

Université de Montréal

**Evaluating perceptual maps of asymmetries for
gait symmetry quantification and pathology
detection**

par

Antoine Moevus

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Decembre, 2014

© Antoine Moevus, 2014.

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé:

**Evaluating perceptual maps of asymmetries for
gait symmetry quantification and pathology
detection**

présenté par:

Antoine Moevus

a été évalué par un jury composé des personnes suivantes:

Derek Nowrouzezahrai
président-rapporteur

Max Mignotte & Jean Meunier
directeur de recherche

Roland Memisevic
membre du jury

RÉSUMÉ

Le mouvement de la marche est un processus essentiel de l'activité humaine et aussi le résultat de nombreuses interactions collaboratives entre les systèmes neurologiques, articulaires et musculo-squelettiques fonctionnant ensemble efficacement. Ceci explique pourquoi une analyse de la marche est aujourd'hui de plus en plus utilisée pour le diagnostic (et aussi la prévention) de différents types de maladies (neurologiques, musculaires, orthopédique, etc.). Ce rapport présente une nouvelle méthode pour visualiser rapidement les différentes parties du corps humain liées à une possible asymétrie (temporellement invariante par translation) existant dans la démarche d'un patient pour une possible utilisation clinique quotidienne. L'objectif est de fournir une méthode à la fois facile et peu dispendieuse permettant la mesure et l'affichage visuel, d'une manière intuitive et perceptive, des différentes parties asymétriques d'une démarche. La méthode proposée repose sur l'utilisation d'un capteur de profondeur peu dispendieux (la Kinect) qui est très bien adaptée pour un diagnostic rapide effectué dans de petites salles médicales car ce capteur est d'une part facile à installer et ne nécessitant aucun marqueur. L'algorithme que nous allons présenter est basé sur le fait que la marche saine possède des propriétés de symétrie (relativement à une invariance temporelle) dans le plan coronal.

Mots clefs : Analyse de la symétrie de la marche, Kinect, trouble locomoteur, positionnement multidimensionnel (MDS), réduction de dimensionnalité non linéaire, carte de couleur perceptuelle, invariance par décalage temporel.

ABSTRACT

The gait movement is an essential process of the human activity and also the result of coordinated effort between the neurological, articular and musculoskeletal systems. This motivates why gait analysis is important and also increasingly used nowadays for the (possible early) diagnosis of many different types (neurological, muscular, orthopedic, etc.) of diseases. This paper introduces a novel method to quickly visualize the different parts of the body related to an asymmetric movement in the human gait of a patient for daily clinical. The goal is to provide a cheap and easy-to-use method to measure the gait asymmetry and display results in a perceptually relevant manner. This method relies on an affordable consumer depth sensor, the Kinect. The Kinect was chosen because this device is amenable for use in small, confined area, like a living room. Also, since it is marker-less, it provides a fast non-invasive diagnostic. The algorithm we are going to introduce relies on the fact that a healthy walk has (temporally shift-invariant) symmetry properties in the coronal plane.

Keywords : Gait asymmetry analysis, Kinect, loco-motor disorders, multidimensional scaling (MDS), nonlinear dimensionality reduction, perceptual color map, temporal shift-invariance.

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
TABLE DES MATIÈRES	v
LISTE DES FIGURES	vii
LISTE DES TABLEAUX	viii
LISTE DES ALGORITHMES	ix
LISTE DES SIGLES	x
CHAPITRE 1 : INTRODUCTION AND STATE OF THE ART . .	1
1.1 Introduction	1
1.2 Previous work	2
1.3 References	6
CHAPITRE 2 : EVALUATING PERCEPTUAL MAPS OF ASYMMETRIES FOR GAIT SYMMETRY QUANTIFICATION AND PATHOLOGY DETECTION	9
2.1 Introduction	10
2.2 Previous Work	10
2.3 Data Description	12
2.4 Method	12
2.4.1 Setup Phase	12
2.4.2 Silhouette Segmentation	14
2.4.3 Dimensionality Reduction	15
2.4.4 Color Space Conversion	17
2.5 Experimental Results	17

2.6	Conclusion	19
2.7	Acknowledgments	20
2.8	References	21
	ANNEXE A : Implementation Notes	23
	Importing Data	23
	Clipping Step	23
	ANNEXE B	26
	Correlation Metric	26
	ANNEXE C	27
	Asymmetry Map Stochastic Local Refinement	27
	ANNEXE D	28
	Embc'14 Poster	28
	ANNEXE E	29
	Local Symmetry Detection in Natural Images using a Particle Filtering Approach	29

LISTE DES FIGURES

1.1	<i>Example of two depth signals for a gait cycle of a patient.</i>	5
2.1	<i>Setup and pre-processing steps. (a) Original depth map. (b) After clipping. (c) After treadmill removal.</i>	15
2.2	<i>Asymmetry map for subject #15, one of the best result of the dataset. ASI is 0.03082 for normal gait, 0.04740 for left asymmetry, and 0.05266 for right asymmetry.</i>	18
2.3	<i>Asymmetry map for subject #05, one of the best result of the dataset. ASI is 0.04164 for normal gait, 0.05858 for left LLD, and 0.06824 for right LLD. With the right LLD (case C), the asymmetry of arm swing is clearly noticeable.</i>	19
2.4	<i>Asymmetry map for subject #09, the worst result of the dataset. The corresponding ASI are 0.05672 for normal gait, 0.05166 for left LLD, and 0.04625 for right LLD. The patient had naturally a strong arm swing but a sole on the left foot seems to help rectifying it.</i>	20
2.5	<i>(a) An input image where non-patient pixels where naively clipped to zero. The high contrast makes details harder to see. (b) An input image where non-patient pixels where clipped to the mean value of patient's pixels for the whole sequence. More details are visible. (c) The distribution of the pixel values of the image for the naive clipping (semi-log scale). (d) The distribution of the pixel values of the image for the smart clipping (semi-log scale). It is important to notice that the whole point of using a default clipping value is to make the distribution of pixel value of the images unimodal and continuous. This will ensure a well contrasted map that is visually significant for human eye.</i>	25

LISTE DES TABLEAUX

2.1	<i>Average and SD (σ) of the ASI for the 17 patients</i>	17
-----	--	----

LIST OF ALGORITHMS

1	Local exploration with Metropolis	27
---	---	----

LISTE DES SIGLES

2D	Two dimensions
3D	Three dimensions
3DGA	3D gait analysis
ASI	Asymmetry Index
DEI	Depth Energy Image
GA	Gait Analysis
IR	Infrared Red
LLD	Length Leg Discrepancy
MDS	Multi-Dimensional Scaling
TOF	Time-Of-Flight

CHAPITRE 1

INTRODUCTION AND STATE OF THE ART

1.1 Introduction

Scientists and medical communities have been interested in the analysis of gait movement for a long time, because, as mentioned in [1,2,3] symmetrical gait is expected in the case of healthy people, whereas asymmetrical gait is a common feature of subjects with loco-motor disorders.

Abnormal or atypical gait can be caused by different factors, either orthopedic (hip injuries [4], bone malformations, etc.), muscular, or neurological (Parkinson's disease, stroke [5], etc.). Consequently, different parts of the body can be involved or affected, which make gait analysis a complex procedure but also a reliable and accurate indicator for early detection (and progression) of a wide range of pathologies. It thus also makes a 3D gait analysis (3DGA) procedure a powerful early clinical diagnostic tool [6] that is reliable and non-invasive, and which has been used successfully until now for a first screening test, detection and tracking of disease progression, joint deficiencies, pre-surgery planning, as well as recovery from post-operative surgery or accident (rehabilitation). It is also worth mentioning that a gait analysis-based diagnostic tool also allows to reduce the costs and amount of surgery per patient [7]. Also, a more appropriate medical prescription can be made by performing a gait analysis before treating a patient. This leads to a better recovery for the patients [6].

But nowadays, with the aging population, clinical diagnostics have to be cheaper, faster and more convenient for clinical [8,9,10] (or home [11]) usage while remaining accurate. However, analyzing a gait video sequence is often difficult, requires time, and subtle anomalies can be omitted by the human eye. Also, videos are not easy to annotate, store and share.

In this work, we design both a reliable and accurate imaging system that is also cheap and easy to set up for daily clinical usage. This diagnostic tool is relying on the fact that the gait of healthy people is symmetrical in the coronal (front) plane (with half a period phase shift) and that asymmetrical gait may be a good indicator of pathologies (and its progression over different periods of time) [1,2,3,5]. More precisely, the goal of our proposed GA-based diagnostic tool is to evaluate a perceptual color map of asymmetries from a video acquired by a depth sensor (Kinect) of a patient walking on a treadmill. The recording plane was the coronal plan in order to exploit the temporally shift-invariant properties of the movement. A perceptual color map of asymmetries is the compression of a patient’s video mapped into a color image in such manner that asymmetries of the body parts related to an asymmetric movement in the human gait cycle may be clearly visible and immediately quantifiable.

1.2 Previous work

Current 3DGA techniques can be divided in two categories : with or without markers.

Among the state-of-the-art marker-based approach, the Vicon motion-tracking and capture system [12] offers millimeter resolution of 3D spatial displacements. Due to its accuracy, it is often used as ground-truth for validation in medical application. On the other hand, the high cost of this system inhibits its widespread usage for routine clinical practices. Basically, optical motion capture system consists in tracking infrared (IR) reflective markers with multiple IR cameras [13]. In this context, some recent work [8] has compared a cheap and simple motion capture system to a high-end system such as Vicon, since the accuracy of higher-end systems is often more than required. They found that the cheaper motion capture hardware provides reliable results for clinical usage, but that the software given by the system was a limitation on the accuracy of the acquisition and the analysis of the results.

That was because the software was not designed for medical application. Optical Motion capture is efficient, but requires a lot of space, time, and expertise to be installed and used. For instance, placing the markers on the patient is prone to localization errors and requires someone who understands both the patient's anatomy and the acquisition system. Also the patient might have to wear a special suit and change outfits, which is constraining both for the patient and for the physician.

Therefore, marker-less systems are a promising alternative for clinical environments and are often regarded as easy-to-set-up, easy-to-use, and non-invasive. They are either based on either stereo-vision [14], structured light [15], or time-of-flight (TOF) [16] technologies. As a stereo-vision application, [17] used two camcorders to extract 3D information of the patients and to measure the gait parameters. Although low-cost, the setup and calibration procedure of the system remains complex and only the lower parts of the body are measured. Also, stereo vision-based systems will not function properly if the patient's outfits lack of texture. However, the Kinect sensor is based on structured light technology which makes it robust to surfaces with no textures. The Kinect remains also compact and affordable. Recent researches have been conducted to test whether the Kinect is suitable for clinical usage or not. The Kinect has two output modes : depth map or skeleton modes. The former consists of an image sequence where the value of each pixel is proportional to the inverse of the depth, whereas the latter is a set of 3D points and edges that represents 20 joints of the human body.

Clark et. al. [18] used the skeleton mode to measure spatial-temporal gait variability (such as the stride duration, speed, etc.) and compared it with data acquired by the high-end Vicon MX system. They found encouraging results but important landmarks were missing on the foot and therefore they conclude that gait parameters have to be chosen carefully. In addition, the Kinect camera was placed facing the patient, without a treadmill. Therefore the system was based on the analysis of only one gait cycle, because the intrinsic working range of the Kinect depth sensor

is between 800mm and 4000mm. This somehow compromised the accuracy and the reliability of their system.

[11] also used the skeleton mode to perform a 3DGA. They made people wear wireless sensors (gyroscopes and pressure sensors) at movement points and walk back and forth along a straight path for approximately 7 minutes. They found that the Kinect was capable of providing accurate and robust results, but only a few gait parameters were tested and further research is under investigation. Finally, it is worth mentioning that none of the methods, using the Kinect skeleton mode, provide a visual feedback of the gait of the patient.

In [19], the authors compared the Kinect with depth map output mode versus a Vicon system. They place 2 Kinects in a different alignment with the subject (facing and on the side) and measured key gait parameters, such as stride duration and length, and speed. They found excellent results with an average difference of less than 5% for both Kinect camera setups. They also found that using the depth map data allows to reduce drastically the computation time for background removal.

In [10], the authors proposed to use a treadmill and a Kinect depth sensor to quantify the gait asymmetry with a low-cost gait analysis system. More precisely, the authors compute an index for quantifying possible asymmetries between the two legs by first dividing each gait cycle in two sub-cycles (relative to the left and right step sub-cycle), and by comparing these two sub-cycles, in term of an asymmetry index (proportional to the difference of depth, over a gait cycle, between the two legs) after a rough spatial and temporal registration procedure. Although the system is able to distinguish whether the patient has a symmetric walk or not, no visualization or information on the location of the asymmetries is provided, unlike our method.

In [9], the Kinect camera is placed at the back of a treadmill and is used to record a video sequence of the patient walk. The authors then simply compute the mean of the obtained depth image sequence (over a gait cycle or a longer period) in order to compress the gait image sequence into one image which is finally called

a depth energy image (DEI). Their results were conclusive since they were able to distinguish both visually and quantitatively asymmetries (a symmetric walk generating a DEI exhibiting a symmetric silhouette, in terms of mean depth and conversely). Nevertheless, this latter strategy is inherently inaccurate since taking the average (mean) depth over a gait cycle does not allow to detect all asymmetric body movements ; indeed, movement variation of some parts of the body can clearly be different and asymmetric while keeping the same mean (in term of mean depth).

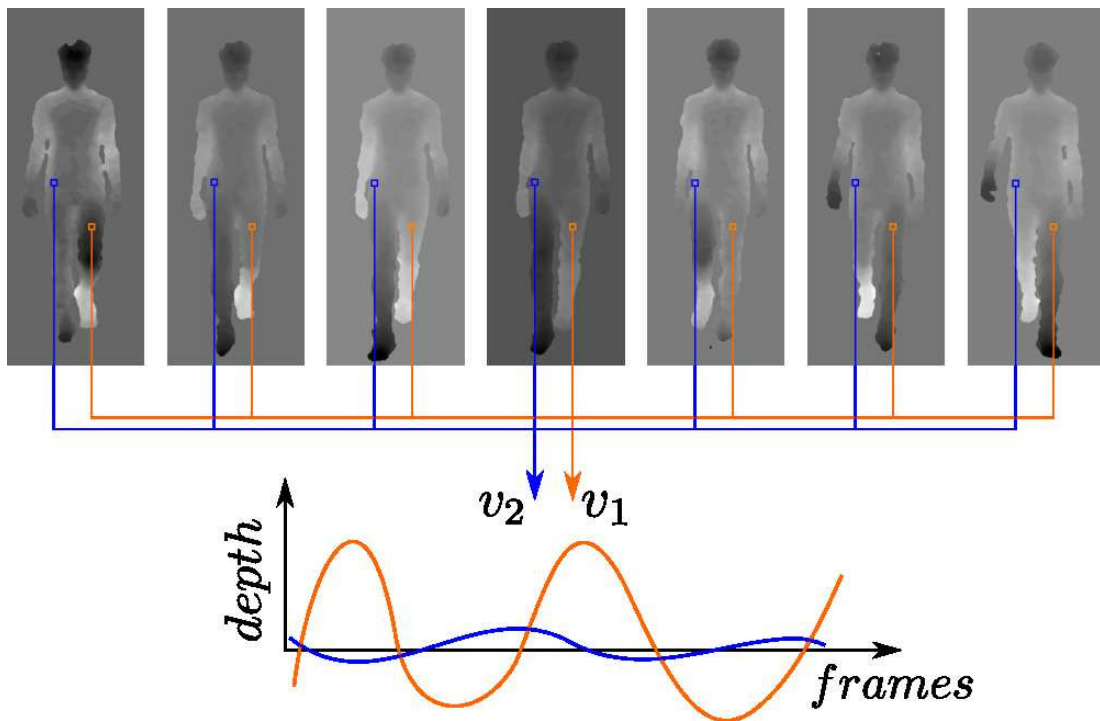


FIG. 1.1: *Example of two depth signals for a gait cycle of a patient.*

In our work, the depth image sequence of the gait, containing a number of gait cycles (wherein each pixel of the video corresponds to a depth signal as a function of time, as shown in Fig. 1.1) is reduced to three dimensions with a Multi-Dimensional Scaling (MDS) mapping [20] using a temporally shift invariant Euclidean distance. This allows us to quickly display the gait image cube into an informative color image (with red, green and blue channels) allowing to visualize the asymmetric body parts of the gait cycle of a patient with a color difference, in a perceptual

color space, which is linearly related to the asymmetry magnitude.

This appealing property is crucial in our case and only respected in the case of the MDS reduction strategy (contrary to all the other existing dimension reduction techniques) which attempts to find an embedding from the initial feature vectors in the high dimensional space such that pairwise distances are preserved in a low dimensional space (as main criterion) *via* the minimization of the so-called *stress* function¹.

1.3 References

[1] J. R. Engsberg, K. G. Tedford, J. A. Harder, and J. P. Mills, “Timing changes for stance, swing, and double support in a recent below-knee-amputee child.” *Pediatric Exercise Science*, vol. 2, no. 3, 1990.

[2] J. Loizeau, P. Allard, M. Duhaime, and B. Landjerit, “Bilateral gait patterns in subjects fitted with a total hip prosthesis,” *Archives of physical medicine and rehabilitation*, vol. 76, no. 6, pp. 552–557, 1995.

[3] J. Hamill, B. Bates, and K. Knutzen, “Ground reaction force symmetry during walking and running,” *Research Quarterly for Exercise and Sport*, vol. 55, no. 3, pp. 289–293, 1984.

