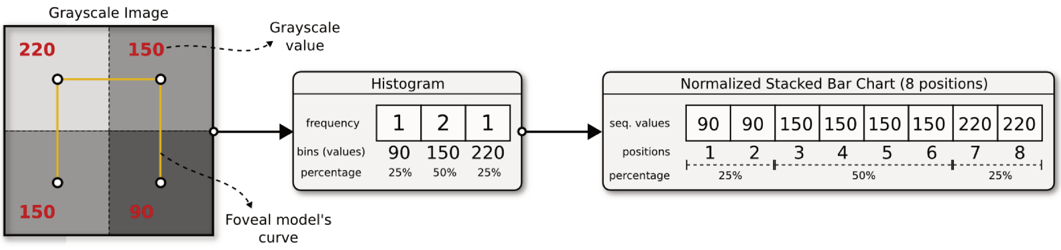


Figure 5.2: Illustration of the normalization procedure for the creation of the normalized stacked bar chart. The chart is a representation of a sequence of collected points according to a model. In this example, we create a normalized stacked bar chart of eight positions based on a sequence comprising four points. We also consider a grayscale image where each pixel has a single color value in the [0, 255] range. First, we compute a color histogram that indicates the frequency of each value in the grayscale image. In this illustrative case, the frequencies for the 90, 150, and 220 values are 25%, 50%, and 25%, respectively. Then, we create the normalized stacked bar chart by arranging the grayscale values along the available eight positions of the chart. Such arrangement takes into account the frequencies of each value in the computed histogram. Thus, for instance, as the 90 value has 50% frequency, it will appear in four of the eight positions of the chart.

The final step to produce the information visualization tool is regarding the use of a color palette. The chosen palette considers GCC or BCC values between 0.31 and 0.5 (i.e., there are 20 bins, each one being represented by a different color). Values outside the interval are rounded to the nearest extreme palette value. According to [52], most of the GCC values are usually found in this range. Thus, we apply the color palette to the normalized stacked bar chart by "painting" each array position with the color defined on the palette. Finally, we combine normalized charts of the cropped images to create the proposed representation.

Figure 5.3 illustrates the construction of the proposed information visualization strategy. The top-most line of the visualization represents the normalized stacked bar chart of the desired pixel representation (GCC or BCC) colored according to the palette's configuration. The next charts (representing other images from the series) are vertically and chronologically stacked below the first one into a single joint representation.

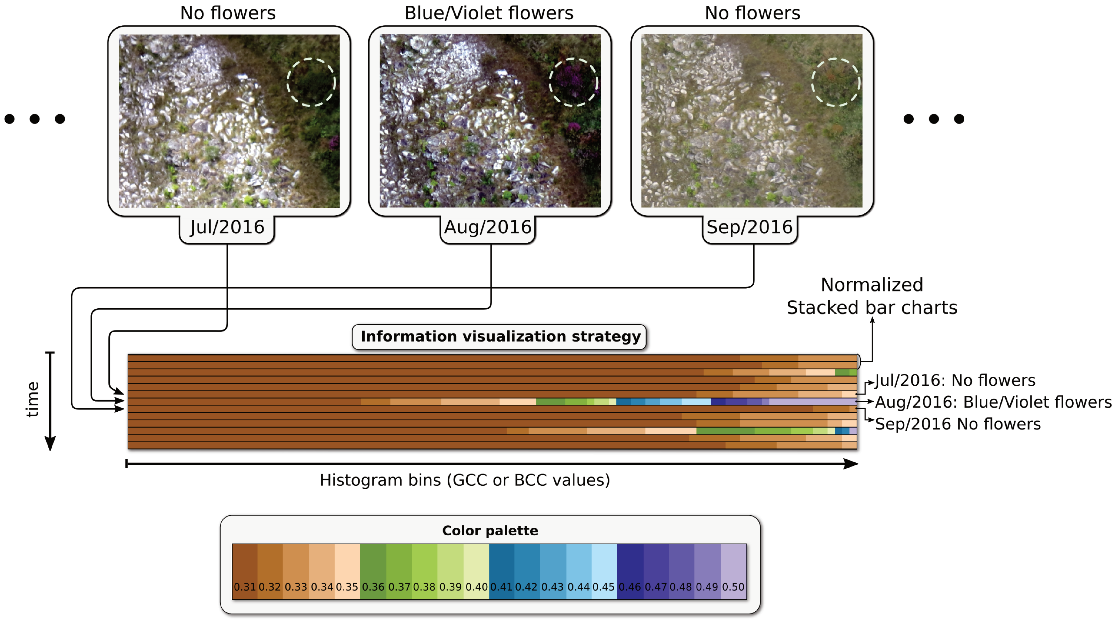


Figure 5.3: The proposed information visualization strategy for a specific cropped area from the VHR image time series. The top-most line of the visualization represents the normalized histogram of the desired pixel representation (GCC or BCC) colored according to the palette's configuration. The next charts (representing other images from the series) are vertically and chronologically stacked bellow the first one.

5.3 Usage Scenarios

We use a dataset of 13 VHR images of a landscape area over one year, from February 2016 to February 2017. The images were captured by means of a Canon SX260 camera mounted on a drone equipment. The camera acquired 4000 × 3000 pixels aerial images of the Serra do Cipo region, which is a Brazilian Cerrado area characterized by high biodiversity and vegetation heterogeneity. These images were later mosaiced into VHR images of size 18817×15623 pixels. This vegetation dataset has been employed in semantic segmentation of plant species [68].

We selected a grid with dimensions $M = 31$ and $N = 32$. The justification for the 31×32 size is that it would to ensure (i) that a reasonable number of medium-size cropped images would be created, and (ii) that the VHR image could be cropped without rejecting any of its rows or columns.

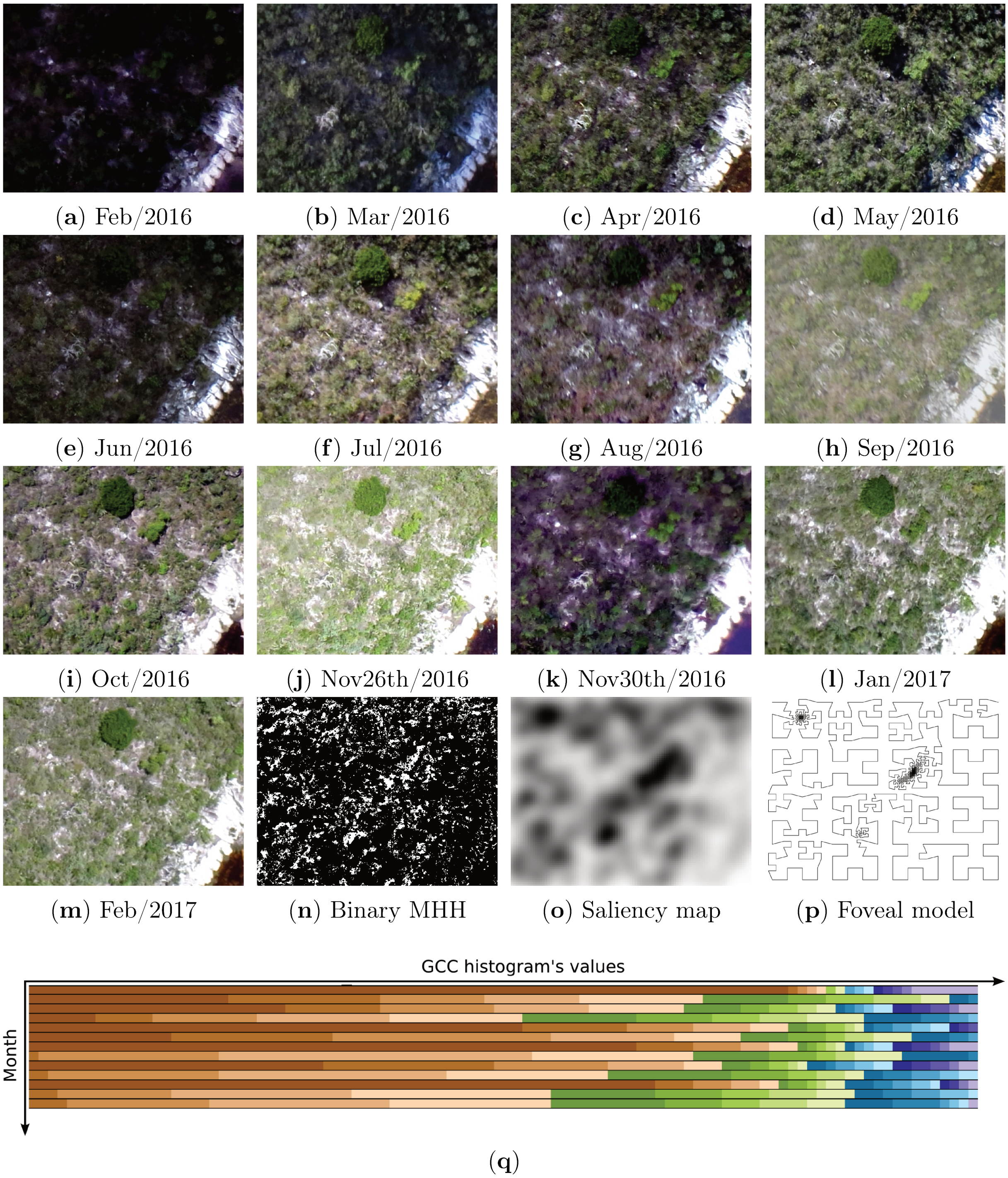
In Figures 5.4 to 5.11, we show some examples of the cropped images from the dataset, the MHHs, the saliency maps, the foveal models, and the information visualization scheme created. In the following, we examine these usage scenarios.

In Figure 5.4, we show an example of the phenological analysis allowed by the use of the framework. The selected area undergoes a green up variation over time, due to the occurrence of a leaf flush phenomenon in certain periods of the year. In addition from the green up phenological metric, we have searched for pixel data following the 010 pattern (see Section 4.2 for details of this concept). Figure 5.4-q is the resulting visualization strategy for the evaluated scenario, showing an increasing on the GCC values in different periods, as expressed in the green-to-blue variation of the palette values. This can be visually verified in the image series. A similar analysis considering the same vegetation area is shown in Figures 5.5, and 5.6 for the 0110, and 01110 patterns, respectively.

By inspecting the provided visualizations, a specialist could conclude, for instance, that the considered vegetation area has more plant individuals undergoing a leaf flush, and changing according to a certain phenological behavior, compared to another area. Another possible information that can be obtained is the period of the year in which the leaf flush occurs. This latter case would be visually expressed as a “peak” of green and blue colors in the provided visualization. Moreover, the visualization may also allow a better comprehension regarding the kind of plant species living on the vegetation area, and compare their current phenological behaviors with those from past years. Still, since the visualization relies on a combination of saliency estimation and foveation, it is possible to obtain a comprehension not only from relevant (foveal) regions, but also from a larger area of the field of view, by also taking peripheral data into account.

In Figures 5.7, 5.8, and 5.9, we exemplify another phenological analysis allowed by the framework. In this analysis, we also examine an area undergoing a green up variation over time, and the 010, 0110, and 01110, respectively. In this example, the leaf flush is less evident, both by inspecting the images and the provided visualizations, which show less green variations.

