



Detection of alterations in historical violins with optical monitoring

Alireza Rezaei

► To cite this version:

Alireza Rezaei. Detection of alterations in historical violins with optical monitoring. Computer Vision and Pattern Recognition [cs.CV]. Université Paris-Saclay, 2022. English. NNT: 2022UPASG054 . tel-03771343

HAL Id: tel-03771343

<https://theses.hal.science/tel-03771343>

Submitted on 7 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Detection of alterations in historical violins with optical monitoring

Détection des altérations des violons historiques par contrôle optique

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580,
Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Traitement du signal et des images
Graduate School : Informatique et sciences du numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **SATIE** (Université Paris-Saclay, ENS Paris-Saclay, CNRS), sous la direction de **Sylvie LE HEGARAT-MASCLE**, Professeur, et le co-encadrement d'**Emanuel ALDEA**, Maître de Conférences.

Thèse soutenue à Paris-Saclay, le 13 juin 2022, par

Alireza REZAEI

Composition du Jury

| | |
|--|------------------------|
| Jon Yngve HARDEBERG Professeur, Norwegian University of Science and Technology (NTNU) | Président |
| Yann GOUSSEAU Professeur, Télécom Paris | Rapporteur & Examineur |
| Olivier LALIGANT Professeur, Université de Bourgogne | Rapporteur & Examineur |
| Piercarlo DONDI Maître de Conférences, University of Pavia | Examineur |
| Sylvie LE HEGARAT-MASCLE Professeur, Université Paris-Saclay | Directrice de thèse |

Titre : Détection des altérations des violons historiques par contrôle optique

Mots clés : Détection des changements, cadre a-contrario, clustering 3D, Conservation préventive.

Résumé : La conservation préventive est le contrôle continu de l'état d'une œuvre d'art pour réduire le risque de dommages et minimiser les restaurations. De nombreuses méthodes ont été proposées pour atteindre cet objectif, soit à partir de données unimodales, soit par combinaison de différentes techniques d'analyse. Dans ce travail, nous présentons deux algorithmes probabilistes de clustering pour la détection d'altérations sur des surfaces vernies, telles que celles des instruments de musique historiques. Les deux méthodes sont reposent sur une approche a-contrario et le critère Nombre de Fausses Alarmes (NFA). La première méthode aborde le problème de la détection de changement entre une paire d'images couleur en analysant leur image de différence. Il considère simultanément l'information spectrale et spatiale avec un seul modèle de bruit.

Le deuxième méthode travaille avec une séquence d'images et analyse l'évolution de les zones altérées entre les images. Les deux méthodes sont robustes au bruit et évitent réglage des paramètres ainsi que toute hypothèse sur la forme et la taille de la modification domaines. Dans les deux cas, des tests ont été effectués sur des séquences d'images UVIFL (images de fluorescence induite par les UV) incluses dans le jeu de données "Violins UVIFL imagery". UVIFL est une technique de diagnostic bien connue, utilisée pour voir les détails d'une surface qui ne sont pas perceptibles à la lumière visible. Les résultats obtenus prouvent la capacité de l'algorithme pour détecter correctement les régions altérées. Des comparaisons avec d'autres les méthodes de clustering de pointe montrent une amélioration à la fois de la Precision et du Recall.

Title : Detection of alterations in historical violins with optical monitoring

Keywords : Change detection, a-contrario framework, 3D clustering, Preventive conservation.

Abstract : Preventive conservation is the constant monitoring of the state of conservation of an artwork to reduce the risk of damage in order to minimise the necessity of restorations. Many methods have been proposed to achieve this goal, generally including a mix of different analytical techniques. In this work, we present two probabilistic clustering algorithms for the detection of alterations on varnished surfaces, in particular those of historical musical instruments. Both methods are based on the a-contrario framework and the Number of False Alarms (NFA) criterion. The first one tackles the problem of detecting changes between a pair of colour images by analysing their difference map. It considers simultaneously grey-level and spatial density information with a single background model.

The second method works with a sequence of images and analyses the evolution of the changed areas between frames. Both methods are robust to noise and avoid parameter tuning as well as any assumption about the shape and size of the changed areas. In both cases, tests have been conducted on UV-induced fluorescence (UVIFL) image sequences included in the "Violins UVIFL imagery" dataset. UVIFL photography is a well-known diagnostic technique used to see details of a surface not perceivable with visible light. The obtained results prove the capability of the algorithm to properly detect the altered regions. Comparisons with other state-of-the-art clustering methods show improvement in both precision and recall.

Abstract

Preventive conservation is the constant monitoring of the state of conservation of an artwork to reduce the risk of damage in order to minimise the necessity of restorations. Many methods have been proposed to achieve this goal, generally including a mix of different analytical techniques. In this work, we present two probabilistic clustering algorithms for the detection of alterations on varnished surfaces, in particular those of historical musical instruments. Both methods are based on the a-contrario framework and the Number of False Alarms (NFA) criterion. The first one tackles the problem of detecting change between a pair of colour images by analysing their difference map. It considers simultaneously grey-level and spatial density information with a single background model. The second method works with a sequence of images and analyses the evolution of the changed areas between frames. Both methods are robust to noise and avoid parameter tuning as well as any assumption about the shape and size of the changed areas. In both cases, tests have been conducted on UV induced fluorescence (UVIFL) image sequences included in the “Violins UVIFL imagery” dataset. UVIFL photography is a well known diagnostic technique used to see details of a surface not perceivable with visible light. The obtained results prove the capability of the algorithm to properly detect the altered regions. Comparisons with other the state-of-the-art clustering methods show improvement in both precision and recall.

Résumé

La conservation préventive est le contrôle continu de l'état d'une œuvre d'art pour réduire le risque de dommages et minimiser les restaurations. De nombreuses méthodes ont été proposées pour atteindre cet objectif, soit à partir de données unimodales, soit par combinaison de différentes techniques d'analyse. Dans ce travail, nous présentons deux algorithmes probabilistes de clustering pour la détection d'altérations sur des surfaces vernies, telles que celles des instruments de musique historiques. Les deux méthodes sont reposent sur une approche a-contrario et le critère Nombre de Fausses Alarmes (NFA). La première méthode aborde le problème de la détection de changement entre une paire d'images couleur en analysant leur image de différence. Il considère simultanément l'information spectrale et spatiale avec un seul modèle de bruit. Le deuxième méthode travaille avec une séquence d'images et analyse l'évolution de les zones altérées entre les images. Les deux méthodes sont robustes au bruit et évitent réglage des paramètres ainsi que toute hypothèse sur la forme et la taille de la modification domaines. Dans les deux cas, des tests ont été effectués sur des séquences d'images UVIFL (images de fluorescence induite par les UV) incluses dans le jeu de données "Violins UVIFL imagery". UVIFL est une technique de diagnostic bien connue, utilisée pour voir les détails d'une surface qui ne sont pas perceptibles à la lumière visible. Les résultats obtenus prouvent la capacité de l'algorithme pour détecter correctement les régions altérées. Des comparaisons avec d'autres les méthodes de clustering de pointe montrent une amélioration à la fois de la Precision et du Recall.

Synthèse de la thèse

L'étude actuelle a été réalisée en collaboration avec le laboratoire Arvedi de diagnostic non invasif de l'université de Pavie en Italie. L'objectif principal était d'utiliser la vision par ordinateur pour surveiller les violons historiques et détecter tout dommage croissant sur leur surface. Le Museo del Violino de Crémone, en Italie, abrite plusieurs violons historiques de premier plan à différents stades de conservation. Tout type d'utilisation use progressivement la surface des violons, enlevant la couche protectrice de l'instrument et exposant la couche de bois à l'air. Les études et les efforts visant à détecter ces parties endommagées le plus tôt possible relèvent de la "conservation préventive".

La conservation préventive est une procédure cruciale dans le domaine du patrimoine culturel et consiste, en général, à surveiller les œuvres d'art et les monuments afin de minimiser les restaurations dont ils font l'objet. Cette pratique est particulièrement complexe et nécessite une approche interdisciplinaire pour interpréter correctement et gérer les effets des altérations chimiques, physiques et biologiques.

Les instruments de musique historiques en bois (tels que les violons ou les altos) sont des œuvres d'art particulières, principalement parce qu'ils sont à la fois conservés dans des musées et joués lors de divers événements. Cela entraîne un risque important d'usure mécanique dans les zones en contact direct avec le corps des musiciens. Des travaux antérieurs ont proposé de multiples techniques analytiques pour aborder la surveillance de ces instruments. Mais, bien qu'elles soient assez précises, elles prennent souvent plus de temps que souhaité. Une procédure plus efficace en termes de temps consistera à analyser régulièrement les images pour identifier rapidement les éventuelles zones altérées, puis à appliquer des techniques spectroscopiques à titre de confirmation.

Pour un contrôle purement optique, nous utilisons des images de fluorescence induite par les UV (UVIFL) prises sur des surfaces en bois présentant une usure croissante. L'idée principale est de tirer parti de la différence entre l'effet de réémission des zones usées et vernies. En outre, contrairement aux images ordinaires en lumière visible, les images UVIFL cachent les altérations superficielles telles que les empreintes digitales et la poussière.

Le processus général de surveillance consiste à capturer régulièrement des images UVIFL pendant une période prolongée au cours de laquelle l'échantillon est susceptible d'être endommagé. Une image originale est prise au début pour enregistrer l'état initial de l'échantillon. Les images ultérieures sont comparées à cette image originale à l'aide d'un algorithme de détection des changements. Toute zone significativement modifiée (jugée comme n'étant pas du bruit ou un artefact) soulignera une possible usure émergente ou croissante.

Dans les images UVIFL, le bruit peut provenir de différentes sources, mais principalement de la réflectance du vernis et aussi d'une erreur d'enregistrement (pour éviter d'endommager le vernis, les violons ne peuvent pas être fixés de manière rigide à une structure de support). De plus, l'usure du vernis peut évoluer de différentes manières en fonction des conditions initiales de la surface et des différentes substances présentes. Cela entraîne plusieurs limitations pour l'algorithme de détection des changements : le nombre de régions modifiées n'est pas connu à l'avance; les régions modifiées peuvent prendre n'importe quelle forme ou taille; et, le bruit et les artefacts doivent être différenciés des zones modifiées. Par conséquent, pour contrôler la qualité de la surface en bois des violons, nous avons besoin d'une procédure de détection rapide des changements qui soit robuste au bruit et qui suppose aussi peu d'informations préalables que possible.

Une comparaison pixel par pixel entre une image donnée de la séquence et l'image originale nous donne une carte de différence en niveau de gris. Cette thèse se concentre sur l'analyse de la carte des différences produite afin de trouver et de regrouper les régions de changement possible, d'attribuer des scores relatifs à chacune d'elles et enfin de suivre leur évolution dans la séquence. Des comparaisons qualitatives et quantitatives ont été faites avec les méthodes de clustering existantes.

Notre travail, présenté dans les chapitres suivants, est principalement basé sur le cadre a-contrario proposé par Desolneux pour trouver des structures significatives dans une image numérique. Le cadre a-contrario est basé sur les lois perceptives de la vision humaine et sur le fait que nous ne percevons aucune structure dans une image avec des valeurs purement aléatoires. La signification de toute structure peut donc être déterminée par la probabilité que cette structure ne soit pas le fruit du hasard.

En résumé, nous apportons plusieurs contributions au problème de la détection de l'usure dans la surveillance optique : Premièrement, englober dans un seul modèle les critères spatiaux et radiométriques que présentent les zones changées afin de comparer une paire d'images UVIFL. Ensuite, proposer un processus de décision basé sur la signification permet de s'affranchir des seuils et des paramètres caractérisant les zones modifiées (forme, nombre, position, etc.). Enfin, proposer

une méthode permettant de prendre en compte l'information temporelle présente dans les séquences d'images UVIFL pour différencier les artefacts statiques des régions d'usure croissantes.

Contents

| | |
|--|-------------|
| Contents | xi |
| List of Figures | xiii |
| List of Tables | xvii |
| 1 Optical monitoring of historic musical instruments | 5 |
| 1.1 Preventive conservation | 5 |
| 1.2 UV Induced Fluorescence (UVIFL) Photography | 9 |
| 1.3 Violins UVIFL imagery dataset | 10 |
| 1.4 Notations and basic assumptions | 15 |
| 1.5 Data pre-processing | 16 |
| 1.6 Difference Map | 18 |
| 1.7 Conclusion | 19 |
| 2 Data clustering algorithms | 21 |
| 2.1 Hierarchical clustering | 22 |
| 2.2 Partition-based clustering | 24 |
| 2.3 Density-based clustering | 26 |
| 2.4 Distribution-based clustering | 27 |
| 2.5 Fuzzy theory based clustering | 29 |
| 2.6 Dimensionality reduction and feature transformation | 30 |
| 2.7 Evaluation metrics | 37 |
| 2.8 Conclusion | 38 |
| 3 A-contrario framework and the number of false alarms | 41 |
| 3.1 Gestalt theory | 41 |
| 3.2 The Helmholtz principle | 46 |
| 3.3 The number of false alarms | 47 |
| 3.4 Applications of the a-contrario framework in computer vision | 48 |
| 3.5 Change detection using the a-contrario framework | 49 |
| 3.6 Conclusion | 52 |

| | | |
|----------|--|-----------|
| 4 | A-contrario framework for cluster detection | 55 |
| 4.1 | Clustering in one step | 56 |
| 4.2 | Robustness evaluation using simulated data | 65 |
| 4.3 | Performance on actual data | 68 |
| 4.4 | Conclusion | 76 |
| 5 | Analysing a multi-temporal image sequence | 77 |
| 5.1 | From input data to 3D point cloud | 77 |
| 5.2 | Clustering the 3D point cloud | 79 |
| 5.3 | Changed area ranking | 81 |
| 5.4 | Experiments and the benefits of the multitemporal aspect | 82 |
| 5.5 | Comparative performance evaluation | 87 |
| 5.6 | Conclusion | 90 |
| | Conclusion and future work | 92 |

List of Figures

| | | |
|------|--|----|
| 1 | Museo del Violino in Cremona, Italy houses many historical violins in need of monitoring. | 2 |
| 1.1 | A typical monitoring plan using multiple techniques in short-term (STM) or long-term (LTM) [32]. | 7 |
| 1.2 | An example of (a) XRF, (b) FTIR, and (c) colourimetry analysis [32]. . . | 8 |
| 1.3 | Example of alteration growing on the top left side of a sample violin back plate (real alterations are highlighted in red, noisy reflections in green): (a) the initial state, reference UVIFL image; (b) same region with some alterations in an early stage, i.e. having limited colour variations that can be confused with noise; (c) same region with a large alteration in an advanced stage exhibiting clear variations in both shape and colour with respect to the reference image. | 9 |
| 1.4 | Cloths dampened with alcohol for each step of the wearing process. . . | 11 |
| 1.5 | The WS01 sequence. | 12 |
| 1.6 | The WS02 sequence. | 13 |
| 1.7 | The SV01 sequence. | 13 |
| 1.8 | The SV02 sequence. | 14 |
| 1.9 | An example of spatial registration: (a) matched SIFT feature points; (b) difference of two frames before registration; (c) difference of two frames after registration. Brighter locations indicate higher differences. . . | 17 |
| 1.10 | An example of reflection removal. (a) the sample frame (b) the produced mask showing the UV artefact pixels. | 18 |
| 1.11 | An example of colour difference map; (a) reference frame I_0 , (b) frame I_{27} , (c) the difference map produced from CIEDE2000 [58]. | 19 |
| 2.1 | (a) Density distribution of a sample data set with a high threshold and (b) its resulting clustering into two clusters (blue and green) and noise (red) [13]. | 27 |
| 2.2 | Architecture of a typical clustering based autoencoder [63]. | 35 |
| 2.3 | General architecture of CDNN-based clustering methods [63]. The network architecture varies between algorithms. | 36 |

| | | |
|-----|--|----|
| 3.1 | Gestalt's grouping laws [23]: (a) the colour constancy law means we see one single black object instead of many connected ones; (b) with the vicinity law we group these objects into two higher level visual objects; (c) we separate this circular area into two regions with different textures according to the similarity law; (d) we perceive a single object against the white background according to the closure law; (e) dark objects are perceived as a curve with the good continuation law; (f) the butterfly shaped dark objects on the left are covered with white rectangles on the right, they are now perceived as disks half occluded by the rectangles, in line with the amodal completion law; (g) we perceive the two parallel curves as the edges of an arm-shaped object with a constant width; (h) the dark objects are symmetrical with respect to a vertical line and perceived together as one object according to the symmetry law; (i) we can interpret the shapes as white ovals on black background or black triangles on white background, the convexity law favours the first option; (j) with the perspective law, we perceive this shape as a 3D object with point d as a vanishing point. | 44 |
| 3.2 | Two examples of grouping laws giving rise to different interpretations [23]: (a) the white dots are perceived, simultaneously, as a part of the grid and as a part of a curve; (b) two incompatible interpretations: the shape to the left can be perceived as two overlapping shapes or merge of two symmetrical shapes given on the right side. | 45 |
| 3.3 | An example of Helmholtz principle in action [23]: A set of four aligned line segments exists in both (a) and (b); however it can only be perceived in image (b). | 46 |
| 4.1 | Comparison for $\tau = 3.0$ between (a) $f_1(x) = \frac{1}{x-\tau}$ and (b) $f_2(x) = 1 + \tanh(\tau - x)$ | 57 |
| 4.2 | 3D point cloud (a) before applying the function f , (b) after applying $f_1(x) = \frac{1}{x-\tau}$ and (c) after applying $f_2(x) = 1 + \tanh(\tau - x)$. The vertical axis represents the (transformed) grey-level values while the other two axis originate from the 2D image plane. | 57 |
| 4.3 | Detection of a single cluster when (a) $c = 0.001$, (b) $c = 0.1$, (c) $c = 1$, (d) $c = 10$ and (e) $c = 500$; with the meaningfulness values of (a) 1228.90, (b) 466.06, (c) 161.74, (d) 7.35, (e) 1.62. The blue circle and the red number indicate the detection of only one cluster. | 62 |
| 4.4 | Experiment 1: a) histogram of the foreground (grey) and background (black) in the first step and b) the last step. | 66 |

| | | |
|------|---|----|
| 4.5 | Experiment 2: a) histogram of the foreground (black) and background (grey) in the first step and b) the last step. | 66 |
| 4.6 | The F-score for the results of the algorithm with different (a) spread for the background (Experiment 1) and (b) mean for the foreground (Experiment 2). | 67 |
| 4.7 | Experiment 1: the detected clusters from (a) to (f) in steps 1,8,12,17,21 and 27. (a) shows the perfect segmentation. | 67 |
| 4.8 | Experiment 2: the detected clusters from (a) to (f) in steps 1,2,3,6,9 and 14. (f) shows the perfect segmentation. | 67 |
| 4.9 | Clustering output from frames 3, 9, 15 and 20 of set WS01 using the proposed NFA clustering (Algorithm 2). | 69 |
| 4.10 | F-score values generated for each frame of the sequence WS01 using different algorithms. | 70 |
| 4.11 | Clustering result of several chosen algorithms performed on the frame 12 of sequence WS01. | 74 |
| 4.12 | Precision-Recall plot for WS01 (a), WS02 (b) and SV01 (c). For a given algorithm (indicated by the colour), each point highlights the performance at a specific time-step of the sequence. | 75 |
| 5.1 | Point clouds \mathcal{P} derived from the sequences WS01 (left), SV01 (middle) and SV02 (right). X and Y axes are in pixels, while the time dimension (Z axis) depends on the factor c_t (in these experiments $c_t = 2$). | 82 |
| 5.2 | Evolution of the detection by using the later time frames t in the sequence WS01. Colour code gives the rank according to significance: red first, green second, blue third. The time domain is the upward axis. | 83 |
| 5.3 | Evolution of the detection by using the later time frames t in the sequence SV01. Same conventions as Figure 5.2. | 84 |
| 5.4 | Evolution of the detection by using the later time frames t in the sequence SV02. Same conventions as Figure 5.2. | 85 |
| 5.5 | The evolution of the meaningfulness value of sample clusters in different runs of the algorithm for sequence WS01 (a), SV01 (b) and SV02 (c). In each run, at most the last 10 frames have been used. | 86 |
| 5.6 | Comparison, on sample frames of each sequence, between a wear detection performed considering only two frames at a time and the multi-temporal approach: first row, original UVIFL image; second row, detected clusters as in Chapter 4; third row, detected clusters applying the multi-temporal analysis; fourth row, ground truth showing the actual worn-out regions. | 89 |