[4] H. Miki, N. Sugano, K. Hagio, T. Nishii, H. Kawakami, A. Kakimoto, N. Nakamura, and H. Yoshikawa, “Recovery of walking speed and symmetrical movement of the pelvis and lower extremity joints after unilateral tha,” *Journal of Biomecha-*

¹Let us recall that MDS is formulated as an optimization problem, where the solution is found as a minimizer of the following cost function also called the *stress* function :

$$\hat{\mathbf{u}} = \arg \min_{\mathbf{u}} \sum_{s,t,s \neq t} \left(\beta_{s,t} - \|\mathbf{u}_s - \mathbf{u}_t\| \right)^2 \quad (1.1)$$

where the summation $\sum_{s,t,s \neq t}$ is over all the pairs of sites. $\beta_{s,t}$ and $\|\mathbf{u}_s - \mathbf{u}_t\|$ denote, respectively, the Euclidean distance associated between the two initial feature vectors s and t in the full dimensional space and between the two feature vectors \mathbf{u}_s and \mathbf{u}_t in the low dimensional (or reduced) space.

nics, vol. 37, no. 4, p. 443–455, Apr 2004.

[5] L. D. Alexander, S. E. Black, K. K. Patterson, F. Gao, C. J. Danells, and W. E. McIlroy, “Association between gait asymmetry and brain lesion location in stroke patients,” *Stroke*, vol. 40, no. 2, pp. 537–544, 2009.

[6] T. A. Wren, G. E. Gorton III, S. Ounpuu, and C. A. Tucker, “Efficacy of clinical gait analysis : a systematic review,” *Gait & posture*, vol. 34, no. 2, pp. 149–153, 2011.

[7] T. A. Wren, M. M. Kalisvaart, C. E. Ghatan, S. A. Rethlefsen, R. Hara, M. Sheng, L. S. Chan, and R. M. Kay, “Effects of preoperative gait analysis on costs and amount of surgery,” *Journal of Pediatric Orthopedics*, vol. 29, no. 6, pp. 558–563, 2009.

[8] B. Carse, B. Meadows, R. Bowers, and P. Rowe, “Affordable clinical gait analysis : An assessment of the marker tracking accuracy of a new low-cost optical 3d motion analysis system,” *Physiotherapy*, vol. 99, no. 4, pp. 347 – 351, 2013.

[9] C. Rougier, E. Auvinet, J. Meunier, M. Mignotte, and J. A. de Guise, “Depth energy image for gait symmetry quantification,” in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 5136–5139.

[10] E. Auvinet, F. Multon, and J. Meunier, “Lower limb movement asymmetry measurement with a depth camera,” in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, Aug 2012, pp. 6793–6796.

[11] M. Gabel, R. Gilad-Bachrach, E. Renshaw, and A. Schuster, “Full body gait analysis with kinect,” in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 2012, pp. 1964–1967.

[12] “Motion capture systems from vicon.” [HTTP://WWW.VICON.COM/](http://www.vicon.com/)

[13] F. Potdevin, C. Gillet, F. Barbier, Y. Coello, and P. Moretto, “The study

of asymmetry in able-bodied gait with the concept of propulsion and brake,” in *9th Symposium on 3D Analysis of Human Movement, Valenciennes, France*, 2006.

[14] N. Lazaros, G. C. Sirakoulis, and A. Gasteratos, “Review of stereo vision algorithms : from software to hardware,” *International Journal of Optomechatronics*, vol. 2, no. 4, pp. 435–462, 2008.

[15] J. Salvi, J. Pages, and J. Batlle, “Pattern codification strategies in structured light systems,” *Pattern Recognition*, vol. 37, no. 4, pp. 827–849, 2004.

[16] M. Hansard, S. Lee, O. Choi, and R. Horaud, *Time-of-flight cameras*. Springer, 2013.

[17] A. Leu, D. Ristic-Durrant, and A. Graser, “A robust markerless vision-based human gait analysis system,” in *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on*, May 2011, pp. 415–420.

[18] R. A. Clark, K. J. Bower, B. F. Mentiplay, K. Paterson, and Y.-H. Pua, “Concurrent validity of the Microsoft Kinect for assessment of spatiotemporal gait variables,” *Journal of biomechanics*, vol. 46, no. 15, pp. 2722–2725, 2013.

[19] E. E. Stone and M. Skubic, “Evaluation of an inexpensive depth camera for passive in-home fall risk assessment.” in *PervasiveHealth*, 2011, pp. 71–77.

[20] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC Press, 2000.

CHAPITRE 2

EVALUATING PERCEPTUAL MAPS OF ASYMMETRIES FOR GAIT SYMMETRY QUANTIFICATION AND PATHOLOGY DETECTION

Cet article a été publié comme l'indique la référence bibliographique et est présenté ici, tel quel, dans la langue originale de publication :

©2014 Antoine Moevus, de :

A. Moevus, M. Mignotte, J. de Guise, J. Meunier,

“Evaluating perceptual maps of asymmetries for gait symmetry quantification and pathology detection”,

36th International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'2014, Chicago, Illinois, USA, Aout 2014.

De plus, la version longue de cet article (non présentée ici) est actuellement en cours de soumission à un journal scientifique pour publication.

Abstract

The gait movement is a complex and essential process of the human activity. Yet, many types of diseases (neurological, muscular, orthopedic, etc.) can be diagnosed from the gait analysis. This paper introduces a novel method to quickly visualize the different body parts related to an (temporally shift-invariant) asymmetric movement in the human gait of a patient for daily clinical usage. The goal is to provide a cheap and easy-to-use method that measures the gait asymmetry and display results in a perceptual and intuitive way. This method relies on an affordable consumer depth sensor, the Kinect, which is very suitable for small room and fast diagnostic, since it is easy to setup and marker-less.

2.1 Introduction

Scientists and medical communities have been interested in the analysis of gait movement for a long time, because, as mentioned in [1,2,3] symmetrical gait is expected in the case of healthy people, whereas asymmetrical gait is a common feature of subjects with loco-motor disorders.

Abnormal or atypical gait can be caused by different factors, either orthopedic (hip injuries [4], bone malformations, etc.), muscular, or neurological (Parkinson’s disease, stroke, etc.). Consequently, different parts of the body can be involved or affected, which make gait analysis a complex procedure. Nevertheless, gait analysis remains a powerful early clinical diagnostic tool that is easy to perform and non-invasive, and has been used until now for detection and tracking of disease progression, joint deficiencies, pre-surgery planning, as well as recovery from post-operative surgery or accident.

But nowadays, with the aging population, clinical diagnostics have to be cheaper, faster and more convenient, while remaining accurate. However, analyzing a gait video sequence is often difficult, requires time, and subtle anomalies can be omitted. Also, videos are not easy to annotate, store and share. The goal of the proposed diagnostic tool is to evaluate a perceptual color map of asymmetries from a video, acquired by a depth sensor (Kinect), and recorded in the coronal (front) plane of a patient walking on a treadmill. A perceptual color map of asymmetries is the compression of a patient’s video mapped into a color image in such manner that asymmetries of the body parts related to an (temporally shift-invariant) asymmetric movement in the human gait cycle may be clearly visible and immediately quantifiable.

2.2 Previous Work

One popular method for gait analysis is motion capture [11] which consists in tracking infrared (IR) reflective markers with multiple IR cameras. This method is

effective but requires a lot of space to be set up along with lots of expertise and time and effort to be installed and used.

In [6], the authors proposed to also use a treadmill and a Kinect depth sensor to quantify the gait asymmetry with a low-cost gait analysis system. More precisely, the authors compute an index for quantifying possible asymmetries between the two legs by first dividing each gait cycle in two sub-cycles (relative to the left and right step sub-cycle), and by comparing these two sub-cycles, in term of an asymmetry index (proportional to the difference of depth, over a gait cycle, between the two legs) after a rough spatial and temporal registration procedure. Although, the system is able to distinguish whether the patient has a symmetric walk or not, no visualization or information on the location of the asymmetries is provided, unlike our method.

In [10], the Kinect camera is placed at the back of a treadmill and is used to record a video sequence of the patient walk. The authors then simply compute the mean of the obtained depth image sequence (over a gait cycle or a longer period) in order to compress the gait image sequence into one image which is finally called a depth energy image (DEI). Their results were conclusive since they were able to distinguish both visually and quantitatively asymmetries (a symmetric walk generating a DEI exhibiting a symmetric silhouette, in terms of mean depth and conversely). Nevertheless, this latter strategy is inherently inaccurate since taking the average (mean) depth over a gait cycle does not allow to detect all asymmetric body movements ; indeed, movement variation of some parts of the body can clearly be different and asymmetric while keeping the same mean (in term of mean depth).

In our work, the depth image sequence of the gait, containing a number of gait cycles (and wherein each pixel of the video corresponds to a depth signal, as a function of time) is reduced to three dimensions with a Multi-Dimensional Scaling (MDS) mapping [12] using a temporally shift invariant Euclidean distance. This allows us to quickly display the gait image cube into an informative color image (with red, green and blue channels) allowing to visualize the asymmetric body parts

of the gait cycle of a patient with a color difference, in a perceptual color space, which is even greater than the asymmetry is large in magnitude.

2.3 Data Description

The dataset consists of multiple sequences of people walking on a treadmill, facing a cheap depth sensor (Kinect). The Kinect sensor outputs 30 depth maps per second (30 *fps*), with a resolution of 640 per 480 pixels. The dataset contains 51 sequences acquired from 17 subjects walking with or without simulated length leg discrepancy (LLD). Every patient had to walk normally (group A), then with a 5 *cm* sole under the left foot (group B), then with the sole under the right foot (group C). Sequences are approximately 5 minutes long and contain around 180 gait cycles. For all sequences, the same relative position between the treadmill and the sensor is kept in order for the subject to be within the same area of images.

2.4 Method

In addition to a prerequisite setup step, the method can be divided into three steps : pre-processing, MDS-based dimensionality reduction and color space conversion.

2.4.1 Setup Phase

Since the scene took place in a non-cluttered room where the treadmill is in the same position relatively to the camera, a 3D bounding box around the subjects can be set. Hence, by retrieving 3D information, such as the position of the treadmill or the patient, we can convert this information back in the 2D image space and segment the patient’s silhouette directly from a depth map.

Therefore, the setup step is to, first, determine the 3D position of the treadmill and the patient by converting a depth map in a 3D point cloud. To do so, the

depth sensor is reasonably considered as a pinhole camera model with intrinsic parameters, K , (see [5,p.30]) defined as :

$$K = \begin{bmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 575.82 & 0 & 240 \\ 0 & 575.82 & 320 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

where f is the focal length in pixels and (c_u, c_v) is the image center in pixels (values given by the manufacturer). From a depth map, a pixel at position $(u, v)^T$ with depth value, d is projected in 3D space, $(X, Y, Z)^T$, using :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = dK^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = d \begin{pmatrix} 1/f * (u - c_u) \\ 1/f * (v - c_v) \\ 1 \end{pmatrix} \quad (2.2)$$

First, the positions of the points around the patient, approximated by a 3D bounding box, is first estimated. Second, the minimal and maximal depth, Z_{min} and Z_{max} , of the 8 points (of the bounding box) are then retrieved. Third, the 8 points were projected back in the 2D image space (see [5,p.30]) where the minimal and maximal vertical and horizontal 2D position value (u_{min} , u_{max} , v_{min} , and v_{max}) are finally estimated.

The necessity of working in 3D space is because of the spatial coherence of objects in the scene. For instance, in 3D space the treadmill is always beneath the subject whereas in an image it overlaps the patient, as shown in Fig. 2.1b. Once this step is done, it is no more necessary to project the depth maps in the 3D space as long as the camera and the treadmill stay at the same relative position. In our case, some small adjustments on the enclosing parameters, u_{min} , u_{max} , v_{min} , v_{max} , Z_{min} and Z_{max} , were needed to encompass all sequences.

2.4.2 Silhouette Segmentation

Now with the required information, the patient can be segmented in each frame of the original gait depth sequence (of N frames).

2.4.2.1 Background removal

Background removal is trivial since the subject is in the middle of the image in a non-cluttered room. Therefore, every pixel outside the range of the enclosing parameters around the subject are clipped to a default value.

2.4.2.2 Treadmill removal

After background removal, the only objects remaining in the image are the treadmill and the patient. Because the treadmill is below the patient, it can be removed by selecting pixel with coordinates superior to a threshold (Y axis is going from top to bottom) and clip them. An equation in the 2D-space can be derived from Eq. (2.2) in order to work directly on the image :

$$Y < T_y \quad (2.3)$$

$$\frac{d}{f}(v - c_v) < T_y \quad (2.4)$$

$$v < \frac{fT_y}{d} + c_v, \quad \text{since } d > 0 \text{ and } f > 0 \quad (2.5)$$

where T_y is the threshold value measured during the setup phase and d is the depth.

Fig. 2.1 visually shows the different steps of the setup and pre-processing stage.

2.4.2.3 Filtering

Finally, the whole sequence is filtered with a 3D (5×5) median filter to remove some aberrations on the contours or on top of the treadmill.

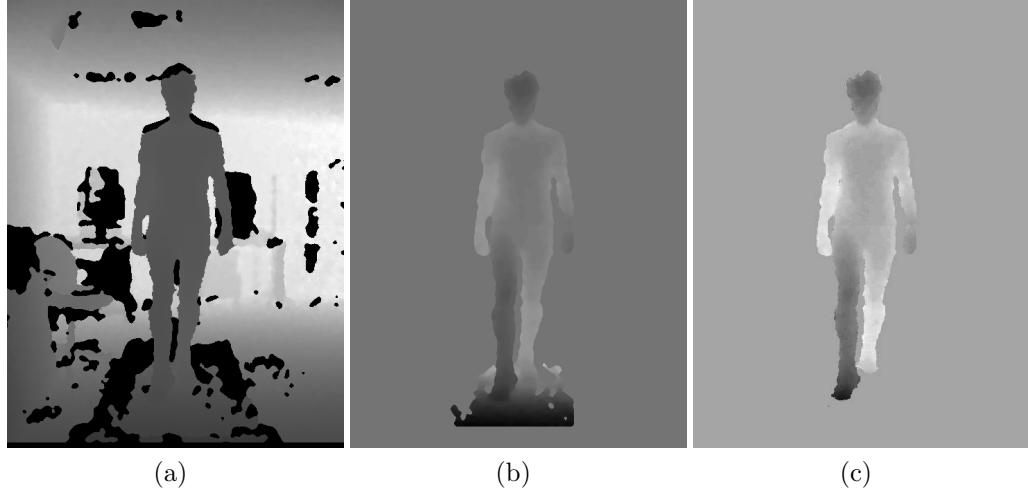


FIG. 2.1: *Setup and pre-processing steps. (a) Original depth map. (b) After clipping. (c) After treadmill removal.*

2.4.3 Dimensionality Reduction

The MDS dimensionality reduction-based mapping technique [12] aims at visualizing the (temporally shift-invariant) asymmetric body parts of the gait cycle of a patient with a color difference, in a perceptual color space, which is linearly related to the asymmetry magnitude⁰. This mapping is achieved by considering each pair of pixels (i.e., pair of N -dimensional depth signals) in the original gait video sequence and by quantifying their (temporally shift-invariant) degree of asymmetry with a temporally shift-invariant pairwise Euclidean distance d_{tsi} between each pair $(\mathbf{s}_1(t), \mathbf{s}_2(t))$ of depth signals :

$$d_{\text{tsi}}(\mathbf{s}_1, \mathbf{s}_2) = \min_{\forall \tau} \left\{ \left(\sum_{t=0}^N \mathbf{s}_1(t + \tau) - \mathbf{s}_2(t) \right)^2 \right\} \quad (2.6)$$

where the maximal value of τ corresponds approximately to the number of frames in a gait cycle.

⁰let us recall that this appealing property is crucial in our case and only respected in the case of the MDS reduction strategy which preserves the between-object (or depth signal) distances as well as possible.

In addition, four points are important to consider in this step :

1. First, it is important to understand that the use of the shift-invariant pairwise Euclidean distance is crucial in this MDS-based mapping step. Indeed, two pixels in the gait video cube, i.e., two depth signals (as a function of the time) with a perfect similar movement but in phase opposition (phase difference of half a gait cycle) like the legs and arms will have to be considered as symmetric with the same (perceptual) color in the final asymmetry map.
2. Second, in order to finally provide a final perceptual color asymmetry visualization map, the MDS mapping is achieved in a perceptual color space, namely the classical CIE 1976 L^*, a^*, b^* (LAB) color space which is approximately perceptually uniform. In this color space, a color difference shall (perceptually) appear twice as large for a measured (temporally shift-invariant) asymmetry value which is twice bigger.
3. Third, as already said, MDS is a dimensionality reduction technique that maps objects lying in an original high N dimensional space to a lower dimensional space (3 in our application), but does so in an attempt that the between-object distances are preserved as well as possible. The original MDS algorithm is not appropriate in our application and more generally for all large scale applications because it requires an entire $N \times N$ distance matrix to be stored in memory (with a $O(N^3)$ complexity). Instead, the FastMap [7] is a fast alternative to the MDS that we have adopted herein with a linear complexity $O(pN)$ (with $p = 3$, the dimensionality of the target space)¹.
4. The above-mentioned FastMap-based mapping method, which exploits an algebraic procedure [8], has the main advantage of being very fast (for large scale applications) but slightly less accurate than a (gradient descent or local stochastic search-based) optimization procedure [8]. For this reason, we decide

¹In FastMap, the axis of target space are then constructed dimension by dimension. More precisely, it implicitly assumes that the objects are points in a p -dimensional Euclidean space and selects a sequence of $p \leq N$ orthogonal axes defined by distant pairs of points (called pivots) and computes the projection of the points onto the orthogonal axes.

TAB. 2.1: Average and SD (σ) of the ASI for the 17 patients

	Normal gait	Left LLD	Right LLD
Average	0.045374	0.053549*	0.055936*
σ	0.008080	0.006274	0.009451

*Paired difference t -test is statistically significant ($p \ll 0.01$)

to refine the estimated asymmetry map given by the FastMap as being the initial starting solution of a stochastic local search (using a local exploration around the current solution and the Metropolis criteria) as proposed in [8].

2.4.4 Color Space Conversion

It is important to mention that, at this stage, we are not assured that the LAB color values of the 3D asymmetry map are not saturated in the RGB space. In order to fix this problem, we use a simple linear stretching of the L , A , B color values such as $L \in [0 : 100]$, and A, B have a maximal amplitude of 100 with a zero mean in order to ensure that a very small number of pixels are outside the RGB color space [8].