In Figures 5.10, and 5.11, the selected areas undergo a flowering phenomenon that is characterized by the increase in the levels of blue color (i.e., blue and violet flowers). In this case, we searched for pixels in the Hue range of the blue color. Moreover, due to the fact that flowers may appear in one month and quickly disappear in the next one, the 010 pattern was chosen to characterize the investigated phenomenon. Figures 5.10-q, and 5.11-q are the resulting visualization strategies for the evaluated scenario. Both strategies show an increasing on the BCC values in different periods (green and blue values). This can also be visually verified in the image series.



(**a**) Feb/2016 (**b**) Mar/2016 (**c**) Apr/2016 (**d**) May/2016

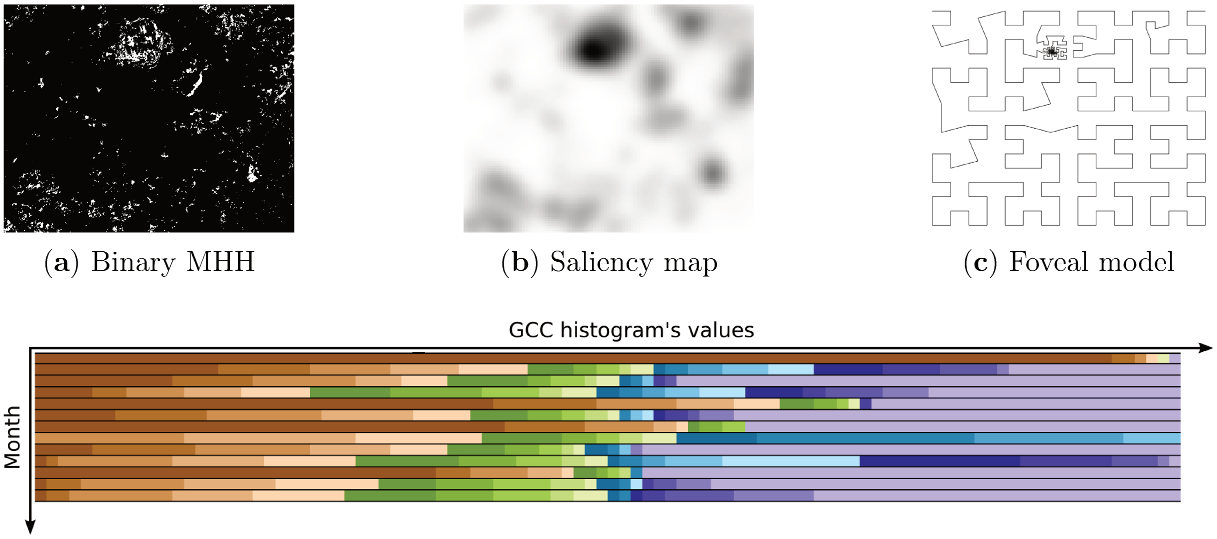
(**e**) Jun/2016 (**f**) Jul/2016 (**g**) Aug/2016 (**h**) Sep/2016

(**i**) Oct/2016 (**j**) Nov26th/2016 (**k**) Nov30th/2016 (**l**) Jan/2017

(**m**) Feb/2017 (**n**) Binary MHH (**o**) Saliency map (**p**) Foveal model

(**q**)

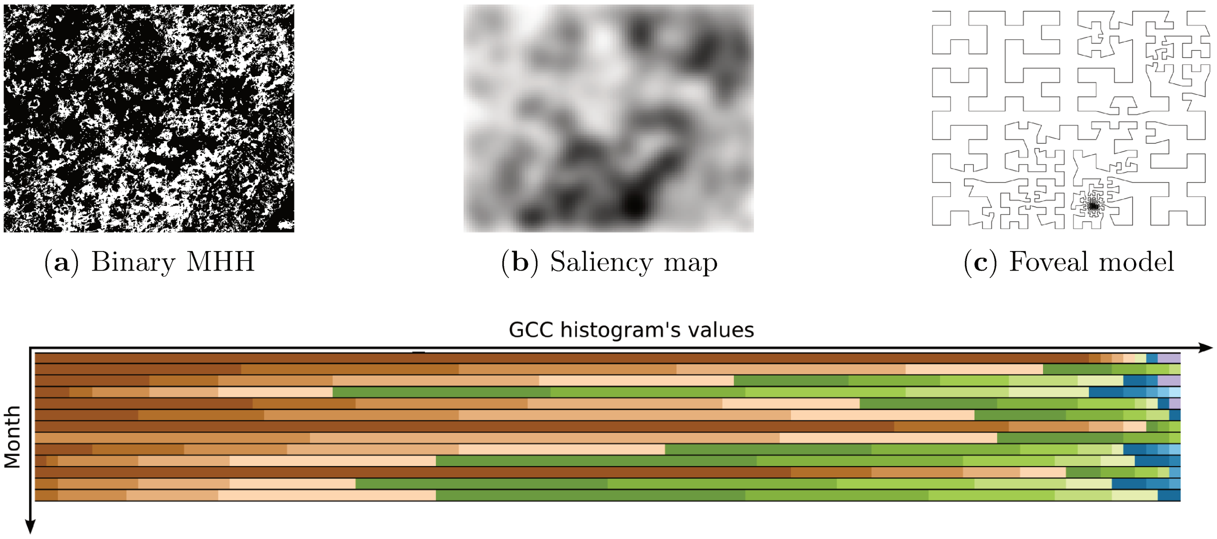
Figure 5.4: Example of the proposed information visualization scheme (**q**) encoding a Green Up phenomenon (**010** pattern) for a specific (cropped) area from the VHR image time series. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**–**m**) is a series of cropped images of a green area, (**n**) is the binary MHH calculated, (**o**) is the saliency map comprising regions of interest, and (**p**) is the foveal model generated.



(**a**) Binary MHH (**b**) Saliency map (**c**) Foveal model

(**d**)

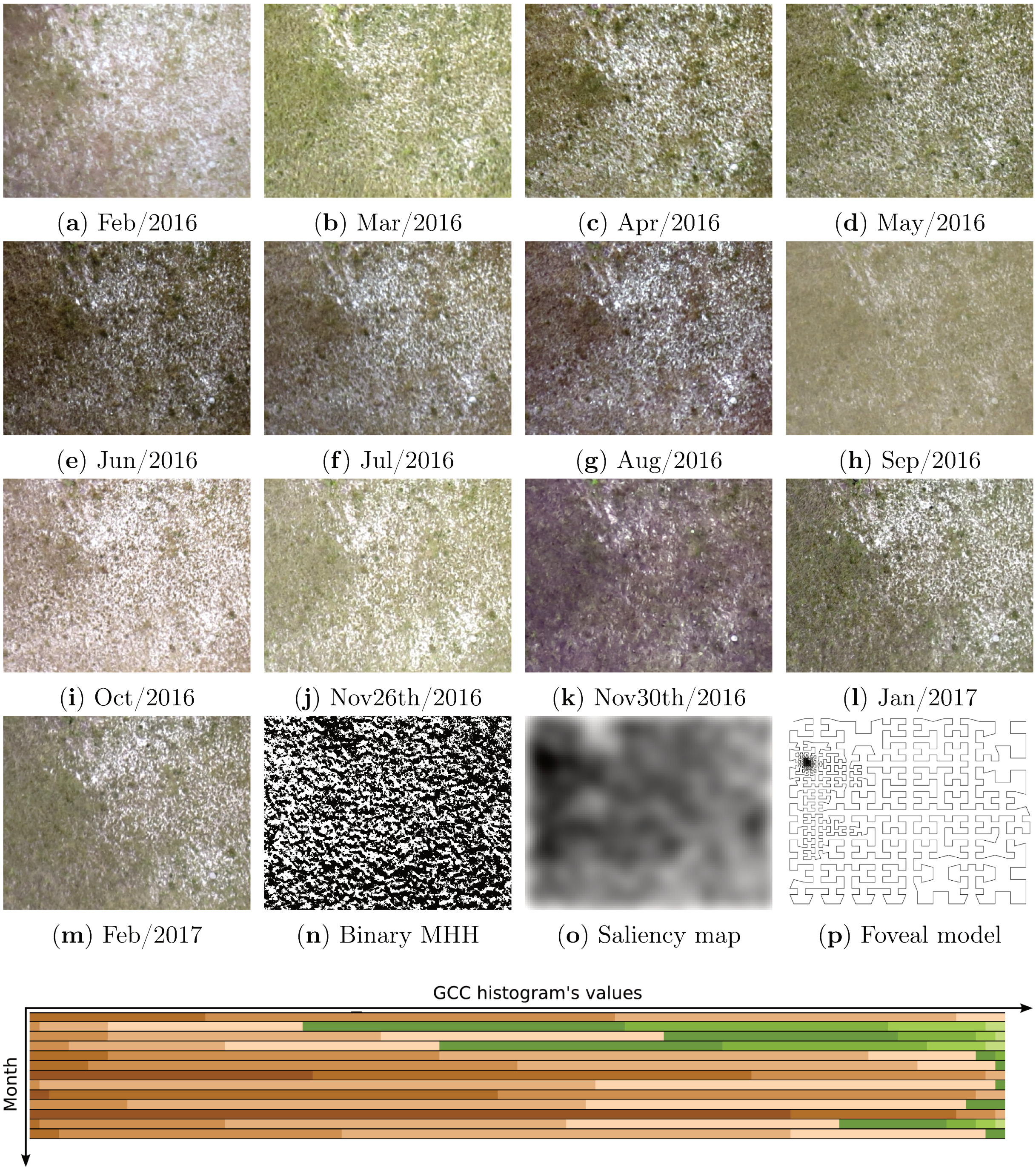
Figure 5.5: Example of the proposed information visualization scheme (**d**) encoding a Green Up phenomenon (**0110** pattern) for a specific (cropped) area from the VHR image time series from Figure 5.4. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**) is the binary MHH calculated, (**b**) is the saliency map comprising regions of interest, and (**c**) is the foveal model generated.



(**a**) Binary MHH (**b**) Saliency map (**c**) Foveal model

(**d**)

Figure 5.6: Example of the proposed information visualization scheme (**d**) encoding a Green Up phenomenon (**01110** pattern) for a specific (cropped) area from the VHR image time series from Figure 5.4. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**) is the binary MHH calculated, (**b**) is the saliency map comprising regions of interest, and (**c**) is the foveal model generated.