List of Tables

| | | |
|-----|--|----|
| 2.1 | Several linkage definitions for agglomerative hierarchical clustering. . | 23 |
| 2.2 | Clustering evaluation indicators [105]. | 38 |
| 4.1 | Average and standard deviation of F-score values for different clustering algorithms on Seq. WS01, WS02 and SV01. Best results are in bold, second best results are underlined. | 71 |
| 4.2 | Comparison between the proposed NFA clustering, Dondi et al. [27], FRFCM+HDBSCAN clustering and the ground truth for some sample frames from set WS01. | 72 |
| 4.3 | Comparison between the proposed NFA clustering, Dondi et al. [27], FRFCM+HDBSCAN clustering and the ground truth for some sample frames from set WS02. | 73 |
| 4.4 | Comparison between the proposed NFA clustering, Dondi et al. [27], FRFCM+HDBSCAN clustering and the ground truth for some sample frames from set SV01. | 73 |
| 5.1 | The average and standard deviation of F-score values for the 3D clustering of each sequence using the proposed algorithm as well as six other clustering methods. First and second-best results are highlighted in green and light green respectively. | 88 |

Introduction

Overview

The current study has been performed in collaboration with the Arvedi Laboratory of Non-Invasive Diagnostics, University of Pavia in Italy. The main objective was set to using computer vision to monitor the historical violins and detect any growing damage on their surface. Museo del Violino in Cremona, Italy houses several high profile historical violins in different stages of preservation. Some examples are shown in Figure 1. Any kind of usage gradually wears the surface of the violins removing the protective layer of the instrument and exposing the wooden layer to air. Studies and efforts to detect these damaged parts as soon as possible fall under the term “preventive conservation”.

Preventive conservation is a crucial procedure in cultural heritage and in general, consists of the monitoring of artworks and monuments to minimise restorations on them [9, 57]. This practice is particularly complex and requires an interdisciplinary approach to correctly interpret and to manage the effects of chemical, physical and biological alterations [39, 70].

Specifically, historical wooden musical instruments (such as violins or violas) are special case artworks; mainly, because they are both held in museums and played in various events. This leads to a major risk of mechanical wear in the areas in direct contact with the musicians’ bodies. Previous works have proposed multiple analytical techniques to tackle the monitoring of these instruments [32, 79]. But, although they are quite accurate, they are often more time consuming than desired. A more time efficient procedure will consist in regular analysis of the images to quickly identify possible altered areas, followed by applying spectroscopic techniques as confirmation.

For a purely optical monitoring, we use UV induced fluorescence (UVIFL) images taken from wooden surfaces containing a growing wear. The main idea is to capitalise on the difference between the re-emission effect of worn-out and varnished areas. In addition, as opposed to regular visible light images, UVIFL images hide superficial alterations such as finger prints and dust.

The general monitoring process is to capture UVIFL images regularly during



Figure 1: Museo del Violino in Cremona, Italy houses many historical violins in need of monitoring.

an extended period of time in which the sample is prone to damage. An original image is taken in the beginning to record the initial state of the sample. Subsequent images are compared to this original image using a change detection algorithm. Any significant changed area (judged not to be noise or artefact) will underline a possible emerging or growing wear.

In UVIFL images, noise may be produced from different sources, but mainly reflectance from the varnish and also registration error (to avoid damage to the varnish, violins cannot be rigidly fixed to a support). Moreover, varnish wear can evolve in different ways depending on the initial conditions of the surface and on the different substances present. This produces several limitations for the change detection algorithm:

- Number of changed regions is not known before hand.
- Changed regions can assume any shape, form or size.
- Noise and artefacts have to be differentiated from changed areas.

Therefore, to monitor the quality of the wooden surface of the violins, we are in need of a fast change detection procedure which is robust to noise and which assumes as little prior information as possible.

A pixel by pixel comparison between a given frame in the sequence and the original image gives us a difference map in grey-level. This thesis is focused on analysing the produced difference map in order to find and cluster regions of

possible change, assign relative scores to each and finally follow their evolution through the sequence. Qualitative and quantitative comparisons have been made to the existing clustering methods.

Our work, presented in the following chapters, is mainly based on the a-contrario framework proposed by Desolneux [21] to find meaningful structures in a digital image. The a-contrario framework is based on perceptual laws of human vision and the fact that we do not perceive any structures in an image with purely random values. The significance of any structure, therefore, can be determined by how unlikely it is for that structure to happen by chance.

In short, we make several contributions to the problem of wear detection in optical monitoring:

- Encompassing in a single model both spatial and radiometric criteria that changed areas present in order to compare a pair of UVIFL images.
- Proposing a significance based decision process allowing us to be free from thresholds and parameters characterising the changed regions (shape, number, position etc.).
- Proposing a method to take into account the temporal information present in UVIFL image sequences to differentiate between static artefacts and growing wear regions.

In the following, we present a brief introduction of the problem domain in Chapter 1, a survey of existing clustering algorithms in Chapter 2, and a theoretical overview of the a-contrario framework in Chapter 3. Then, in Chapter 4 we introduce our change detection and clustering process between a pair of colour images. Finally, Chapter 5 contains our proposition for detecting growing changes across a multi-modal image sequence.

Publications

This is the list of the publications done during this study:

- **3D clustering for detection of alterations in multi-temporal images of historical violins**, Revision in progress, ACM Journal on Computing and Cultural Heritage, 2021, Alireza Rezaei, Sylvie Le Hégarat-Masclé, Emanuel Aldea, Piercarlo Dondi and Marco Malagodi
- **A-contrario framework for detection of alterations in varnished surfaces**, Journal of Visual Communication and Image Representation, 2021, Alireza Rezaei, Sylvie Le Hégarat-Masclé, Emanuel Aldea, Piercarlo Dondi and Marco Malagodi

- **Analysis of multi-temporal image series for the preventive conservation of varnished wooden surfaces**, International Symposium on Visual Computing, 2021, Alireza Rezaei, Sylvie Le Hégarat-Masclé, Emanuel Aldea, Piercarlo Dondi and Marco Malagodi
- **One step clustering based on a-contrario framework for detection of alterations in historical violins**, International Conference on Pattern Recognition, 2020, Alireza Rezaei, Sylvie Le Hégarat-Masclé, Emanuel Aldea, Piercarlo Dondi and Marco Malagodi
- **Detecting alterations in historical violins with optical monitoring**, Quality Control by Artificial Vision, 2019, Alireza Rezaei, Emanuel Aldea, Piercarlo Dondi, Marco Malagodi and Sylvie Le Hégarat-Masclé

Chapter 1

Optical monitoring of historic musical instruments

Contents

| | |
|--|-----------|
| 1.1 Preventive conservation | 5 |
| 1.2 UV Induced Fluorescence (UVIFL) Photography | 9 |
| 1.3 Violins UVIFL imagery dataset | 10 |
| 1.4 Notations and basic assumptions | 15 |
| 1.5 Data pre-processing | 16 |
| 1.6 Difference Map | 18 |
| 1.7 Conclusion | 19 |

1.1 Preventive conservation

The main aim of preventive conservation procedures is to reduce the risk of alteration of historical artefacts and to avoid the natural ageing of materials (such as pigments, organic binders, or protective layers), with the consequence of a general reduction of the need to carry out operations of restoration. A preventive conservation plan involves several different operations including the study of the environmental areas of buildings, micro-climatic conditions, materials, airborne pollutants, and even the effect of the visitors in the museum [99].

Preventive conservation procedures have been carried out since 1980s, and thus, nowadays, there is an extensive scientific literature concerning the evaluation of the indoor micro-climatic conditions in museums, ancient palaces, or depositories [9, 86, 89], and it is now well known that the main factors that can accelerate the degradation processes of artworks are temperature, dampness, relative humidity,

and pollutants. Bad conservative conditions can increase the risk of damages to the materials of the exposed artworks, leading, for example, to chromatic variations, modifications in the organic structures of the binders or mechanical alterations of the stratigraphic system.

In the last years, the research for preventive conservation plans has aimed to develop new standards and protocols to control the indoor environmental parameters and, at the same time, to check day by day the conservation state of artworks and of their original materials [18, 50, 70].

During the centuries, a lot of cultural assets, such as furniture or archaeological finds, have lost their original employment and today they are preserved as historic and artistic exemplars from the past centuries. Nowadays, those kinds of objects are considered at the same level of other more “standard” artworks, like paintings or statues, and thus they are included in preventive conservation plans. However, in some cases, the use of those objects can still be performed, as for musical instruments in general and violins specifically. Musical instruments represent a particular class of artworks, on which conservation is not limited to their materials but also concerns the preservation of their acoustic quality [11, 34]. The regular use of these instruments brings about new problems concerning the conservation of materials:

- Dirt deposits on the musical instruments, depending on the climatic conditions such as the presence of pollutants, the dirt particle size, or the varnish surface roughness.
- Direct contact with the player that may prompt dirt deposition and adhesion on the surface, due to the exposure to bad conditions during the performance.
- Varnish wear due to direct contact with the violinist’s skin which contains sweat and acid compounds.

Furthermore, the analysis of their surface is particularly challenging. First of all, the varnishes are generally highly reflective, thus, noisy reflections, that can be confused for alterations, are common during photo acquisition. Secondly, varnish wear can evolve in different ways depending on both the initial conditions of the surface and on the different substances present. Finally, the surface can be very complex and stratified due to multiple restorations having occurred during centuries (which is very common for historical violins).

As a result of these conditions, it became important to develop innovative non-invasive diagnostic procedures to monitor material transformation and to check the conservation state of the musical instruments, especially regarding the wearing of the varnishes. The combined use of multiple analytical techniques

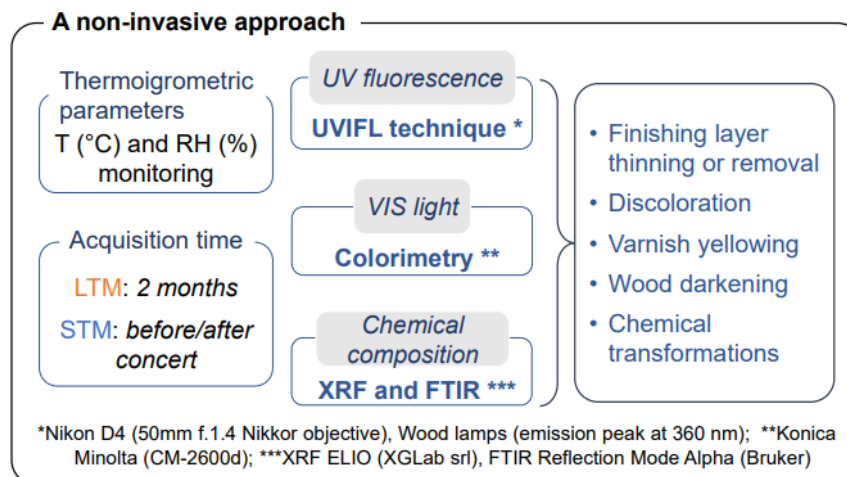


Figure 1.1: A typical monitoring plan using multiple techniques in short-term (STM) or long-term (LTM) [32].

has been proved beneficial for handling of these issues [32, 79]. For example, a non-invasive monitoring process proposed in [32] consists of three different techniques (illustrated in Figure 1.1) performed in a long-term format (every two months) or short-term (before/after a concert). These methods include:

- *UV fluorescence imaging.* In this technique, the sample is illuminated by UV light and the re-emission is captured by a RGB camera. Section 1.2 goes into more details about this method.
- *Colourimetry.* This analysis is done by measuring the colour difference (ΔE) of selected group of points between two different times using for example a portable spectrophotometre. The colour data is gathered usually in the $L^*a^*b^*$ colour space and the differences are calculated using a preferred formula. High difference values may indicate change in those specific points [32]. Figure 1.2c illustrates colour difference values for six different spots on the surface of a historical violin when measured before and after a concert.
- *Chemical composition analysis (XRF and FTIR).* X-ray fluorescence (XRF) is an analytical technique used to determine the elemental composition of materials. Comparison of XRF findings in different times can give us an indication of the changes the material has gone through (Figure 1.2a).

Fourier transform infrared spectroscopy (FTIR) is a method used to acquire an infrared (IR) spectrum of emission and absorption of the sample. First, the IR radiation is passed through the sample and then, the detector picks up the resulting signal. The absorbed and transmitted portions of the radiation depend on the chemical composition of the sample. Studying the variations of

the output spectrum between different times is a reliable way to detect surface change on the sample (Figure 1.2b).

While chemical analysis gives us more exact information about the variations of the surface, it is not always easy to interpret those findings specially because each method is strong in some areas but weak in others.

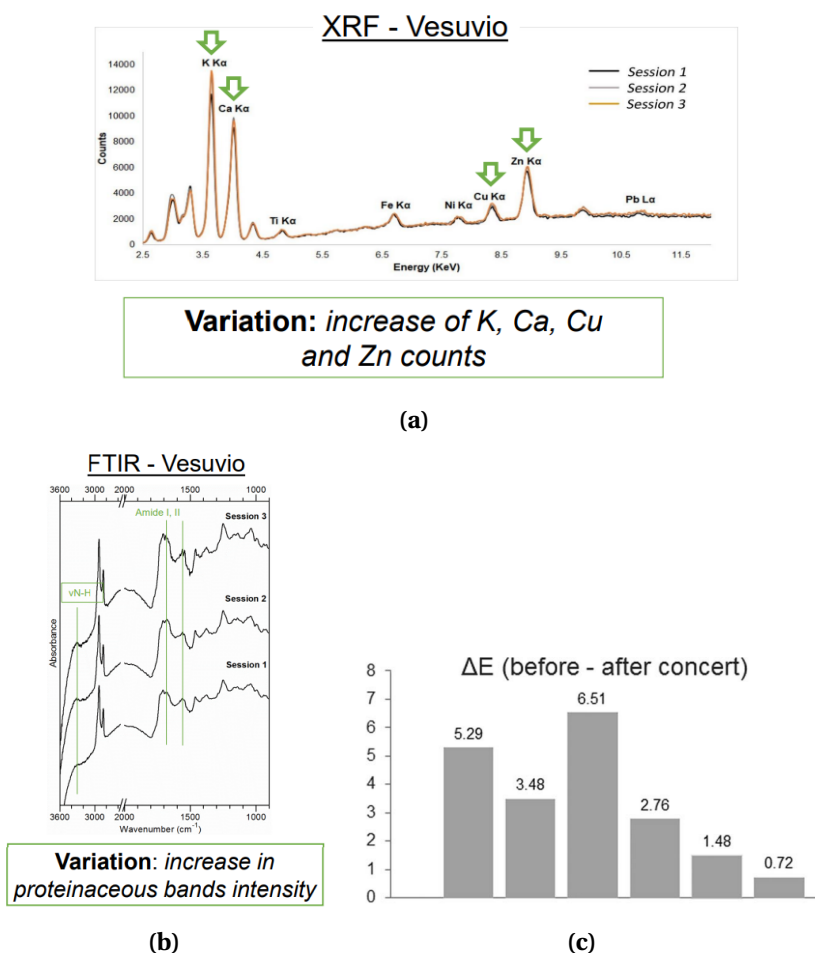


Figure 1.2: An example of (a) XRF, (b) FTIR, and (c) colourimetry analysis [32].

However, this preventive conservation approach is very time consuming and can become tedious and prone to human errors if multiple instruments are being monitored at the same time. A more efficient procedure should consist of regular but rapid optical analyses of images of instruments to quickly identify the *possible* altered areas. The result of this analysis will indicate where a more thorough multi-modal analysis (i.e., the application of spectroscopic examinations) is needed. Image acquisition is very fast compared to other chemical-physical examinations, thus, ideally, using image processing, it would be possible to frequently examine in a limited time the state of conservation of one or more

artworks, focusing the human attention only on those that are genuinely at risk.

1.2 UV Induced Fluorescence (UVIFL) Photography

In the Cultural Heritage field, UVIFL photography is one of the most commonly employed non-invasive diagnostic techniques. It is based on the properties of some organic substances that react to UV radiation (generally in the UV-A range, between 315 and 400 nm), re-emitting radiations in the visible spectrum (400 - 700 nm) [46, 87]. This process allows us to see details not visible with a standard illumination, such as, in the case of historical violins, substances commonly adopted as binders, pigments, adhesives, or material used for retouching [10].