2.5 Experimental Results

This section presents the asymmetry maps obtained for patients with or without (simulated) pathologies. Sequences of 300 frames have been used (longer sequences did not yield better results). This corresponds approximately to 5 or 8 gait cycles depending on the subject’s speed and step size. On average for all images, the correlation score [8] for the compression of 300 frames to 3 channels is $93.5\% \pm 2\%$ which shows us that the FastMap-based MDS procedure is able to preserve, due to its non-linearity property, a large quantity of information of the original image sequence.

Asymmetries can be easily detected visually, as shown by Fig. 2.2, 2.3, and

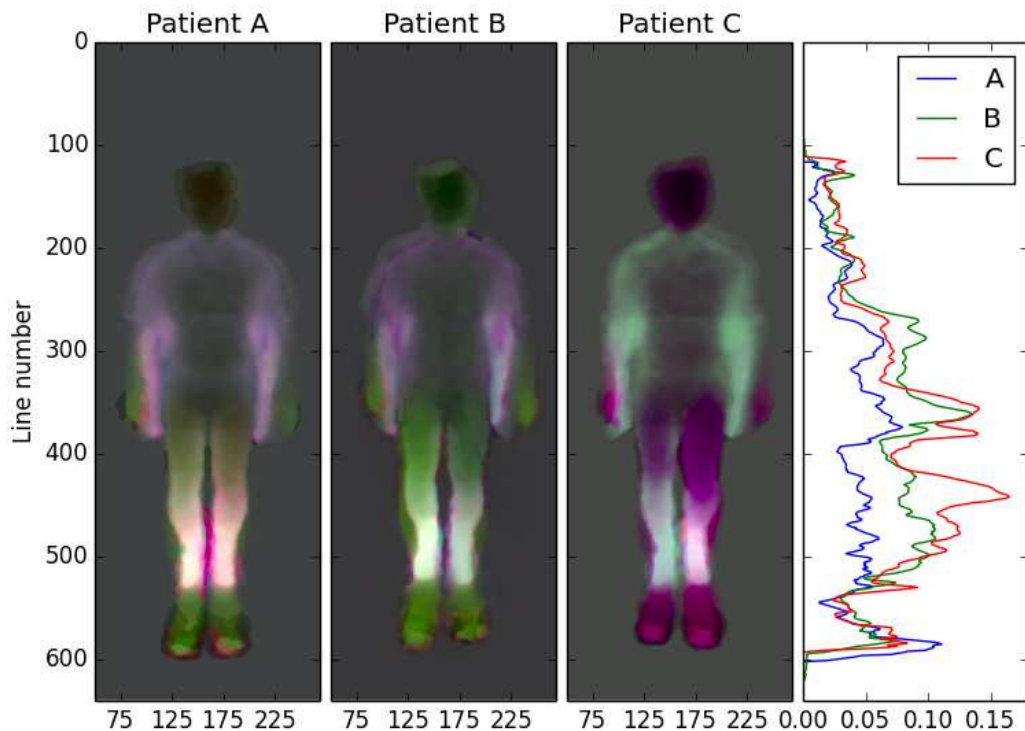


FIG. 2.2: *Asymmetry map for subject #15, one of the best result of the dataset. ASI is 0.03082 for normal gait, 0.04740 for left asymmetry, and 0.05266 for right asymmetry.*

2.4, but also quantitatively. To do so, the mean of biggest mirrored differences² is computed for each line of the map, which yields a vertical curve³. Then, by taking the mean value of the curve, as a global asymmetry index (ASI) is computed.

Table 2.1 shows the average and standard deviation of the ASI for the three groups of patient. The statistical difference for the paired t test were highly significant for both left and right legs LLD group ($p \ll 0.01$). This demonstrates

²For a line \mathbf{k} of width w , the set of mirrored differences is : $\{\|p_{i,\mathbf{k}} - p_{w-i,\mathbf{k}}\|_2^2, \forall i \in [0, w/2]\}$ where $p_{i,j}$ is a pixel at position (i, j) .

³To estimate this curve, images are centered a first time based on the position of the neck, then more accurately by seeking the minimum area of the curve around the axis of symmetry.

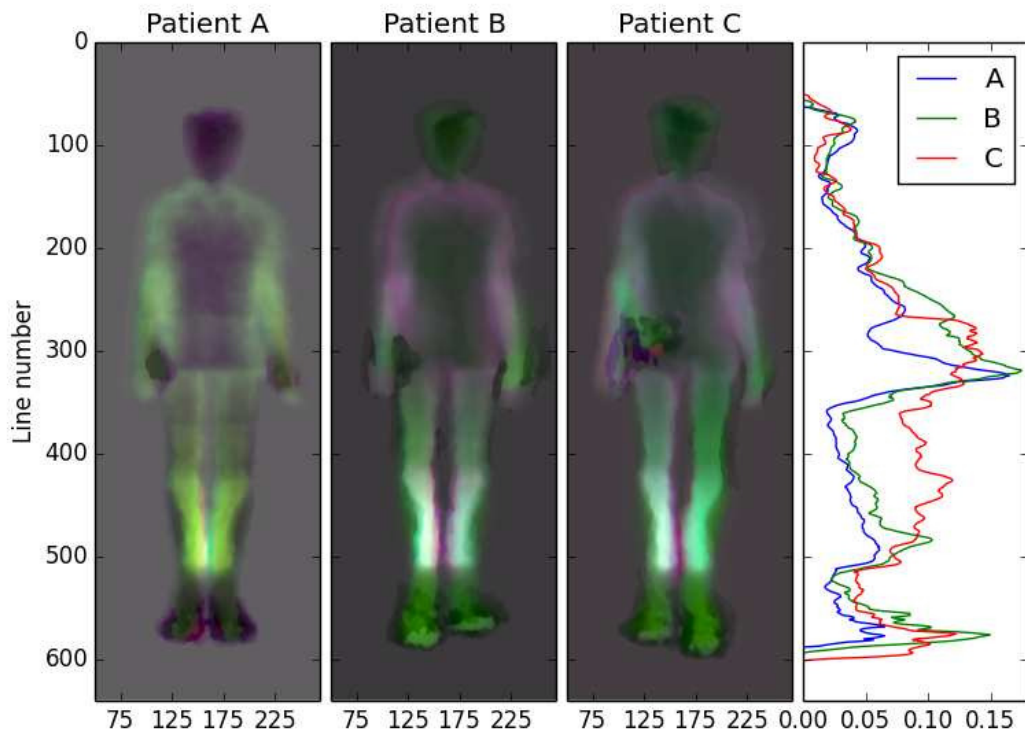


FIG. 2.3: *Asymmetry map for subject #05, one of the best result of the dataset. ASI is 0.04164 for normal gait, 0.05858 for left LLD, and 0.06824 for right LLD. With the right LLD (case C), the asymmetry of arm swing is clearly noticeable.*

that this method can efficiently detect gait symmetry. In practice, three patients had a higher ASI for their normal gait than with the sole (Fig. 2.4). By looking at their videos, the authors have noticed that those patients already had a visible gait asymmetry (one arm swinging more than the other, tilted shoulders, etc.).

2.6 Conclusion

In this paper, we have presented a new gait analysis system, based on a depth sensor, which estimates a perceptual color map providing a quick overview of existing asymmetry existing in the gait cycle of a patient and an index (ASI), that was proved statistically significant ($p \ll 0.01$). While being cheap, markerless, non-

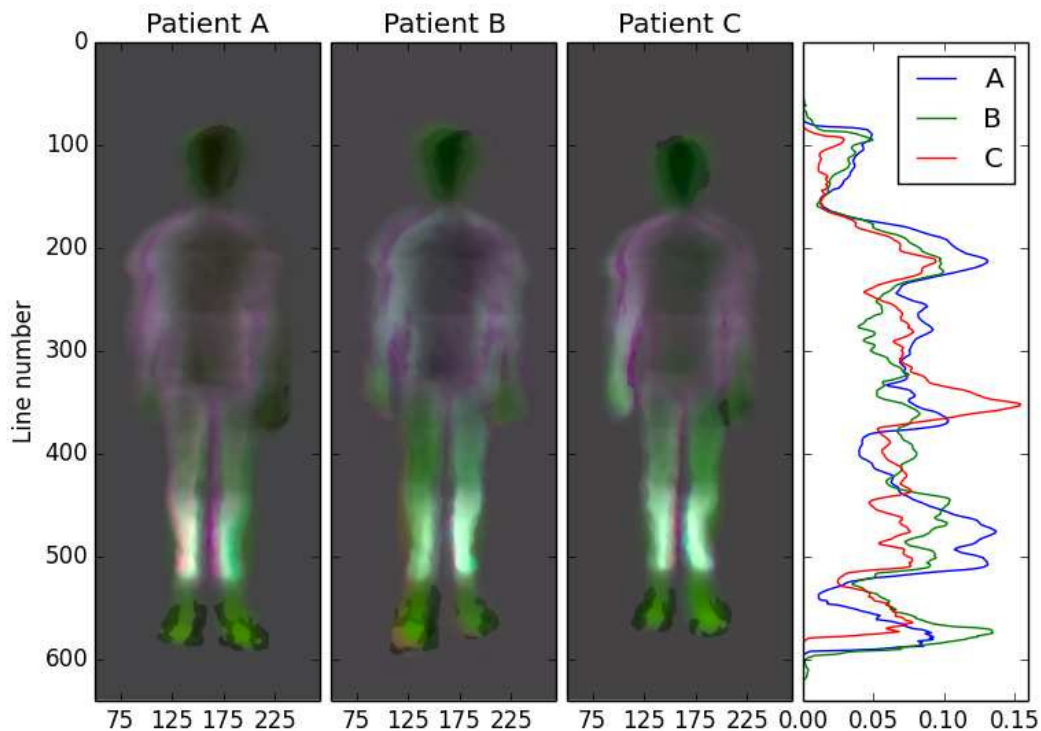


FIG. 2.4: *Asymmetry map for subject #09, the worst result of the dataset. The corresponding ASI are 0.05672 for normal gait, 0.05166 for left LLD, and 0.04625 for right LLD. The patient had naturally a strong arm swing but a sole on the left foot seems to help rectifying it.*

invasive, easy to set up and suitable for small room and fast diagnostic, this new gait analysis system offers a readable and flexible tool for clinicians to analyze gait characteristics which can be easily exploited for disease progression, recovery cues from post-operative surgery or might be used for other pathologies where gait asymmetry might be a symptom.

2.7 Acknowledgments

Authors would like to thank E. Auvinet for his help with the dataset [6].

2.8 References

- [1] J. R. Engsborg, K. G. Tedford, J. A. Harder, and J. P. Mills, “Timing changes for stance, swing, and double support in a recent below-knee-amputee child.” *Pediatric Exercise Science*, vol. 2, no. 3, 1990.
- [2] J. Loizeau, P. Allard, M. Duhaime, and B. Landjerit, “Bilateral gait patterns in subjects fitted with a total hip prosthesis,” *Archives of physical medicine and rehabilitation*, vol. 76, no. 6, pp. 552–557, 1995.
- [3] J. Hamill, B. Bates, and K. Knutzen, “Ground reaction force symmetry during walking and running,” *Research Quarterly for Exercise and Sport*, vol. 55, no. 3, pp. 289–293, 1984.
- [4] H. Miki, N. Sugano, K. Hagio, T. Nishii, H. Kawakami, A. Kakimoto, N. Nakamura, and H. Yoshikawa, “Recovery of walking speed and symmetrical movement of the pelvis and lower extremity joints after unilateral tha,” *Journal of Biomechanics*, vol. 37, no. 4, p. 443-455, Apr 2004.
- [5] J. Ponce and D. Forsyth, *Computer vision : a modern approach*. United States of America : Prentice Hall, 2003.
- [6] E. Auvinet, F. Multon, and J. Meunier, “Lower limb movement asymmetry measurement with a depth camera,” in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 2012, pp. 6793–6796.
- [7] C. Faloutsos and K.-I. Lin, *FastMap : A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*. Proceedings of ACM SIGMOD, vol. 24, no. 2, pp. 163-174, 1995.
- [8] M. Mignotte, “A bicriteria optimization approach based dimensionality reduction model for the color display of hyperspectral images,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 50, no. 2, pp. 501–513, 2012.
- [9] A. K. Jain, *Fundamentals of digital image processing*. Prentice-Hall, Inc.,

1989.

[10] C. Rougier, E. Auvinet, J. Meunier, M. Mignotte, and J. A. de Guise, “Depth energy image for gait symmetry quantification,” in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 5136–5139.

[11] F. Potdevin, C. Gillet, F. Barbier, Y. Coello, and P. Moretto, “The study of asymmetry in able-bodied gait with the concept of propulsion and brake,” in *9th Symposium on 3D Analysis of Human Movement, Valenciennes, France*, 2006.

[12] F.T. Cox and M.A.A. Cox, *Multidimensional Scaling*, Chapman and Hall CRC, 2000

ANNEXE A

Importing Data

Each sequence is saved in a binary format and compressed with gzip (NB : ending with the “.gz” extension). Each pixel is meant to be an 16-bit unsigned integer (uint16) and represents the depth in millimeter. So as to recover an image from binary to uint16, three steps are required : the first one is to get to the starting byte of the image, the second one is to read and convert the pixels of the image, and the final one is to store each pixel value. To recover the i^{th} image, the starting point is evaluated as follows :

$$\text{image}_i = 8 + (24 + 640 \times 480) \times i$$

Each pixel is stored as a 16-bit little-endian (which can be represented as for 4 digits in hexadecimal base). Therefore, 4 bytes are read sequentially. We give them the notation $p_{j_{read}}^i$. And then, they are converted into a decimal base according to the following scheme :

$$\begin{aligned} p_{j_{read}}^i &= [A123]_{\text{little-endian}}^{\text{HEX}} \rightarrow 10 \times 16^1 + 1 \times 16^0 + 2 \times 16^3 + 3 \times 16^2 \\ &= [9121]^{\text{DECIMAL}} = p_j^i \end{aligned}$$

Where \rightarrow design the conversion of the read data to the decimal base. At this point, the pixel p_j^i is stored as a uint16 in a table. This table represents the i^{th} image and once the i^{th} image is fully recovered, it is saved in a tree-dimensional array at the i^{th} position.

Clipping Step

This step takes place when the silhouette of the patient is recovered.

It is important to recall that the core of the method, which is MDS-based,

aims at mapping the original high N -dimensional depth signals to a lower 3D space, but also attempt to preserve the between-depth signal distances as well as possible. Preserving the between-depth signal distances will result in a visually more significant map. The map will be more contrasted and the color used are going to be well distributed. Therefore, the clipping value of the non-patient pixels have to be set carefully.

In other words, the MDS algorithm will work well if no big artificial pairwise distances are created, because those distances will induce a squeezing effect on the other informative distances. For instance, if the relative value of the background differs a lot from the patient's pixel values, the asymmetries between the right and left legs might not be distinguishable. Therefore the default value has to be a central value, such as the depth of the chest. In this work, the default clipping value is computed by taking the mean of all the depth value belonging to the patient in the whole sequence.

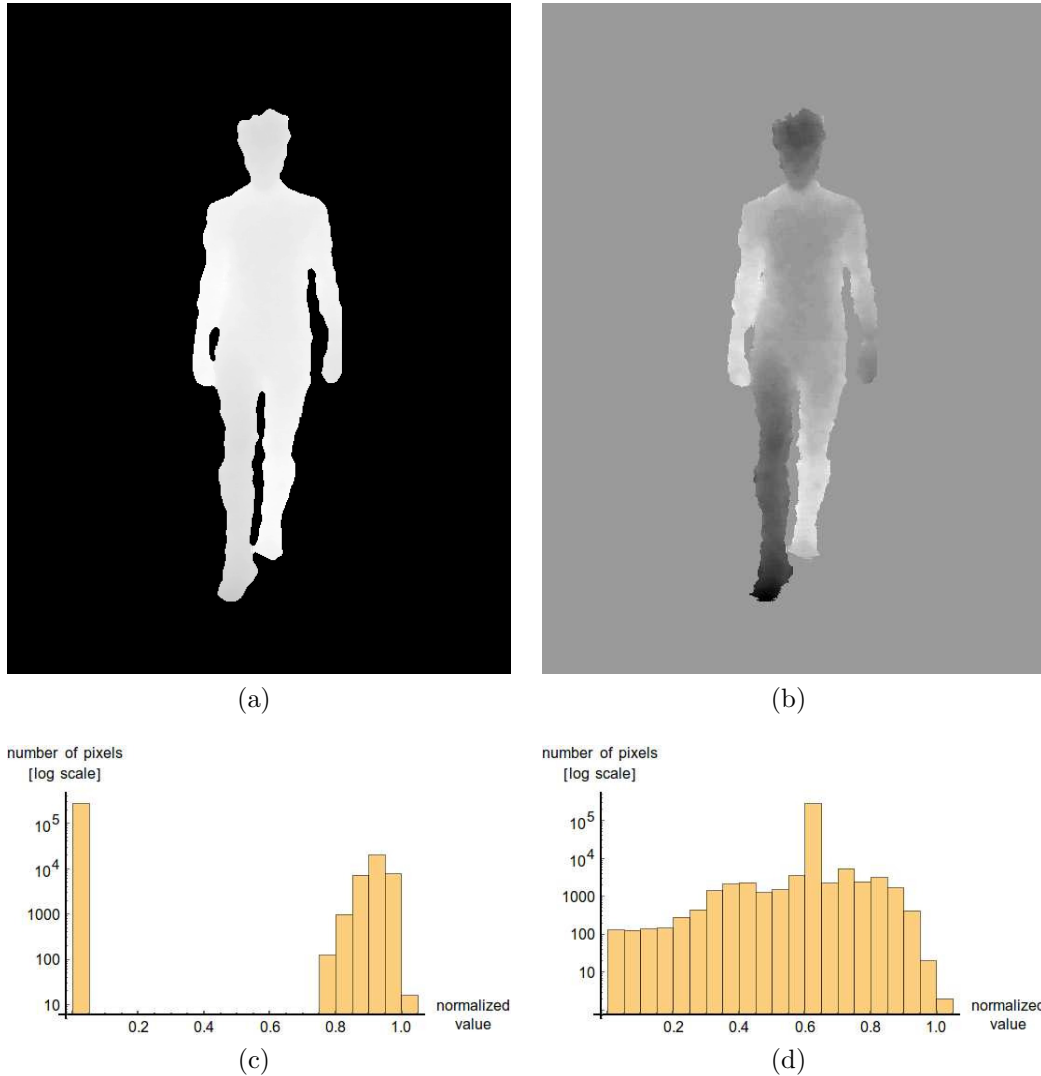


FIG. 2.5: (a) An input image where non-patient pixels were naively clipped to zero. The high contrast makes details harder to see. (b) An input image where non-patient pixels were clipped to the mean value of patient’s pixels for the whole sequence. More details are visible. (c) The distribution of the pixel values of the image for the naive clipping (semi-log scale). (d) The distribution of the pixel values of the image for the smart clipping (semi-log scale). It is important to notice that the whole point of using a default clipping value is to make the distribution of pixel value of the images unimodal and continuous. This will ensure a well contrasted map that is visually significant for human eye.