(**a**) Feb/2016 (**b**) Mar/2016 (**c**) Apr/2016 (**d**) May/2016

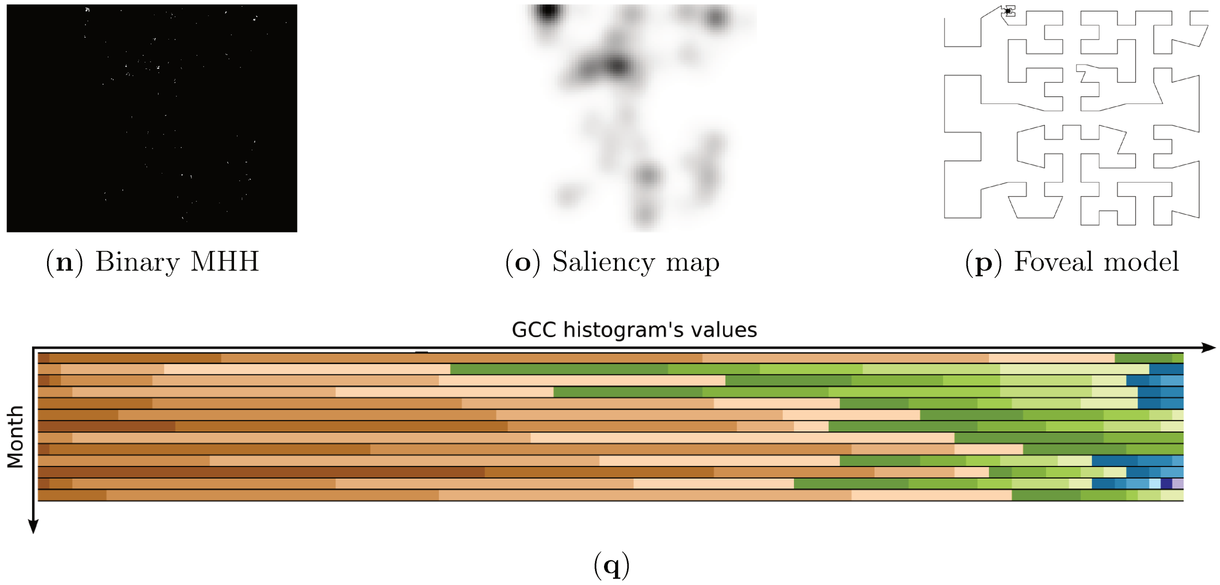
(**e**) Jun/2016 (**f**) Jul/2016 (**g**) Aug/2016 (**h**) Sep/2016

(**i**) Oct/2016 (**j**) Nov26th/2016 (**k**) Nov30th/2016 (**l**) Jan/2017

(**m**) Feb/2017 (**n**) Binary MHH (**o**) Saliency map (**p**) Foveal model

(**q**)

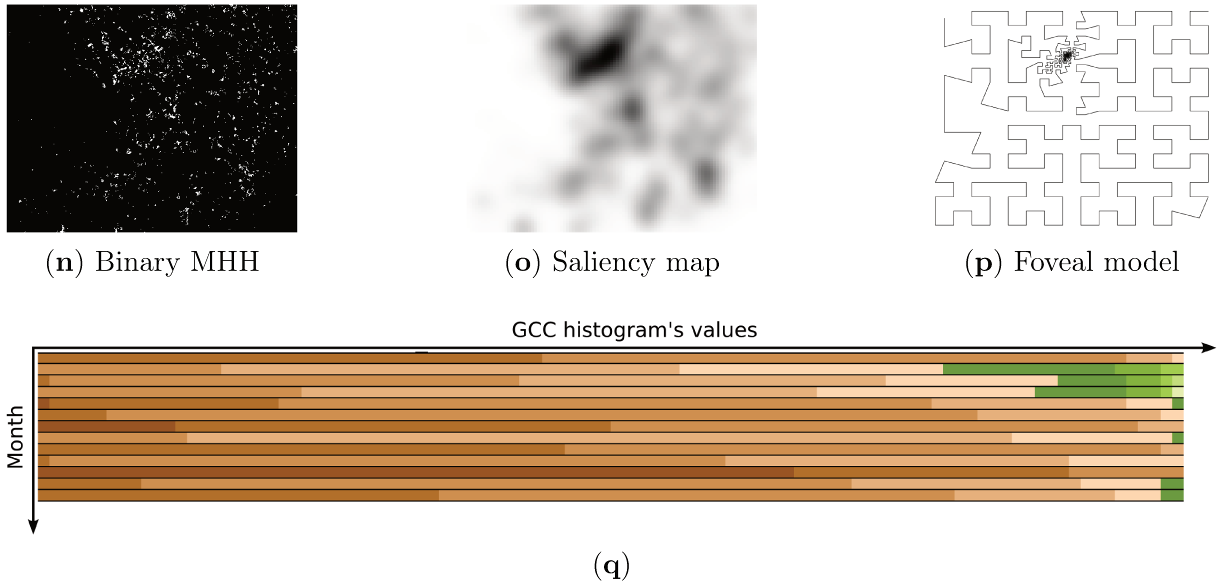
Figure 5.7: Example of the proposed information visualization scheme (**q**) encoding a leaf flush phenomenon (**010** pattern) for a specific area from the VHR image time series. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**–**m**) is a series of cropped images of a green area, (**n**) is the binary MHH calculated, (**o**) is the saliency map comprising regions of interest, and (**p**) is the foveal model generated.



(**n**) Binary MHH (**o**) Saliency map (**p**) Foveal model

(**q**)

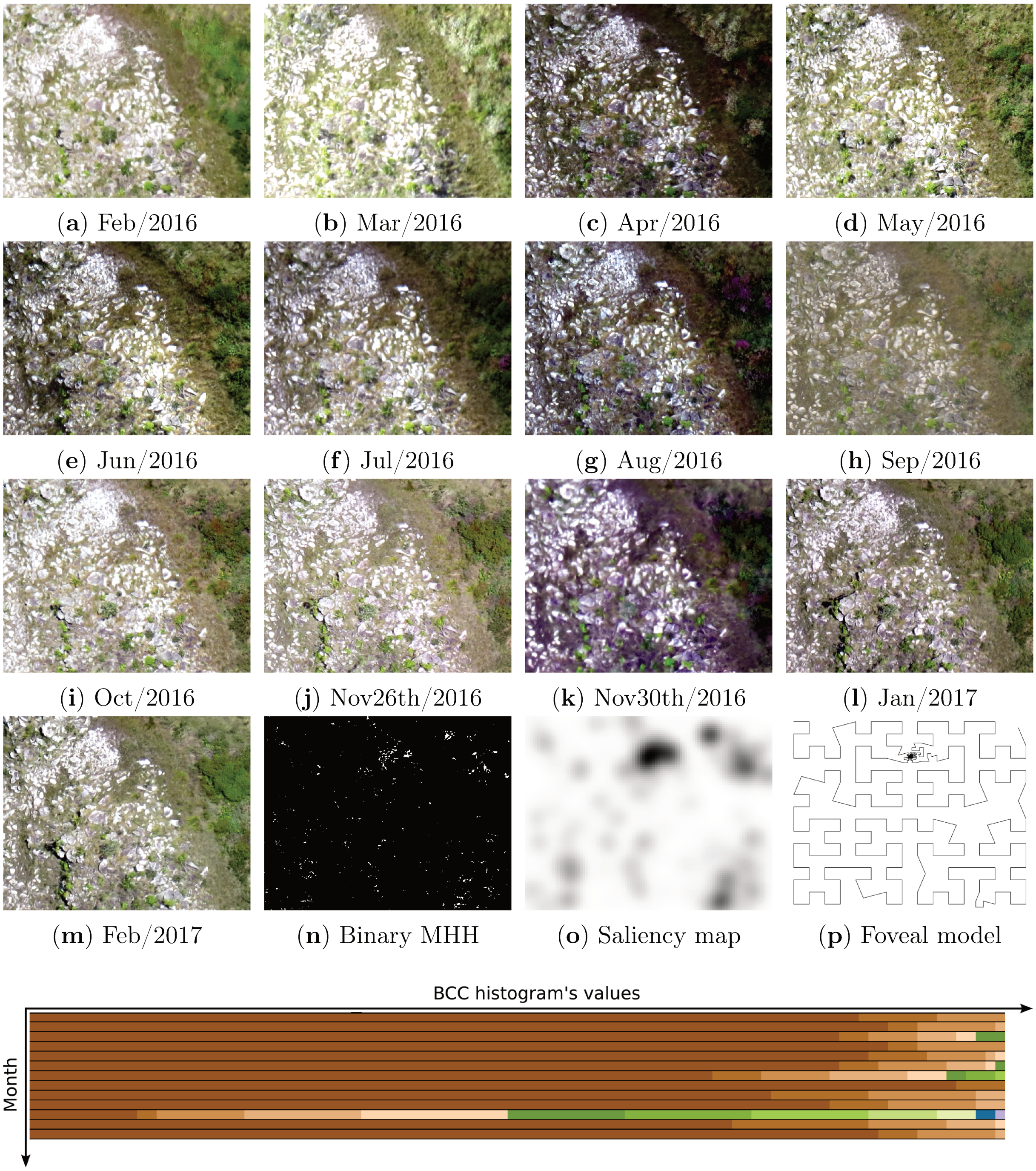
Figure 5.8: Example of the proposed information visualization scheme (**d**) encoding a leaf flush phenomenon (**0110** pattern) for a specific area from the VHR image time series from Figure 5.7. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**) is the binary MHH calculated, (**b**) is the saliency map comprising regions of interest, and (**c**) is the foveal model generated.



(**n**) Binary MHH (**o**) Saliency map (**p**) Foveal model

(**q**)

Figure 5.9: Example of the proposed information visualization scheme (**d**) encoding a leaf flush phenomenon (**01110** pattern) for a specific area from the VHR image time series from Figure 5.7. The phenomenon is caught by investigating and processing images in the GCC space. Time ranges from February 2016 to February 2017. (**a**) is the binary MHH calculated, (**b**) is the saliency map comprising regions of interest, and (**c**) is the foveal model generated.



(**a**) Feb/2016 (**b**) Mar/2016 (**c**) Apr/2016 (**d**) May/2016

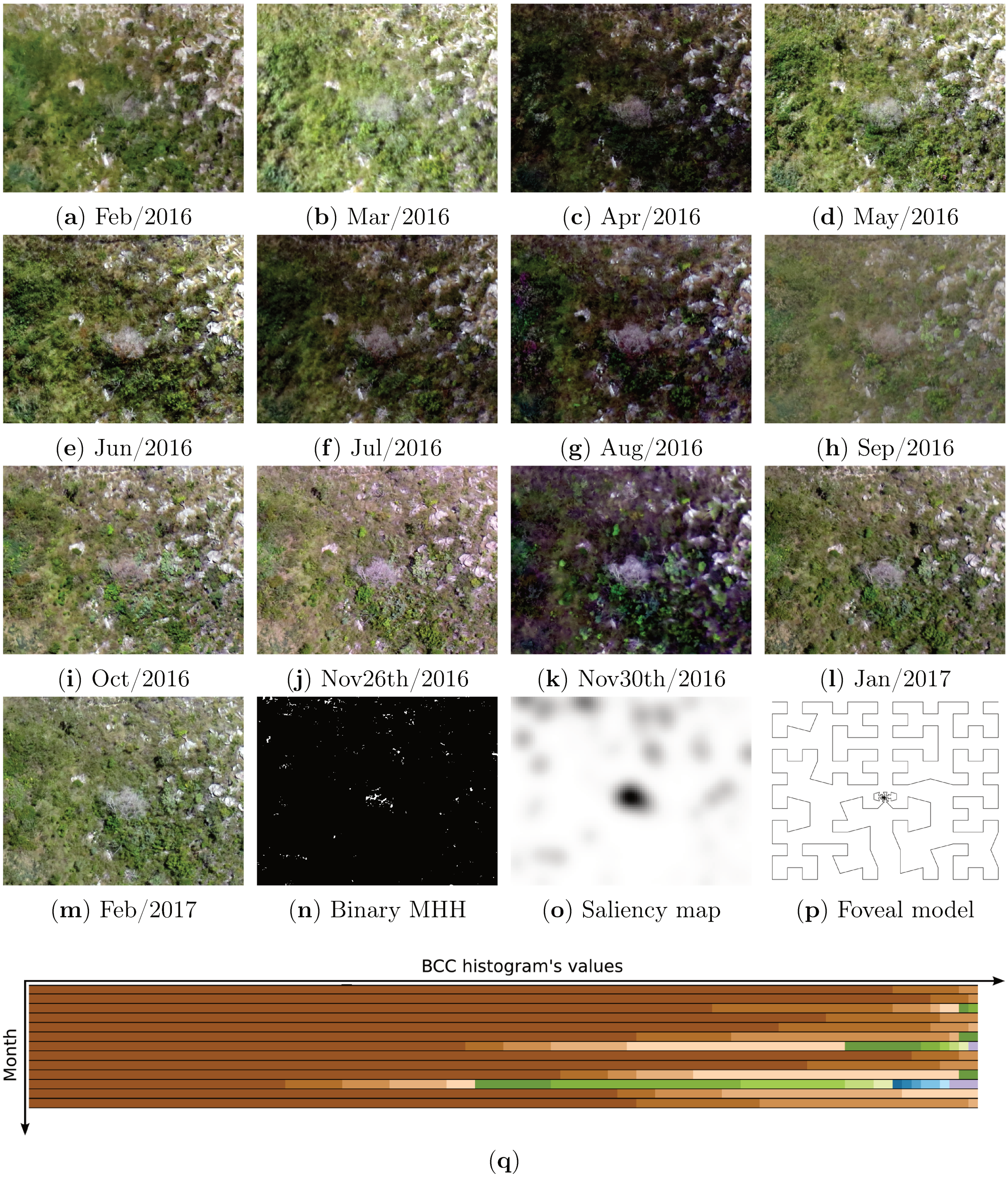
(**e**) Jun/2016 (**f**) Jul/2016 (**g**) Aug/2016 (**h**) Sep/2016