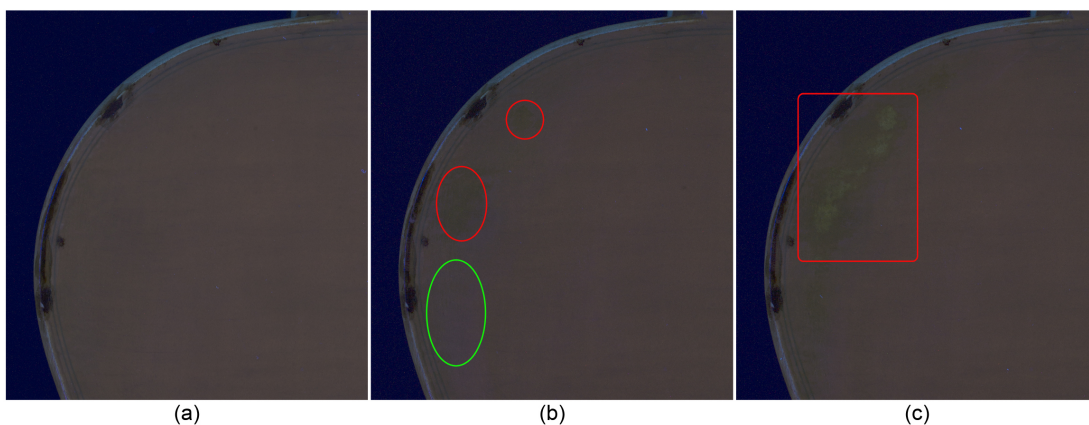


Figure 1.3: Example of alteration growing on the top left side of a sample violin back plate (real alterations are highlighted in red, noisy reflections in green): (a) the initial state, reference UVIFL image; (b) same region with some alterations in an early stage, i.e. having limited colour variations that can be confused with noise; (c) same region with a large alteration in an advanced stage exhibiting clear variations in both shape and colour with respect to the reference image.

However, even if fluorescent materials produce characteristic colour responses (e.g., a yellow fluorescence can be attributed to oils and a green one to protein substances such as hide glues [8]), the fluorescence phenomenon alone is not sufficient to unambiguously identify materials. In fact, the varnishes adopted in violin making include very heterogeneous mixtures of substances, that, added to the various restorations occurred during centuries, create very complex emission spectra. Hence, there is not a unique colour-substance match. Moreover, UV radiation penetrates only the superficial layers of varnishes, thus the underlying ones cannot be detected by this technique. Given these limitations, UVIFL images are mainly employed as a preliminary analysis [26] to spot some regions of interest where to apply more precise, but slower, spectroscopic techniques, such as XRF [80], or FTIR spectroscopy [44], to fully characterise the materials present.

In the current scenario of a constant monitoring process, UVIFL photography can be particularly helpful to spot new superficial alterations, since, when the wear removes the outer layers of varnishes, the lower layers start becoming visible producing slightly different fluorescence colours. Thus, the occurrence of a colour variation in a region while the surroundings remain unchanged is a clear hint of a possible alteration. Of course, it is important to take into account the possible presence of noisy reflections (due to the high reflective varnishes), that can sometimes occur even with a rigorous acquisition process and that can be mistakenly interpreted as alterations, especially in the early stages, as illustrated on Figure 1.3.

1.3 Violins UVIFL imagery dataset

In this study, we use UV induced fluorescence (UVF or UVIFL) images collected in the “Violins UVIFL imagery” dataset¹ [27]. This dataset contains UVIFL images of both historical and sample violins. Regular acquisitions have been performed on two historical violins held in Museo del Violino in Cremona (Italy), “Carlo IX” (c.1566) made by Andrea Amati and “Vesuvio” (1727) made by Antonio Stradivari. However, they did not show any new wear areas (only “Vesuvio” showed a very slight alteration on its back plate). Thus, for the wear monitoring purpose, we considered four artificially created sample sequences containing images of artificially altered samples for the study of various possible alterations over a long-term use.

The alterations were created scrubbing the surface with a cloth damped with alcohol to reproduce, as faithfully as possible, the effect of mechanical wear during playing (Figure 1.4). The alteration process was repeated multiple times. At each step we took (at least) three photos of the samples, for safety, to exclude errors due to accidental wrong acquisitions.

The first artificial sequence, called WS01 and shown on Figure 1.5, is a wood sample which simulates an alteration in an area with intact varnish. This set contains one reference image of the initial state of the sample and 20 altered frames.

The second artificial sequence, called WS02 (cf. Figure 1.6), is a wood sample which simulates an alteration in an area with a thin layer of varnish. This set contains one reference image of the initial state of the sample and 8 altered frames.

The third sequence, called SV01 (cf. Figure 1.7), contains images of the lower part of the back plate of a sample violin. This set simulates the growing of wear starting from an area already ruined and consists of one reference image and 20 altered frames.

The fourth sequence, called SV02 (cf. Figure 1.8), contains images of the top left

¹<https://vision.unipv.it/research/UVIFL-Dataset/>

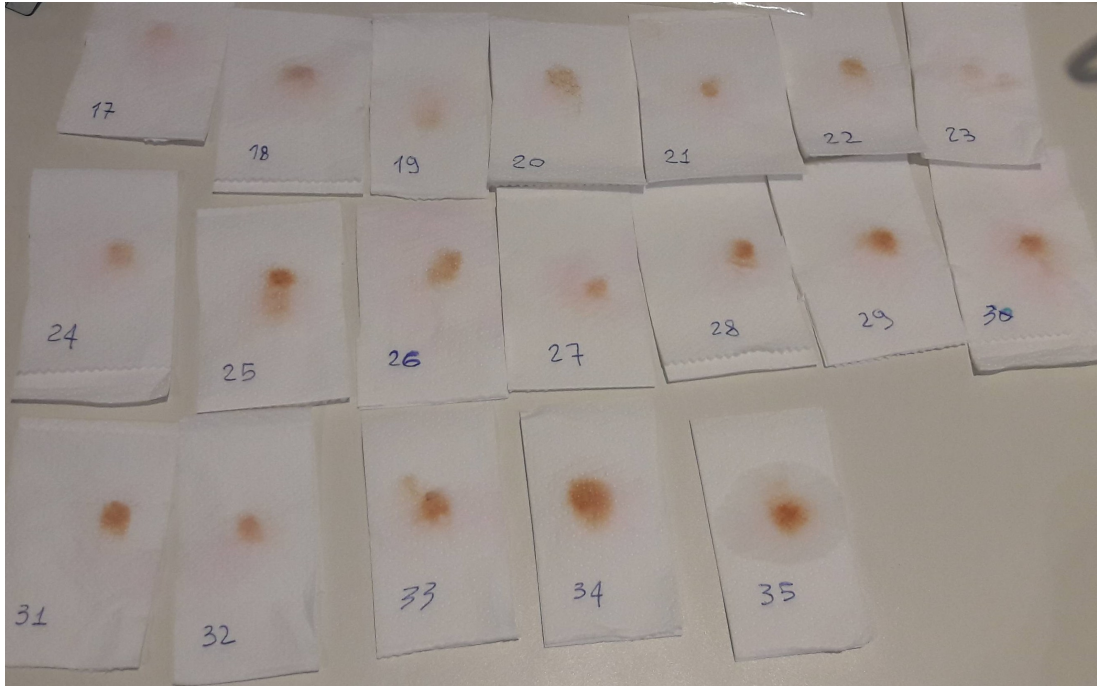


Figure 1.4: Cloths dampened with alcohol for each step of the wearing process.

part of the back plate of a sample violin. This set simulates a very slow alteration on a region with a thick layer of intact varnish. This sequence contains one reference image and 35 altered frames.

All the images were acquired following a rigorous acquisition protocol designed to minimise, as much as possible, the presence of ambient noise [28, 29]. The wood samples were placed on a small support to maintain them stable during the shot, while the instruments were placed on an ad-hoc rotating platform that allow us to move them precisely at the needed angle. The photos were taken with a Nikon D4 full-frame digital camera with a 50 mm $f/1.4$ Nikkor objective, 30s exposure time, aperture $f/8$, ISO 400. We used two wooden lamp tubes (Philips TL-D 36 W BBL IPP low-pressure Hg tubes, 40 Watt, emission peak $\sim 365nm$) as UV-A lighting source. The lamps were oriented at 45 degrees to uniformly illuminate all the surface of the samples/instruments. Note that, even with such a rigorous acquisition protocol, some noise can still occur, especially in the most rounded part of the violins.

Finally, to make the evaluation of the algorithm possible, the ground truth for each frame has been created manually from careful visual inspection. In the absence of any exact (due intrinsically to the wear construction process) knowledge on the boundaries of wear regions, they have been intentionally overestimated. In other words, the produced ground truth masks can look different if done by another expert; so, a loose interpretation of the wear boundaries is necessary.

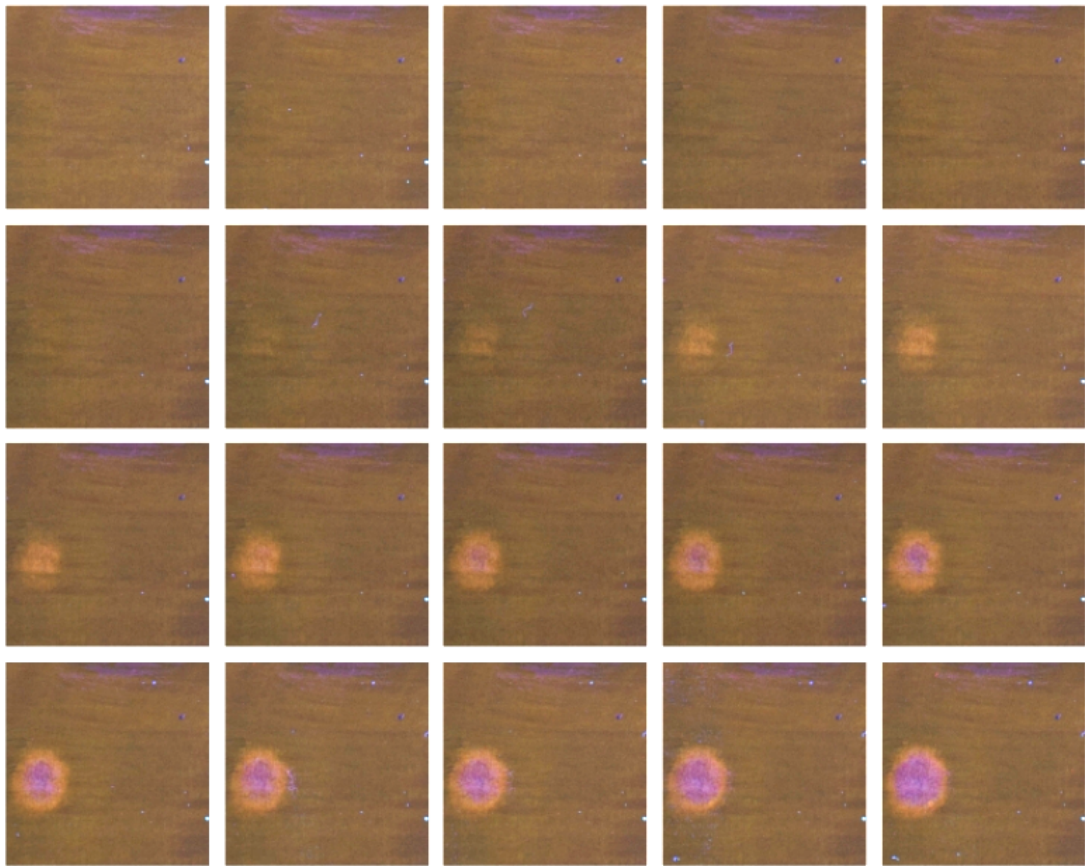


Figure 1.5: The WS01 sequence.

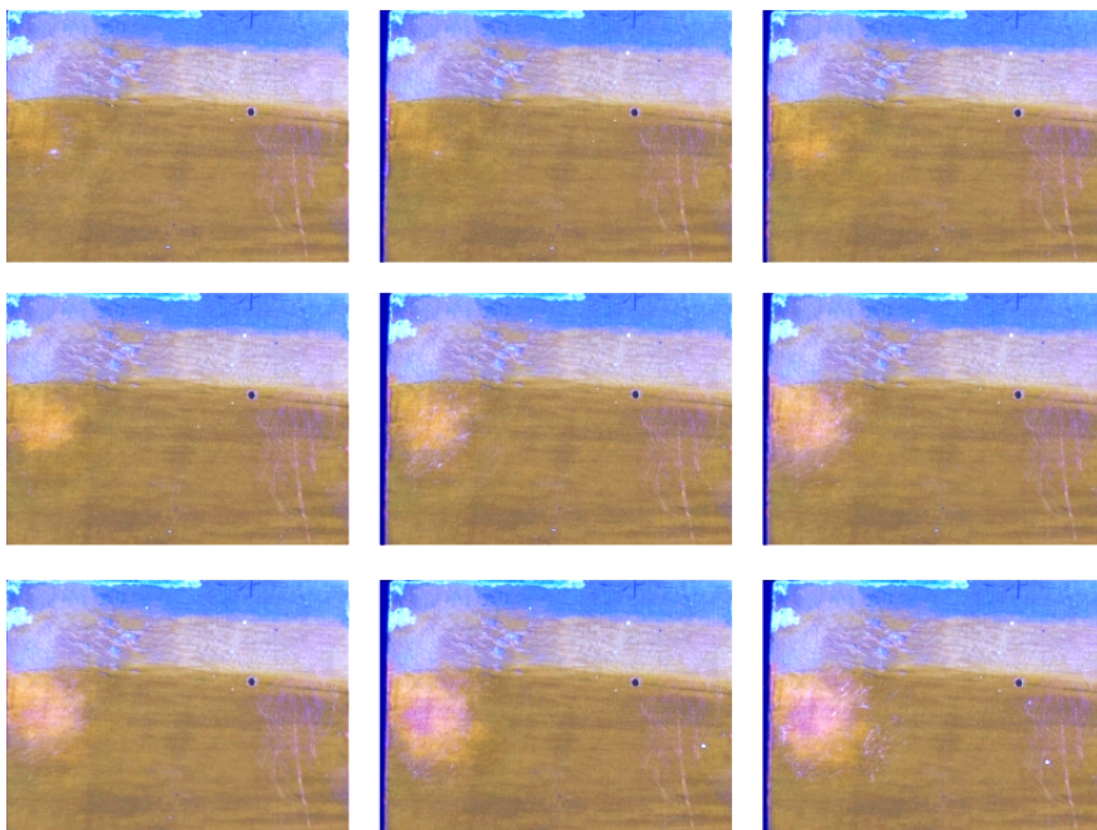


Figure 1.6: The WS02 sequence.

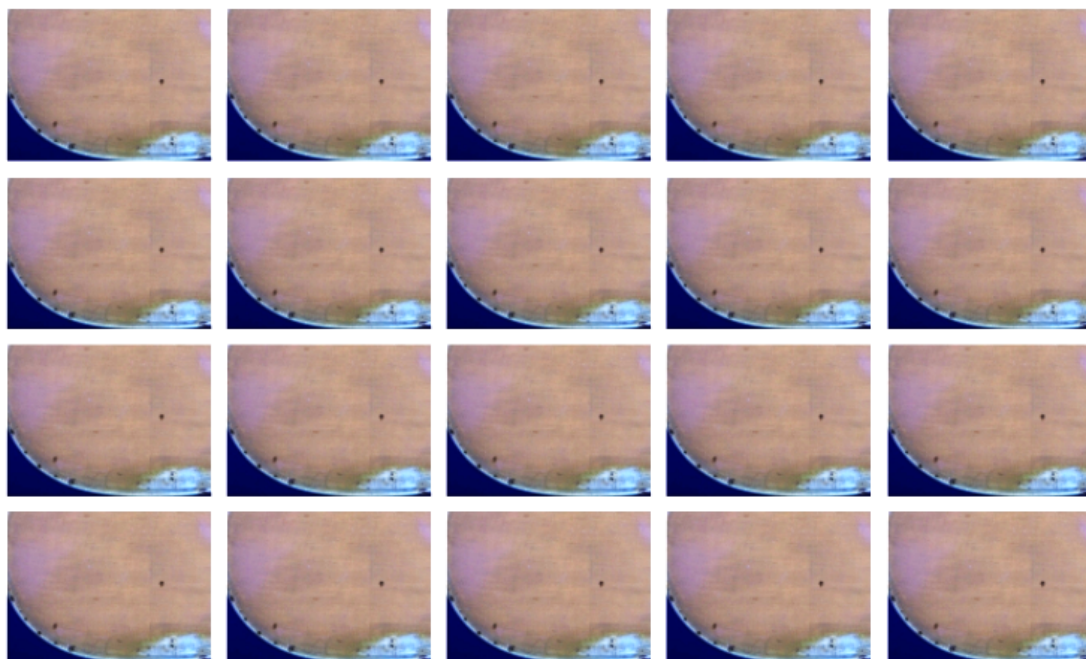


Figure 1.7: The SV01 sequence.