ANNEXE B

Correlation Metric

We can evaluate the efficiency of the FastMap technique in its ability to reduce the dimensionality reduction of a gait sequence by computing the correlation metric [2] which is simply the correlation of the temporally shift-invariant Euclidean distance between each pairwise vectors in the high (N -)dimensional space (let X be this vector) and their corresponding (pairwise) Euclidean distances in the low (3D) dimensional space (let Y be this vector). The correlation ρ can be estimated by the following equation :

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{X^t Y / |X| - \bar{X} \bar{Y}}{\sigma_X \sigma_Y} \quad (2.7)$$

where X^t , $|X|$, \bar{X} and σ_X respectively represent the transpose, cardinality, mean, and standard deviation of X . This correlation factor (Pearson) will specifically quantify the degree of linear dependence between the variables X and Y and quantify how the FastMap technique is able to give a low dimensional mapping in which each object is placed such that the between-object distances (in the original high dimensional space) are preserved as well as possible [1]. A perfect correlation $\rho = 1$ indicates a perfect relationship between original data and reduced data and a correlation of $\rho = 0$ indicates a total loss of information.

References

[1] T. F. Cox and M. A.. A. Cox, *Multidimensional scaling*. Chapman and Hall CRC Press, 2000.

[2] P. Jacobson and M.R. Gupta, *Design goals and solutions for display of hyperspectral images*, IEEE Trans. Geosci. Remote Sens., 2005

ANNEXE C

Asymmetry Map Stochastic Local Refinement

As indicated in Section 2.4.3, the FastMap-based mapping method (which exploits an algebraic procedure) has the main advantage of being very fast for large scale applications but slightly less accurate than an optimization procedure. For this reason, we decide to refine the estimated asymmetry map given by the FastMap as being the initial starting solution of a stochastic local search using a local exploration around the current solution and the Metropolis criteria with the following algorithm :

Algorithm 1
Local exploration with Metropolis

E	Energy function to be minimized
T_l	Temperature at Iteration step l
a	Cooling schedule parameter
r	Radius of exploration, real $\in]0, 1]$
T_0, T_f	Initial and final temperature
L_{\max}^S	Maximal number of iterations

1. Initialization : $a \leftarrow \left(\frac{T_f}{T_0}\right)^{\frac{1}{L_{\max}^S}}$

2. Local Exploration

while $l < L_{\max}^S$ **do**

for each pixel and each color channel with value x_s at site s **do**


- Compute $\Delta_{\text{Energy}} = E(y_s) - E(x_s)$ with $y_s \in [x_s - r : x_s + r]$ and $y_s \in [0.0 : 1.0]$
- **If** ($\Delta_{\text{Energy}} < 0$) Replace x_s by y_s
- **Else** Replace x_s by y_s with probability $\triangleright \exp\left(-\frac{\Delta_{\text{Energy}}}{T_l}\right)$

$l \leftarrow l + 1$ and $T_l \leftarrow T_0 a^l$

Algorithm 1: Local exploration with Metropolis


ANNEXE D

Embc'14 Poster



Université
de Montréal

Evaluating perceptual maps of asymmetries for gait symmetry quantification and pathology detection



Antoine Moevus¹, Max Mignotte¹, J.A. de Guise², and Jean Meunier¹

¹ Department of Computer Science and Operations Research (DIRO), University of Montreal, Canada.
² Laboratoire de Recherche en Imagerie et Orthopédie (LIO), CRCHUM and École de Technologie Supérieure, Canada.

Abstract: Human gait is a complex and essential pattern of movements for mobility. Moreover, many types of diseases (neurological, muscular, orthopedic, etc.) can be diagnosed from gait analysis. This paper introduces a novel method to quickly visualize (a)symmetric motion of different body parts of a **patient walking on a treadmill** for daily clinical usage. The method relies on the analysis of **depth maps** provided by a low-cost easy-to-use depth sensor.

Method

Step 1: Converting a depth map to 3D points
Step 2: Human silhouette segmentation
 Step 3: Dimensionality reduction to a 3D color space

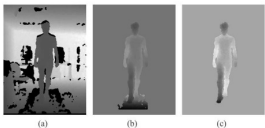
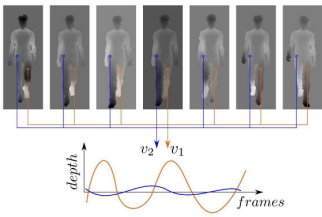


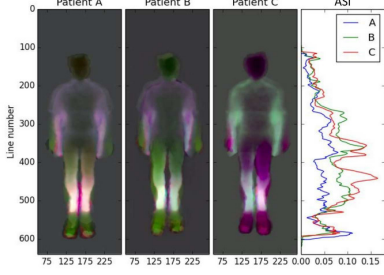
Fig. 1: Setup and pre-processing steps. (a) Original depth map. (b) After clipping. (c) After treadmill removal.

Step 3.1: For each pair of pixels, compute (a)symmetry with time-shift-invariant distance d_{tsi} between their N-dimensional depth signals $s_1(t)$ and $s_2(t)$:

$$d_{tsi}(s_1, s_2) = \min_{\tau} \left\{ \sum_{t=0}^N (s_1(t + \tau) - s_2(t))^2 \right\}$$


Step 3.2: MDS mapping from N-D to the 3-D perceptually uniform color space L*a*b*^{*}. MDS preserves distances.

➤ **Asymmetry index (ASI):** Mean difference between original L*a*b* map and its mirror image



Experimental Results

Dataset:
 Subjects walking on a treadmill, facing a Kinect

- Normal walk (group A)
- 5 cm sole under the left foot (group B)
- 5 cm sole under the right foot (group C)

Results:

- Asymmetries can be easily detected visually
- Statistically discriminative index (ASI)

TABLE I: Average and SD (σ) of the ASI for the 17 patients

	Normal gait	Left LLD	Right LLD
Average	0.045374	0.053549*	0.055936*
σ	0.008080	0.006274	0.009451

*Paired difference t test is statistically significant ($p << 0.01$)

Conclusion

- New gait analysis system with depth maps
- Cheap, markerless, non-invasive, easy to use, suitable for small room and fast diagnosis
- Color map of gait (a)symmetries + discriminative index (ASI)
- Applications: Assessment of pathologies (affecting gait) and pre- and post-operative (e.g. knee or hip) surgeries

36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society
 EMBC 2014, Chicago

ANNEXE E

Local Symmetry Detection in Natural Images using a Particle Filtering Approach

This research work was done in collaboration with Nicolas Widynski and the related paper was accepted and published by IEEE Transactions on Image Processing, 23(12) :5309-5321, December 2014.

Contrary to the previously described gait analysis system which estimates a perceptual color map providing a quick overview of existing asymmetry existing in the gait cycle of a patient and offering a readable and flexible tool for clinicians for pathology detection and disease progression, we propose now an original algorithm to detect smooth local symmetries and the contours of ribbon-like object in natural images ¹.

¹As second author, my own contribution to this work was to help in defining the local geometric model of sequential and well-formed ribbons used in our Bayesian detection procedure based on a particle filtering algorithm. To this end, this contribution included an analysis of previous works proposed in the symmetry and medial axis detection literature before making an overview of the definition of the different types of ribbons (see Fig. 1) along with their inherent properties (see Section II). Finally, I help in defining a simple generative but yet expressive model of ribbons, which can be sequentially detected and generated with a Bayesian framework (in which the particle filtering is expressed) and which is based on the Brooks definition of ribbons combined with constraints (that the generator makes equal angles with the contours) similar to the ones formulated in the Brady ribbons. This model ensures a reliable and well-posed detection solution of smooth local symmetries in a Bayesian framework. My contribution was also to help in defining a strategy demonstrating the ability of the approach to only detect class-specific ribbons using local prior geometric information and also to participate to the revisions and corrections of the manuscript.

Local Symmetry Detection in Natural Images using a Particle Filtering Approach

Nicolas Widynski, Antoine Moevus, and Max Mignotte

Abstract—In this work, we propose an algorithm to detect smooth local symmetries and contours of ribbon-like objects in natural images. The detection is formulated as a spatial tracking task using a particle filtering approach, extracting one part of a structure at a time. By using an adaptive local geometric model, the method can detect straight reflection symmetries in perfectly symmetrical objects as well as smooth local symmetries in curved elongated objects. Also, the proposed approach jointly estimates spine and contours, making it possible to generate back ribbon objects. Experiments for local symmetry detection have been conducted on a recent extension of the Berkeley Segmentation Datasets. We also show that it is possible to retrieve specific geometrical objects using intuitive prior structural information.

Index Terms—Particle filter, local symmetry detection, ribbon detection.

I. INTRODUCTION

IN human perception, symmetry is a key element for object recognition as it is one of the fundamental law of the Gestalt theory. Its importance is easy to underline as it is ubiquitous in nature: plants, animals, humans; as well as in man-made objects, buildings, and art [1], [2].

In computational science, reflection symmetry detection has received an increasing amount of attention in the last decade. In this paper, we focus on *local symmetries*, which locally approximate reflection symmetries. Two points form a local symmetry if the angles between their outward normal and the line joining the points are the same [3], [4]. This definition covers *global* as well as *purely local* symmetries. The former refers to a symmetry that holds a whole object, whereas the latter relies on a subset of an object. In other terms, although the definition is local, it does not mean that the detected symmetry cannot be global. While global reflection symmetries can be difficult to retrieve due to noise, occlusion and local deformations, purely local symmetries offer more flexibility and may be more suited for natural images.

A symmetry axis is a spatial characteristic of the shape. However, the symmetry axis does not encode the scale and contours of the shape. On the other hand, contours only can hardly describe regional properties (e.g. symmetry) and descriptions (e.g. “elongated and curve”) [3]. Ribbon-like objects, or simply *ribbons*, are a representation of 2D plane shapes. A ribbon can be defined by a smooth local symmetry curve, called *spine* (the black plain curve in Figure 1), and

a geometric figure, such as a segment or a disk, called the *generator* (the black dashed structure in Figure 1). Ribbon shapes benefit of a dual representation: a contour-based and a region-based representation.

In this article, we propose to detect both contour and symmetry curves of ribbon objects in natural images. Ribbons being strongly related to the notion of local symmetry, they are in fact common in natural images, as they can describe a large amount of objects [1]. They are also of interest for the detection of specific tubular structures, for example in medical image analysis (vessels, arteries, colon, ...) [5]–[9], and remote sensing (urban structures, road networks) [10], [11]. Ribbons being a basic element structure that represents well any local object part, their detection could also be employed to retrieve more sophisticated non-ribbon objects using a higher abstraction level detection framework: contours can be used for object segmentation [12], [13]; local symmetries can serve object recognition by providing candidates for structural-based graph matching techniques [14], [15]; and local contours and symmetries could be jointly exploited by a dedicated algorithm, as it has been done in [16] in the context of knowledge transfer between object classes, and in [17], [18] in the context of object detection using a hierarchical representation.

Our approach spatially tracks symmetries and contours using a local geometric model of ribbons. Local symmetries are assumed to form smooth 1D lines (spines). The use of an iterative algorithm for this task is particularly well-suited to extract smooth local symmetries and to preserve local connectivity properties. We propose to implement a sequential Monte Carlo method, namely a particle filter, to extract one piece of the structure at a time. The particle filter naturally embeds a prior and a spatial transition, which enable to geometrically control the structure of the shape to be extracted. The particle filter is also able to maintain several hypotheses during time, thus overcoming local ambiguities that may happen in occlusions and clutter, for example. The last component of the proposed recursive Bayesian approach is the likelihood function, which is adaptive to the image, in order to confer more importance on visually salient geometric patterns.

This paper is organized as follows. In Section II, we present a brief analysis of previous works proposed in the symmetry and medial axis detection literature. In Section III, we describe our model of ribbons. Applied sequentially, our model provides consistent and well-formed ribbons. We also define the ribbon prior and transition probability density functions. In Section IV, we detail the contour and object features. These features rely on the information of local gradient, oriented gradient, and textural color. Their purposes are twofold: to

Copyright (c) 2013 IEEE. Personal use of this material is permitted.

N. Widynski, Antoine Moevus, and Max Mignotte are with Department of Computer Science and Operations, University of Montreal, C.P. 6128, succ. Centre-Ville, Montreal (Quebec), Canada, H3C 3J7.

evaluate the accuracy of the localization of a contour pair candidate, and to ensure that the ribbon spine is inside an object. In Section V, we explain the ribbon detection algorithm. The detection framework originates from the recent work introduced by Widynski and Mignotte [19], [20], in which the authors proposed to detect contours using a particle filtering technique. We adapt this methodology to our purpose. In Section VI, we show the results obtained on the LS-BSDS300. We also illustrate examples of symmetrical object retrieval using a prior information on their structure. We finally conclude in Section VII.

II. RELATED WORK

We focus on symmetry and medial axes detection algorithms in the recent literature, before making an overview of the definition of a ribbon.

A. Symmetry Detection

Reflection, rotation, and translation symmetries have been the subject of numerous detection methods these recent years. Among the three, reflection symmetry is definitely the most common, as it may be the most ubiquitous in the real world. Reflection symmetry detection methods can be separated into two groups: the first starts from segmented data, and aims at coherently gathering pieces in order to form symmetries with nice geometric properties. The second does not require any pre-processing step, and thus aims at extracting local symmetry features directly from the image, possibly using a shape model. In this section, we mainly focus on un-segmented reflection symmetry detection methods. The reader can find extended bibliography on these subjects in [1], [21].

Literature on shape representation methods has been particularly prolific these recent years [22], [23]. Their scope of application is appealing: object detection, data compression, tracking, segmentation, ... Among these methods, Trinh and Kimia addressed the object detection problem by extracting medial axis of specific objects in natural images [24]. The shape is learned from a dataset and is modeled using a graph representation. Since they rely on a structural prior rather than a potentially changing appearance model, such methods are very promising in object detection. But although skeletons contain local symmetries [23], [25], these methods are not intended to extract them all since the detections are driven by a specific shape model.

Detection of symmetries without integrating a model has notably been studied in [21], [26], [27]. These papers aim at finding major symmetry axes from extracted feature points. Thus, these methods are more adapted to recover global reflection symmetries rather than multiple local symmetries from images [28]. In [29], Podolak *et al.* proposed a planar reflective symmetry transform to measure the degrees of symmetries of arbitrary 3D shapes. All the planes passing through the object bounding volume being evaluated, this representation creates an interesting bridge between local and global symmetries, yet it remains to be applied to the detection problem in natural images.

Recently, Tsogkas and Kokkinos proposed a soft local symmetry detector using a learning-based approach operating jointly on several scales [30]. The features are based on the well known gPb contour detection algorithm [12]. Although we compare our approach to theirs, the problems are not exactly the same. They focus on local symmetry detection, and not on ribbon detection, meaning that their approach cannot recover an object nor a scale information. The authors also proposed to use the Berkeley Segmentation Dataset (BSDS300) [31] to learn and evaluate their local symmetry detection algorithm. To create the symmetry ground truth dataset, they first manually selected elongated objects from the set of human-annotated segments available in the BSDS300. Then, they used a skeletonization algorithm to extract the medial structure of the objects. We further refer to this dataset as the LS-BSDS300, which stands for Local Symmetries in the Berkeley Segmentation Dataset 300. As in [30], we also compare our approach to the methods proposed by Lindeberg [32] and Levinshtein *et al.* [33]. The former defines a multi-scale ridge detector, hence yielding to a one-dimensional output of the structure. The latter approximates symmetrical regions using fitted ellipses from which it retrieves the major axes.

The problem of detecting symmetries and ribbons together has been previously addressed in the context of the detection of thin elongated structures such as roads, vessels or other tubular objects [5]–[8], [10], [34], but to our knowledge, not in natural images. The comparison with our approach is even more relevant in the case of [5]–[7], since the proposed methods employ a particle filtering technique to extract vessels and arteries in 3D computation tomography data. However, the aforementioned methods are often semi-automatic, thus making them designed for single extraction purposes, and/or are object-specific, which make their adaptation in natural images not trivial.

In this work, we stress that simultaneously retrieving the contours and their associated local symmetries into a single framework is an interesting challenge as it aims at representing objects by compact and consistent structures at different levels of abstraction. Our goal is also to propose a generic geometric model that could be easily exploited to extract specific ribbons from natural images. This prior could be based on many basic filtering criteria, such as the length of the symmetry, its minimal/maximal curvature, the object area, its perimeter, its thickness, ...