(**i**) Oct/2016 (**j**) Nov26th/2016 (**k**) Nov30th/2016 (**l**) Jan/2017

(**m**) Feb/2017 (**n**) Binary MHH (**o**) Saliency map (**p**) Foveal model

(**q**)

Figure 5.10: Example of the proposed information visualization scheme (**q**) encoding a flowering phenomenon (**010** pattern) for a specific area from the VHR image time series. The phenomenon is caught by investigating and processing images in the BCC space. Time ranges from February 2016 to February 2017. (**a**–**m**) is a series of cropped images of an area containing blue flowers, (**n**) is the binary MHH calculated, (**o**) is the saliency map comprising regions of interest, and (**p**) is the foveal model generated.



(**a**) Feb/2016 (**b**) Mar/2016 (**c**) Apr/2016 (**d**) May/2016

(**e**) Jun/2016 (**f**) Jul/2016 (**g**) Aug/2016 (**h**) Sep/2016

(**i**) Oct/2016 (**j**) Nov26th/2016 (**k**) Nov30th/2016 (**l**) Jan/2017

(**m**) Feb/2017 (**n**) Binary MHH (**o**) Saliency map (**p**) Foveal model

(**q**)

Figure 5.11: Example of the proposed information visualization scheme (**q**) encoding a flowering phenomenon (**010** pattern) for a specific area from the VHR image time series. The phenomenon is caught by investigating and processing images in the BCC space. Time ranges from February 2016 to February 2017. (**a**–**m**) is a series of cropped images of an area containing blue flowers, (**n**) is the binary MHH calculated, (**o**) is the saliency map comprising regions of interest, and (**p**) is the foveal model generated.

5.4 Final Considerations

In this chapter, we have provided our third contribution regarding the use of foveal image models in CV applications. We have proposed an extension to our previous framework for creating foveal image models presented in Chapter 4. Our approach takes into account foveation and visual attention to the development of an information visualization technique that allows more palpable visual inspections of VHR image time series. This was made possible due to a reduction in the volumes of data induced by the use of foveal models in conjunction with the proposed information visualization strategy. We validate our information visualization approach in the remote sensing field, particularly in phenology investigations aiming at comprehending the life cycle events of plants. We have employed a dataset of VHR images (covering a large landscape area) captured with the aid of a drone. Our approach may provide specialists with a visualization tool that allows easier visual image inspections related to the phenological events represented in a sequence of VHR images that, otherwise, would need much more effort to be properly analysed.

Chapter 6

Conclusions

In this chapter, we summarize this thesis by presenting an overview of the scientific contributions in the light of the hypothesis and research questions formulated. We also describe possibilities of future research initiatives derived from our research.

6.1 Scientific Contributions

In the “era of data”, one of the main challenges is how to have these data properly stored for further processing and analysis. In particular, digital images are a good example of such data, because they can be quickly and easily obtained, even in high resolutions. Daily applications available in our smartphones and tablets make intense use of these uniform images to provide us with the desired outputs. The cost of handling these huge amounts of data, materialized in image sequences or videos, is closely related to the limitations in resources of some computing environments and platforms. Given that the lack of proper approaches to manage storage and energy resources may hinder the user’s experience, considering alternatives for uniform images, such as the ones induced by our foveal models, is something that deserves exploration.

In this thesis, we have investigated the use of foveal models in CV applications. We have introduced some approaches to deal with the huge volumes of image data these applications usually have to handle. In the following, we summarize our contributions in terms of each research question previously formulated.

1. **RQ1:** Would the use of foveal image models lead to savings in computational resources, such as memory storage and energy?

   Yes, they do. First, in Chapter 3, we proposed a framework to generate and handle application-oriented retinal image models (ARIM). We validated the framework in a CV application of the biometry domain and showed that the use of ARIMs induce savings in storage and energy resources, at the cost of some drop in the application’s accuracy. We also showed that foveal and peripheral regions of ARIMs can be exploited individually via different pixel representations. This convenience contributed to the lower energy consumption induced by ARIMs. Second, in Chapter 4, we investigated the idea of foveal models in the field of plant phenology. The approach can be seen as an extension of our first contribution in which more abstract and general information from the application's domain is considered, instead of specific 2-d spatial parameters. The devised foveal models contain peripheral (less dense) and foveal (dense) regions, and can be used to reconstruct the 2-d image or as a sequence of 1-d points. We evaluated both scenarios and concluded that the models provide significant storage savings.

2. **RQ2:** Does the use of foveal image models heavily impact in the accuracy rates of the application?

   No, they do not. In Chapter 3, we showed that the use of ARIMs induce savings in storage and energy resources, at the cost of some drop in the application's accuracy. This is expected, since ARIMs were used to resample images in a space-variant manner, and so there are different degrees of spatial resolution across the image space. Still, we believe that the accuracy rates of the evaluated biometry application were satisfactory to allow the application to effectively operate. A similar conclusion was obtained in Chapter 4, where the foveal models were employed under two perspectives: the 2-d reconstructed image and the 1-d sequence of points provided by the Hilbert curve. In both avenues, our approach maintained good Pearson-correlation rates with the original uniform images in an application from the plant phenology field.

3. **RQ3:** How to adapt the proposed frameworks to different application scenarios?

   In Chapter 3, we described a generic framework that can be adapted to several kinds of applications. To accomplish this, an early analysis of the inherent aspects of the application's domain is paramount to the further steps of the framework. Thus, we believe that the framework provides flexibility to some degree. Additionally, given that our second contribution is an extension of the first one, the same flexibility remains. Thus, there are many possibilities to adapt the frameworks to other application scenarios, some of which we present in Section 6.2.

4. **RQ4:** How to devise foveal image models by taking into account relevant image regions found automatically?

   As explained in question RQ1, instead of relying on static 2-d spatial parameters, our second contribution considers more abstract and general information from the application's domain. In this sense, we selected two abstract information from the plant phenology field: a phenological metric and a binary behavior pattern. The former encodes the phenophase (in our case, a leaf flush) being analyzed, whereas the latter encodes a phenological change of interest. Our method combines these elements with motion history histograms (an approach from the object detection literature) to spot out 2-d foveal regions in a binary change map. The map is thus produced from the examination of an image time series. Next, we created a foveal model by sampling the image's space with a Hilbert curve. In summary, this process require few spatial (low-level) parameters, allowing us to devise image models in a more automatic manner by means of high-level parameters.

5. **RQ5:** How to combine foveal image models with information visualization techniques?

   In Chapter 5, we tackled the challenge of handling VHR image time series. In this sense, we extended upon our second contribution by conceptualizing the framework idea in terms of two properties of the HVS: foveation and visual attention (saliency estimation). The goal of the approach was to provide an information visualization technique based on these HVS properties to aid visual inspections of huge volumes of data in VHR image time series. We showed that the proposed technique decreases the amount of data to be examined, therefore aiding the execution of visual tasks related to plant phenology. Such reduction in data volumes also lead to huge memory storage savings, when comparing the proposed visualizations to the full image time series, a common approach in phenology studies. In summary, we effectively combined the framework from Chapter 4 with a visualization technique.

The obtained results from the investigations performed in the context of the research questions have helped us to confirm the following hypothesis underlying this thesis:

The use of foveal image models might induce a reduction in storage usage and energy consumption of computer vision applications, while also keeping satisfactory accuracy rates, and possibly allowing easier visual inspections of series of image data.

The work is associated with the following papers published in international journals:

- Silva, E.; Torres, R.S.; Pinto, A.; Li, L.T.; Vianna, J.E.S.; Azevedo, R.; Goldenstein, S. **Application-Oriented Retinal Image Models for Computer Vision**. *Sensors* 2020, 20, 3746. [82]
- Silva, E.; Torres, R.S.; Alberton, B.; Morellato, L.P.C.; Silva, T.S.F. **A Change-Driven Image Foveation Approach for Tracking Plant Phenology**. *Remote Sens.* 2020, 12, 1409. [83]

6.2 Future Work

The work presented herein can be extended in several ways. In the following, we describe future research opportunities for each one of our contributions.

6.2.1 On the first contribution

With regards to our first contribution (Chapter 3), future work may be concerned with the evaluation of the framework in other CV applications, such as:

- **Remote sensing and surveillance:** In this kind of application (Figure 6.1), instead of processing full-size images, an ARIM containing foveas at the locations of the plant species of interest could be devised. Also, by taking into account the expected phenological behavior of the species over a year, the periphery could have a non-uniform point distribution in which near-foveal areas would present higher point densities than more distant ones. This layout would favor technical investigations, because uncertainty plays a big role in phenological variations, as an expression, for example, of anthropogenic factors. Thus, a non-uniform, but still dense region near the foveas could be useful to account for these variations. We emphasize that, although we have evaluated a similar scenario in our second contribution (Chapter 4), we could still evaluate the use of ARIMs in a remote sensing application.

- **Industrial applications:** In an assembly line (Figure 6.2), inspections may be conducted to check whether products follow the specifications and requirements previously established. These tasks are very important to ensure the quality of the final products. Hence, an automatic analysis of images of these products taken at some time intervals is a viable CV application. In this light, a retinal model could be devised to provide more efficiency to the process. For example, a central fovea with full, uniform resolution could have two other foveas at each side. This pair of lateral foveas would present lower resolution than the central one, so as to capture some early details of the object coming from any side. If a more detailed investigation becomes necessary, the central fovea may be activated. Furthermore, top and bottom peripheral regions could be used to capture any movement to help in the foveal analysis (e.g., human or robotic interventions occurring at the moment).