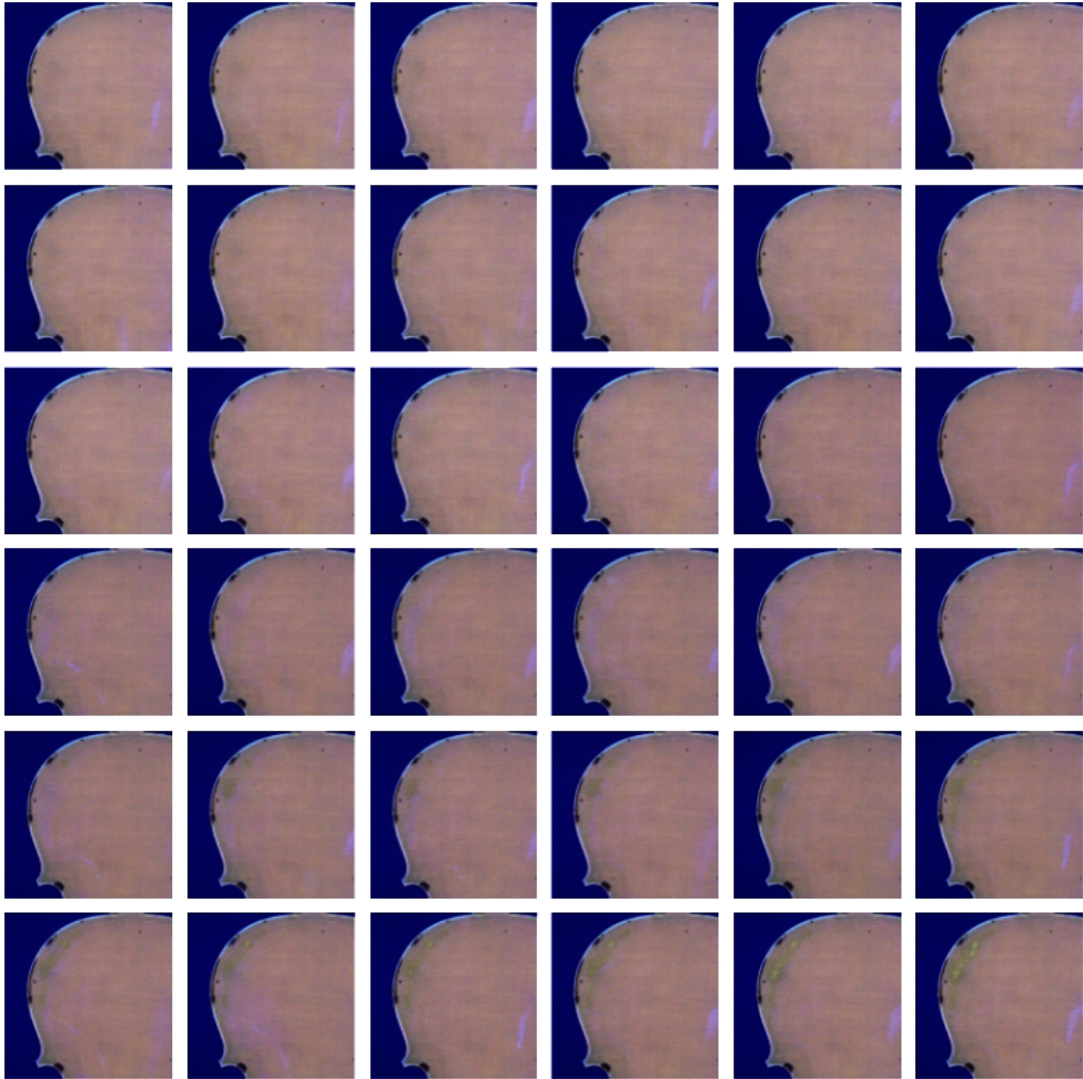


Figure 1.8: The SV02 sequence.

1.4 Notations and basic assumptions

Some notations have already been introduced. Let us recall and complete them:

- $\mathcal{K} \subsetneq \mathbb{N}_+$ denotes the set of image indices; For notation convenience, in the following we assume $\mathcal{K} = \llbracket 0, K-1 \rrbracket$;
- $\mathcal{I} = \{I_i, i \in \mathcal{K}\}$ is the image series, so that I_0 is the original image, that all the subsequent frames will be compared with;
- $\Delta I = \{\Delta I_i, i \in \llbracket 1, K-1 \rrbracket\}$ is the difference map series.

Wear detection is a semantic segmentation problem with two semantic classes which are the unchanged areas and the wear region(s). However, due to some noise or artefact, such a problem is not straightforward. Indeed, considering images independently, some artefacts cannot be distinguished from wear, so that only the temporal evolution of such areas allows for their distinction. Specifically, let us denote by C_1 the unchanged areas, background or untouched wooden surface, by C_2 the noise or artefacts due to reflectance, device error or human error, and by C_3 the surface wear.

Based on the considered application, we can make the following assumptions:

- If C_3 areas exist, they are at least a few pixels large;
- Two C_3 areas are considered as separate clusters if they are divided by enough empty space;
- $\forall i \in \mathcal{K}$, if a C_3 area is present in image I_i , it will also be present in subsequent images $I_j, j \in \{i+1, \dots, K-1\}$;
- $\forall i \in \{1, \dots, K-2\}$, C_2 areas do not grow from image I_i to I_{i+1} .

We specifically avoid to make any assumption about the distribution of the change values, explicit shape of the wear, and number of the wear regions. As a result, searching for a parametric form (such as disk, tile, etc.) will not solve our problem.

Therefore, in this work, we propose to split our initial problem into two sub-problems as follows:

1. Semantic segmentation of each image in ΔI considering the two following classes: C_1 and the disjunction $\{C_2, C_3\}$ (since they are assumed indistinguishable);
2. From areas labelled $\{C_2, C_3\}$, construct spatio-temporal clusters in \mathcal{P}^{K-1} and refine their label between C_2 or C_3 .

1.5 Data pre-processing

In a typical optical monitoring process for cultural heritage objects, images are taken every few weeks or months (or any predefined period of time). This helps us to find any unwanted change on the subject but it also presents many challenges: the image acquisition is done potentially by different operators and under different environmental conditions; therefore, it is natural to find spatial and spectral inconsistencies between images. The following are the most common sources of this problem:

1. *Illuminant, camera and sample position/orientation.* Any small change to the position or the orientation of the camera (sample, or the illuminant) can produce misalignment and more importantly reflection artefacts. Careful documentation of the process can alleviate (or perhaps eradicate) these issues.
2. *Equipment deterioration over time.* Depending on the overall length of the monitoring process, we may encounter gradual deficiencies in the performance of the camera or the illuminant. It is vital to detect these issues beforehand and replace the faulty equipment.
3. *Human error.* The capturing process itself may introduce noise, artefacts or spatial/spectral misalignment; mainly through the mistakes of the operator. An effective way to limit these issues is to repeat each capture a few times and choose the best one later in the computation phase.

In addition to careful consideration of the previous points during the acquisition, we perform several pre-processing steps to align the images as best as possible. The details of these steps depend on the application, i.e. the surface which we are monitoring. In the current work, our data only contains images of varnished wooden surfaces, applicable to various historical music instruments. The following steps are designed for this specific application:

1. *Illumination correction.* Despite setting the same illumination hardware setup and configuration for each image capture, the illumination variation from one acquisition session to another may be of same order of magnitude as the colour variations due to wear appearance or increase. Then, for each frame $I_t, t > 0$, for each colour channel, we normalise the mean pixel values with the reference frame I_0 value, i.e. subtract the difference between means ($\mathbb{E}[I_t] - \mathbb{E}[I_0]$, where $\mathbb{E}[\cdot]$ is the expectation operator approximated here by the average of image pixels).

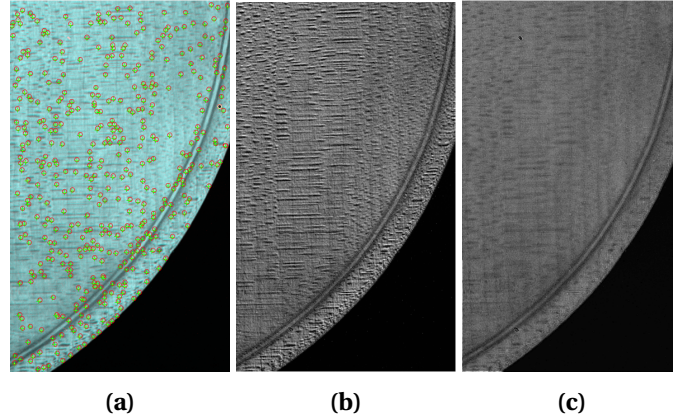


Figure 1.9: An example of spatial registration: (a) matched SIFT feature points; (b) difference of two frames before registration; (c) difference of two frames after registration. Brighter locations indicate higher differences.

2. *Spatial registration.* In order to spatially match the samples, we rely on extracting and matching SIFT [56] features in the original image and also in the subsequent frames. Many approaches exist for registering deformable objects performing general transformations[19, 101], however our captures have been performed on a rigid object in a controlled environment. Although some residual rotations may subsist, the transformation for each image pair may be approximated using small translation and scaling components (the proper alignment of the samples with respect to the camera is easier to perform during the capturing process). Figure 1.9 shows the detected SIFT features in a sample pair of images and the result of the registration. The figure also illustrates the pairs of matched features which are, in our case, very tightly coupled. These matched points are used to estimate a proper transformation between the two frames using a robust estimation method [90]. After estimating the right transformation from the matched features and applying it to the moving image we assume that the alignment is achieved at pixel level.
3. *Reflection removal.* The UV reflection on the surface of the samples usually appears in the purple/blue section of the visible range. Therefore, to remove the most intrusive reflections in our RGB images, we filter out the pixels that have significantly higher blue values than green ($B \gg G$) or have higher red and blue than green ($B > G$ & $R > G$). Note that, depending on the application, these operations might or might not be necessary, and that the general aim is simply to increase the correlation between change and the investigated process (in our case, the wear). Figure 1.10 shows the result of this process on a sample UVIFL image.

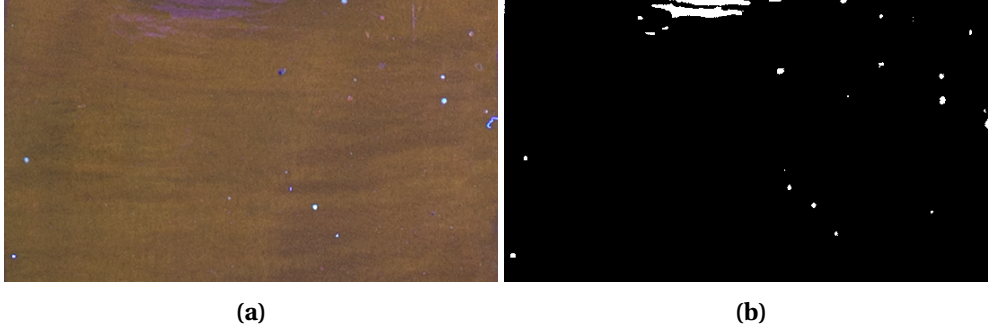


Figure 1.10: An example of reflection removal. (a) the sample frame (b) the produced mask showing the UV artefact pixels.

1.6 Difference Map

The first step in our change detection process is to calculate the difference map sequence ΔI . Each ΔI_i is a pixel by pixel difference map between any considered frame I_i and the reference frame I_0 . The image ΔI_i should be a grey-level image defined on the pixel domain $\mathcal{P} \subseteq \mathbb{N}^2$.

There are many different ways to derive the difference maps depending on the application and the sample being monitored. Change detection neural networks, for example, can be used to produce such a map [65, 96, 110] but they are useful only in specific applications with considerable amount of labelled data, which clearly is not our case.

In the case of varnished wooden surfaces, since pixel-level alignment can be achieved quite easily, we can consider pixel by pixel colour difference as a reasonable metric. When comparing the colour of two pixels, we are interested in how similar or different they are perceived to the human eye. The $L^*a^*b^*$ colour space is designed to relate visual differences between two colours to a measure of Euclidean distance. The colour difference formulas associated with $L^*a^*b^*$ gives us a quantitative representation of this visual difference. In the current work, we have used the CIEDE2000 [58] formula to compute the difference map.

The captured images are all in conventional RGB format, so the first step is to convert the RGB values into the $L^*a^*b^*$ colour space. From there, we use the CIEDE2000 [58] formula with all its constants set to 1. Figure 1.11 gives an example of difference map for one single frame compared to the reference frame. There are two things to note in this figure: firstly, the obvious reflection areas including the background have been removed in the pre-processing step; so they appear completely black; Secondly, even the unchanged areas on the surface of the violin produce some values in the difference map (values > 0). Then, any further developed method will have to consider the presence of such background noise, e.g. by comparing pixel local values with respect to the rest of the image.

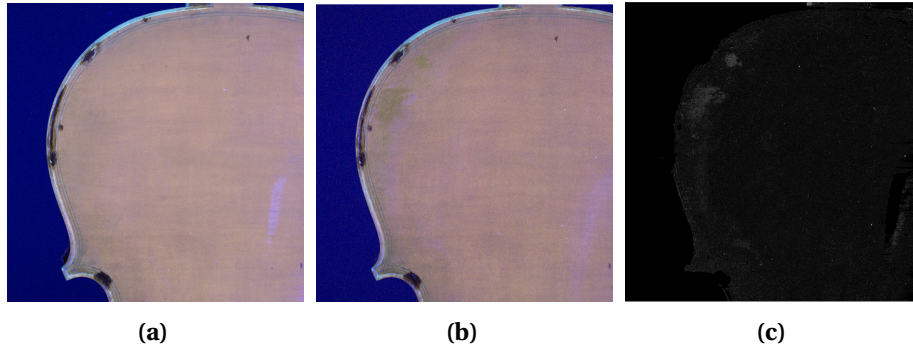


Figure 1.11: An example of colour difference map; (a) reference frame I_0 , (b) frame I_{27} , (c) the difference map produced from CIEDE2000 [58].

It is worth noting that, in our experience, all three CIELAB-based colour difference formulas CMC [16], CIE94 [60] and CIEDE2000 [58] work correctly for our application. They do not produce identical values but give us the same relative relationship between the pixels. In addition, more in-depth colour difference models specific to the wooden surfaces can be developed to potentially boost the performance of the wear detection. This, however, falls outside the scope of the current study.

1.7 Conclusion

In this chapter, we presented a quick introduction to preventive conservation and optical monitoring to specify the challenges and limitations which make this study necessary. In addition, the wear detection problem was introduced formally along with several notations used in the following chapters. Several data pre-processing steps were explained which are responsible for calculating the input to our proposed clustering algorithms in Chapters 4 and 5.

To demonstrate the difficulties of clustering a given dataset, the next chapter goes through several families of clustering algorithms and iterates their strengths and weaknesses.

Chapter 2

Data clustering algorithms

Contents

| | |
|--|-----------|
| 2.1 Hierarchical clustering | 22 |
| 2.2 Partition-based clustering | 24 |
| 2.3 Density-based clustering | 26 |
| 2.4 Distribution-based clustering | 27 |
| 2.5 Fuzzy theory based clustering | 29 |
| 2.6 Dimensionality reduction and feature transformation | 30 |
| 2.6.1 Principle Component Analysis | 30 |
| 2.6.2 Spectral Graph theory | 31 |
| 2.6.3 Kernel algorithms | 32 |
| 2.6.4 Deep neural networks | 34 |
| 2.7 Evaluation metrics | 37 |
| 2.8 Conclusion | 38 |

In this work, we are interested in unsupervised classification or clustering of the input data, which means that no labelled data is available for learning. More formally, clustering is a way of separating a finite unlabelled dataset in an unknown area into a finite set of structures such that [45, 106]:

- *points* (or objects), in the same cluster, are as similar to each other as possible;
- points, in different clusters, are as different as possible to each other;
- and finally, similarity and dissimilarity is measured in a clear and well-defined manner.

A typical clustering process involves the following steps [106]:

- Selecting a set of object representative features from the input dataset. These features are used to distinguish between patterns of different clusters. In general, they need to be robust to noise and easy to interpret. Proper selection of features can simplify the following steps in the design process.
- Clustering the data based on derived object features and application domain characteristics.
- Evaluating the results using a predefined metric.
- Explaining the acquired results in a practical sense; what does each cluster represent?

In this chapter, we take a quick look at several families of clustering algorithms. Each one performs using specific features of the targeted data. In addition, they present different benefits and shortcomings.

2.1 Hierarchical clustering

The goal of this family of clustering methods is to organise the dataset into a hierarchical structure based on a proximity matrix. The output is usually depicted as a binary tree or a dendrogram where the root node represents the whole dataset and the leaf nodes are each individual objects, called points in the following. Each horizontal cut of this structure will represent a possible flat (non-hierarchical) clustering of the data. There are two main ways to generate this structure: agglomerative or divisive clustering methods. Agglomerative clustering starts from a state in which every single point is a cluster; then step by step, it merges the closest clusters together until only one cluster is left which comprises all points. Divisive clustering operates in the opposite way and starts with the whole data as one cluster. Successive divisions are performed until every cluster has only one point [106]. In practice, agglomerative methods are mostly preferred to the divisive methods.

Here, we take a look at some widely-used agglomerative clustering algorithms. The general process shared by all methods in this family is as follows [106]:

1. Starting from N points as singleton clusters, we calculate the proximity matrix.
2. Find the two closest clusters based on the proximity matrix and merge them into a new cluster.
3. Update the proximity matrix for the new clusters.
4. Repeat steps 2 and 3 until we have only one cluster containing all the objects.

Table 2.1: Several linkage definitions for agglomerative hierarchical clustering.

| Name | Formula | Explanation |
|------------------|--|--|
| Single linkage | $d(A, B) = \min_{a \in A, b \in B} d(a, b)$ | The distance between two clusters is the distance between their closest members. |
| Complete linkage | $d(A, B) = \max_{a \in A, b \in B} d(a, b)$ | The distance between two clusters is the distance between their most dissimilar members. |
| Average linkage | $d(A, B) = \frac{1}{ A \times B } \sum_{a \in A, b \in B} d(a, b)$ | The distance between two clusters is the average distance between their members. |
| Centroid linkage | $d(A, B) = d(\bar{a}, \bar{b})$ | The distance between two clusters is the distance between their respective centroids. |

The differentiating factor is the way they compute the distance between two clusters, called linkage. Some widely used linkage definitions include single, average, complete and centroid linkage [31]. Table 2.1 explains these definitions in more detail.

Hierarchical clustering methods, in general, are not robust to noise and outliers; therefore, without additional features, they are expected to perform poorly on datasets with noisy data. In addition, as the clustering process suggests, each object is considered only once and assigned to a particular cluster; therefore, a possible mistake will not be corrected later on [106]. Depending on the application, another disadvantage can be their requirement for the number of clusters as input. On the other hand, they have the advantage of unveiling the inherent hierarchical relationships present in the input data. They, also, can be used for datasets with any arbitrary shapes [105].

2.2 Partition-based clustering

Partition based methods produce a flat (non-hierarchical) clustering in an iterative manner, by computing, in each step, a cluster representative for each cluster based on the data points in that cluster. K-means [59] is the most famous example of this family [105]. The general process for K-means is as follows:

1. Choose k centres $c_i, 0 < i \leq k$ randomly or based on prior knowledge from the set of data points \mathcal{X} .
2. Assign each point in \mathcal{X} to the cluster with the nearest centre c_i .
3. Compute for each cluster \mathcal{C}_i its new centre: $c_i = \frac{1}{|\mathcal{C}_i|} \sum_{x \in \mathcal{C}_i} x$.
4. Repeat steps 2 and 3 until there is no change for any cluster.

As we can see, the general process is very simple to implement and execute. This makes it an attractive choice for datasets containing compact and (hyper)spherical clusters. In addition, since it can be executed very rapidly, it is a viable choice for large datasets [105].