B. Ribbon

The definition of a local symmetry is related to the notion of ribbon. Past literature [1], [3], [35], [36] reports several definitions of local symmetries, each one being associated with a different type of ribbon [4], [37]. According to the definition proposed in [4], a ribbon is a plane shape generated by translating a geometric figure, the *generator*, along a plane curve, the *spine*. To ensure that the generated shape is a proper ribbon, the generation process should respect the following rules [37]:

- the size and the orientation of generators should vary smoothly along the spine;

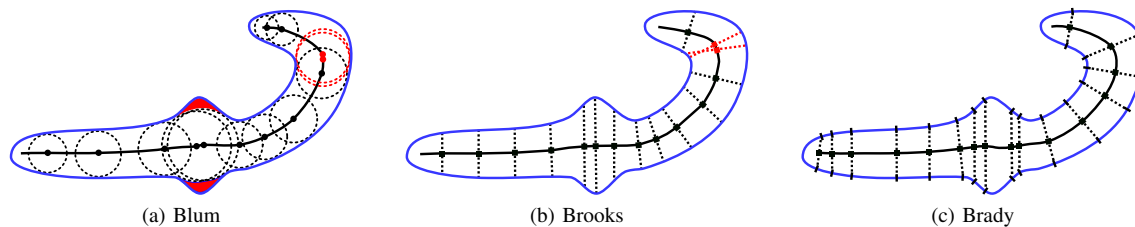


Fig. 1: Examples of spine's recovery for the three well-known types of ribbons. (a) Blum ribbon does not fully recover a shape with a high local positive curvature or strong turn. The red highlighted areas are missed and the red circles intersect at the same point, hence, this shape cannot be recovered with a Blum ribbon. (b) With the Brooks ribbon, generators also intersect. (c) Only the Brady ribbon can fully recover the shape. It is important to notice that Brady ribbons were specifically dedicated to recovery.

- generators should not intersect each others;
- generators should not contain one another.

Smooth variations of the size and the orientation imply the continuity of the shape contours. The two latter rules ensure that the shape is well formed: the non-intersection rule forbids objects with holes (which are not ribbons objects), and the non-inclusion rule imposes each generator center to contribute to the global shape of the ribbon (this way, the ribbon structure is very similar to that of the spine).

Figure 1 illustrates the most popular types of ribbons: *Blum ribbons*, *Brooks ribbons*, and *Brady ribbons*. Each of these is defined by a different generator: *Blum ribbons* use a circle as a generator [35], *Brooks ribbons* use a line segment with a fixed angle (often perpendicular) to the spine [36] and *Brady ribbons* use a line segment whose extremities make equal angles with the contour of the ribbons [3].

Rosenfeld [37] explains that these ribbons have different properties because they were designed for different purposes. For instance, Blum ribbons have the property to be uniquely recoverable from the contours, *i.e.* only one spine can be recovered from a given ribbon. However, these ribbons can represent a limited class of objects. Brooks ribbons are more flexible than Blum's, and are easy to generate, but are locally ambiguous due to their lack of constraint on the contour layouts. Thus, they may not result in a unique recovery of the spine. Finally, Brady ribbons define the most flexible class of ribbons. They describe well a wide range of shapes, but their generation is neither unique nor obvious compared to other ribbons. To generate Brooks and Blum ribbons only the size (radius) of the generator, l , is required before sweeping it out on the spine, whereas for Brady ribbons the slope of the generator with the spine, $\tan \theta$, is also required. Moreover, the end points of the generator are the sides of the ribbon and their slopes are equal to [37]:

$$\frac{d(y \pm l \sin \theta)}{d(x \pm l \cos \theta)} = \frac{y' \pm l' \sin \theta \pm l \theta' \cos \theta}{1 \pm l' \cos \theta \mp l \theta' \sin \theta}$$

with u' the local derivative of u . These slopes are referred to by $\tan \alpha_a$ and $\tan \alpha_b$, and they must respect the equal angle condition, meaning that $(\alpha_a + \alpha_b)/2 = \theta$. In practice, this equation is generally not easy to solve and this confirms that Brady's intention was not to define a generative model of ribbons.

The process of retrieving ribbons from images is different regarding the type of ribbon considered. Blum and Brady ribbons belong to the bottom-up category, since their detection algorithm may be divided into the following procedures: 1) a pre-processing step is used to extract binary contours, 2) multiple local symmetries are obtained according to their respective definitions, and finally, 3) the middle points on the axes of symmetry, the locus, are joined all together to obtain the medial axis. Brooks ribbons may be used in a top-down approach, by 1) generating candidates, and 2) validating or invalidating the candidates by matching their local characteristics to the image. The former type of approach may not be suited in the case of natural images, since retrieving binary contours may produce erroneous results, and yet these noisy measurements may increase the detection errors of the ribbon extraction algorithm. Conversely, generating candidates and then estimating their likelihood is not vulnerable to propagation errors from a pre-processing step and may benefit from a higher level of decision process.

To summarize, we expect that a ribbon satisfies the following three properties: 1) it should be expressive enough in order to be able to model a large range of ribbon-like objects; 2) it must be generative in order to be able to propagate ribbon candidates; 3) it should not be ambiguous (otherwise, it could lead to an ill-posed problem). Blum ribbons offer neither 1) nor 2), since they can represent a limited class of objects and are not generative. Brooks ribbons do not offer 3), since they are highly ambiguous. Finally, Brady ribbons do not offer 2). Therefore, we need to define a simple generative but yet expressive model of ribbons.

In our detection algorithm, ribbons are detected sequentially: candidate parts of a ribbon are generated and then weighted in a Bayesian framework. We base our ribbon definition on that of Brooks, but we add the constraint that the generator makes equal angles with the contours. This constraint is formulated in the Brady ribbons, and results in reducing the local ambiguities of the spine location. This way, we define a convenient generative ribbon model which can easily reconstruct contours of the shape and keep a good description capacity of smooth local symmetries for a large class of ribbons.

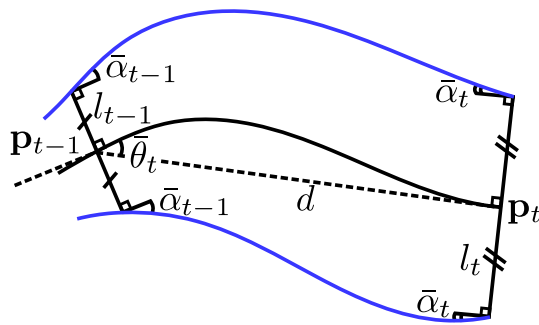


Fig. 2: Geometric model of a ribbon $\mathbf{r}_t = (\mathbf{p}_{t-1}, \theta_t, l_t, \bar{\alpha}_t)$. θ_t is the orientation shift $|\theta_t - \theta_{t-1}|$, l_t is the generator radius, and $\bar{\alpha}_t$ the deviation angle from θ_t , i.e. $\alpha_t = \theta_t \pm \bar{\alpha}_t$ with α_t being the angles between the contour curves and the abscissa axis. Contour points generated at the spine points \mathbf{p}_{t-1} and \mathbf{p}_t define local symmetries. Spine and contours are interpolated using cubic Hermite splines.

III. A MODEL OF RIBBONS

We present in this section a geometric model used to represent ribbons. Since we want to efficiently retrieve these structures from natural images, and be able to generate suitable candidates, a good geometric model would confer a trade-off between simplicity and power expressivity.

A. Geometric Model

Let us introduce some notations. Since the detection method is stochastic, the random variables are indexed by the time t . However, in our case, the tracking is not spatio-temporal but only spatial, thus, the time t is a t^{th} step iteration of our detection algorithm. Hence the set $\{\mathbf{p}_t, t \in \mathbb{N}\}$ represents a sequence of spine points. A spine point is a local smooth symmetry point, with respect to the Brady's definition [3], i.e. it is defined as the barycenter of the line segment joining two contour points that makes equal angles with the sides of the shape. The smooth spine is interpolated using a cubic Hermite spline whose data points and tangents are defined by the spine points. Hence, the tangent direction at the point \mathbf{p}_t is equal to $(\mathbf{p}_t - \mathbf{p}_{t-1}) / \|\mathbf{p}_t - \mathbf{p}_{t-1}\|$.

Let θ_t be the angle defined by the tangent direction, i.e. θ_t is the angle between \mathbf{p}_t and the abscissa axis with respect to the center point \mathbf{p}_{t-1} . We consider that the distance between two consecutive spine points is a parameter d of the method, so that the point \mathbf{p}_t is defined by the polar system (θ_t, d) and with respect to \mathbf{p}_{t-1} . We would also like to retrieve the symmetric contours from the spine. To do that, we add a local symmetry radius parameter, l_t , into our model. At a certain point \mathbf{p}_t , the contour points are defined along the normal of $(\mathbf{p}_{t-1}, \mathbf{p}_t)$, i.e. at an angle $\theta_t \pm \pi/2$ and at a distance l_t . These contour points are the extremities of the ribbon at time t . The angles between their respective contour curves and the abscissa axis are defined as $\alpha_t = \theta_t \pm \bar{\alpha}_t$, which is consistent with the Brady ribbons. The angles α_t are expressed as a deviation angle from θ_t because we assume that for most of natural ribbons, contours are locally parallel to the spine points, implying in

that case $\bar{\alpha}_t = 0$. Finally, the contours are interpolated using cubic Hermite splines, whose angles of the tangent are defined by the angles α_t . The model is illustrated in Figure 2.

B. State Model

In conformity with the notation system introduced in the previous section, the vector state \mathbf{s}_t of a spine axis at a time t is defined as $(\mathbf{p}_t, \theta_t, d)$, where \mathbf{p}_t is the spine point, θ_t the angular coordinate, and d the radial coordinate. These three variables define the axis of a local symmetry. Since the point \mathbf{p}_t is defined by the first point \mathbf{p}_0 and the polar coordinates (θ_t, d) , we discard it from the notation system. Plus, the distance d is fixed, and thus does not need to be estimated. In these conditions, the spine trajectory $\mathbf{s}_{1:t}$ is simply characterized by the sequence $(\mathbf{p}_0, \theta_1, \dots, \theta_t)$. However, in Section V, we will see that the first point of a trajectory is not necessarily \mathbf{p}_0 , since the trajectory corresponds to a set of spines instead of only one spine.

From the spine axis \mathbf{s}_t , we can define a ribbon portion, \mathbf{r}_t , as $\mathbf{r}_t = (\mathbf{s}_t, l_t, \bar{\alpha}_t) = (\theta_t, l_t, \bar{\alpha}_t)$, where l_t is the radius of the ribbon, and $\alpha_t = \theta_t \pm \bar{\alpha}_t$ the angles at the contours.

C. Probabilistic Model

There are two ways to define the prior and transition distributions $p(\mathbf{r}_t)$ and $p(\mathbf{r}_t | \mathbf{r}_{t-1})$. The first one is to learn these distributions on a ribbon shapes database. The spine can for example be retrieved using the Brady's algorithm [3], and the other conditional distributions can be approximated using a shape reconstruction error criterion. However, in this work, we propose to define generic distributions whose parameters are estimated by a trial-and-error procedure on the LS-BSDS300 training dataset. The reason is that in this way, the shape parameters can intuitively be tuned, which is suitable when one wants to detect a specific category of objects based on some shape criteria, as we will experiment in Section VI.

1) *Prior Model*: We define the prior distribution $p(\mathbf{r})$ as a product of independent terms:

$$p(\mathbf{r}) = p(\mathbf{p}_0) p(\bar{\alpha}) p(\theta) p(l) \\ = \mathcal{U}(\mathbf{p}_0; \Omega) \mathcal{N}(\bar{\alpha}; 0, \sigma_{\bar{\alpha}}) \mathcal{U}(\theta; [0, 2\pi[) \mathcal{U}(l; [l^m, l^M]) \quad (1)$$

Here we assume that the prior distribution of the spine point \mathbf{p}_0 follows a uniform distribution in the image domain Ω . The distribution of the angle θ is uniform in $[0, 2\pi[$, meaning that no initial direction is suggested. The prior distribution $p(l)$ is uniform in the interval $[l^m, l^M]$. The angle $\bar{\alpha}$ on the contours follows a normal distribution centered in 0 (in that case the tangent contours are equal to the one of the spine), and of standard deviation $\sigma_{\bar{\alpha}}$.

2) *Transition Model*: The joint transition distribution is used to propagate a spatial information from $t-1$ to t and is defined as:

$$p(\mathbf{r}_t | \mathbf{r}_{t-1}) = p(l_t, \bar{\alpha}_t | \mathbf{s}_t, l_{t-1}) p(\mathbf{s}_t | \mathbf{s}_{t-1}) \\ = p(\bar{\alpha}_t) p(\theta_t | \mathbf{r}_{t-1}, \theta_{t-1}, l_t) p(l_t | l_{t-1}) \\ = \mathcal{N}(\bar{\alpha}_t; 0, \sigma_{\bar{\alpha}}) \mathcal{N}(\theta_t; \theta_{t-1}, \sigma_{\theta}) \mathcal{N}(l_t; l_{t-1}, \sigma_l) \quad (2)$$

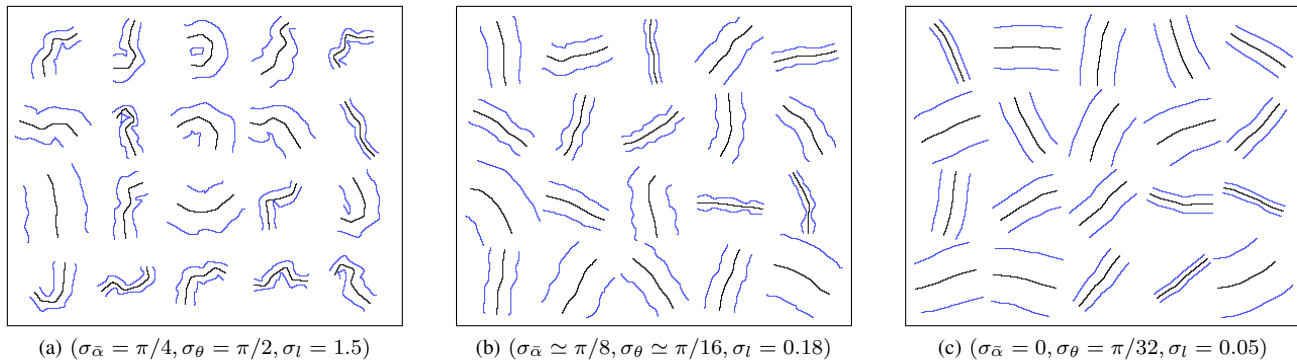


Fig. 3: Simulation of ribbons using different parameters for the prior and transition models. Other parameters have been set to $d = 9$ pixels, $t = 1, \dots, 5$, for all the simulations. Each simulation has been laid out on a 75×75 pixel grid. The simulations in (b) correspond to the estimated parameters on the LS-BSDS300 training dataset.

The transition of the radius is defined as a truncated normal distribution $\tilde{N}_{l^m, l^M}(l_t; l_{t-1}, \sigma_l)$, with $[l^m, l^M]$ the support interval. The transition of θ_t is then defined as a normal truncated distribution $\tilde{N}_{r_t}(\theta_t; \theta_{t-1}, \sigma_\theta)$ whose endpoints are computed using $(\mathbf{r}_{t-1}, \theta_{t-1}, l_t)$ in order to induce a generator (connecting line of the two contour point extremities at t) that does not intersect with the one at $t-1$. This ensures generating locally well formed ribbons, and implies that the thickness of a generated ribbon cannot exceed twice the radius of curvature of its spine. Figure 3 illustrates ribbons generated using the prior model (Eq. 1) and the transition model (Eq. 2) for several combinations of parameters.

IV. COMPUTING FEATURES ON RIBBONS

In this section, we propose to extract five symmetry features. We note $\overleftarrow{c}(\mathbf{r})$, $\overrightarrow{c}(\mathbf{r})$, the left and right contours, respectively, associated with the medial axis \mathbf{r} , and $\overleftrightarrow{c}(\mathbf{r})$ whether it can be one or the other. The sides of the contours are arbitrarily defined relatively to the direction of traversal of the ribbon. The time subscript t is omitted for more clarity. The first three features are computed near the contours: the local gradient f^1 , the oriented gradient f^2 , and the textural gradient f^3 . The textural features f^4 and f^5 ensure that the medial axis is correctly located inside an object. All the features are illustrated in Figure 4.

A. Contour Features

First, we need to define a merging operator to combine a pair of contour features $(f^i(\overleftarrow{c}(\mathbf{r})), f^i(\overrightarrow{c}(\mathbf{r})))$. We define it as the harmonic mean between these two quantities:

$$f^i(\mathbf{r}) = \frac{2f^i(\overleftarrow{c}(\mathbf{r}))f^i(\overrightarrow{c}(\mathbf{r}))}{f^i(\overleftarrow{c}(\mathbf{r})) + f^i(\overrightarrow{c}(\mathbf{r}))}, \forall i = 1, 2, 3$$

The choice of a harmonic mean relies on the fact that when computed on just two values, this merging operator is a good trade-off between a minimum and an average operator. It forces the contour features to get a high and consistent response, but it is also less drastic than a minimum operator when one of the two contours is cluttered or occluded.

1) *Local Gradient*: The local gradient is a smoothed gradient norm $|g * \nabla I|$ of the image I . The feature $f^1(\overleftrightarrow{c}(\mathbf{r}))$ is computed along a contour $\overleftrightarrow{c}(\mathbf{r})$ of length L :

$$f^1(\overleftrightarrow{c}(\mathbf{r})) = \frac{1}{L} \sum_{i=1}^L |\nabla g * I(\overleftrightarrow{c}^i(\mathbf{r}))|$$

where $\overleftrightarrow{c}^i(\mathbf{r})$ is the i^{th} point of the contour $\overleftrightarrow{c}(\mathbf{r})$. For an image I , we take the maximum gradient value on each point among the different channels.

2) *Oriented Gradients*: We propose to compute a histogram distance between two oriented gradient patches. The histogram of oriented gradients [38] summarizes a textural information. The distance between two adjacent patches of oriented gradients is known to give a good indication of the presence of a contour [12]. The procedure proposed here is slightly different from that of the pioneers Dalal and Triggs [38] and has been used in [39]. For a point $\overleftrightarrow{c}^i(\mathbf{r})$ of a contour $\overleftrightarrow{c}(\mathbf{r})$, we consider the two sides of its normal segment, one related to the outside the object, n_{out} , and one related to the inside, n_{in} . Let $h_{\text{O}}[a] = \{h_{\text{O}}^r[a]\}_{r=1}^R$ be the histogram of a set of pixels a , where r is the bin index of a histogram of length $R = 4 \times R_{\text{M}}$, with 4 the number of considered orientations (vertical, horizontal, and two diagonals) and R_{M} the number of magnitude bins. The distance between two adjacent oriented gradient patches is defined as:

$$f^2(\overleftrightarrow{c}(\mathbf{r})) = \frac{1}{L} \sum_{i=2}^{L-1} d_{\text{B}}(h_{\text{O}}[n_{\text{in}}(\overleftrightarrow{c}^i(\mathbf{r}))], h_{\text{O}}[n_{\text{out}}(\overleftrightarrow{c}^i(\mathbf{r}))])$$

with $d_{\text{B}}(h[a], h[b]) = \left[1 - \sum_{r=1}^R \sqrt{h^r[a] h^r[b]}\right]^{1/2}$ the Bhattacharyya distance between two histograms. In practice, we consider normal segments with 5 pixels width in order to get enough samples to compute the histograms.