- **Action recognition applications:** Another relevant CV application able to exploit the benefits provided by the use of ARIMs refers to automatic systems for detecting driver's fatigue and other facial actions (Figure 6.3). This system could comprise a wide-lens camera to capture both the driver's and the side passenger's faces. Thus, a retinal model to represent such a scenario could have two foveas located in the left and right sides of the image space, whereas the remaining regions would constitute the periphery. Since the main idea of the system might be the analysis of facial actions of the driver, "his fovea" would have full resolution. In contrast, the passenger's fovea would have low resolution, be significantly smaller, or even constitute the periphery, where motion detection may take place. An integrated system could also emit some kind of warning signal based on face expressions, signs of drowsiness, or hazardous behavior/movement from the driver/passengers, or even based on the presence of objects inside the vehicle via motion analysis.

Further initiatives evaluating ARIMs in different CV applications may also come across new challenges regarding the flexibility of models. For example, taking the face authentication application considered in our experiments, there might be circumstances when faces should be recognized outside the foveal regions. In these cases, two approaches could be used. First, an ARIM could be built so as to have more foveas conveniently located in strategic image regions. Another possibility is creating different models to be exchanged with each other according to some criteria. Thus, when the optical flow analysis suggests that the individual is near the foveal area or partially inside it, other models having larger or more foveas could be used to spot out the face. These alternative models would be applied only in specific cases where the application's effectiveness could potentially be compromised. In a broader perspective, we believe the trade-off between accuracy and energy/storage consumption induced by the collaborative use of different models, specifically tailored for other CV applications, is a promising line of investigation for future work. Another possibility is to represent the periphery of ARIMs as super-pixel-like artifacts (Voronoi cells) that could be filled with the grayscale pixel value at each cell's central point in the original image (Figure 6.4). The analysis of degraded peripheral regions represented in grayscale might also be applied to the aforementioned application domains.

With respect to the hardware side, a promising avenue is the effective integration of our approach into an FPGA responsible for resampling uniform images according to some predefined or dynamic space-variant models. The models could be computed at the FPGA or by software, in which case an efficient communication mechanism between these layers should be implemented. To provide more flexibility, the collaborative use of different models and the use of computer graphics techniques to dynamically reshape them (as with the case of deformable surface models) are relevant investigations for further work. In these scenarios, a more complex repertoire of variables would need to be considered, including the costs of computing and adapting the models, resampling uniform images inside the FPGA, and the trade-off among accuracy, storage, and energy. Even with all of these variables in the field, we believe that such an infrastructure could still favor savings in the use of computational resources. Another idea that deserves some study is the use of our models along with alternative technologies for data storage, such as Network Attached Storage systems. In this setup, bandwidth resources may benefit from the use of the space-variant image representation induced by our models.

6.2.2 On the second contribution

Our second contribution (Chapter 4) may be the target of future initiatives both on the hardware and software sides. First, examining space-variant imaging sensors and FPGA devices that are able to handle foveation at the hardware layer is a promising research avenue, as these could boost the autonomy of the image acquisition process, particularly in remote areas. In this context, evaluating energy consumption levels of different foveation procedures from the literature, performed on variable hardware and software platforms, would be invaluable to the field. Second, a mathematical formulation for directly retrieving a 2-d point from a 1-d sequence point based on the non-linear Hilbert curve also deserves investigation. This is a challenging approach that could allow direct evaluations of particular areas in the 1-d sequence. Third, the proposed foveal models could be based on different SFCs, such as the Peano curve. In this case, a detailed analysis and comparison of the benefits and drawbacks of these curves and their impact of the reconstructed images and 1-d sequences would be necessary.

6.2.3 On the third contribution

Our third contribution (Chapter 5) may be extended in some ways. The proposed approach can be exploited in other domains where sequences of high-resolution images need to be analyzed, such as those related to video-based surveillance systems. Moreover, different information visualization strategies relying on the foveal models could be devised. In this vein, several visualization techniques may be experimented and combined into novel, more robust ones which may employ diverse graphical elements, spatial pixel setups, color palettes/schemes, etc. For instance, a radial-like setup could be employed [60, 61] for the stacked bar chart representation. Finally, another plausible investigation could be conducted to improve the generation of saliency maps by integrating low-level visual cues into the process, such as the shape and size of elements.

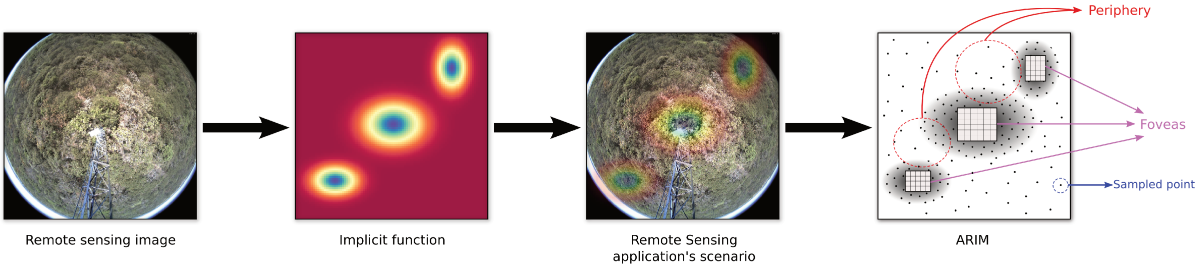
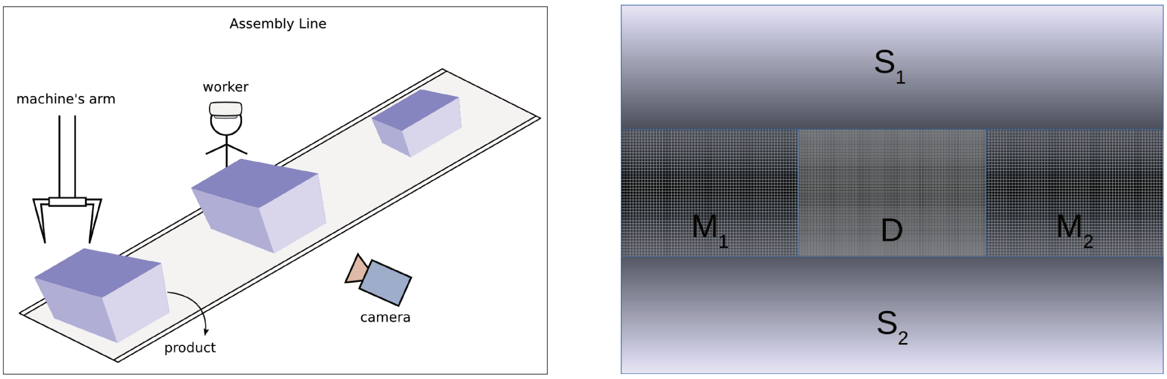
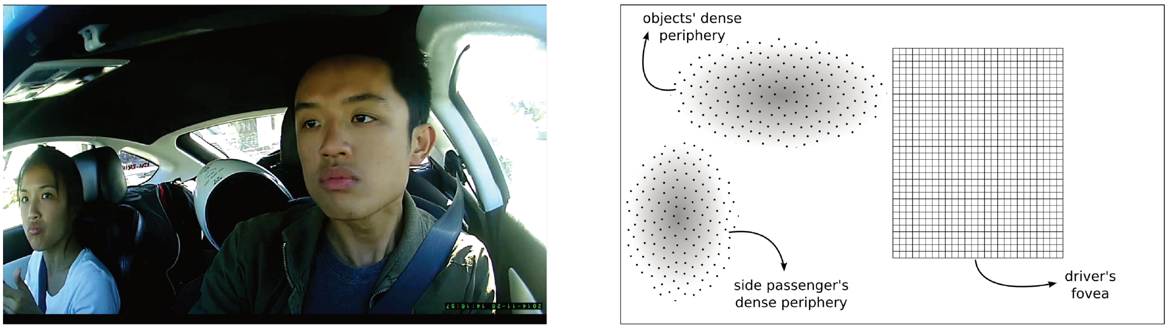


Figure 6.1: Illustration of the remote sensing application scenario using ARIMs. The idea may be appropriated for plant phenology tasks.



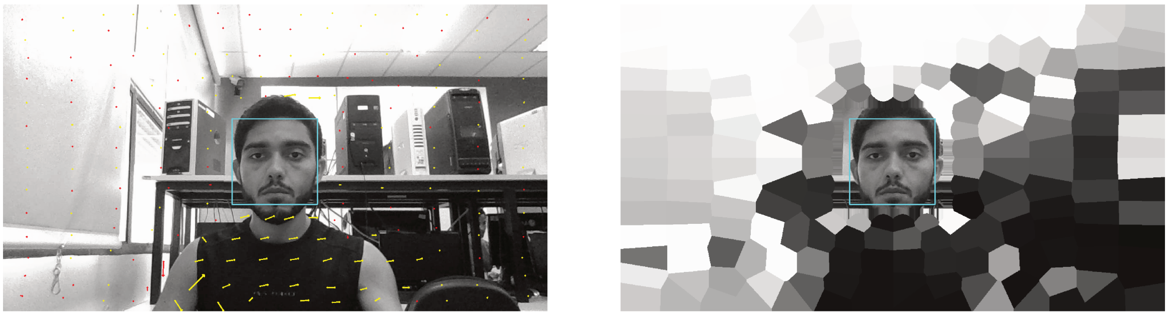
(**a**) (**b**)

Figure 6.2: Illustration of the industrial application scenario (**a**) using ARIMs (**b**). The idea may be appropriated for assembly line inspection tasks.



(**a**) (**b**)

Figure 6.3: Illustration of the action recognition application scenario (**a**) using ARIMs (**b**). The idea may be appropriated for driver's fatigue and other facial expression recognition tasks. Image (**a**) is a frame from the Brain4Cars Dataset [44].



(**a**) (**b**)

Figure 6.4: Image reconstruction based on an ARIM that uses a grayscale pixel representation on the periphery. The original image (**a**) is reconstructed (**b**) with voronoi cells, each of which has a single grayscale color.

Bibliography

[1] D.J. Abel and J.L. Smith. A data structure and algorithm based on a linear key for a rectangle retrieval problem. *Computer Vision, Graphics, and Image Processing*, 24(1):1–13, 1983.

[2] N. Abramson, editor. *Information Theory and Coding*. McGraw-Hill, New York, 1963.

[3] Emre Akbas and Miguel P. Eckstein. Object detection through search with a foveated visual system. *PLOS Computational Biology*, 13(10):1–28, 10 2017.

[4] Bruna Alberton, Jurandy Almeida, Raimund Helm, Ricardo da S. Torres, Annette Menzel, and Leonor Patricia Cerdeira Morellato. Using phenological cameras to track the green up in a cerrado savanna and its on-the-ground validation. *Ecological Informatics*, 19:62 – 70, 2014.

[5] Bruna Alberton, Ricardo da S. Torres, Leonardo F. Cancian, Bruno D. Borges, Jurandy Almeida, Greice C. Mariano, Jefersson dos Santos, and Leonor Patricia Cerdeira Morellato. Introducing digital cameras to monitor plant phenology in the tropics: applications for conservation. *Perspectives in Ecology and Conservation*, 15(2):82–90, 2017.

[6] Mir Ashfaque Ali and S. A. Ladhake. Overview of space-filling curves and their applications in scheduling. *Intl. Journal of Advances in Engineering & Technology*, 1:148–154, 2011.

[7] David Alleysson and Nathalie Guyader. *Retinal Processing: From Biology to Models and Applications*, pages 29–52. 08 2015.

[8] Jurandy Almeida, Jefersson A. dos Santos, Bruna Alberton, Leonor Patricia C. Morellato, and Ricardo da Silva Torres. Phenological visual rhythms: Compact representations for fine-grained plant species identification. *Pattern Recognition Letters*, 81:90–100, October 2016.