The standard version of the algorithm comes with several downsides; as a result, many extensions and variants have been proposed to rectify these issues. The first issue is the selection of the initial set of partitions, i.e. seed selection. The final clustering result can vary depending on the choice of initial seeds. Different seed selection alternatives have been proposed [36, 48, 59], which try to replace the random selection with their own process and as a result improve the algorithm's robustness, effectiveness and convergence speed. The second important issue of k-means is its sensitivity to outliers and noise. Even a single outlier, far from the centre, can distort the shape of a cluster and as a result change outcome of the algorithm. Variants such as [7] and [48] try to improve this aspect of the algorithm by ignoring the clusters with few points or by considering only actual data points as centres for each cluster [106].

A widely implemented variant of k-means is K-means++ [5]. This algorithm adds a randomised seeding technique and improves the accuracy and speed in most cases. Specifically, it chooses the starting centres at random from the points in the dataset, but with each point weighted based on its proximity to the closest centre already chosen. More specifically, replace the first step of the classic K-means with the following:

1. Select one centre c_1 randomly from the set of data points \mathcal{X} .

2. Let $D(x)$ be the distance from the data point x to the closest centre already chosen. Select a new centre c_i by choosing x from \mathcal{X} with probability $\frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2}$.
3. Repeat step 2 until k centres have been chosen.

An important subset of partition-based algorithms is the affinity propagation (AP) methods. Similarly to k -means, these methods (in an iterative fashion) search for a set of exemplar data points to represent a good partitioning of the data. The difference, however, is that they do not require an initial guess for the representative data points (cluster centres in k -means). This way, they avoid the problems associated with sensitivity to the initial guess. The basic AP process considers all data points simultaneously as potential representative points. Then, assuming each data point is a node in a network, it recursively sends (propagates) real-valued messages between nodes until a good set of representative data points and their corresponding clustering is found [37].

As input, AP takes the similarity matrix S where $s(i, k)$ shows how well the data point x_k is suited to represent data point x_i ; and, for example, it can be set to negative squared euclidean distance: $s(i, k) = -|x_i - x_k|^2$. In addition, if $i = k$ then $s(k, k)$ is an input value for x_k which indicates how suitable it is to be a representative point. If there is no a-priori knowledge, then, every $s(k, k)$ is set to a common value. This value can influence the number of clusters produced by the algorithm (for example, if it is set to the minimum of input similarities, AP will produce a small number of clusters) [37].

The propagation process involves two kinds of messages: the *responsibility* and the *availability*. The responsibility message $r(i, k)$ from node i to candidate representative node k conveys how well-suited node k is to represent node i taking into consideration all the other candidates for node i . The availability message $a(i, k)$ from candidate representative node k to node i conveys how suitable it would be for node i to choose node k as its representative taking into consideration the support other nodes give to node k . In the first iteration, the availabilities are set to zero and the responsibilities are computed as follows [37]:

$$r(i, k) = s(i, k) - \max_{j: j \neq k} (a(i, j) + s(i, j)). \quad (2.1)$$

Then, the availabilities are updated for node k by adding the self-responsibility $r(k, k)$ to the sum of positive responsibilities received from other points:

$$a(i, k) = \begin{cases} \min \{0, r(k, k) + \sum_{j: j \notin \{i, k\}} \max(0, r(j, k))\}, & \text{if } i \neq k \\ \sum_{j: j \neq k} \max(0, r(j, k)), & \text{otherwise.} \end{cases} \quad (2.2)$$

At the end of each iteration, the combined value of availability and responsibility for each node suggests its suitable representative: for node i , the value of k that maximises $a(i, k) + r(i, k)$ indicates its representative (which can be i itself). The message passing continues until a convergence condition is reached; meaning that the representative decisions do not change anymore [37].

In general, AP algorithms have the advantage of being simple and easy to implement while showing good robustness to outliers. In addition, the exact number of clusters is not required to be set in advance. On the other hand, AP suffers from high computational complexity and therefore is not suitable for very large data sets. It is also worth noting that AP can be sensitive to the input parameters $s(k, k)$ discussed earlier [105].

2.3 Density-based clustering

Density-based (DB) clustering approach relies on the idea that high density areas present in the data represent each cluster. Compared to some other clustering families, DB methods do not need the number of clusters as input, nor do they produce results with low within-cluster dissimilarities. This means the output clusters can be of any shape (convex or concave). Each cluster is a set of data points spread over the data space separated from other clusters by regions of low density data points. In an ideal clustering, noise and outliers remain outside of all clusters [13].

Considering the probability density function for the data set, a density-based clustering output can be viewed as a threshold through the function. Regions with higher probability density are the resulting clusters and the rest of the data is ignored as outliers [13]. Figure 2.1 shows a sample 2D data set and its corresponding density distribution. In this case, the density threshold has produced two clusters and a large number of discarded points.

DB algorithms usually only differ in the definition of density and how to consider two objects connected. The main classic examples include: DBSCAN [30], OPTICS [4] and Mean-shift [17]. In general, these methods have medium to high time complexity; and their main challenge is to find suitable density threshold(s), either local or global. Many improvements have been proposed to alleviate this issue and introduce new benefits. For example, hierarchical DB methods can help with the density threshold problem. *HDBSCAN* [14] or “Hierarchical Density-Based Spatial Clustering with Application with Noise” is an example of these methods. It provides a complete clustering hierarchy of all possible density based clusters. In addition, it can produce a non-hierarchical clustering without the need for the number of clusters as input by maximising the overall stability of the extracted

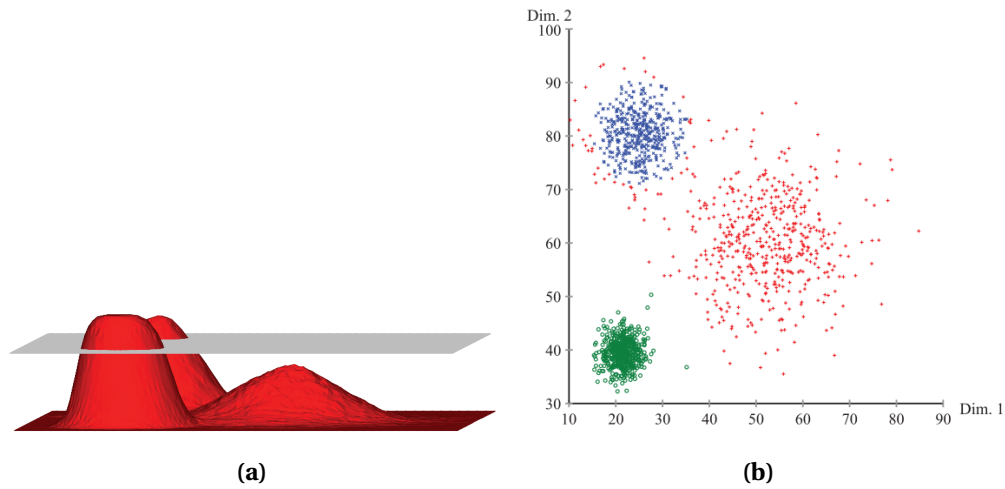


Figure 2.1: (a) Density distribution of a sample data set with a high threshold and (b) its resulting clustering into two clusters (blue and green) and noise (red) [13].

clusters.

Another clustering algorithm with similar ideas to density-based methods is “Clustering by fast search and find of density peak” [77]. This method combines distance and density for its clustering criterion. More specifically, it assumes the following about the cluster centres:

- they have a higher local density than their neighbours;
- they are far from points with higher densities.

Based on these two ideas, the algorithm constructs a decision plot which shows the local density of each data point and the shortest distance between that point and points with higher local density. In the end, data points with relatively high values for both criteria become the cluster centres and the remaining points are added to the closest cluster [77].

This method has the advantage of simplicity and robustness to outliers; and in addition, can work with arbitrarily shaped clusters. However, its time complexity is relatively high; and the process of choosing centres from the decision plot can be very subjective [105].

2.4 Distribution-based clustering

This family of methods assumes that in a dataset containing points generated from different probability distributions, and the clustering objective is that the points following the same distribution should belong to the same cluster [105]. The clusters can represent varying distribution types, or alternatively the same type

but with different parameters. If we assume (or know) the distribution type before hand, then the clustering would be equal to estimating the parameters of several distributions present in the data [106].

Formally, consider for each cluster $\mathcal{C}_i, i \in [1, K]$, the prior probability $P(\mathcal{C}_i)$ and the conditional probability $p(x|\mathcal{C}_i, \theta_i)$, where θ_i is the parameter vector to be estimated. The mixture probability density for the whole dataset is computed as:

$$p(x|\theta) = \sum_{i=1}^K p(x|\mathcal{C}_i, \theta_i)P(\mathcal{C}_i), \quad (2.3)$$

where $\theta = (\theta_1, \dots, \theta_K)$.

To find the posterior probability for assigning a data point to a particular cluster, we only need the parameter vector θ . To construct the mixture densities, the most important and widely used type is the Gaussian mixtures. This is mainly due to its simple and concise representation which requires only two parameters: the mean and the variance. Furthermore, the Gaussian density is symmetric and assumes the least prior knowledge when estimating an unknown probability density with a given mean and variance [113].

To estimate the parameters θ , the most popular statistical approach is the Maximum likelihood (ML) estimation. To that end, the algorithm finds the parameters which *maximise* the probability of generating all the observations. This probability is given by the following joint density function:

$$p(x_1, \dots, x_N|\theta) = \prod_{j=1}^N p(x_j|\theta); \quad (2.4)$$

or its logarithmic version:

$$l(\theta) = \sum_{j=1}^N \ln p(x_j|\theta). \quad (2.5)$$

Generally, the solution to this maximisation problem cannot be found analytically; therefore, sub-optimal estimation methods are needed to obtain an acceptable approximation [106]. However, there exist many practical problems with the approximation process [113]:

- The log-likelihood function (Eq. 2.5) can have non-unique solutions; meaning that multiple sets of parameters produce largest local maximum.
- There is a possibility for numerous local maximum solutions which are not the global maximum.
- The numerical methods used to solve the maximum likelihood problem may be sensitive to the initial values and produce different solutions each time.
- The number of components in the mixture must be known before hand.

The most popular approximation method for solving Equation 2.5 is the Expectation Maximisation (EM) algorithm. EM consists of two main parts: an expectation step and a maximisation step. The first step (E-step) provides an expectation of the unknown variables using the current estimate of the parameters; then, the maximisation step (M-step) calculates a new estimate of the parameters. Both steps are repeated until convergence to a solution [64]. The EM process considers the dataset to be incomplete and divides each data point into two parts: the observable features and the missing data [106]. Following this assumption, it produces a series of parameter estimates $\{\theta^0, \theta^1, \dots, \theta^T\}$, where T is the convergence state reached by the following steps [64]:

1. Set $t = 0$. Choose initial parameter θ^0 .
2. *E-step*: estimate the unobserved data using θ^t .
3. *M-step*: Compute maximum likelihood estimate of parameter θ^{t+1} using the estimated data.
4. $t = t + 1$. Go to step 2 if not converged.

As mentioned before, the EM process suffers from sensitivity to the initial parameter choice (θ^0), converging to a non-optimum answer, and a low convergence speed [106].

2.5 Fuzzy theory based clustering

Unlike the previously mentioned algorithms, fuzzy clustering methods assign each point to every cluster with a varying degree of membership. This is opposite to *hard* or *crisp* clustering which assigns each point to at most one cluster. This is mostly useful in cases where there is ambiguity in the data and the boundaries among the clusters are not well defined [106].

A classic example of fuzzy methods is *Fuzzy c-means* or *FCM*. In short, FCM tries to assign membership to each data point based on the its distance to the cluster centres. The closer each data is to a particular cluster centre the higher its membership will be to that cluster. The membership values for each data point are real values belonging to $[0, 1]$ and add up to 1.0. The process of finding the cluster centres and updating the membership values is repeated until convergence. The following is the general steps of FCM [88]:

1. initialise the matrix $U^{(0)} = [u_{ij}]$. u_{ij} is the membership value of data point x_i in the cluster j .

2. Calculate the centres vector $C^{(k)}$ based on $U^{(k)}$.
3. Compute $U^{(k+1)}$ based on new cluster centres.
4. if $|U^{(k+1)} - U^{(k)}| > \epsilon$ then $k = k + 1$ and go to step 2.

This basic FCM process suffers from the following limitations [88]:

- sensitivity to the initial membership values $U^{(0)}$;
- high time complexity;
- inability to deal with noise and outliers (points with low membership values).

Various alterations and extensions have been proposed to deal with the problem of time complexity and noise sensitivity of FCM [12, 15, 40]. In a recent example, in the specific case of image segmentation, [52] proposes FRFCM (Fast and Robust FCM), an improved version of FCM which have two advantages: firstly, reducing the time complexity by using a faster membership filtering instead of distance calculations between data points and cluster centres; and secondly, increasing the robustness to noise by smoothing the input data with morphological reconstruction [93].

2.6 Dimensionality reduction and feature transformation

Most conventional clustering methods struggle to produce acceptable results for high dimensional data. The main reason is the inefficiency of their similarity measures in finding the existing patterns in the data. Additionally, high dimensions can cause high computational complexity when clustering a large dataset. To deal with these problems, there exist several methods for dimensionality reduction and feature transformation where the original data is mapped to a new feature space. In this space, the data would be easier and/or faster to separate using the existing conventional measures [63]. In the following sections, we go through several linear and non-linear transformation techniques.

2.6.1 Principle Component Analysis

Principle Component Analysis (PCA) is a linear multivariate technique which analyses a given dataset and extracts its important information. It provides a new set of orthogonal variables called principal components which act as a low dimensional version of the original dataset. Each principal component is a linear combination

of the input data points. The first component has the highest possible variance; the second one has to be orthogonal to the first and again with the highest possible variance. The other components follow the same pattern [1].

Given a d -dimensional input dataset $\{x_i\}, i \in \llbracket 1, n \rrbracket$, the transformed data $\{y_i\}$ is computed as:

$$y_i = A^T(x_i - \mu), \quad (2.6)$$

where μ is the sample mean of the input data. To construct the orthogonal transformation matrix A (whose dimensionality is $d \times d$) the eigenvectors of sample covariance matrix Q are used. Q is computed as:

$$Q = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T. \quad (2.7)$$

Eigenvalue decomposition of $Q = AVA^T$ gives the eigenvectors of Q . The diagonal matrix V holds the eigenvalues of Q which describe the variance of observations towards the corresponding eigenvector. The largest eigenvalues indicate their eigenvectors as principal components of the input data. By choosing the first d' ($d' < d$) principal components we can map the input data into a low dimension space. The ideal value for d' is chosen by minimising the squared reconstruction error E_r given as:

$$E_r = \|x - \mu - yA_{d'}\|^2, \quad (2.8)$$

where $A_{d'}$ is the transformation into the ideal sub-space and constructed by the first d' eigenvectors [49]. In clustering, PCA can be used as a pre-processing tool to make the input dataset less complex by producing a low-dimensional representation. However, PCA is only able to consider second order correlation between data points; non-linear methods are able to take into account higher order correlations.

2.6.2 Spectral Graph theory

The main idea in graph-based clustering techniques is to consider each point in the data set as a node or vertex and the similarity values between them as weighted edges. This way, the clustering problem transforms into a graph partition problem. Therefore, the clustering result should minimise the total weight of connections between clusters while maximising the total weight of connections within each cluster [94, 105].

In addition, spectral clustering methods have the ability to deal with high dimensional datasets by producing an alternate set of data points with lower (and manageable) dimensionality (using a non-linear transformation). The general process of spectral clustering can be summarised in the following [69]:

1. Define the similarity matrix A describing the similarity between each two points in the dataset.
2. Calculate the Laplacian matrix L using the similarity matrix A .
3. Calculate the first k eigenvectors of L .
4. In the matrix Y formed by each eigenvector as a column, consider the rows as new data points and cluster them using another clustering method, e.g. k-means.
5. Assign the original data point s_i to cluster j if and only if the i th row of Y belongs to cluster j .

Note that applying k-means from the beginning on the input dataset may result in unsatisfactory results because of the general shape of the clusters present in the data.

Several studies have proposed improvements on different parts of the general process. We can divide these improvements into two sets depending on which step of the original algorithm they target. The first set of algorithms [71, 72, 95] tries to improve the construction of the data affinity graph in the first step to make it robust to noise and outliers and to ultimately improve the clustering results. The second group [69, 83, 103] tries to improve data grouping after the data affinity graph is constructed [112].

In the first group, the performance of spectral clustering greatly depends on the choice of the data affinity graph (the similarity matrix). Constructing this graph is not a trivial task due to the inherent ambiguity and complexity of the input data. For example, [112] proposes a robust affinity graph which takes into account only the most informative features in the feature space. The method works in an unsupervised manner (i.e. no need for ground truth annotation); and in addition, it is robust to noise and irrelevant features.

In general, spectral clustering methods are suitable for data sets with arbitrary shapes and high dimension. On the other hand, they suffer a high time complexity, unclear process of similarity matrix construction and eigenvector selection; and they need the number of clusters as an input [105].

2.6.3 Kernel algorithms

As mentioned earlier, many conventional similarity measures are incapable of finding complex patterns present in the data. Kernel-based methods try to solve this problem by transforming the data to a higher dimensional space. In other words,

it is possible to linearly separate non-linear patterns present in the data by first, non-linearly transforming it to a higher feature space [106].

Let us first define a positive definite kernel, also called a Mercer kernel:

Definition 1. Let $X = \{x_1, \dots, x_n\}$, $x_i \in \mathbb{R}^d$ be a nonempty set. Function $K : X \times X \rightarrow \mathbb{R}$ is called a positive definite kernel if and only if:

1. $K(x, y) = K(y, x)$,
2. $\sum_{i=1}^n \sum_{j=1}^n c_i c_j K(x_i, x_j) \geq 0$, $\forall n \geq 2$,

where $c_r \in \mathbb{R} \forall r = 1, \dots, n$.