3) *Textural Gradients*: The last contour feature integrates the color distribution of the neighborhood of a contour to help determine whether the candidate is located on a true contour or not. We consider two CIE Lab color histograms of adjacent patches, ideally one located inside the contour, the other one outside. The dimension of the outside patch is fixed, while the inside patch depends on the radius of the object. Let h_{T} be a

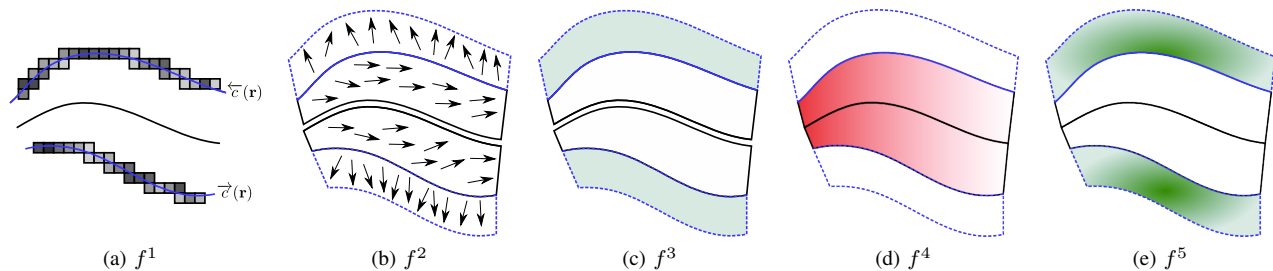


Fig. 4: Features. $\overleftarrow{c}(\mathbf{r})$ and $\overrightarrow{c}(\mathbf{r})$ are the left-side and right-side contours, respectively. The figures represent the five proposed features $f^j, j = 1, \dots, 5$. (a) f^1 is the local gradient on both sides of the contours. (b) f^2 is a combination of two histogram distances of the oriented gradients computed inside and outside the object. (c)-(e) The three last features are extracted from the color histograms of these regions: f^3 ensures the presence of the contours by a combination of the two distances inside / outside the object, the other ones ensure the presence of an object by a distance inside / inside (f^4) and another outside / outside (f^5).

textural color histogram of length R equals $R_T \times R_T \times R_T$, with R_T the number of bins by channel. The distances between the pair of histograms along the barycenter contour curve is defined as:

$$f^3(\overleftarrow{c}(\mathbf{r})) = d_B(h_T[n_{in}(\overleftarrow{c}^b(\mathbf{r}))], h_T[n_{out}(\overleftarrow{c}^b(\mathbf{r}))])$$

with $\overleftarrow{c}^b(\mathbf{r})$ the barycenter point of the contour $\overleftarrow{c}(\mathbf{r})$. In practice, to limit the additional computational cost of considering a radius-dependent patch size, the regions are approximated by considering rectangles centered on the barycenter points, and 16 possible orientations in $[0, \pi]$. This feature is computed at four different size patches in order to integrate a more global information.

B. Object Features

The feature f^4 ensures that the medial axis divides a region into parts sharing the same textural information. It helps detecting symmetries inside consistent regions. This is done by considering color histograms extracted from two patches located on the inner sides of the contour:

$$f^4(\mathbf{r}) = 1 - d_B(h_T[n_{in}(\overleftarrow{c}^b(\mathbf{r}))], h_T[n_{in}(\overrightarrow{c}^b(\mathbf{r}))])$$

Finally, the feature f^5 assumes that the object is superposed to a unique background. The same hypothesis is made in [30], and from experiments, we found that it greatly reduces the number of undesired symmetry detections in the background. We also use color histograms to check that the background histograms on each contour side match:

$$f^5(\mathbf{r}) = 1 - d_B(h_T[n_{out}(\overleftarrow{c}^b(\mathbf{r}))], h_T[n_{out}(\overrightarrow{c}^b(\mathbf{r}))])$$

V. RIBBON DETECTION ALGORITHM

In this Section, we present a local symmetry detection algorithm using a particle filtering technique. The algorithm is based on previous developments in which we proposed the PFCF framework [19], [20], standing for Particle Filter for Contour Detection, in order to detect contours in natural images. We turn the ribbon detection problem into the estimation of the distribution of the joint local reflection

symmetries and radiuses $\mathbf{r}_{1:t} = (\theta_{1:t}, l_{1:t}, \bar{\alpha}_{1:t})$ considering a set of observations \mathbf{y} . The detection algorithm is composed of three estimation steps:

- The approximation of the likelihood functions, which are defined using the features described in Section IV.
- The approximation of the initialization distribution of the local symmetries $p(\mathbf{r}_1|\mathbf{y})$ is performed using a Monte-Carlo sampling approach. Ribbons are first generated using the prior distributions defined in Section III, and are then weighted using the likelihood functions. The joint likelihood function and the initialization distribution are both estimated during this step.
- The approximation of the trajectory distribution $p(\mathbf{r}_{1:t}|\mathbf{y})$ is performed using a sequential Monte-Carlo technique, the so-called particle filter. For each step t , ribbons are first generated using an importance function that is related to the transition distributions defined in Section III, and are then weighted using, in particular, the same joint likelihood function that was used in the initialization step. Each time a ribbon has been entirely extracted, the detection is automatically reinitialized on a new ribbon. Local symmetries are therefore iteratively and spatially tracked until the particle filter reaches a stopping criterion.

A. Likelihood Function

This section presents how estimating tail distributions of the features, or complementary cumulative distribution functions of the features, are used to overcome two classical drawbacks of particle filtering techniques. First, if a particle gets lost, there is no mechanism in the particle filter to handle it. This may happen, for example, when the extraction of the last ribbon was completed at $t-1$, and then needs to move on a new spine. This is partially due to the weight normalization step, intrinsic to particle filters, which avoid any global comparison. Second, particle filters do not embed a stopping criterion, which is necessary in our application. We denote by $\mathbf{y}^j(\mathbf{r})$ the observation associated to the ribbon \mathbf{r} . The observation $\mathbf{y}^j(\mathbf{r}_t)$ is given by the tail distribution associated to the feature j :

$$\mathbf{y}^j(\mathbf{r}_t) = P(\mu^j > f^j(\mathbf{r}_t)) \quad (3)$$

where μ^j is a random variable taking values in \mathbb{R}^+ and f^j is the j^{th} feature previously defined in Section IV. In this case, the tail distribution of the j^{th} feature response can be seen as a false alarm distribution. Its value tends to 0 as the feature response tends to $+\infty$. In this last example, we say that the feature response is *meaningful*, conveying the idea that significant events, according to the human perception, are rare. This is the idea of the Helmholtz principle, and has notably been used in an *a contrario* framework [40], [41]. Using this contextual information, these J distributions will be exploited to decide whenever some particles need to be reassigned to new objects, and when to stop the detection algorithm.

We define the likelihood joint $p(\mathbf{y}|\mathbf{r}_t)$ which measures the adequation between the joint observation $\mathbf{y} = (\mathbf{y}^1, \dots, \mathbf{y}^J)$, and the state \mathbf{r}_t at a time t . Although the vector observation \mathbf{y}^j designates observations from the whole set of ribbons included in an image, the likelihood is only evaluated on the ribbon \mathbf{r}_t [20], [42]:

$$p(\mathbf{y}|\mathbf{r}_t) = \prod_{j=1}^J p(\mathbf{y}^j(\mathbf{r}_t)) \quad (4)$$

where the observations $(\mathbf{y}^1, \dots, \mathbf{y}^J)$ are assumed conditionally independent given (\mathbf{r}_t) . The marginal likelihood $p(\mathbf{y}^j(\mathbf{r}_t))$ is defined using its associated tail distribution:

$$p(\mathbf{y}^j(\mathbf{r}_t)) \propto \exp\left(-\lambda^j P(\mu^j > f^j(\mathbf{r}_t))\right) \quad (5)$$

where f^j is the j^{th} feature defined in Section IV, and $\lambda^j \in \mathbb{R}^+$ a coefficient that weights the importance of the feature j . The marginal likelihood $p(\mathbf{y}^j(\mathbf{r}_t))$ gives low density values for high values of the tail distribution, i.e., the density values are low for common events, thus giving more weight to rare relevant events. The tail distributions are estimated using a Monte Carlo procedure, along with the initialization distribution, as we will see in the next section.

B. Initialization Distribution

Particle filters assume that an initialization distribution is known. In real-world applications, this might not be the case, so we need to estimate it. We could also simply use the prior distribution $p(\mathbf{r}_1)$ defined in Section III, but since by definition they are not conditioned by the observation \mathbf{y} , it could be inefficient. This leads the estimating of the initialization distribution $p(\mathbf{r}_1|\mathbf{y})$ using a classic Bayesian decomposition:

$$\begin{aligned} p(\mathbf{r}_1|\mathbf{y}) &\propto p(\mathbf{y}|\mathbf{r}_1)p(\mathbf{r}_1) \\ &= p(\mathbf{y}|\mathbf{r}_1)p(\bar{\alpha}_1)p(\theta_1)p(l_1)p(\mathbf{p}_0) \end{aligned} \quad (6)$$

with the prior distribution $p(\mathbf{r}_1)$ defined in Equation 1 and the likelihood $p(\mathbf{y}|\mathbf{r}_1)$ defined in Equation 4. To alleviate the problem of dimensionality, generated ribbons from the initialization distribution are rectangular, i.e. $l_0 = l_1$ and $\bar{\alpha}_0 = \bar{\alpha}_1$. The tail and initialization distributions are approximated using a Monte Carlo importance sampling procedure by:

- Generating samples $\{\mathbf{r}_1^{(n)}\}_{n=1, \dots, N_i}$ according to the prior distribution $p(\mathbf{r}_1)$;
- Evaluating the feature responses $\{f^j(\mathbf{r}_1^{(n)}), \forall j = 1, \dots, 5\}_{n=1, \dots, N_i}$ of every sample, which results in

the approximation of the tail distributions $\{P(\mu^j > f^j(\mathbf{r}_t)), \forall j = 1, \dots, 5\}$;

- Computing the joint likelihood $p(\mathbf{y}|\mathbf{r}_1^{(n)})$ of every sample, which results in the approximation of the initialization distribution $p(\mathbf{r}_1|\mathbf{y})$.

The initialization distribution implies that the spatial tracking of the ribbons can begin at any location of an object. As we will see in Section V-C, the potential negative effect of this behavior is coped by the sequential aspect of the detection algorithm that allows the tracking to pursue the detection on other object parts at different times, and by the use of multiple particle filters to add diversity to the posterior distribution. Moreover, the method implicitly favors to find the longest symmetry axis of a ribbon: if the probability of initialization is identical at any location of the ribbon axes, the effect of this tendency is proportional to the length of the symmetry axes. This means that minor axes can still be detected, but with a lower probability. From a ribbon representation perspective, this is consistent with the fact that one generally wants to limit the ambiguities of the ribbon model, and that the longest axes should be preferred, since they represent the object more intuitively [37].

C. Detection Algorithm: Trajectory Distribution

First, we need to make an adjustment of the hidden state. In order to detect all the ribbon objects of an image, the algorithm needs a mechanism to detect when the extraction of a ribbon has been completed. Let j_t be a binary random variable that designates the current work state of the ribbon extraction. If $j_t = 0$, the spatial tracking goes on normally. Otherwise, it means that the extraction of the current object is completed and needs to be reassigned to a new object. This point will be further clarified in the current section. The purpose of this section is to estimate the trajectory distribution $p(\mathbf{x}_{1:t}|\mathbf{y})$ using a particle filtering technique, with $\mathbf{x}_t = (\mathbf{r}_t, j_t) = (\theta_t, \bar{\alpha}_t, l_t, j_t)$. The modeling of the state and the definition of the prior and transition distributions in the original PFCF approach [19] are conceptually different from the one proposed here.

In sequential Monte Carlo methods, $(\mathbf{x}_t)_{t \in \mathbb{N}^+}$ and $(\mathbf{y}_t)_{t \in \mathbb{N}^+}$ are modeled as stochastic processes, where the index t represents the time. In image processing, particle filters are essentially known for tracking applications. Basically, the goal is to estimate the position of an object over time by empirically approximating the posterior density distribution and next by computing the Monte Carlo expected value. However, in the context of symmetry and ribbon detections, the tracking is purely spatial as it aims at estimating the positions of ribbons by locally propagating small parts of ribbons. Thus, the time index is removed from (\mathbf{y}_t) , as the observations could all be gathered at the beginning of the estimation process. The recursive trajectory estimation consists in estimating the following posterior distribution:

$$p(\mathbf{x}_{1:t}|\mathbf{y}) \propto p(\mathbf{x}_{1:t-1}|\mathbf{y})p(\mathbf{y}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad (7)$$

where the likelihood function $p(\mathbf{y}|\mathbf{x}_t) = p(\mathbf{y}|\mathbf{r}_t)$ has been defined in Equation 4. The transition density distribution

$p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is defined as:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = p(\mathbf{r}_t|\mathbf{r}_{t-1})p(j_t) \quad (8)$$

with the ribbon transition $p(\mathbf{r}_t|\mathbf{r}_{t-1})$ as defined in Equation 2. We assume that the transition of the jump variable is independent from its past state, i.e. $p(j_t|j_{t-1}) = p(j_t)$. This way, $p(j_t)$ implicitly controls the length of the detected spines. We note $\kappa = p(j_t = 1)$ the probability to make a jump.

The particle filter aims at recursively approximating the posterior distribution $p(\mathbf{x}_{1:t}|\mathbf{y})$ using a finite set of N samples $\{\mathbf{x}_t^{(n)}\}_{n=1,\dots,N}$. The resulting empirical distribution is [43]:

$$P_N(d\mathbf{x}_{1:t}|\mathbf{y}) = \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_{1:t}^{(n)}}(d\mathbf{x}_{1:t}) \quad (9)$$

where $\delta_{\mathbf{x}_{1:t}^{(n)}}(\cdot)$ is a Dirac mass centered on a hypothetical state realization $\mathbf{x}_{1:t}^{(n)}$ of the state $\mathbf{x}_{1:t}$, also called particle, $w_t^{(n)}$ is its weight, and $d\mathbf{x}_{1:t}$ is an event of infinitesimal support. The recursion estimation of the trajectory distribution can be carried out by three steps: 1) propagating the particle set from $t-1$ to t using an importance function; 2) updating the particle weights using in particular the likelihood, and resampling the particles if needed [43]; 3) checking if the particle filter has completed its task, i.e. if no further ribbon needs to be extracted. These three steps are detailed below.

1) *Generating the Particles:* The first step of a particle filter algorithm consists in propagating the particles from the previous particle set $\{\mathbf{x}_{t-1}^{(n)}\}_{n=1,\dots,N}$. Particles are generated using a function of importance $q(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{y})$:

$$q(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{y}) = q(\mathbf{r}_t|\mathbf{r}_{1:t-1}, j_t, \mathbf{y}) q(j_t|\mathbf{x}_{t-1}, \mathbf{y}) \quad (10)$$

The ribbon proposal density is proportional to the initialization density (Eq. 6) if a jump is made. Otherwise, the proposal density is proportional to the ribbon transition (Eq. 2). Also, it depends on a trajectory term that gives little probability to echo detections:

$$q(\mathbf{r}_t|\mathbf{r}_{1:t-1}, j_t, \mathbf{y}) = \frac{1}{n_q} f(\mathbf{s}_t, \mathbf{s}_{1:t-1}) \times [j_t p(\mathbf{r}_t|\mathbf{y}) + (1 - j_t) p(\mathbf{r}_t|\mathbf{r}_{t-1})] \quad (11)$$

The trajectory constraint $f(\cdot)$ is set to 10^{-5} , if any point of the spine curve defined by \mathbf{s}_t is closer than a Manhattan distance of 5 with any spine point of the past ribbons $\mathbf{s}_{0:t-1-k}$, with k the length of the current ribbon, otherwise $f(\cdot)$ is set to 1. While the spines from current and past ribbons cannot intersect, the ribbons themselves can. This way, the algorithm can detect symmetries at different scales, which may for example happen with objects presenting local and global symmetries. There is no restriction on the current ribbons, since it is already done for the past state in the definition of $p(\mathbf{r}_t|\mathbf{r}_{t-1})$. Hence, self-crossing ribbons are unlikely, but remain possible in order to add a little flexibility in the detection process. Generating the particles can be done using a rejection sampling, i.e. generating a candidate $\mathbf{s}_t^{(n)}$ with the right-side of Equation 11, and accepting it with a probability of $f(\mathbf{s}_t^{(n)}, \mathbf{s}_{1:t-1}^{(n)})$. Finally, the normalizing term n_q can be approximated using an importance sampling method, but we

found in practice that ignoring it resulted in a non-significant bias, while saving great computational time.

The jump proposal density $q(j_t|\mathbf{x}_{t-1}, \mathbf{y})$ uses the probabilities of the tail distributions (Eq. 3) computed at $t-1$ to determine if the ribbon \mathbf{r}_{t-1} is still relevant. Therefore, a jump is made with probability $\sum_{j=1}^J \tilde{\lambda}^j P(\mu^j > f^j(\mathbf{r}_{t-1}))$, where $\tilde{\lambda}^j = \lambda^j / \sum_{k=1}^J \lambda^k$.