[9] H. Araujo and J. M. Dias. An introduction to the log-polar mapping [image sampling]. In *Proceedings of the Second Workshop on Cybernetic Vision*, pages 139–144, Dec 1996.

[10] Tetsuo Asano. Digital halftoning algorithm based on random space-filling curve. In *Proceedings of the Intl. Conference on Image Processing*, volume 1, pages 545–548, Sep 1996.

[11] Tetsuo Asano, Desh Ranjan, Thomas Roos, Emo Welzl, and Peter Widmayer. Space-filling curves and their use in the design of geometric data structures. *Theoretical Computer Science*, 181(1):3–15, jul 1997.

[12] R. Ascher and G. Nagy. A means for achieving a high degree of compaction on scan-digitized printed text. *IEEE Transactions on Computers*, 23(11):1174–1179, nov 1974.

[13] Michael Bader. *Space-Filling Curves: An Introduction with Applications in Scientific Computing*, volume 1. Springer-Verlag, 1st edition, 2013.

[14] L. R. Bahl and H. Kobayashi. Image data compression by predictive coding ii: Encoding algorithms. *IBM Journal of Research and Development*, 18(2):172–179, 1974.

[15] Donald G. Bailey. and Christos-Savvas Bouganis. *Vision Sensor with an Active Digital Fovea*, pages 91–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[16] A. Basu and K. Wiebe. Enhancing videoconferencing using spatially varying sensing. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(2):137–148, Mar 1998.

[17] Kenan Bektas, Arzu Cöltekin, Jens Krüger, and Andrew T. Duchowski. A Testbed Combining Visual Perception Models for Geographic Gaze Contingent Displays. In E. Bertini, J. Kennedy, and E. Puppo, editors, *Eurographics Conference on Visualization (EuroVis) - Short Papers*. The Eurographics Association, 2015.

[18] Fabio Berton, Giulio Sandini, and Giorgio Metta. *Anthropomorphic visual sensors*. American Scientific Publishers, USA, Stevenson Ranch, CA, 2006.

[19] Marc Bolduc and Martin D. Levine. A review of biologically motivated space-variant data reduction models for robotic vision. *Computer Vision and Image Understanding*, 69(2):170–184, 1998.

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

[21] Ali Borji. Saliency prediction in the deep learning era: Successes and limitations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 08 2019.

[22] James Bornholt, Todd Mytkowicz, and Kathryn S. Mckinley. The model is not enough: Understanding energy consumption in mobile devices. *Power (watts)*, 1(2):3, 2012.

[23] Hue-Ling Chen and Ye-In Chang. Neighbor-finding based on space-filling curves. *Journal of Information Systems*, 30(3):205–226, May 2005.

[24] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu. Big data for remote sensing: Challenges and opportunities. *Proceedings of the IEEE*, 104(11):2207–2219, Nov 2016.

[25] Revital Dafner, Daniel Cohen-Or, and Yossi Matias. Context-based space filling curves. In *Eurographics Conference*, 2000.

[26] Ashraf Darwish. Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications. *Future Computing and Informatics Journal*, 3(2):231 – 246, 2018.

[27] Fernando de Goes, Siome Goldenstein, and Luiz Velho. A simple and flexible framework to adapt dynamic meshes. *Computers & Graphics*, 32(2):141-148, 2008.

[28] Harish Doraiswamy, Juliana Freire, Marcos Lage, Fábio Miranda, and Cláudio T. Silva. Spatio-temporal urban data analysis: A visual analytics perspective. *IEEE Computer Graphics and Applications*, 38(5):26–35, 2018.

[29] F. Ferrari, J. Nielsen, P. Questa, and G. Sandini. Space variant imaging. *Sensor Review*, 15(2):18–20, 1995.

[30] Andrew Floren and Alan C. Bovik. Chapter 14 - foveated image and video processing and search. In Joel Trussell, Anuj Srivastava, Amit K. Roy-Chowdhury, Ankur Srivastava, Patrick A. Naylor, Rama Chellappa, and Sergios Theodoridis, editors, *Academic Press Library in Signal Processing: Volume 4*, volume 4 of *Academic Press Library in Signal Processing*, pages 349 – 401. Elsevier, 2014.

[31] Guillermo Gallego, Tobi Delbruck, Garrick Michael Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jorg Conradt, Kostas Daniilidis, and et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[32] Wilson S. Geisler and Jeffrey S. Perry. A real-time foveated multiresolution system for low-bandwidth video communication. In *Proc. SPIE*, pages 294–305, 1998.

[33] Fabian Göbel, Peter Kiefer, Ioannis Giannopoulos, Andrew T. Duchowski, and Martin Raubal. Improving map reading with gaze-adaptive legends. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, pages 29:1–29:9, New York, NY, USA, 2018. ACM.

[34] Siome Goldenstein, Christian Vogler, and Luiz Velho. Adaptive deformable models for graphics and vision. *Computer Graphics Forum*, 24(4):729–741, 2005.

[35] S. Golomb. Run-length encodings. *IEEE Transactions on Information Theory*, 12(3):399–401, 1966.

[36] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice-Hall, Inc., 3rd edition, 2008.

[37] Martin González, Antonio Sánchez-Pedraza, Rebeca Marfil, Juan A. Rodríguez, and Antonio Bandera. Data-driven multiresolution camera using the foveal adaptive pyramid. *Sensors*, 16(12), 2016.

[38] Leo John Grady. *Space-variant Computer Vision: A Graph-theoretic Approach.* PhD thesis, Boston University, Boston, MA, USA, 2004.

[39] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[40] Rolando Herrero and Vinay K. Ingle. Space-filling curves applied to compression of ultraspectral images. *Signal, Image and Video Processing*, 9(6):1249–1257, 2013.

[41] Rolando Herrero and Vinay K. Ingle. Ultraspectral image preprocessing with tridimensional space filling curves. *Open Transactions on Information Processing*, 1(1):27–42, 2014.

[42] Yanbo Huang, Zhong xin CHEN, Tao YU, Xiang zhi HUANG, and Xing fa GU. Agricultural remote sensing big data: Management and applications. *Journal of Integrative Agriculture*, 17(9):1915 – 1931, 2018.

[43] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[44] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3182–3190, 2015.

[45] Rehna Jameela and M. Kumar. Wavelet based image coding schemes : A recent survey. *International Journal on Soft Computing*, 3, 09 2012.

[46] Cheung-Woon Jho and Won-Hyung Lee. Digital halftoning with single pixel error diffusion using random space filling curve. *Advanced Science and Technology Letters*, 39:153–158, 2013.

[47] F. Jurie. A new log-polar mapping for space variant imaging. *Pattern Recognition*, 32(5):865 – 875, 1999.

[48] Kashif Nizam Khan, Mikael Hirki, Tapio Niemi, Jukka K. Nurminen, and Zhonghong Ou. Rapl in action: Experiences in using rapl for power measurements. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, 3(2), March 2018.

[49] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[50] Stephen Klosterman, Eli Melaas, Jonathan A. Wang, Arturo Martinez, Sidni Frederick, John O'Keefe, David A. Orwig, Zhuosen Wang, Qingsong Sun, Crystal Schaaf, Mark Friedl, and Andrew D. Richardson. Fine-scale perspectives on landscape phenology from unmanned aerial vehicle (uav) photography. *Agricultural and Forest Meteorology*, 248:397 – 407, 2018.

[51] H. Kobayashi and L. R. Bahl. Image data compression by predictive coding i: Prediction algorithms. *IBM Journal of Research and Development*, 18(2):164–171, 1974.

[52] Roger A. Leite, Lucas Mello Schnorr, Jurandy Almeida, Bruna Alberton, Leonor Patricia C. Morellato, Ricardo da S. Torres, and João L.D. Comba. Phenovis - a tool for visual phenological analysis of digital camera images using chronological percentage maps. *Inf. Sci.*, 372(C):181–195, December 2016.

[53] Jan-Yie Liang, Chih-Sheng Chen, Chua-Huang Huang, and Li Liu. Lossless compression of medical images using hilbert space-filling curves. *Computerized Medical Imaging and Graphics*, 32(3):174–182, 2008.

[54] Chang-Chou Lin and Wen-Hsiang Tsai. Visual cryptography for gray-level images by dithering techniques. *Pattern Recognition Letters*, 24(1–3):349 – 358, 2003.

[55] Shao-Ping Lu and Song-Rai Zhang. Saliency-based fidelity adaptation preprocessing for video coding. *J. Comput. Sci. Technol.*, 26(1):195–202, January 2011.

[56] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, page 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[57] Clifford Lynch. Big data: How do your data grow? *Nature*, 455(7209):28–29, 2008.

[58] Rebeca Marfil, Antonio Jesús Palomino, and Antonio Bandera. Combining segmentation and attention: a new foveal attention model. In *Frontiers in Computational Neuroscience*, 2014.

[59] G. C. Mariano, N. C. Soares, L. P. C. Morellato, and R. D. S. Torres. Change frequency heatmaps for temporal multivariate phenological data analysis. In *2017 IEEE 13th International Conference on e-Science (e-Science)*, pages 305–314, Oct 2017.

[60] Greice C. Mariano, Bruna Alberton, Leonor Patricia C. Morellato, and Ricardo da Silva Torres. Radialpheno: A tool for near-surface phenology analysis through radial layouts. *Applications in Plant Sciences*, 7(6):e01253, 2019.

[61] Greice C. Mariano, Vanessa G. Staggemeier, Leonor Patricia Cerdeira Morellato, and Ricardo da S. Torres. Multivariate cyclical data visualization using radial visual rhythms: A case study in phenology analysis. *Ecological Informatics*, 46:19 – 35, July 2018.

[62] Hongying Meng, Nick Pears, Michael Freeman, and Chris Bailey. *Motion History Histograms for Human Action Recognition*, pages 139–162. Springer London, London, 2009.

[63] Mohamed F. Mokbel and Walid G. Aref. Irregularity in multi-dimensional space-filling curves with applications in multimedia databases. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, pages 512–519. ACM, 2001.

[64] Leonor Patrííicia Cerdeira Morellato, Bruna Alberton, Swanni T. Alvarado, Bruno Borges, Elise Buisson, Maria Gabriela G. Camargo, Leonardo F. Cancian, Daniel W. Carstensen, Diego F.E. Escobar, Patrícia T.P. Leite, Irene Mendoza, Nathália M.W.B. Rocha, Natalia C. Soares, Thiago Sanna Freire Silva, Vanessa G. Staggemeier, Annia Susin Streher, Betânia C. Vargas, and Carlos A. Peres. Linking plant phenology to conservation biology. *Biological Conservation*, 195:60 – 72, 2016.

[65] Patrícia Morellato, Bruna Alberton, Jurandy Almeida, Jefersson Alex, Greice Mariano, and Ricardo Torres. e-phenology: monitoring leaf phenology and tracking climate changes in the tropics. *EGUGA*, page 12020, 2014.

[66] Hugo Vieira Neto, Diogo Rosa Kuiaski, and Gustavo Benvenutti Borba. Efficient retina-like resampling from cartesian images. In *VII Workshop de Visão Computacional (WVC2011)*, pages 91–104, 2011.

[67] Yajun Niu, Jun Chang, Fengxiang Lv, Benlan Shen, and Weilin Chen. Low-cost dynamic real-time foveated imager. *Appl. Opt.*, 56(28):7915–7920, Oct 2017.