Let $\phi : X \rightarrow \mathcal{F}$ be a mapping from input space X to a high dimension feature space \mathcal{F} ; then, a positive definite kernel K can be computed as [33]:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j). \quad (2.9)$$

In practice, one important feature that makes the kernel method work is the fact that we are able to compute the Euclidean distances in \mathcal{F} without knowing ϕ [33, 67]:

$$\begin{aligned} \|\phi(x_i) - \phi(x_j)\|^2 &= (\phi(x_i) - \phi(x_j)) \cdot (\phi(x_i) - \phi(x_j)) \\ &= \phi(x_i) \cdot \phi(x_i) + \phi(x_j) \cdot \phi(x_j) - 2\phi(x_i) \cdot \phi(x_j) \\ &= K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j). \end{aligned} \quad (2.10)$$

This means that a distance in the target feature space is a function of the input data points. There are several examples of Mercer kernel function, including [33]:

- linear kernel: $K^{(1)}(x_i, x_j) = x_i \cdot x_j$ which results in $\phi = I$, the identity function;
- polynomial kernel of degree p : $K^{(p)}(x_i, x_j) = (1 + x_i \cdot x_j)^p$, $p \in \mathbb{N}$;

- Gaussian kernel: $K^{(g)}(x_i, x_j) = e^{\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)}$, $\sigma \in \mathbb{R}$.

Using the kernel method, there are two main ways to adapt the conventional clustering methods: firstly, by kernelisation of the metric, which means we look for cluster centres in input space but we compute the distances in the feature space; and secondly, by computing the cluster centres in feature space [33].

To summarise, the kernel methods can help conventional clustering techniques to be more successful with complex datasets. In addition, they can help with analysing noise and separating overlapping clusters. Kernel FCM [102, 111] and Kernel k-means [81] are examples of this process. On the other hand, they can be sensitive to the type of kernel function and its parameters. In general, kernel methods suffer from high computational complexity [105].

2.6.4 Deep neural networks

Owing to its property of highly non-linear transformation, deep neural networks can be used to transform a complex dataset into a more clustering-friendly space [63]. There are several different network architectures which have been proposed to derive this feature representation. We go through the most widely used architectures in the following sections.

2.6.4.1 Autoencoder

Autoencoder (AE) is a useful tool for unsupervised data representation. An AE consists of two main parts: an encoder which maps the input data into the target feature space; and a decoder which reconstructs the data using the target features. The aim is to minimise the reconstruction loss. Normally, the target representation has a smaller dimensionality than the input data; therefore, an AE is likely to extract the most salient features of the data [63].

Given f_ϕ as the encoding function and g_θ as the reconstruction function, the objective is to achieve the following minimisation [63]:

$$\min_{\phi, \theta} L_{rec} = \min \frac{1}{n} \sum_{i=1}^n \|x_i - g_\theta(f_\phi(x_i))\|^2, \quad (2.11)$$

where $\{x_i, i \in [1, n]\}$ is the set of input data points, and ϕ and θ are the function parameters to be optimised. The clustering based on an AE works by joint training on reconstruction loss (L_{rec}) and a clustering loss (L_c). The clustering loss can be any of the conventional clustering metrics: k-means, agglomerative, etc. (cf. Figure 2.2). Deep Clustering Network (DCN) [108], Deep Embedding Network (DEN) [43], Deep Continuous Clustering (DCC) [82], and Deep Embedded Regularised Clustering (DEPICT) [38] are some examples of clustering networks based on autoencoders.

In general, AE based methods have the advantage of being easy to implement and to explain; meaning that they can be combined with almost all clustering methods. As a result, they are the most common deep clustering network architectures. The use of reconstruction loss is another advantage which guaranties that the solutions are non-trivial and feasible. The computational complexity depends mainly on the clustering loss utilised for the training. However, the network is limited in term of depth to remain computationally feasible. The training based on both reconstruction and clustering losses also means that the parameter to balance the two needs extra fine tuning [63].

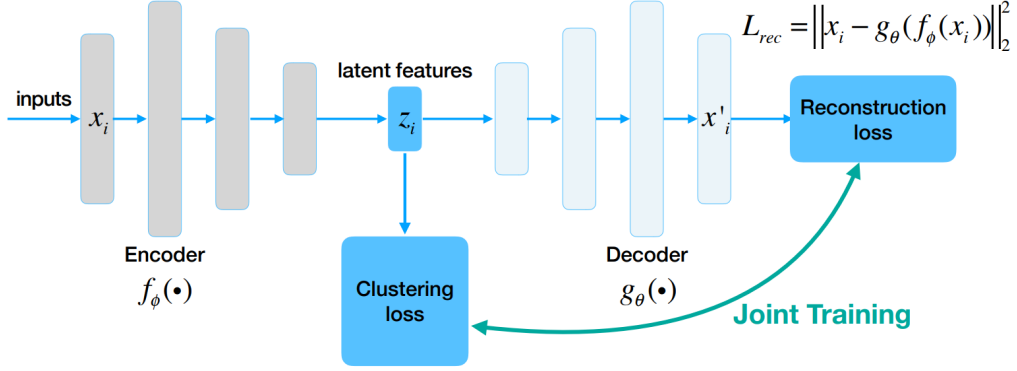


Figure 2.2: Architecture of a typical clustering based autoencoder [63].

2.6.4.2 Clustering Deep Neural networks

Clustering deep neural networks (CDNN) use only a clustering loss (L_c) to train the network which can be a fully connected network, a convolutional network, or another choice. The training is done in three main ways [63]:

- *Unsupervised pre-trained network:* Training a network on the input data for feature extraction; then fine tuning it using the clustering loss. As an example, Deep Embedded Clustering (DEC) [104] uses an autoencoder to learn a mapping from the data space to lower dimensional feature space and then, in the feature space, iteratively minimises the clustering loss.
- *Supervised pre-trained network:* Combining networks pre-trained on a dataset (e.g., ImageNet) and classical clustering algorithms. For example, in the domain of image clustering, Clustering Convolutional Neural Network (CCNN) [42] uses a pre-trained convolutional network (from ImageNet) to extract the features of its candidate cluster centres, while using K-means to update input samples of each cluster.
- *Non-pre-trained network:* Using only a well-designed clustering loss for feature extraction. As an example, for image clustering, Joint Unsupervised Learning (JULE) [109] tries to combine a convolutional network for representation learning and agglomerative clustering for assigning those representations to proper clusters in a hierarchical structure. During the training, the image representations and clusters are updated jointly in a recurrent process, such that better representations lead to better clustering and better clustering provides better image representations.

Unlike autoencoders, CDNN-based methods do not use any reconstruction loss to ensure non-trivial solutions; so the responsibility falls to the way they design the

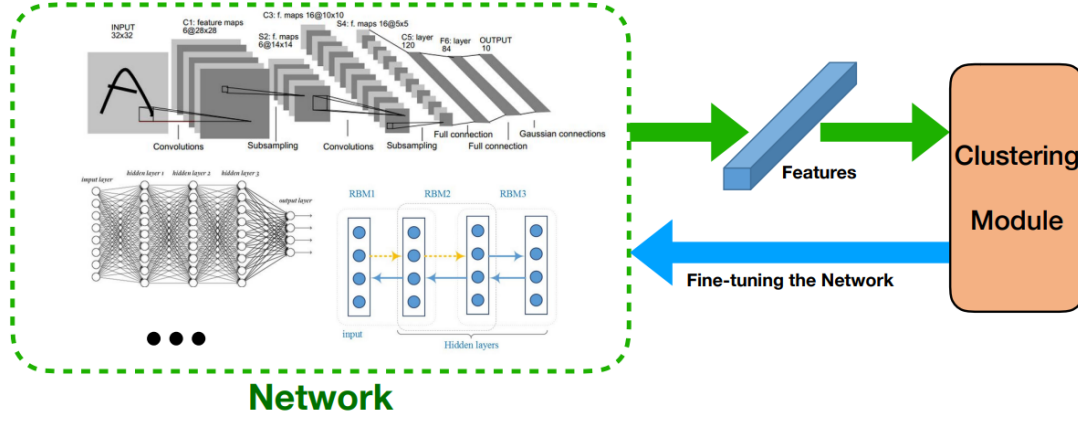


Figure 2.3: General architecture of CDNN-based clustering methods [63]. The network architecture varies between algorithms.

clustering loss and how they initialise the network (one of the three ways mentioned above). In addition, compared to AE methods, they have the capacity to run deeper networks and use pre-trained networks to extract more discriminative features; which means they are particularly suitable for large-scale datasets.

2.6.4.3 Generative Adversarial Network

A Generative Adversarial Network (GAN) consists of two networks: a *generator* network which generates new data from the input samples; and a *discriminator* network which tries to judge if an input is a real sample or it is generated [47]. The discriminator produces a real-valued score between 0 and 1 specifying how likely it is that the sample is real (scoring 1 when it considers the sample an actual data sample). Both networks play an adversarial game where the generator's objective is to fool the discriminator by producing better and better samples, and the discriminator improves its understanding of the data to be able to identify generated samples [47, 63]. The adversarial game can be formulated as a minmax optimisation [63]:

$$\min_G \max_D \mathbb{E}_x [\log D(x)] + \mathbb{E}_z [\log (1 - D(G(z)))], \quad (2.12)$$

where x is a data sample, z is a generated sample, D is discriminator and G is the generator.

As an example, Categorical Generative Adversarial Network (Cat-GAN) [85] is a clustering method which applies the GAN framework to multiple classes instead of two (generated or real). In Cat-GAN the discriminator D classifies the real data points into a given number of clusters k and stays uncertain about the samples

produced by the generator. Meanwhile, the generator G tries to generate data points belonging to exactly one specific cluster.

In general, GAN-based methods suffer from high computational complexity and may even fail to converge to a solution. Furthermore, the generator network may produce a very plausible output and decide to continuously generate the same output (or a small set of outputs). This is a specific failure called mode collapse. Another convergence failure is when the discriminator is too good and it provides not enough information for the generator to train [63].

2.7 Evaluation metrics

To compare the performance of any two clustering algorithms, we need some objective evaluation metrics. There are four widely used indicators which use the final clustering output to compute a performance score allowing comparison between any two algorithms. The building blocks of all four indicators are the measurements of *true positives (TP)*, *false positives (FP)*, *true negatives (TN)* and *false negatives (FN)*. These measurements are done by making use of the ground truth provided for the input dataset. Table 2.2 shows these four evaluation indicators and their formulas.

Table 2.2: Clustering evaluation indicators [105].

| Name | Formula | Explanation |
|---------------------------|--|--|
| F score | $\begin{cases} P = \frac{TP}{TP + FP}, \\ R = \frac{TP}{TP + FN}, \\ F_{\beta} = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R} \end{cases}$ | P is the precision, and R is the recall. β is a constant which indicates the importance of P and R with respect to each other. $\beta = 1$ gives the F_1 score in which precision and recall have the same importance. |
| Rand indicator | $RI = \frac{TP + TN}{TP + FP + FN + TN}$ | The denominator is equal to the total number of points in the dataset. |
| Jaccard indicator | $J(A, B) = \frac{ A \cap B }{ A \cup B } = \frac{TP}{TP + FP + FN}$ | Measures the similarity of two sets A and B. $ X $ is the number of elements in the set X. $0 \leq J(A, B) \leq 1$. |
| Fowlkes Mallows indicator | $FM = \frac{\sqrt{P \times R}}{\sqrt{\frac{TP}{TP + FP} \times \frac{TP}{TP + FN}}} =$ | P is the precision, and R is the recall. Higher FM values indicate greater similarity to the ground truth clustering. |

2.8 Conclusion

In this chapter, we presented a quick survey of several widely used families of data clustering algorithms. From classic methods such as k-means and fuzzy c-means to more recent propositions based neural networks, we demonstrated the difficulties faced by these algorithms to properly perform clustering.

The quality of the clustering is judged based on several factors. Firstly, how close the produced cluster are to real relations between the data points, measured by one of the introduced metrics. Secondly, how robust is the process to noise and outlier data? The algorithms that can deal with outliers better, find inherent relationships among data points easier. Finally, is it extendable to large datasets with high dimensional data? Many classic methods work very well on regular datasets but fail to perform on higher dimensions or when processing high number of data points. To solve this problem, several methods was introduced for dimensionality reduction and feature transformation.

In addition, each clustering algorithm comes with certain limitations: some need the number of clusters present in the data as input; some need initial guesses for different parameters; and some perform with a high time complexity. For each family of data clustering methods, strengths and weaknesses was mentioned to demonstrate their possible applications.

The next chapter introduces the a-contrario framework and the number of false alarms criterion which is the basis of our proposed clustering algorithm.

Chapter 3

A-contrario framework and the number of false alarms

Contents

| | |
|--|----|
| 3.1 Gestalt theory | 41 |
| 3.2 The Helmholtz principle | 46 |
| 3.3 The number of false alarms | 47 |
| 3.4 Applications of the a-contrario framework in computer vision . . | 48 |
| 3.5 Change detection using the a-contrario framework | 49 |
| 3.5.1 Seed detection | 50 |
| 3.5.2 Clustering the seeds | 51 |
| 3.6 Conclusion | 52 |

3.1 Gestalt theory

One of the oldest questions tackled by psychologist was: Why and how we interpret stimuli arriving at our eyes as familiar shapes (e.g., straight lines, polygons and curves)? After all, every scene produced in our eyes and brain consists of only a set of dots each corresponding to a retina cell. How a set of dots is related to the mathematical representation of an infinitely continuous line? Therefore, the problem consists of the identification between a group of incoming stimuli and a physical object or mathematical representation of a shape. This identification process follows a set of general laws called *the principles of visual reconstruction* [23].

The Gestalt theory is an attempt to state and explain the principles of visual reconstruction. In his paper in 1923, Max Wertheimer goes through two sets

of organising laws: *grouping laws* and the principles governing the interaction (collaboration and conflicts) of grouping laws. These grouping laws start from the lowest level and recursively make more complicated groupings [23]. Decades of research in Gestalt theory have given us a rich collection of different grouping laws. Most of the initial research was done on human perception and not on computer vision. However, apart from some specific details, the Gestalt grouping laws can be just as useful when working with camera frames.

Working with the pixels of a digital image as the starting point, every time a group of points or previously formed groups have some characteristics in common, they join and form a larger visual object, a *gestalt*. The followings are the basic grouping laws [23]:

- *Colour constancy*: Regions where luminance or colour does not change strongly are seen as a whole (Figure 3.1a).
- *Vicinity*: Objects with small distance to each other with respect to the rest are grouped together (Figure 3.1b).
- *Similarity*: Similar looking objects are grouped together to form a higher level object (Figure 3.1c).
- *Closure*: Interior of a closed curve is seen as a separate object from the background (Figure 3.1d).
- *Good continuation*: We tend to perceive objects in alignment as smooth, uninterrupted contours (Figure 3.1e).
- *Amodal completion*: If one curve stops another curve (creating a T-junction), we interpret the interrupted curve as a part of the boundary of an object partly occluded (Figure 3.1f).
- *Constant width*: Two parallel curves are perceived as the boundaries of an object with a constant width (Figure 3.1g).
- *Symmetry*: A set of objects which are symmetric with respect to a straight line are grouped together (Figure 3.1h).
- *Convexity*: Any convex curve (not necessarily closed) indicate the boundary of a convex object against the background (Figure 3.1i).
- *Perspective*: A group of concurring lines are perceived as parallel lines in a 3D scene and their meeting point as a vanishing point of the scene (Figure 3.1j).

To perceive a complex visual object, these grouping laws cooperate from small to large scale. As a result, they may agree or conflict with each other along the way. Conflicts give rise to different interpretations and groupings of the same objects. These interpretations can be valid simultaneously (Figure 3.2a) or one interpretation may invalidate the other ones (cf. Figure 3.2b) [23].

In computer vision, when working with a digital image, pixels are the starting points for the gestalt grouping procedure. An image is finite and the data it provides is discrete; therefore, the geometric information extracted from an image is never *certain*. Every detection and localisation of a shape (e.g., lines, angles, curves, polygons) comes with a degree of *precision* [23]. In short, in computer vision, perceptual organisation is the process of evaluating and assigning significance to every potential grouping of features in an image [55].

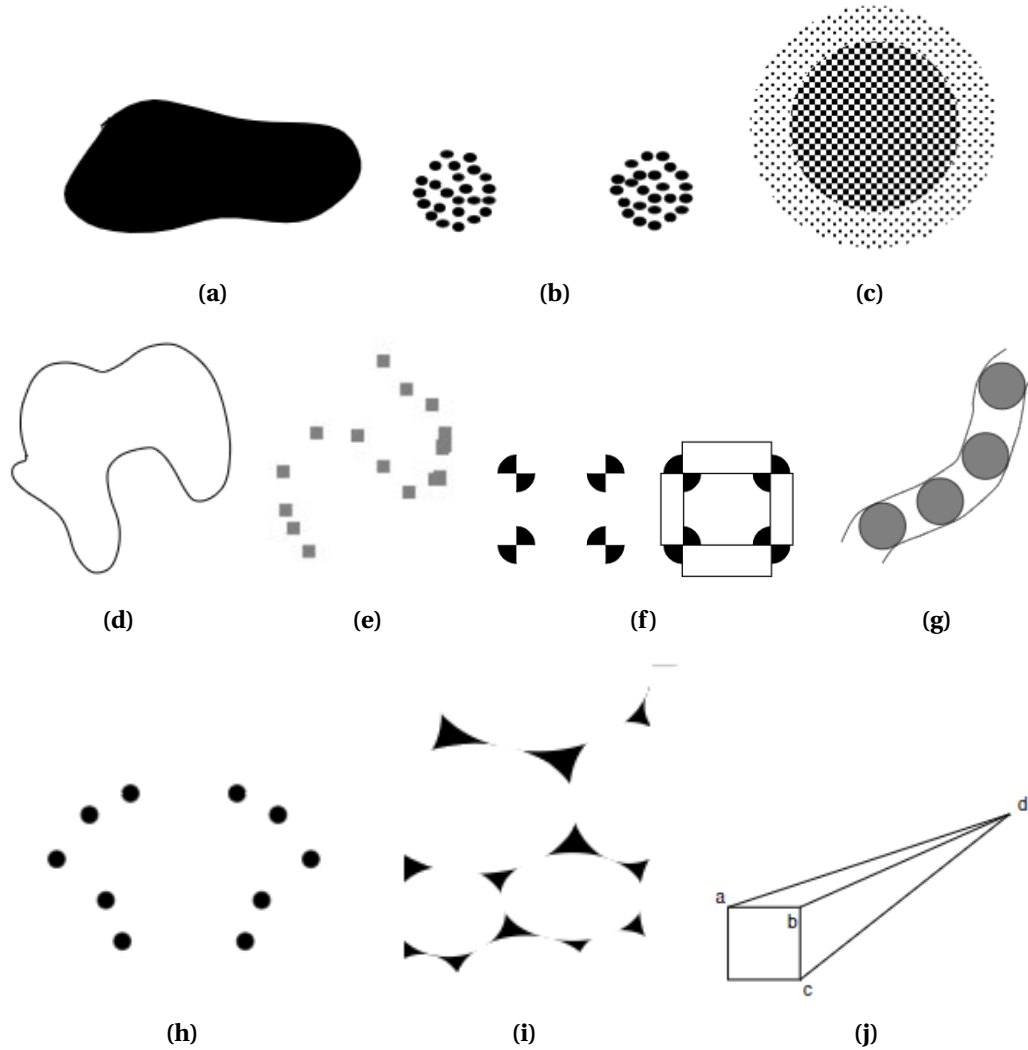


Figure 3.1: Gestalt's grouping laws [23]: (a) the colour constancy law means we see one single black object instead of many connected ones; (b) with the vicinity law we group these objects into two higher level visual objects; (c) we separate this circular area into two regions with different textures according to the similarity law; (d) we perceive a single object against the white background according to the closure law; (e) dark objects are perceived as a curve with the good continuation law; (f) the butterfly shaped dark objects on the left are covered with white rectangles on the right, they are now perceived as disks half occluded by the rectangles, in line with the amodal completion law; (g) we perceive the two parallel curves as the edges of an arm-shaped object with a constant width; (h) the dark objects are symmetrical with respect to a vertical line and perceived together as one object according to the symmetry law; (i) we can interpret the shapes as white ovals on black background or black triangles on white background, the convexity law favours the first option; (j) with the perspective law, we perceive this shape as a 3D object with point d as a vanishing point.