2) *Computing the Particle Weights:* The joint likelihood (Eq. 4), the transition model (Eq. 8) and the proposal distribution (Eq. 10) are used to weigh the particles:

$$w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(\mathbf{y}|\mathbf{r}_t^{(n)}) p(\mathbf{x}_t^{(n)}|\mathbf{x}_{t-1}^{(n)})}{q(\mathbf{x}_t^{(n)}|\mathbf{x}_{1:t-1}^{(n)}, \mathbf{y})} \quad (12)$$

with $\sum_{n=1}^N w_t^{(n)} = 1$. Thus, a particle is assigned to a high weight when its likelihood is high, and when prior and importance function are in agreement. This shows the importance of defining an importance function dependent on both the transition and the observation, and whose support includes the one of the filtering distribution. The particles may then be resampled when their effective number $[\sum_{n=1}^N (w_t^{(n)})^2]^{-1}$ becomes lower than a threshold, typically $0.75 \times N$ [43].

3) *Algorithm Completion:* A particle filter is automatically stopped when the number of jumps reaches a learned threshold. The number of jumps being related to the meaningfulness of an extracted ribbon, this means that the algorithm automatically stops when all relevant ribbons have been extracted from the image. Also, L several independent particle filters are used to estimate the trajectory distribution in order to add diversity to the trajectory, hence yielding $\{\mathbf{r}_t^l = (\mathbf{p}_t^l, \theta_t^l, l_t^l, \bar{\alpha}_t^l, j_t^l)\}_{l=1,\dots,L; t=1,\dots,t_t}$, with t_l the last step performed by the l^{th} particle filter. Let $\bar{\mathbf{r}}_{1:t_t}^l$ be the ribbon trajectory of the particle of maximum weight of the l^{th} particle filter, and $\bar{w}_{t_t}^l$ its associated weight. The soft ribbon map is a real-valued image in $[0, 1]$, giving the probability of a pixel $z \in \Omega$ to be part of a ribbon:

$$O(z) = \frac{1}{L} \sum_{l=1}^L \bar{w}_{t_t}^l \mathbf{1}_{\bar{\mathbf{r}}_{1:t_t}^l}(z) \quad (13)$$

with l the index of a particle filter, and $\mathbf{1}_{\mathbf{r}}(z) = 1$ if z belongs to the ribbon \mathbf{r} , 0 otherwise. Contour and spines soft maps can be accordingly computed replacing \mathbf{r} by the adequate variables in Equation 13.

VI. EXPERIMENTS

This section presents some experimental results in symmetry, contour, and ribbon detections. First, we talk about implementation details, parameters tuning, and their influence.

A. Implementation

The proposed algorithm uses parameters of three different kinds. They originate from: 1) the geometric model, 2) the likelihoods, 3) the particle filter framework. For the geometric model, parameters were either optimized using a trial-and-error procedure on the LS-BSDS300 training dataset for the detection of local symmetries (Section VI-B) or manually set

(although they also could be learned) for the detection of specific ribbons (Section VI-C). The trial-and-error procedure gave the best results for ($d = 9, \sigma_{\bar{\alpha}} = 0.4, \sigma_{\theta} = 0.2, \sigma_l = 0.18$). Closely related to the geometric model since it controls the length of the spines, the prior probability of making a jump was estimated to $\kappa = 0.0005$. The minimal radius l^m was manually set to 3 and the maximal radius l^M to $\max(w, h)/2$, which ranges the detection thickness from thin tubular objects to objects as wide as the image. For the likelihoods, the weights $\{\lambda^j\}_j$ and the histogram computation internal parameters have also been optimized using a trial-and-error procedure, and were found optimal for ($\lambda^1 = 25, \lambda^2 = 10, \lambda^3 = 25, \lambda^4 = 13, \lambda^5 = 13$), for color histograms of $6 \times 6 \times 6$ bins and for oriented gradient histograms of 8 bins by direction, respectively. These parameters have been fixed for all the different experiments. As for the particle filters, $N_i = 10^7$ samples in the tail and initialization distributions and 150 particle filters of 150 particles were needed to achieve the estimation of the posterior distribution. These parameters have been set to offer a good trade-off between estimation accuracy and computation time. Finally, a particle filter stops when $t \geq 100$ and its proportion of jump on the last 200 steps reaches 14%. These parameters have been estimated using a trial-and-error procedure.

The algorithm is well suited for parallel programming: the samples in the approximation of the tail and initialization distributions are independent and identically distributed (i.i.d.), the samples in a particle filter are also i.i.d. (before the normalization step), and the particle filters are mutually independent. Hence, the algorithm has the advantage of getting benefits from any hardware based on multi-threaded CPUs and/or GPU cards. Our implementation has been done in Cuda, and runs in about 22 seconds on a Nvidia GTX 670 GPU card. In contrast, the code of Tsogkas and Kokkinos [30] is a mixed C++ and Matlab implementation and takes about 270 seconds on a quad cores i7-920 (2.66 GHz) processor. Code and results are available at <http://www.iro.umontreal.ca/~mignotte/pfsd/>.

B. Local Symmetry Detection

Since our approach deals with local symmetry detection, we first compare our results on the LS-BSDS300 to the ones of [30], [32], [33]. The LS-BSDS300 test dataset contains 89 of the 100 test images of the BSDS300 [30]. The LS-BSDS300 ground truth is extracted from an automatic skeletonization algorithm from the manual selections of symmetrical objects. For these objects, the skeleton medial axes are assumed to be good approximations of the local symmetry curves. The ground truth score is indicated by a plain red curve at $F = 0.73$. The results of Lindeberg's method [33] come from the implementation proposed in [17]. The F-Measure curves are illustrated in Figure 5, and pairwise score comparisons are presented in Table I. We performed the experiment several times and obtained a variance of 2.3×10^{-6} attributed to the stochastic nature of our algorithm. Our approach (FM: 0.422) compares favorably to Levinshtein *et al.* [33] (FM: 0.356) and Lindeberg [32] (FM: 0.36) algorithms, and obtains a slightly inferior F-Measure score compared to the one obtained

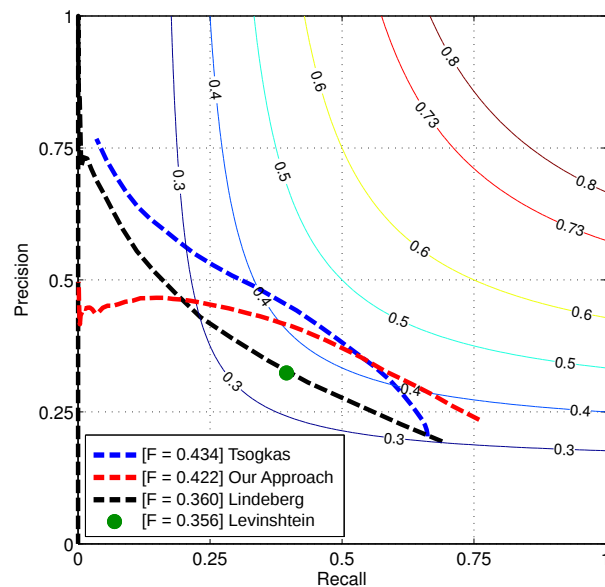


Fig. 5: Scores obtained by state-of-the-art methods and our proposed approach on the LS-BSDS300. The F-Measure curves correspond to the harmonic means of the recall and the precision scores. The final score is the optimal threshold computed among the 89 test images.

	Levinshtein	Lindeberg	Tsogkas	Our approach	All
Levinshtein	–	28	11	14	4
Lindeberg	61	–	19	23	9
Tsogkas	78	70	–	56	47
Our approach	75	66	33	–	29

TABLE I: One-to-one and one-against-all comparisons of symmetry detection methods on the LS-BSDS300. Scores are compared based on the optimal threshold obtained by each image. Each row points out the number of images that obtain better scores than the method in the corresponding column. The last column presents a one-against-all comparison.

by Tsogkas and Kokkinos [30] (FM: 0.434). One-against-all comparisons show that Tsogkas and Kokkinos method obtains the best performance on 47 of the 89 test images. Our approach obtains the highest scores for 29 images. Figure 6 depicts the ground truth for several images and their associated results for the comparison methods and ours. For Lindeberg, Tsogkas and Kokkinos, and our method, thresholds were optimized over the whole dataset. The strength of the approach proposed by Tsogkas and Kokkinos is to result in well structured spine detection, even for the thresholded maps. In comparison, our method produces less misled spines originating from textures and contours. Figure 7 presents several grayscale results of the extracted spines and their corresponding contours. The detected spines are the ones evaluated by the LS-BSDS300. We also display a degree of symmetry in an image by defining it as the ratio between the area of the extracted ribbons and the number of pixels in the image. We can especially see that images of human constructions, such as statues and buildings, have high degrees of symmetry.

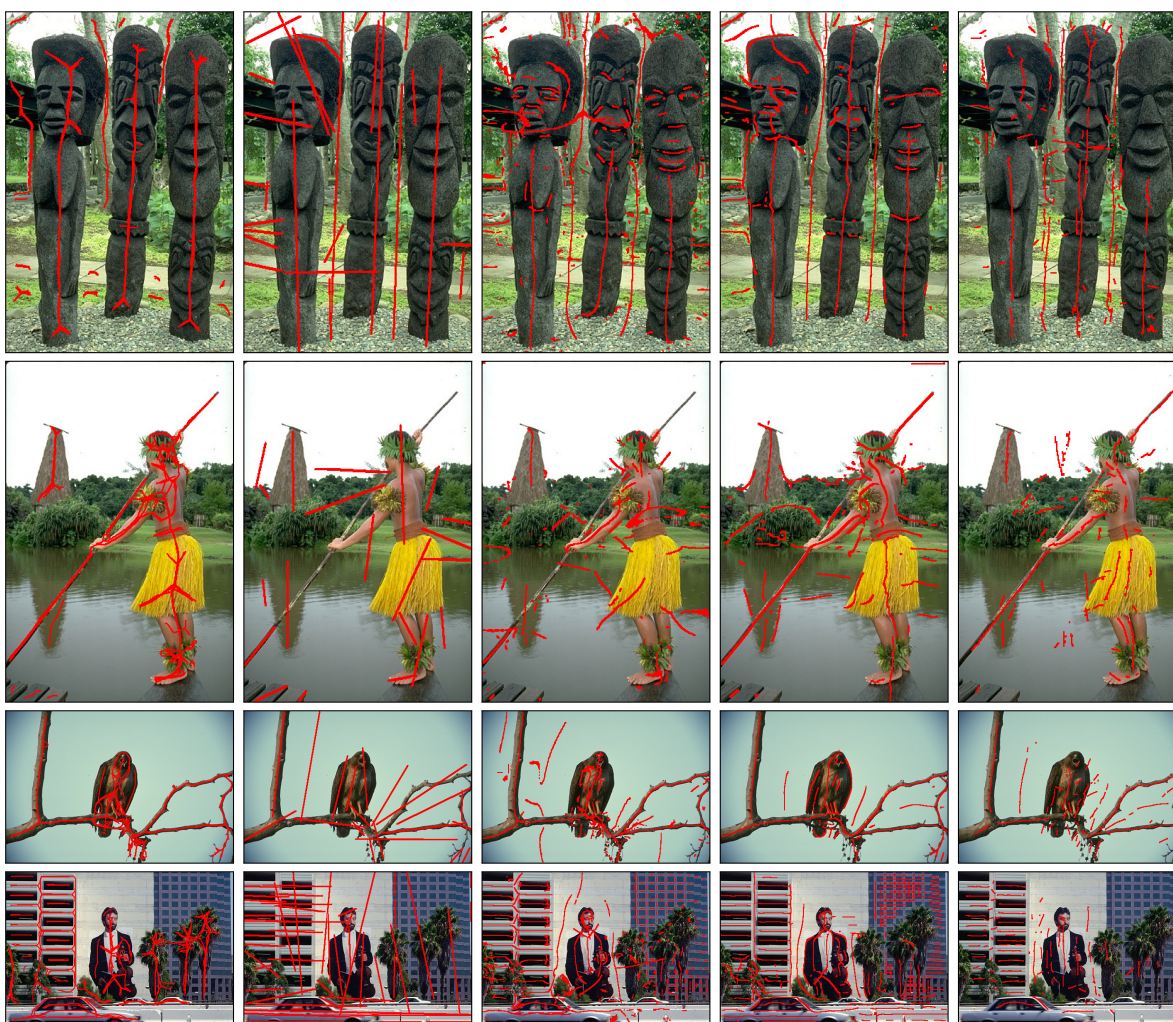


Fig. 6: Local symmetry ground truth (1st column on the left) and several thresholded detection results obtained by (2nd to the last column on the right): Levinstein *et al.* [33], Lindeberg [32], Tsogkas and Kokkinos [30], and our approach, respectively. Thresholds were optimized over the whole dataset.

As mentioned in [30], it is relevant to compare the algorithms [30], [32], [33] with ours since all these algorithms aim at detecting local symmetries. However, it is worth noticing that these algorithms produce different outputs: the detector proposed by Levinstein *et al.* [33] approximates symmetric objects by fitted ellipses from which symmetric axes are retrieved, thus the output is binary, whereas in [30], [32], the outputs are one-dimensional, and in grayscale. For the proposed method, the output is also a probability, but on ribbons, not just spines. The two-scale parameters and the symmetry axes extracted by the Levinstein *et al.* approach could be directly used as a preprocessing step of a segmentation algorithm. As for the other approaches, which include ours, a thresholding step may be needed before applying a segmentation algorithm. As it happens, this threshold may be the one maximizing the F-Measure score in Figure 5. Our approach not only provides the spines but also the structural information of the object, namely its scale and its contours (Figure 7). Finally, although the precision-recall curves obtained by Tsogkas and Kokkinos [30] and our approach are very different, the F-Measure scores are

quite similar. Moreover, we can see from Table I that these methods perform well for different images. This can certainly be explained by the distinct natures of these two algorithms: one is a data-driven learning method, the other is a geometric-based model method. This could suggest that the former could be embedded as a likelihood into our Bayesian approach to achieve better results.

Failure modes of the proposed approach include incorrect and meaningless symmetry axis detections, e.g. between the raven and the vertical twig (last image of the third row in Figure 7), or in the snow (in the middle of the second row in the same figure). This behavior can be explained by our definition of symmetry, which is purely local, and thus both the geometric model and the likelihood functions have not the adequate level of abstraction to discriminate relevant from insignificant symmetry axes. To overcome these limitations, we propose to integrate prior geometric information in order to detect specific ribbons. This is the subject of the following section.

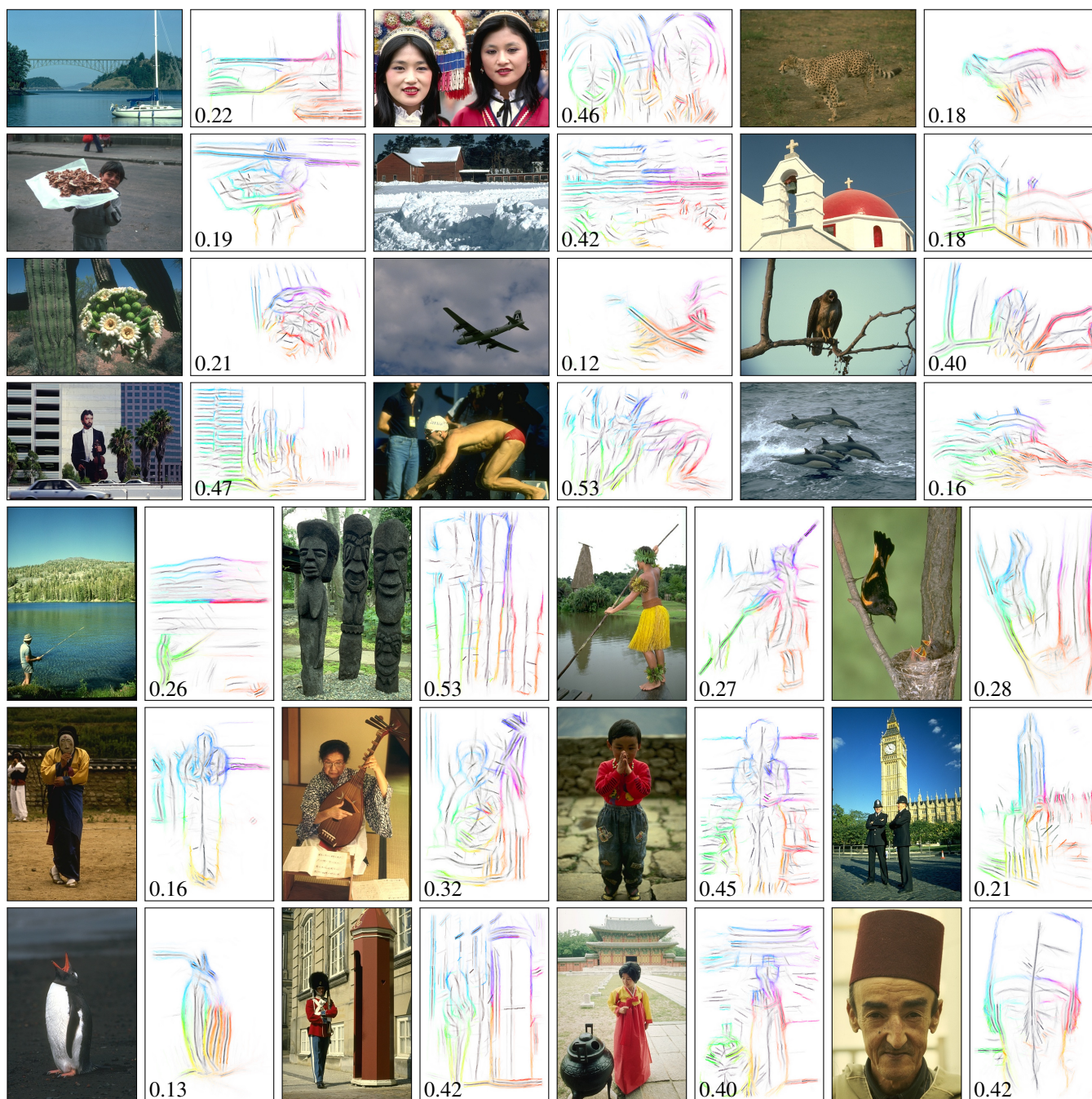


Fig. 7: Examples of local symmetry (black) and contour (color) detections obtained by the proposed algorithm on the BSDS300. Values correspond to the degrees of symmetry estimated on the images. Contour points associated with the same spine point share the same color.