[68] K. Nogueira, J. A. dos Santos, L. Cancian, B. D. Borges, T. S. F. Silva, L. P. Morellato, and R. d. S. Torres. Semantic segmentation of vegetation images acquired by unmanned aerial vehicles using an ensemble of convnets. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3787–3790, July 2017.

[69] Fernando Pardo, Isaac Llorens, Francisco Micó, and José A. Boluda. Space variant vision and pipelined architecture for time to impact computation. In *Proceedings of the 5th International Workshop on Computer Architectures for Machine Perception, 2000*, pages 122–126, 2000.

[70] Frieda Parker. Space-filling curves. `http://www.dblab.ntua.gr/~gtsat/collection/SFCs/SFC_Paper.pdf`, 2008.

[71] David B. Phillips, Ming-Jie Sun, Jonathan M. Taylor, Matthew P. Edgar, Stephen M. Barnett, Graham M. Gibson, and Miles J. Padgett. Adaptive foveated single-pixel imaging with dynamic supersampling. *Science Advances*, 3(4), 2017.

[72] Bradley C Reed, Mark D Schwartz, and Xiangming Xiao. Remote Sensing Phenology: Status and the Way Forward. In Noormets A., editor, *Phenology of Ecosystem Processes*, pages 231–246. Springer, New York, NY, 2009.

[73] Andrew D. Richardson, Bobby H. Braswell, David Y. Hollinger, Julian P. Jenkins, and Scott V. Ollinger. Near-surface remote sensing of spatial and temporal variation in canopy phenology. *Ecological Applications*, 19(6):1417–28, sep 2009.

[74] Jihoon Ryoo, Kiwon Yun, Dimitris Samaras, Samir R. Das, and Gregory Zelinsky. Design and evaluation of a foveated video streaming service for commodity client devices. In *Proceedings of the 7th International Conference on Multimedia Systems*, MMSys '16, pages 1–6, New York, NY, USA, 2016. ACM.

[75] David Salomon. *Data Compression: The Complete Reference*, volume 1. Springer-Verlag, 2nd edition, 2012.

[76] Giulio Sandini and Vincenzo Tagliasco. An anthropomorphic retina-like structure for scene analysis. *Computer Graphics and Image Processing*, 14(4):365–372, 1980.

[77] Brian Scassellati. Eye finding via face detection for a foveated, active vision system. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, pages 969–976. American Association for Artificial Intelligence, 1998.

[78] Eric L. Schwartz. Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biological Cybernetics*, 25(4):181–194, 1977.

[79] M. D. Schwartz. *Phenology: An Integrative Environmental Science.* Springer, 2013.

[80] G. Ruthwik Sharma, G. Shirisha, and Dr. M. Narsing Yadav. Visual cryptography using space filling curve ordered dithering with adaptive clustering. *International Journal of Advanced Trends in Computer Science and Engineering*, 3(1):569 – 574, 2014.

[81] C. Shi, J. Zhang, and Y. Zhang. A novel vision-based adaptive scanning for the compression of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1336–1348, March 2016.

[82] Ewerton Silva, Ricardo da S. Torres, Allan Pinto, Lin Tzy Li, José Eduardo S. Vianna, Rodolfo Azevedo, and Siome Goldenstein. Application-oriented retinal image models for computer vision. *Sensors*, 20(13):3746, Jul 2020.

[83] Ewerton Silva, Ricardo da S. Torres, Bruna Alberton, Leonor Patricia C. Morellato, and Thiago S. F. Silva. A change-driven image foveation approach for tracking plant phenology. *Remote Sensing*, 12(9):1409, Apr 2020.

[84] F Smeraldi and J Bigun. Retinal vision applied to facial features detection and face authentication. *Pattern Recognition Letters*, 23(4):463 – 475, 2002.

[85] Oliver Sonnentag, Koen Hufkens, Cory Teshera-Sterne, Adam M. Young, Mark Friedl, Bobby H. Braswell, Thomas Milliman, John O'Keefe, and Andrew D. Richardson. Digital repeat photography for phenological research in forest ecosystems. *Agricultural and Forest Meteorology*, 152:159 – 177, 2012.

[86] Richard Szeliski. *Computer vision: algorithms and applications.* Texts in computer science. Springer, New York, London, 2011.

[87] Javier V. Traver, Alexandre Bernardino, Plinio Moreno, and José Santos-Victor. *Appearance-Based Object Detection in Space-Variant Images: A Multi-model Approach*, pages 538–546. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[88] V. Javier Traver and Alexandre Bernardino. A review of log-polar imaging for visual perception in robotics. *Robotics and Autonomous Systems*, 58(4):378–398, 2010.

[89] V. Javier Traver and Filiberto Pla. Log-polar mapping template design: From task-level requirements to geometry parameters. *Image and Vision Computing*, 26(10):1354–1370, 2008.

[90] J. Tu, H. Sui, W. Feng, and Q. Jia. Detecting facade damage on moderate damaged type from high-resolution oblique aerial images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(12):5598–5607, Dec 2017.

[91] Jonas Valantinas. On the use of space-filling curves in changing image dimensionality. *Information, Technology and Control (ITC)*, 34(4):354–354, 2005.

[92] Levi Valgaerts. Space-filling curves: An introduction. `http://wwwmayr.informatik.tu-muenchen.de/konferenzen/Jass05/courses/2/Valgaerts/Valgaerts_paper.pdf`, 2005.

[93] Luiz Velho and Jonas de Miranda Gomes. Space filling curve dither with adaptive clustering. In *Proc. of SIBGRAPI*, pages 1–9, 1992.

[94] Luiz Velho and Jonas Gomes. Stochastic screening dithering with adaptive clustering. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 273–276, New York, NY, USA, 1995. ACM.

[95] Luiz Velho and Jonas de Miranda Gomes. Digital halftoning with space filling curves. *SIGGRAPH Comput. Graph.*, 25(4):81–90, Jul 1991.

[96] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511–I–518, 2001.

[97] Richard S. Wallace, Ping-Wen Ong, Benjamin B. Bederson, and Eric L. Schwartz. Space variant image processing. *International Journal of Computer Vision*, 13(1):71–90, 1994.

[98] Brian A. Wandell. *Foundations of Vision*. Sinauer Associates, Incorporated, USA, Sunderland, MA, 1995.

[99] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era: An in-depth survey, 2020.

[100] Yixiu Wang, Bin Wang, Xiaofeng Wu, and Liming Zhang. Scanpath estimation based on foveated image saliency. *Cognitive Processing*, 18(1):87–95, Feb 2017.

[101] Welch. A technique for high-performance data compression. *Computer*, 17(6):8–19, 1984.

[102] Wei Wen, Ondřej Kajínek, Siamak Khatibi, and Goce Chadzitaskos. A common assessment space for different sensor structures. *Sensors*, 19(3), 2019.

[103] Stewart W. Wilson. On the retino-cortical mapping. *International Journal of Man-Machine Studies*, 18(4):361 – 389, 1983.

[104] Ian H. Witten and R.M. Neal. Using peano curves for bilevel display of continuous-tone images. *Computer Graphics and Applications, IEEE*, 2(3):47–52, May 1982.

[105] Yongkang Wong, Shaokang Chen, Sandra Mau, Conrad Sanderson, and Brian C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 81–88. IEEE, June 2011.

[106] Shivanthan A. C. Yohanandan, Adrian G. Dyer, Dacheng Tao, and Andy Song. Saliency preservation in low-resolution grayscale images. *CoRR*, abs/1712.02048, 2017.

[107] Yuefeng Zhang. Space-filling curve ordered dither. *Computers & Graphics*, 22(4):559 – 563, 1998.

[108] Yuefeng Zhang and Robert E. Webber. Space diffusion: An improved parallel halftoning technique using space-filling curves. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 305–312, New York, NY, USA, 1993. ACM.

[109] Qi Zhao and Christof Koch. Learning saliency-based visual attention: A review. *Signal Processing*, 93(6):1401 – 1407, 2013. Special issue on Machine Learning in Intelligent Image Processing.

[110] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

Appendix A

Algorithm for Generating ARIMs

In this appendix, we present the algorithmic steps for the generation of an ARIM, as described in the context of our first contribution (Chapter 3). The input parameters are:

- **num_u**: number of uniform points in the foveal region;
- **num_nu**: number of non-uniform points in the periphery;
- **fov_reg**: coordinate of the top-left and bottom-right points delimiting the squared foveal region;
- **fov_res**: the resolution of the foveal region;
- **asp_r**: the aspect ratio of the image;
- **imp_f**: the implicit function;
- **max_i**: the maximum number of iterations for generating the ARIM.

The process starts with Algorithm 1. It first creates an initial uniform 2-d distribution of points comprising foveal and peripheral regions. Then, a Delaunay triangulation is calculated to find the neighborhood relationship between those points. Next, the optimization takes place during several iterations. Each iteration calculates the velocities of all points to their neighbors (as we show in Algorithm 2), and the total energy of the system. After that, the iteration also updates (i) the arrangement of points by applying the computed velocities (forces), and (ii) the neighbors of all points by computing another Delaunay triangulation. The iteration process ends when it reaches the maximum number of iterations required.

Algorithm 1 Creates an ARIM (a list of 2-d points).

```
function createARIM(num_u, num_nu, fov_reg, fov_res, asp_r, imp_f, max_it)
    pts ← initialUniformDistribution(num_u, num_nu, fov_reg, asp_r)    ▷ creates an initial uniform distribution for the foveal and peripheral regions
    neighs ← {}    ▷ initializes a list of neighbors for each initial point
    delaunayTri(pts, neighs)    ▷ computes the Delaunay triangulation of the initial points, and stores the neighbors of each point in the list
    vels ← {}    ▷ initializes a list of velocities for each point-neighbor relationship
    it ← 0    ▷ initializes the iteration counter
    while (it ≤ max_it) do
        calculateVelocities(pts, neighs, imp_f, vels)    ▷ calculates the velocities for each point-neighbor relationship and stores the values in the correspondent list
        energy ← addVelocities(vels)    ▷ adds up all the previously calculated velocities
        updatePoints(pts, energy)    ▷ updates the coordinates of all points
        delaunayTri(pts, neighs)    ▷ updates the Delaunay triangulation
        it ← it + 1    ▷ increments the iteration counter
    end while
    return pts
end function
```

Algorithm 2 Calculates the velocities of points when creating an ARIM.

```
procedure calculateVelocities(pts, neighs, imp_f, vels)
    for each p ∈ pts do
        i_p ← pts.indexOf(p)    ▷ the position of p in the points array
        for each n ∈ neighs[i_p] do
            i_n ← pts.indexOf(n)    ▷ the position of n in the neighbors list
            delta ← euclideanNorm(p − n) ∗ (eval(imp_f, p) + eval(imp_f, n))    ▷ computes the partial delta value of the energy optimization by calculating the euclidean norm and evaluating the implicit function
            vels[i_p][i_n] ← vels[i_p][i_n] + delta    ▷ adds the delta to the current velocity
        end for
    end for
end procedure
```

Appendix B

Log-polar Images for Comparison

In this appendix, we illustrate a log-polar mapping procedure applied over sample images from the ChokePoint Dataset [105]. In Figure B.1, it is possible to see the visual differences between the cortical images (smaller images in the log-polar domain), and the retinal (inverse log-polar) images, which are the cortical images mapped back to the cartesian domain.