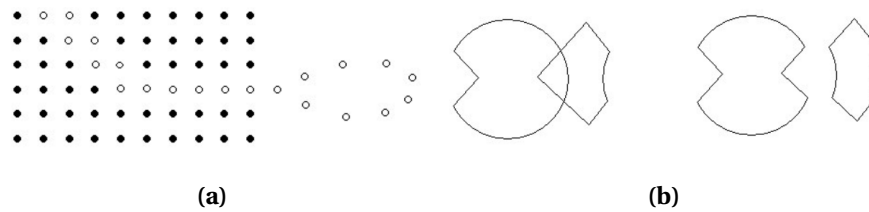


Figure 3.2: Two examples of grouping laws giving rise to different interpretations [23]: (a) the white dots are perceived, simultaneously, as a part of the grid and as a part of a curve; (b) two incompatible interpretations: the shape to the left can be perceived as two overlapping shapes or merge of two symmetrical shapes given on the right side.

3.2 The Helmholtz principle

The Helmholtz principle, in its simplest form, states that we do not perceive any structure in a random image. Alternatively, it says that a structure is perceived when a significant deviation from randomness occurs [6, 23]. Figure 3.3 provides an example of this principle: We can perceive the four segment alignment in Figure 3.3b but not in Figure 3.3a. This is due to the fact that the alignment in Figure 3.3a is not exceptional considering the total number of segments present; i.e. it could have happened by chance. The same cannot be said about Figure 3.3b; here, there are only 31 segments present, so the alignment of four of them cannot be coincidental.

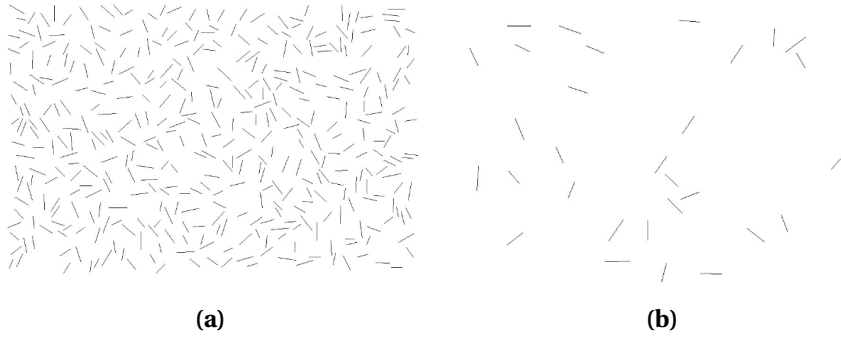


Figure 3.3: An example of Helmholtz principle in action [23]: A set of four aligned line segments exists in both (a) and (b); however it can only be perceived in image (b).

Given a group of n objects O_1, O_2, \dots, O_n in an image, we may observe a common quality in k of these objects. A valid question at this point is to ask “has this quality appeared by chance or is it significant?”. If it is, then we can meaningfully group those objects together. The Helmholtz grouping principle states that if the expectation of observing such arrangement is low enough then their grouping is considered *meaningful* [20]. Then, a very useful process, proposed by A. Desolneux in her seminal work [23], would be:

- to assume a-priori that the considered quality is uniformly and randomly distributed over all n objects;
- under this assumption, check if the observed states of the objects are likely to occur or not;
- if not, surmise *a-contrario* that the observation is the result of a grouping process (a gestalt).

As a result, we can divide image objects or relationships between objects into two sets: those which occur through accident and those which are the result of a meaningful structure [55]. Formally, a meaningful event is defined as:

Definition 2 (ϵ -meaningful event [20]). *An event is ϵ -meaningful if the expectation of the number of occurrences of this event is less than ϵ under the a contrario uniform random assumption. If $\epsilon \leq 1$ then the event is simply meaningful.*

The expectation mentioned in Definition 2 can only be calculated in the context of some assumption regarding the distribution of surrounding objects [55]. This assumption is called the naïve assumption or the naïve model. Knowing the probability of a given arrangement happening by chance, it is obvious that a smaller value for this probability entails a *causal* interpretation for that arrangement [55]. As we are interested to find any causal links in an image, therefore we can simply look for any sign of non-independence. Naturally, a suitable naïve model would be the independently and randomly positioned objects in the image [55, 100]. Obviously, this model is neither accurate nor realistic; it simply describes an image in which no structure will be detected [41].

3.3 The number of false alarms

Assuming a generic quality observable in an image, the Definition 2 can be formulated in the following way. Given the objects $O_i, i \in \llbracket 1, n \rrbracket$ present in an image, the probability of object O_i having the considered quality is denoted by p . Since we have an independence assumption, the probability of at least k objects having the same quality is computed by the tail of the binomial distribution [21]:

$$B(p, n, k) = \sum_{i=k}^n \binom{n}{i} p^i (1-p)^{n-i}. \quad (3.1)$$

Then, the expectation of the considered quality happening by pure chance, or *the number of false alarms (NFA)* is calculated by multiplying Equation 3.1 by the number of tests (N_T) performed on the image. The value of N_T will depend on the considered quality and the way testing is performed on the image. In the end, an event is ϵ -meaningful if [21]:

$$NFA(p, n, k) = N_T B(p, n, k) \leq \epsilon. \quad (3.2)$$

Different arrangements in an image can be evaluated using the NFA criterion. Evidently, we can compare the meaningfulness of two arrangements A and B by comparing their NFA values: A is more meaningful if and only if $NFA_A < NFA_B$. Then, all the arrangements present in an image can be sorted based on their meaningfulness; however, to filter out overlapping arrangements the concept of maximal meaningful events is introduced:

Definition 3 (Maximal meaningful event [20]). *An event A is maximal if*

1. $\forall B \subset A, \text{NFA}(B) \geq \text{NFA}(A),$
2. $\forall B \supset A, \text{NFA}(B) > \text{NFA}(A).$

A is maximal meaningful if it is both maximal and meaningful.

Maximal meaningful events, i.e. lines, segments, edges, clusters, etc. form the results of an a-contrario detection process on an image.

3.4 Applications of the a-contrario framework in computer vision

The a-contrario framework introduced by Desolneux et al. [20] has proven its efficiency in image and video analysis in a variety of detection problems. Generally, the detection is performed by rejecting a *naive model* which describes the statistic of the unstructured data also known as a noise model.

For example, in texture analysis, [41] introduces a noise model with and without colour for detection of spots in a textured background in two applications: medical mammograms and stains on pieces of clothing.

In motion detection, [91] proposes a naive model to describe a scene without any moving objects. Then it uses the a-contrario framework to detect and localise the moving objects in the image. [92] extends this method to work with a temporal image sequence of moving objects.

Another application is for edge and line detection. Several methods use the naive assumption that observing an edge is very unlikely in a randomly and independently distributed image. In [98], the authors propose the addition of high level features to the a-contrario framework by using 'edgelets' (a set of connected pixels) instead of individual pixels. In a similar work, [2] proposes *EDLines*: a fast line segment detector which includes a line validation step based on the Helmholtz principle to help prune any false detection. Another work in quality inspection field, [3] proposes a crack detection algorithm based on a-contrario modelling which is robust to motion blur and works with different crack shapes. Finally, [54] proposes a line segment detector for SAR (Synthetic Aperture Radar) images which is able to withstand the strong multiplicative noise characteristic of these images. To achieve this, they introduce a novel background model which is specific to SAR images instead of the existing models which work with optical images. This model takes into account the spatial dependencies between local orientations using Markov chain (as opposed to the Independence assumption).

The a-contrario framework has also been used in structure from motion (SfM) algorithms. In [66], the authors use the a-contrario methodology to propose adaptive thresholds for model estimation in SfM in place of the usual globally-fixed thresholds. They reach a better precision using adaptive thresholds and remove the need for an initial guess for the threshold values.

In a similar fashion, gestalt grouping principles have been used in computer vision for tasks related to the higher level perceptual organisation of scenes. For example, [62] uses large scale perceptual grouping principles in combination with pixel-wise spectral analysis to process geographic thermal data. In another work, [107] uses gestalt laws to create a model of visual attention from low level to high level (a bottom-up model).

In the current study, we are interested in the applications dealing with the detection of changed areas across multi-temporal images. In early studies, the a-contrario framework and spectral invariant features have been used to detect meaningful changes between two satellite images of the same area taken at different times [53]. An a-contrario approach has also been proposed for change detection in three dimensional multi-modal medical images such as Magnetic Resonance sequences [78]. In another study, [76] uses the a-contrario framework for the definition of a criterion assessing the level of coherence in a sequence of images for detecting sub-pixel changes in a time-series of satellite images. Similarly, [35] further investigates this approach by using exchangeable random variables instead of relying on the independent and identically distributed assumption. All these works focus on the grey level values (and their changes) so that the considered naive model deals with grey level discrepancy.

3.5 Change detection using the a-contrario framework

In this section, to demonstrate the power of the a-contrario framework, we present an initial solution to our change detection problem. The process consists of two different naive models and two separate computations of NFA which are based on existing works of Desolneux et. al [21] and Robin et al. [76]. The following is written based on our paper named “Detecting alterations in historical violins with optical monitoring” [74].

In order to instantiate the a-contrario perception concept through a NFA criterion, two elements have to be defined: the ‘naive’ model that represents the statistics of the model to reject (the H_0 hypothesis) and the feature on which these statistics apply. Both depend on the considered data. However, since the naive model represents the absence of structure, we can choose it as representing the *wide spreading* of the samples, so that it will be rejected once the observations appear

unlikely close with respect to the naive model.

Dealing with change detection, the decision of a change is due to the observation of a surprisingly high density of differences within local features. Such a definition can be interpreted as gathering two criteria: at pixel scale, high differences in feature images and, at region/area scale, high density of previously detected ‘high differences’. In other words, we propose a two-step approach that firstly detects seeds as pixels likely to belong to a change area, and secondly detects dense areas of seeds. For each step, we use a specific NFA criterion.

3.5.1 Seed detection

Starting from the colour difference image ΔI (Section 1.6) defined on the pixel domain $\mathcal{P} \subset \mathbb{N}^2$, we consider the naive model \mathcal{M}_{col} to derive the set of seeds, called $\mathcal{S} \subseteq \mathcal{P}$, representing the pixels likely to belong to the changed areas. Specifically, denoting by $|X|$ the cardinality of a set X ,

Definition 4 (Naive model \mathcal{M}_{col}). *The image ΔI is a random field of $|\mathcal{P}|$ independent centred Gaussian variables $\mathcal{N}(0, \sigma^2)$.*

According to \mathcal{M}_{col} , the distribution of the sum of the squared values (SSV) on a sub-domain $\mathcal{D} \subseteq \mathcal{P}$, $v_{\mathcal{D}} = \sum_{s \in \mathcal{D}} [\Delta I(s)]^2$, is a χ^2 law with $|\mathcal{D}|$ degrees of freedom. Then, the probability $\mathbb{P}_{\mathcal{M}_{col}}(v_{\mathcal{D}}, \sigma)$ of observing a SSV lower than $v_{\mathcal{D}}$ by chance is given by the regularised incomplete Gamma function, and the Number of False Alarms associated to a sub-domain \mathcal{D} having $v_{\mathcal{D}}$ SSV is [3, 76]

$$\text{NFA}_1(\mathcal{D}, \sigma, |\mathcal{P}|) = |\mathcal{P}| \binom{|\mathcal{P}|}{|\mathcal{D}|} \mathbb{P}_{\mathcal{M}_{col}}(v_{\mathcal{D}}, \sigma) \quad (3.3)$$

where $\binom{a}{b}$ is the binomial coefficient.

Then, minimising NFA defined by Equation (3.3), the result depends on the parameter σ that controls the noise level in \mathcal{M}_{col} . In this study, similarly to prior works [3, 76], it is computed by calculating the second moment of the image: $\sigma^2 = \mathbb{E}(x - \mu)^2$ where μ is the statistical mean of the pixel values of the image ΔI . Therefore, Equation 3.3 will turn to:

$$\text{NFA}_1(v_{\mathcal{D}}, |\mathcal{D}|, \sigma, |\mathcal{P}|) = |\mathcal{P}| \binom{|\mathcal{P}|}{|\mathcal{D}|} \frac{1}{\Gamma\left(\frac{|\mathcal{D}|}{2}\right)} \int_0^{\frac{v_{\mathcal{D}}}{2\sigma^2}} e^{-t} t^{\frac{|\mathcal{D}|}{2}-1} dt, \quad (3.4)$$

where $\Gamma(x)$ is the gamma function.

Assuming $\hat{\mathcal{D}} = \text{argmin}_{\mathcal{D} \subseteq \mathcal{P}} \text{NFA}_1(\mathcal{D}, \sigma, |\mathcal{P}|)$. Since the naive model represents the inconsistency in the data, $\hat{\mathcal{D}}$ is the set of pixels that are ‘surprisingly’ structured under the naive model assumption \mathcal{M}_{col} , i.e. the pixels presenting ‘surprisingly’ low

ΔI values, so that the set of seeds \mathcal{S} is the complementary of $\hat{\mathcal{D}}$ with respect to set \mathcal{P} : $\mathcal{S} = \mathcal{P} \setminus \hat{\mathcal{D}}$. Algorithm 1 explains this process in more detail.

3.5.2 Clustering the seeds

Then, having derived \mathcal{S} and represented it under the form of a binary image, we aim to detect the most significant cluster(s) of seeds. In this study, we compare two approaches: the first one assumes a parametric geometric shape of the changed areas (e.g. rectangular tiles, strips, rings, etc. like in Le Hégarat-Masclé et al. [51]), while the second approach considers a general shape clustering scheme proposed in Desolneux et al. [21]

In both cases, the considered naive model \mathcal{M}_{bin} represents the absence of spatially consistent subset(s) of seeds. Specifically,

Definition 5 (Naive model \mathcal{M}_{bin}). *The set of seeds \mathcal{S} is a random set of $|\mathcal{S}|$ independent uniformly distributed variables over the image lattice \mathcal{P} .*

Under uniform distribution model \mathcal{M}_{bin} , denoting by p_O , the prior probability that a seed belongs to a parametric object O , the probability $\mathbb{P}_{\mathcal{M}_{bin}}(p_O, |\mathcal{S}|, \kappa)$ of observing κ seeds within O by chance is given by the tail of the binomial distribution, and the Number of False Alarms [21] is

$$NFA_2(p_O, |\mathcal{S}|, \kappa) = N_{test} \sum_{i=\kappa}^{|\mathcal{S}|} \binom{|\mathcal{S}|}{i} p_O^i (1 - p_O)^{|\mathcal{S}|-i} \quad (3.5)$$

In the previous equation, p_O is estimated by the ratio between the area of object O with respect to the whole image area. Note the slight difference with a NFA like in [25] derived assuming a Bernoulli distribution of parameter p for pixel binary values, so that the probability to have a given number κ of seeds among a given number $\#O$ of pixels is a Binomial distribution of parameter p and $NFA_2(p, \#O, \kappa) = N_{test} \sum_{i=\kappa}^{\#O} \binom{\#O}{i} p^i (1 - p)^{\#O-i}$, with p approximated by the ratio between the seed and the pixel numbers and $\#O$ the pixel number of object O .

In the case of a clustering approach, instead of constraining the object in terms of parametric form, a thick low resolution curve free of any seed and surrounding the object is required. Thus, denoting by \mathcal{C} a cluster, its relative area $a(\mathcal{C})$ with respect to the whole image area is also the probability of a seed to belong to \mathcal{C} under the naive model \mathcal{M}_{bin} , whereas the probability of a seed not to belong to \mathcal{C} is $1 - a(\mathcal{C}) - a(\delta\mathcal{C})$, where $a(\delta\mathcal{C})$ is the relative area of the empty thick low resolution contour surrounding \mathcal{C} . Transposing Desolneux' formula [21] with our notations,

$$NFA_2(|\mathcal{C}|, |\mathcal{S}|, a(\mathcal{C}), a(\delta\mathcal{C})) = M_{test} \sum_{i=|\mathcal{C}|}^{|\mathcal{S}|} \binom{|\mathcal{S}|}{i} [a(\mathcal{C})]^i [1 - a(\mathcal{C}) - a(\delta\mathcal{C})]^{|\mathcal{S}|-i} \quad (3.6)$$

Algorithm 1 Estimation of seeds: pixels likely to be changed. Input: grey-level difference image ΔI

```

1:  $n \leftarrow |\Delta I|$ 
2:  $v \leftarrow$  a vector of size  $n$ 
3:  $v \leftarrow \Delta I. / \max(\Delta I)$ 
4:  $v \leftarrow v^2$ 
5:  $v \leftarrow \text{sort}(v)$ 
6: for  $i \leftarrow 2 : n$  do
7:    $v(i) \leftarrow v(i) + v(i - 1)$ 
8:    $\text{nfa}(i) \leftarrow$  compute NFA using the Equation 3.4 with its parameters set as
      $(v(i), i + 1, \sigma, n)$ 
9: end for
10:  $(\text{minNFA}, \text{minNFAIndex}) \leftarrow \min(\text{nfa})$ 
11:  $\hat{\mathcal{D}} \leftarrow$  set of  $\text{minNFAIndex}$  pixels with the lowest values in  $\Delta I$ 
12:  $\mathcal{S} \leftarrow$  pixels in  $\Delta I$  but not in  $\hat{\mathcal{D}}$ 

```

In Eq. (3.5) and (3.6), N_{test} and M_{test} are the numbers of tests that control the average number of false alarms [23]. Conversely to the case of the first NFA (cf. Section 3.5.1), here, like in Desolneux et al. [21], we take these numbers constant for a given image, i.e. independent of O of \mathcal{C} , so that they are not involved in NFA minimisation.