C. Ribbon Detection

In this section, we address the problem of detecting symmetric objects. As opposed to state-of-the-art symmetry detection methods [21], [26], [27], [30], [32], [33], our approach yields to a ribbon detection output.

1) *Filtering the Posterior Trajectory*: A first way to recover specific ribbons from an image is to filter the output trajectory $\{\mathbf{r}_i^t\}_{t=1, \dots, L; i=1, \dots, t_i}$ using some geometric criteria. This is illustrated in Figure 8. The ribbons have been detected from

the trajectories displayed in Figure 8(b) using criteria such as minimum/maximum ribbon radiuses, and minimum/maximum spine lengths. In Figure 8(d)-(f), radius and length have been measured for specific objects of the image, and their corresponding range parameters have been set to $\pm 10\%$ of their values. The analysis of the posterior trajectory distribution could also be exploited in an object recognition system based on symmetric-, or ribbon-object parts.

2) *Constraining the Geometric Model*: The proposed geometric model is designed in such way that each component

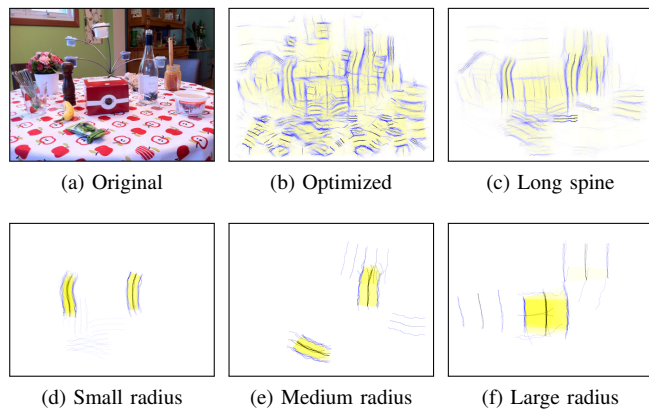


Fig. 8: Ribbon detection. Contours (blue), symmetries (black), and objects (yellow). The brightness of the color indicates the probability of detection. (b) Result using the parameters optimized on training data of the LS-BSDS300. (c) Trajectories with a spine length less than 36 pixels have been filtered from (b). (d-f) Trajectories with a spine length between 45 and 72 pixels and radiuses between (d) 10 and 15, (e) 15 and 25, and (f) 35 and 50 have been kept from (b).

can be intuitively tuned in order to constrain the detection. In Figure 9, we study the effect of tuning the geometric model parameters on the detected spines. Figure 9(b) shows the results obtained using the parameters optimized on the training set of the LS-BSDS300 (Section VI-B). This set of parameters detects curved and local symmetries. As expected, lowering the probability of jumps κ leads to longer spines (Figure 9(d)). Straight and more global symmetries can be retrieved by tuning the spine curvature and the distance parameter d (Figure 9(g)). By also tuning the radius parameter, one can obtain a cone or a cylinder geometric model (Figure 9(h)).

The last experiment is an attempt to demonstrate the ability of the approach to only detect class-specific ribbons using local prior geometric information. This prior geometric information is directly embedded into the geometric model. Hence, only the objects satisfying this prior information are detected by the algorithm. This is illustrated in Figure 10. The top row is especially challenging as the original image contains a lot of symmetric objects with comparable structures. The pepperpot, the box, and the bottle have been detected on three independent runs of the algorithm, each one corresponding to specific constraints on the geometric model. This figure also illustrates the detection of ribbon objects from the BSDS300. Objects have been detected by tuning several parameters of the detection algorithm: the geometrical parameters $(l^m, l^M, \sigma_l, \sigma_\theta)$, and the probability of jumps κ , the latter having an impact on the detected spine lengths. The other parameters have been set identically for all the images. Namely, σ_α equals 0.4, and the particle filter stops when $t \geq 20$ and its proportion of jump on the last 200 steps reaches 6%. Moreover, detections with a spine length of less than 45 pixels have been filtered. The other parameters remained unchanged from the set of optimized parameters. For example, the doll in the first image

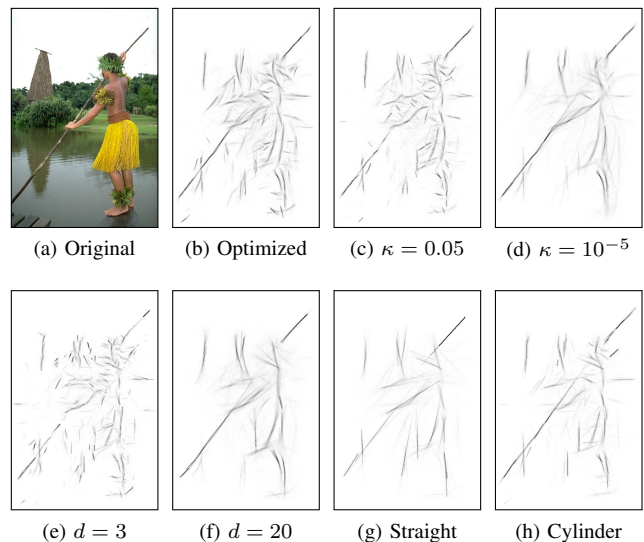


Fig. 9: Parameter influence. (b) Parameters have been optimized on the training set of the LS-BSDS300. (c)-(f) These results have been obtained by changing one parameter of (b). (g),(h) These results have been obtained by changing several parameters of (b): (g) ($d = 20, \sigma_\theta = 0$), and (h) ($\sigma_{\bar{\alpha}} = 0.04, \sigma_\theta = 0.02, \sigma_l = 0.02$).

of this set has been detected by setting the radius interval $[l^m, l^M]$ at $[25, 40]$, the standard deviation of the radius σ_l at 5, and the standard deviation of the orientation σ_θ at 0.005. Compared to filtering the posterior trajectory, constraining the geometric model to detect specific ribbons has the advantage of being computationally efficient: the detection is stopped earlier as it focuses only on the ribbons satisfying the geometric model. This experiment suggests that constraining the geometric model could be employed to detect class-specific ribbon objects. For practical purposes, these local geometric parameters could also be learned directly from a dataset of specific objects.

VII. CONCLUSION

We proposed in this article a ribbon detection algorithm using a particle filtering method. Ribbons have the particularity of exhibiting smooth local symmetries. Our contributions were threefold: the generative geometric model, which can represent a large quantity of objects; the feature extraction, which is designed for the local reflection symmetries in natural images; and the ribbon detection algorithm, which is an adaptation of the work proposed in [19], [20]. Experiments conducted on the LS-BSDS300 [30] and on ribbon detection demonstrated the ability of the approach to extract low level features (local symmetries and contours) and symmetric, ribbon-like objects. These results could be used as an input of a segmentation or class-specific object detection algorithm.

The aim of this article is to detect local symmetries using local geometric models. In contrast to a global geometric model, the use of a local geometric model is suitable to represent a large variety of ribbon objects. However, a local



Fig. 10: Ribbon detection. Contours (blue), symmetries (black), and objects (yellow). The brightness of the color indicates the probability of detection. Ribbons are detected by tuning the geometric model parameters and the prior probability of jumps.

geometric model is often poorly specific, which can be a limitation when one wants to recover complex ribbon shapes. It is also a limitation when one is only interested by the detection of global symmetries. In this case, state-of-the-art global symmetry detection algorithms proposed in [21], [26], [27] are more naturally suited.

These limitations bring several perspectives to this work, in which the geometric model could be adapted to serve different aims. First, modeling more global structural information would enable to detect class-specific ribbons. This global prior information could be, for example, structure-based [23], [24], contour-based [44]–[46], or appearance-based [47], [48]. Second, adapting the likelihood in order to integrate spatial texture information, such as it has been done in [21], [26], [27] using SIFT points, would be relevant for global symmetry detection and for 3D shape detection. In this latter case, it would also require to model skewed symmetries as a subclass of Brooks ribbons [36], as proposed in [4].

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments that greatly contributed to improve this paper.

REFERENCES

[1] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. V. Gool, "Computational symmetry in computer vision and computer graphics," *Foundations and Trends in Computer Graphics and Vision*, vol. 5, pp. 1–195, 2010. **1, 2**

[2] G. Kootstra, A. Nederveen, and B. D. Boer, "Paying attention to symmetry," in *British Machine Vision Conference*, 2008, pp. 1115–1125. **1**

[3] M. Brady and H. Asada, "Smoothed local symmetries and their implementation," *The International Journal of Robotics Research*, vol. 3, no. 3, pp. 36–61, 1984. **1, 2, 3, 4**

[4] J. Ponce, "On characterizing ribbons and finding skewed symmetries," *Computer Vision, Graphics, and Image Processing*, vol. 52, no. 3, pp. 328–340, 1990. **1, 2, 13**

[5] C. Florin, N. Paragios, and J. Williams, "Particle filters, a quasi-Monte Carlo solution for segmentation of coronaries," in *Medical Image Computing and Computer Assisted Intervention*, 2005, pp. 246–253. **1, 2**

[6] —, "Globally optimal active contours, sequential Monte Carlo and on-line learning for vessel segmentation," in *European Conference on Computer Vision*, 2006, pp. 476–489. **1, 2**

[7] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea, "Medial-based Bayesian tracking for vascular segmentation: Application to coronary arteries in 3D CT angiography," in *IEEE International Symposium on Biomedical Imaging*, 2008, pp. 268–271. **1, 2**

[8] M. Schaap, R. Manniesing, I. Smal, T. Van Walsum, A. Van Der Lugt, and W. Niessen, "Bayesian tracking of tubular structures and its application to carotid arteries in CTA," in *Medical Image Computing and Computer Assisted Intervention*, 2007, pp. 562–570. **1, 2**

[9] J.-H. Jang and K.-S. Hong, "Detection of curvilinear structures and reconstruction of their regions in gray-scale images," *Pattern Recognition*, vol. 35, no. 4, pp. 807–824, 2002. **1**

[10] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 113–125, 1998. **1, 2**

[11] X. Hu and C. Tao, "A reliable and fast ribbon road detector using profile analysis and model-based verification," *International Journal of Remote Sensing*, vol. 26, no. 5, pp. 887–902, 2005. **1**

[12] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011. **1, 2, 5**

[13] M. Mignotte, "A non-stationary MRF model for image segmentation from a soft boundary map," *Pattern Analysis and Applications*, vol. 17, no. 1, pp. 129–139, 2012. **1**

[14] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, "Shock graphs and shape matching," *International Journal of Computer Vision*, vol. 35, no. 1, pp. 13–32, 1999. **1**

[15] M. Pelillo, K. Siddiqi, and S. W. Zucker, "Matching hierarchical structures using association graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1105–1120, 1999. **1**

[16] M. Stark, M. Goesele, and B. Schiele, "A shape-based object class model for knowledge transfer," in *IEEE International Conference on Computer Vision*, 2009, pp. 373–380. **1**

[17] I. Kokkinos, P. Maragos, and A. Yuille, "Bottom-up & top-down object detection using primal sketch features and graphical models," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1893–1900. **1, 9**

[18] I. Kokkinos and A. Yuille, "Inference and learning with hierarchical shape models," *International Journal of Computer Vision*, vol. 93, no. 2, pp. 201–225, 2011. **1**

[19] N. Widynski and M. Mignotte, "A particle filter framework for contour detection," in *European Conference on Computer Vision*, vol. LNCS 7572, 2012, pp. 780–794. **2, 6, 7, 12**

[20] —, "A multiscale particle filter framework for contour detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. To Appear. **2, 6, 7, 12**

[21] S. Lee and Y. Liu, "Curved glide-reflection symmetry detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 266–278, 2012. **2, 11, 13**

[22] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002. **2**

[23] N. H. Trinh and B. B. Kimia, "Learning prototypical shapes for object categories," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2010, pp. 1–8. **2, 13**

[24] —, "Skeleton search: Category-specific object recognition and segmentation using a skeletal shape model," *International Journal of Computer Vision*, vol. 94, no. 2, pp. 215–240, 2011. **2, 13**

[25] S. Thrun and B. Wegbreit, "Shape from symmetry," in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1824–1831. **2**

[26] J. Liu and Y. Liu, "Curved reflection symmetry detection with self-validation," in *Asian Conference on Computer Vision*, 2010, pp. 102–114. **2, 11, 13**

[27] G. Loy and J.-O. Eklundh, "Detecting symmetry and symmetric constellations of features," in *European Conference on Computer Vision*, 2006, pp. 508–521. **2, 11, 13**

- [28] M. Park, S. Leey, P.-C. Cheny, S. Kashyap, A. A. Butty, and Y. Liuy, "Performance evaluation of state-of-the-art discrete symmetry detection algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8. [2](#)
- [29] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser, "A planar-reflective symmetry transform for 3D shapes," in *ACM Transactions on Graphics*, vol. 25, no. 3, 2006, pp. 549–559. [2](#)
- [30] S. Tsogkas and I. Kokkinos, "Learning-based symmetry detection in natural images," in *European Conference on Computer Vision*, vol. LNCS 7572, 2012, pp. 41–54. [2](#), [6](#), [9](#), [10](#), [11](#), [12](#)
- [31] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *IEEE International Conference on Computer Vision*, vol. 2, 2001, pp. 416–423. [2](#)
- [32] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117–156, 1998. [2](#), [9](#), [10](#), [11](#)
- [33] A. Levinstein, S. Dickinson, and C. Sminchisescu, "Multiscale symmetric part detection and grouping," in *IEEE International Conference on Computer Vision*, 2009, pp. 2162–2169. [2](#), [9](#), [10](#), [11](#)
- [34] D. Terzopoulos, A. Witkin, and M. Kass, "Symmetry-seeking models and 3D object reconstruction," *International Journal of Computer Vision*, vol. 1, no. 3, pp. 211–221, 1988. [2](#)
- [35] H. Blum, "A transformation for extracting new descriptors of shape," *Models for the perception of speech and visual form*, vol. 19, no. 5, pp. 362–380, 1967. [2](#), [3](#)
- [36] R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *Artificial intelligence*, vol. 17, no. 1, pp. 285–348, 1981. [2](#), [3](#), [13](#)
- [37] A. Rosenfeld, "Axial representations of shape," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 2, pp. 156–173, 1986. [2](#), [3](#), [7](#)
- [38] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893. [5](#)
- [39] M. Mignotte, "MDS-based multiresolution nonlinear dimensionality reduction model for color image segmentation," *Neural Networks*, vol. 22, no. 3, pp. 447–460, 2011. [5](#)
- [40] A. Desolneux, L. Moisan, and J. Morel, "Edge detection by Helmholtz principle," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 271–284, 2001. [7](#)
- [41] N. Widynski and M. Mignotte, "A Contrario edge detection with edgelets," in *IEEE International Conference on Signal and Image Processing Applications*, 2011, pp. 421–426. [7](#)
- [42] P. Pérez, A. Blake, and M. Gangnet, "Jetstream: Probabilistic contour extraction with particles," in *IEEE International Conference on Computer Vision*, vol. 2, 2001, pp. 524–531. [7](#)
- [43] A. Doucet, N. De Freitas, and N. Gordon, Eds., *Sequential Monte Carlo methods in practice*. Springer, 2001. [8](#)
- [44] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of adjacent contour segments for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 36–51, 2008. [13](#)
- [45] C. Xu, J. Liu, and X. Tang, "2D shape matching by contour flexibility," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 180–186, 2009. [13](#)
- [46] T. Ma and L. J. Latecki, "From partial shape matching through local deformation to robust global shape similarity for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1441–1448. [13](#)
- [47] T. Deselaers, B. Alexe, and V. Ferrari, "Localizing objects while learning their appearance," in *European Conference on Computer Vision*, 2010, pp. 452–466. [13](#)
- [48] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006. [13](#)



de Recherche du Centre Hospitalier de l'Université de Montréal (CRCHUM), Quebec, Canada.

Nicolas Widynski received the engineering degree from Ecole Pour l'Informatique et les Techniques Avancées and the image processing master's degree from the University Pierre and Marie Curie, both in Paris, France, in 2007. He completed his Ph.D. degree at Télécom ParisTech and Laboratoire d'Informatique de Paris 6 in 2010. He was a Postdoctoral Fellow at the Département d'Informatique et de Recherche Opérationnelle (DIRO), University of Montreal, Quebec, Canada, from 2011 to 2013. He is currently a Postdoctoral Fellow at the Centre



Antoine Moevus Antoine Moevus received a double bachelor degree in Electrical Engineering and Computer Science from cole Polytechnique de Montréal (Montréal, Canada) and École Supérieure d'Électricité (Gif-sur-Yvette, France) in 2012. He completed a Master Degree in computer vision under the supervision of Max Mignotte and Jean Meunier at the Département d'Informatique et de Recherche Opérationnelle (DIRO), University of Montreal, Quebec, Canada, in 2014.



Max Mignotte received the DEA (Postgraduate degree) in digital signal, image and speech processing from the INPG University, Grenoble, France, in 1993 and the Ph.D. degree in electronics and computer engineering from the University of Bretagne Occidentale (UBO) and the digital signal laboratory (GTS) of the French Naval Academy, France, in 1998.

He was an INRIA Post-Doctoral Fellow with University of Montreal (DIRO), Montreal, QC, Canada, from 1998 to 1999. He is currently with the Computer Vision & Geometric Modeling Lab, DIRO, University of Montreal, as an Associate Professor (Professeur agrégé). He is also a member of Laboratoire de recherche en imagerie et orthopédie, Centre de recherche du CHUM, Hôpital Notre-Dame (LIO) and Researcher at CHUM. His current research interests include statistical methods, Bayesian inference and hierarchical models for high-dimensional inverse problems such as segmentation, parameters estimation, fusion, shape recognition, deconvolution, 3-D reconstruction, and restoration problems.