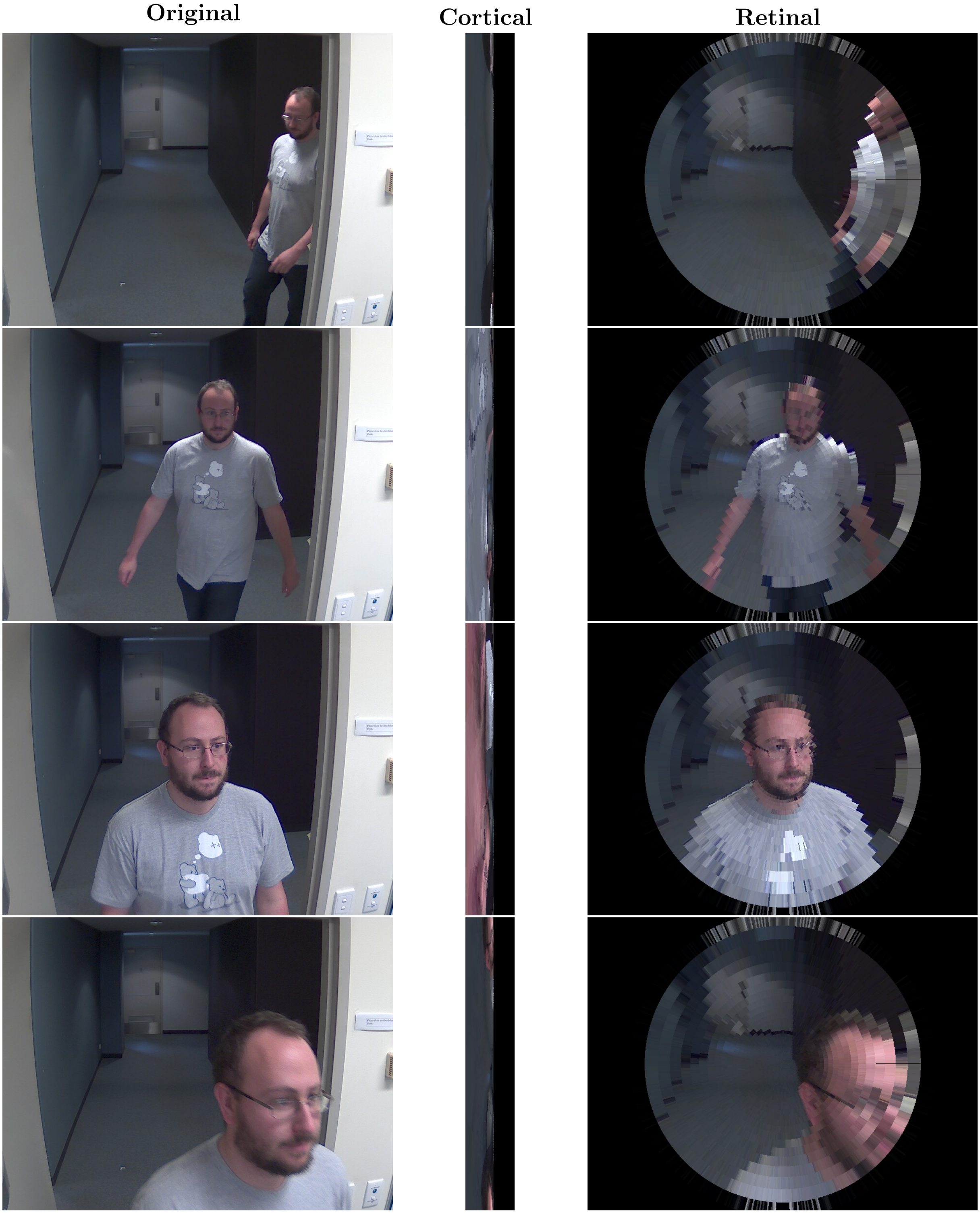


Figure B.1: Examples of images reconstructed with a log-polar mapping. Left-column images are the original ones. Middle-column images are the cortical versions after a log-polar mapping procedure. Right-column images are the retinal images after an inverse log-polar mapping procedure has been performed on their cortical versions. The original images come from the ChokePoint Dataset [105].

Appendix C

Motion Analysis Algorithm

In this appendix, we present the optical-flow-based motion analysis algorithm developed in the context of our first contribution (Chapter 3). The input parameters are:

- **pts_pos**: fixed position of sampled points provided by the model;
- **nu_bstream**: the optical flow for points belonging to the non-uniformly sampled region;
- **min_mag**: magnitude threshold;
- **min_perc**: threshold on the percentage of pixels that exceed the magnitude threshold.

Although this approach might not be very efficient in some circumstances, it has proven itself adequate for our initial tests. Taking the decision based on multiple frames, using new sets for the top or bottom pixels, and possibly finding the best magnitude/percentage thresholds according to the requirements of the application may potentially improve the algorithm.

Algorithm 3 Detects motion towards the image center.

```
procedure MOTIONDET($pts\_pos, nu\_bstream, min\_mag, min\_perc$)
    $n\_points \leftarrow pts\_pos.size()$                                  ▷ total number of points
    $count\_left \leftarrow 0, \quad count\_motion\_left \leftarrow 0$
    $count\_right \leftarrow 0, \quad count\_motion\_right \leftarrow 0$
    $sum\_angle\_left \leftarrow 0, \quad sum\_angle\_right \leftarrow 0$
    $i \leftarrow 0$

    while ($i \leq n\_points$) do
        $i \leftarrow i + 1$
        if $pts\_pos[i] \in$ NON-UNIFORM REGION then
            $dx \leftarrow nu\_bstream[i].x - pts\_pos[i].x$                ▷ $x$ offset
            $dy \leftarrow nu\_bstream[i].y - pts\_pos[i].y$                ▷ $y$ offset
            $angle \leftarrow arctan(dy, dx)$                        ▷ vector angle in radians
            $mag \leftarrow \sqrt{dx^2 + dy^2}$                                ▷ vector magnitude
            if ($pts\_pos[i]$ is a LEFT POINT) then
                $count\_left \leftarrow count\_left + 1$
                if ($mag \geq min\_mag$) then
                    $count\_motion\_left \leftarrow count\_motion\_left + 1$
                    $sum\_angle\_left \leftarrow sum\_angle\_left + angle$
                end if
            else if ($pts\_pos[i]$ is a RIGHT POINT ) then
                $count\_right \leftarrow count\_right + 1$
                if ($mag \geq min\_mag$) then
                    $count\_motion\_right \leftarrow count\_motion\_right + 1$
                    $sum\_angle\_right \leftarrow sum\_angle\_right + angle$
                end if
            end if
        end if
    end while

    $perc\_left \leftarrow \frac{count\_motion\_left}{count\_left}$
    $perc\_right \leftarrow \frac{count\_motion\_right}{count\_right}$
    $mean\_left \leftarrow \frac{sum\_angle\_left}{count\_motion\_left}$
    $mean\_right \leftarrow \frac{sum\_angle\_right}{count\_motion\_right}$

    if ($perc\_left \geq min\_perc$ AND $mean\_left \in [-45^o, 45^o]$) then
        return TRUE                         ▷ motion to the center (from the left side)
    else if ($perc\_right \geq min\_perc$ AND $mean\_right \in [135^o, 225^o]$) then
        return TRUE                        ▷ motion to the center (from the right side)
    else
        return FALSE                                         ▷ no motion to the center
    end if
end procedure
```

Appendix D

Technical Information and Parameters

In this appendix, we show technical information, such as the methods and parameters, used in the implementation of the CV application developed in the context of our first contribution (Chapter 3). Table D.1 displays these information.

Table D.1: Technical information regarding the implemented CV application. Only non-default parameter values are shown.

	Theoretical Reference	**Library**	**Method**	**Input Parameters**

Face Detection	Viola-Jones [96]	OpenCV 3.0.0	detectMultiScale	scaleFactor = 1.1
				minNeighbors = 3

Face Recognition	DNN model [39] + 1-NN	Dlib 19.16 [49]	get_face_chip_details	size = 150
				padding = 0.25

Optical Flow	Lukas-Kanade [56]	OpenCV 3.0.0	cvCalcOpticalFlowPyrLK	winSize = (31, 31)
				maxLevel = 3
				criteria.maxCount = 20;
				criteria.epsilon = 0.03
				minEigThreshold = 0.001

Appendix E

Algorithm for Generating Foveal Models

In this appendix, we detail the algorithm for sampling a 2-d space according to the relevance of regions. Here, we call the map that encodes the relevance of regions as a saliency map (from Chapter 5). However, the Gaussian KDE term (from Chapter 4) could be used instead without prejudice to the presented ideas, because both terms refer to the same concept.

Our sampling approach is closely related to quadtrees and a Hilbert curve. Quadtrees have been used to generate models that describe the geometry of 2-d objects, approximating their shapes. It consists of recursively breaking a 2-d space in smaller squares to match the properties of the object in question and the desired level of detail. A convenience of the approximation via quadtrees is that it can also be realized by means of constructing a Hilbert curve that fills a 2-d space. Both strategies yield the same sampling, with the difference that the curve-based offers us an improved ordering for visiting the sampled points (i.e., the final "trees" diverge).

Algorithm 4 presents our sampling procedure. The process relies on fitting a Hilbert curve in the image's domain by taking into account the relevance of the regions encoded in the saliency map. In summary, we must provide the width and height of a 2-d area in which the sampling will occur, and its mid position on the cartesian space. We are also required to inform the saliency map to guide the sampling, as well as the "mode" (the elementary Hilbert pattern configuration that ensures the continuity of the curve). Next, we proceed by recursively dividing the 2-d area in four subsquares around the current square's midpoint. The division, however, is conditioned on the Euclidean distance of the current midpoint to its two closest neighbors (in the curve) being both greater than the value of the saliency map at the midpoint. This is a gradual refinement that continues as long as the distances between sampled points are greater than one pixel.

Algorithm 4 Samples a 2-D region using a Hilbert curve.

```
procedure HILBERTSAMPLING(width, height, p_m, s_map, mode)
    width ← width/2
    height ← height/2
    p_1 ← (p_m.X − width/2, p_m.Y − height/2)            ▷ p_m is the current midpoint
    p_2 ← (p_m.X − width/2, p_m.Y + height/2)
    p_3 ← (p_m.X + width/2, p_m.Y + height/2)
    p_4 ← (p_m.X + width/2, p_m.Y − height/2)
    list_pts ← {p_1, p_2, p_3, p_4}                      ▷ puts the points in a list
    reorder_list_pts(list_pts, mode)                     ▷ reorders the list based the mode
    next_modes ← getNextModes(mode)                      ▷ returns the next four modes
    final_pts ← {}                                       ▷ list to store the final sampled points
    for each p_i ∈ list_pts do
        d_1 ← euclideanDistance(p_i, getFirstNeighbor(p_i))
        d_2 ← euclideanDistance(p_i, getSecondNeighbor(p_i))
        if d_1 ≥ 1 AND d_2 ≥ 1 then
            if evaluate(s_map, p_i) ≤ d_1 AND eval(s_map, p_i) ≤ d_2 then
                fin_pts ← fin_pts+hilbertSampling(width, height, p_i, s_map, next_modes)
            else
                fin_pts ← fin_pts + {p_i}
            end if
        else
            return {}
        end if
    end for
    return fin_pts
end procedure
```