Numerically, each cluster \mathcal{C} is formed by traversing the minimum spanning tree created from the seed points \mathcal{S} . Then using Eq. 3.6, for each cluster we compute the meaningfulness (Equation 3.7). Finally, the detected areas are separate clusters with maximum meaningfulness.

$$S(|\mathcal{C}|, |\mathcal{S}|, a(\mathcal{C}), a(\delta\mathcal{C})) = -\log(\text{NFA}_2(|\mathcal{C}|, |\mathcal{S}|, a(\mathcal{C}), a(\delta\mathcal{C}))) \quad (3.7)$$

3.6 Conclusion

In this chapter, we presented a theoretical background on the a-contrario framework in computer vision and its roots in Gestalt psychology. Gestalt principles of perception govern how we group structures together and make higher level objects. We introduced the Helmholtz principle which states that we perceive a structure when a significant deviation from randomness occurs (in a live scene or a digital image). This helps computer vision researchers to judge whether or not an arrangement in an image is meaningful or has happened by chance.

In addition, we iterated several applications of the a-contrario framework for detection tasks such as the detection of lines, edges and textures. In relation to our problem of change detection, we presented a two-step approach for clustering

a grey level image. This method is based on existing literature on the a-contrario framework and uses two separate NFA computation processes.

In the next chapter, we propose a one-step clustering algorithm along with extensive quantitative evaluations using simulated and real images.

Chapter 4

A-contrario framework for cluster detection

Contents

| | |
|---|-----------|
| 4.1 Clustering in one step | 56 |
| 4.1.1 Grey-level transformation | 56 |
| 4.1.2 Distance between two points | 58 |
| 4.1.3 Number of False Alarms | 59 |
| 4.1.4 Calculating the lower and upper volumes | 60 |
| 4.1.5 Most meaningful clusters | 60 |
| 4.1.6 Implementation | 61 |
| 4.2 Robustness evaluation using simulated data | 65 |
| 4.3 Performance on actual data | 68 |
| 4.3.1 Evaluation regarding 3D clustering | 68 |
| 4.3.2 Comparison with 2D segmentation | 70 |
| 4.4 Conclusion | 76 |

In Section 3.5, we described a two step clustering method based on two separate naive models and computations of NFA. This approach works well enough as long as the seed detection step does not miss any wear pixels. To make the wear detection process more consistent and robust to noise, we propose a one step approach in which the clustering takes into account the spatial *and* the spectral features at the same time. This way, the difference image ΔI is used as an input straight to the clustering algorithm without the need to first use a thresholding method on the values.

This chapter is written based on our paper “A-contrario framework for detection of alterations in varnished surfaces” [75] published in the journal of Journal of Visual Communication and Image Representation (JVCIR).

4.1 Clustering in one step

Our problem boils down to segmenting the difference image ΔI (Section 1.6), with respect to semantic classes, one of which representing the unchanged area. As previously stated, we propose to rely on a single naive model which will account for both radiometric and spatial criteria characterising a wear area that is present in ΔI_i images.

The basic idea is to extend the meaningfulness concept specifying that a cluster is all the more significant that it is very dense (i.e., its points are ‘surprisingly’ close) not only spatially but also in terms of grey-level differences. Now, to include grey-level features in a-contrario detection, we could either adapt the naive model to grey-level features in case of unstructured data (no change in our case), or adapt the grey-level values so that the uniform distribution can be used as naive model as usual. In this work, we adopt the second approach.

Considering grey-level differences, low values correspond (mainly) to no change and high values (mainly) to changes, so that a grey-level transform is required to meet the assumption that a change can be detected as surprisingly structured or dense. Then, using the cluster NFA based on distance (described in Section 3.5.2), the proposed method also needs the specification of the considered distance. These two points are presented in the next subsections before the presentation of the NFA computation and the cluster detection algorithm.

4.1.1 Grey-level transformation

Let us first enumerate the desirable properties of the required grey-level transformation for ΔI pixel values: following the transformation, (i) the grey-level values of pixels belonging to unchanged areas should be stretched, (ii) the grey-level values of pixels belonging to change areas should be similar and (iii) close to zero. This last property aims at controlling not only the relative values of grey-level differences but also their absolute values. Then, the grey-level function (f) that we can consider has to:

- be decreasing;
- spread not significant grey-level differences so that uniform distribution will be acceptable.

In this study, two f functions, denoted here by f_1 and f_2 , were evaluated: $\forall x \geq \tau, f_1(x) = \frac{1}{x-\tau}, f_2(x) = 1 + \tanh(\tau - x), \forall x < \tau, f_1(x) = f_2(x) = +\infty$. The parameter τ has been introduced to allow us to control the number of points considered in the following steps of the algorithm. Indeed, the values of ΔI which are lower than τ will result in an infinite distance (cf. Section 4.1.2) so that they are simply discarded; therefore, the higher the τ the less points the algorithm considers. Note that, in order to remain parameter free (but at the expense of memory and computational resources), one can set this parameter equal to zero.

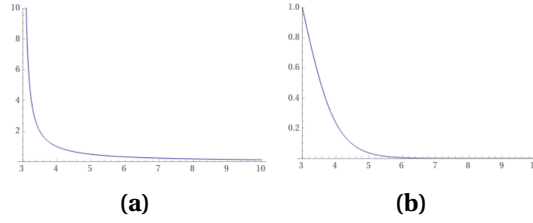


Figure 4.1: Comparison for $\tau = 3.0$ between (a) $f_1(x) = \frac{1}{x-\tau}$ and (b) $f_2(x) = 1 + \tanh(\tau - x)$.

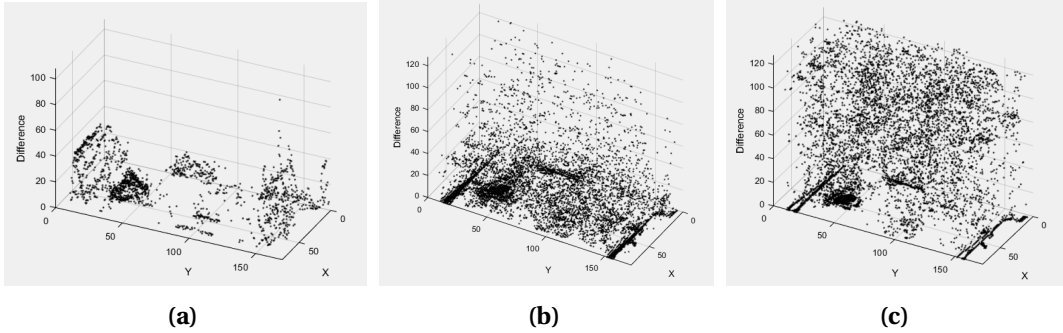


Figure 4.2: 3D point cloud (a) before applying the function f , (b) after applying $f_1(x) = \frac{1}{x-\tau}$ and (c) after applying $f_2(x) = 1 + \tanh(\tau - x)$. The vertical axis represents the (transformed) grey-level values while the other two axis originate from the 2D image plane.

We focus on these two functions since, while having the desirable properties, the resulting value spread is quite different. The inverse function gives a gradual decline while the tanh function provides a more sudden drop for high values (see Figure 4.1). The point clouds provided considering respectively each of the two functions are shown in Figure 4.2; both provide the expected discrepancy of the difference values, the tanh function result appearing somehow more uniformly distributed. Therefore, in our experiments, we use the tanh function allowing better consistency with the considered naive model.

4.1.2 Distance between two points

The cluster detection using the a-contrario approach is based on point distance [23]. In order to take into account both spatial proximity and (transformed) grey-level differences, the proposed distance is a weighted sum of two terms: the 2D spatial distance and a term representing the modified grey-level of each point. Thus, the proposed distance is defined as the minimal path length among the paths relating two points and passing through the $z = 0$ plane with z being the grey-level axis. In this way, we can enforce that points with higher grey-level values after f -transformation be considered farther apart compared to points with lower f -transformed grey-level values. Since the grey-level values and spatial distance are inherently in different scales, we use the scale factor $c \in \mathbb{R}_+$ to control the weight of the spatial term with respect to the grey-level one in the distance definition. The choice of c value is further discussed in Section 4.1.6.

Denoting by y_i the value at pixel $i \in \mathcal{P}$, by $z_i = f(y_i)$ its transformed grey-level value and by $D_{sp}(i, j)$ the 2D spatial distance between the locations of pixels i and j , $\forall (i, j) \in \mathcal{P}^2$, if $i = j$, $D(i, j) = 0$, and otherwise

$$D(i, j) = \sqrt{(D_{sp}(i, j))^2 + c \times (z_i^2 + z_j^2)}. \quad (4.1)$$

Let us specify that, without the constraint “if $i = j$, $D(i, j) = 0$ ”, D would be only a meta-metric. Specifically, among the three properties that a distance metric should satisfy: 1. symmetry, 2. identity of indiscernibles and 3. triangle inequality, the second one is not verified. For triangle inequality, using the positiveness of D , then the fact that 2D spatial distance (Euclidean or approximation) is a metric and finally the positiveness of square function, we have: $\forall (i, j, k)$ three different pixels,

$$\begin{aligned} (D(i, j) + D(j, k))^2 &= D^2(i, j) + D^2(j, k) + 2D(i, j) \times D(j, k), \\ &\geq D^2(i, j) + D^2(j, k), \\ D^2(i, j) + D^2(j, k) &= D_{sp}^2(i, j) + D_{sp}^2(j, k) + c \times (z_i^2 + z_j^2 + z_j^2 + z_k^2), \\ &\geq D_{sp}^2(i, k) + c \times (z_i^2 + 2z_j^2 + z_k^2), \\ &\geq D_{sp}^2(i, k) + c \times (z_i^2 + z_k^2). \end{aligned}$$

Therefore $(D(i, j) + D(j, k))^2 \geq D^2(i, k)$, from which we get $D(i, j) + D(j, k) \geq D(i, k)$ since the square function is an increasing function.

However, the identity of indiscernibles does not hold since $D(i, j) = 0 \Rightarrow i = j$, but the opposite ($i = j \Rightarrow D(i, j) = 0$) is not true except for pixels such that $z_i = z_j = 0$.

Finally, a cluster $\mathcal{C} \subseteq \mathcal{P}$ is defined as a set of close points with respect to the

distance value d : $\mathcal{C} \subseteq \mathcal{P}$ is the set of points i such that

$$\forall i \in \mathcal{C}, \begin{cases} \exists j \in \mathcal{C} \text{ s.t. } D(i, j) \leq d, \\ \forall j' \in \mathcal{P} \setminus \mathcal{C}, D(i, j') > d. \end{cases}$$

Note that for a given d there may be several distinct clusters satisfying the previous definition. Inversely, for a given cluster \mathcal{C} , there is a range of distances leading to \mathcal{C} that allows us to associate an inner border and an outer border to cluster \mathcal{C} . In the following, we denote $d_{min}(\mathcal{C})$ and $d_{max}(\mathcal{C})$ the bounds of this interval.

4.1.3 Number of False Alarms

The Number of False Alarms (NFA) is based on the considered naive model which in our case is the uniform distribution:

Definition 6 (Naive model \mathcal{M}). *The set of points \mathcal{S} is a random set of $|\mathcal{S}|$ independent uniformly distributed variables over the 3D (2D + grey-level) space of the image.*

Note that a key point of a-contrario approaches is that the naive model does not have to be accurate, but it only has to be contradicted in the case of the target structured data (wear in our application).

The Number of False Alarms is computed by extending the NFA proposed in [21] for 2D cluster detection. Considering here a 3D space, the 2D surface areas are replaced by 3D volumes and the 2D distance by the distance defined in Equation (4.1) so that, for any cluster \mathcal{C} of 3D (2D + grey-level) points,

$$\text{NFA}_{\mathcal{M}}(\mathcal{C}, M) = N_{test} \sum_{i=k}^M \binom{M}{i} \underline{V}_{\mathcal{C}}^i (1 - \bar{V}_{\mathcal{C}})^{M-i}, \quad (4.2)$$

where k is the number of points in the cluster, M is the total number of points and $\underline{V}_{\mathcal{C}}$ and $\bar{V}_{\mathcal{C}}$ are the lower and upper bounds of the relative volume of the cluster with respect to the whole image cube volume. Therefore $1 - \bar{V}_{\mathcal{C}}$ represents the volume of the region that is definitely outside the cluster while $\underline{V}_{\mathcal{C}}$ represents the volume of the region which is definitely inside the cluster. These volumes are obtained by relying on morphological operations as specified in the next section. Finally, N_{test} that is a normalisation term equivalent to the number of tests coefficient is set as a constant so that it does not impact the NFA minimisation and can be discarded if one is interested only in the cluster ordering with respect to their NFA-based meaningfulness, which is actually our case.

4.1.4 Calculating the lower and upper volumes

Let us first define the distance between a cluster \mathcal{C} and a single point i as the minimum distance between all the points in \mathcal{C} and i :

$$D(i, \mathcal{C}) = \min_{j \in \mathcal{C}} D(i, j).$$

Then, we define $\delta(\mathcal{C})$ as the radius of the cluster \mathcal{C} . Let $\mathcal{P}(i, j)$ be a path between two given points i and j , i.e. an ordered list of successive points with the first one being i and the last one being j : $\mathcal{P}(i, j) = (k_0, k_1, \dots, k_l)$, $k_0 = i$, $k_l = j$; and assume k_n and k_{n+1} are any two consecutive points on the path $\mathcal{P}(i, j)$. Then, we denote by $\delta(\mathcal{P})$ the maximum distance between two consecutive points on \mathcal{P} : $\delta(\mathcal{P}) = \max_{n \in \{1, |\mathcal{P}|\}} D(k_{n-1}, k_n)$ the value for δ is computed as follows:

$$\delta(\mathcal{C}) = \max_{(i, j) \in \mathcal{C}^2} \left(\min_{\mathcal{P}(i, j)} (\delta(\mathcal{P}(i, j))) \right).$$

In practice, $\delta(\mathcal{C})$ can be computed more easily using an iterative hierarchical structure for all the possible clusters present in the data. Section 4.1.6 describes this process in more detail.

We also define $\delta'(\mathcal{C})$ as the distance between \mathcal{C} and the closest point outside \mathcal{C}

$$\delta'(\mathcal{C}) = \min_{j \notin \mathcal{C}} D(j, \mathcal{C}).$$

The lower and upper volumes of \mathcal{C} are computed by performing a 3D mathematical morphological dilation [84]:

- The lower region is the dilation of the union of the points in \mathcal{C} by a ball structuring element having the radius $\delta/2$. Let us then denote by $\underline{V}_{\mathcal{C}}$ the volume of this region divided by the volume of the image cube.
- The upper region is the dilation of the union of the points in \mathcal{C} by a ball structuring element having the radius δ' . Let us then denote by $\overline{V}_{\mathcal{C}}$ the volume of this region divided by the volume of the image cube.

It is worth noting that since we have used a modified distance formula (Equation 4.1), all dilation operations have to be done using this custom distance. Further details will be discussed in Section 4.1.6.

4.1.5 Most meaningful clusters

After each cluster has an assigned NFA, we compute the meaningfulness for each cluster:

$$\mathcal{S}_{\mathcal{M}}(\mathcal{C}, \mathcal{M}) = -\log(\text{NFA}_{\mathcal{M}}(\mathcal{C}, \mathcal{M})). \quad (4.3)$$

In the following, only comparing cluster significance values at given value M and naive model \mathcal{M} , we shorten significance notation as $\mathcal{S}(\mathcal{C}) = \mathcal{S}_{\mathcal{M}}(\mathcal{C}, M)$.

By construction of the minimum spanning tree, for any pair of considered clusters \mathcal{C} and \mathcal{K} , either $\mathcal{C} \cap \mathcal{K} = \emptyset$ or $\mathcal{C} \subsetneq \mathcal{K}$ or $\mathcal{K} \subsetneq \mathcal{C}$. Then, to avoid redundant results (detection of the same cluster several times), we focus on *maximal* clusters such that a cluster $\mathcal{C} \in \mathcal{P}$ is said *maximal* if [21]

$$\begin{cases} \forall \mathcal{K} \subsetneq \mathcal{C}, \mathcal{S}(\mathcal{K}) < \mathcal{S}(\mathcal{C}), \text{ and} \\ \forall \mathcal{K} \supsetneq \mathcal{C}, \mathcal{S}(\mathcal{K}) \leq \mathcal{S}(\mathcal{C}). \end{cases}$$

In Algorithm 2, meaningfulness *maximality* is handled as a constraint: to be added to the list of *maximal meaningful* clusters \mathbf{C} , a cluster must not intersect any of the clusters already in \mathbf{C} .

4.1.6 Implementation

The algorithm starts with the creation of a minimum spanning tree using the points derived from the difference image ΔI (computed using Equation (4.1)). The spanning tree is constructed as follows [21]: we initialise a graph whose nodes are finite-coordinate points, there are no edges, and all distances between pairs of points are pre-computed. The spanning tree is then constructed in an iterative way, during which, at each iteration

- (i) we select the two nearest nodes among the unconnected nodes and
- (ii) we create an edge between these two nodes.

For an easier derivation of clusters of connected points, we also introduce a hierarchical representation of this iterative process, in which, at each iteration, the nearest nodes A and B are merged in a parent node which stores as well, the minimum distance between pairs of points one in A and one in B . In the algorithm 2, we call a node along with all its children nodes a subtree.

Then, each node is considered as a potential cluster. As mentioned earlier, for each cluster, we compute two separate dilations, one being performed for the lower bound and one for the upper bound region. These dilations are done in 3D by using Equation (4.1) as the distance. As expected, this step produces different shapes than those obtained by using the standard dilation based on a 3D-ball structuring element or Euclidean distance. The volumes of these regions are involved in the computation of the NFA and of the meaningfulness values for each cluster. For saving computational time (and since preliminary tests did not show a difference in the derived ordering of the clusters), we only compute NFA up to scale owing to N_{test} which boils down to deriving meaningfulness values up to a shift. The

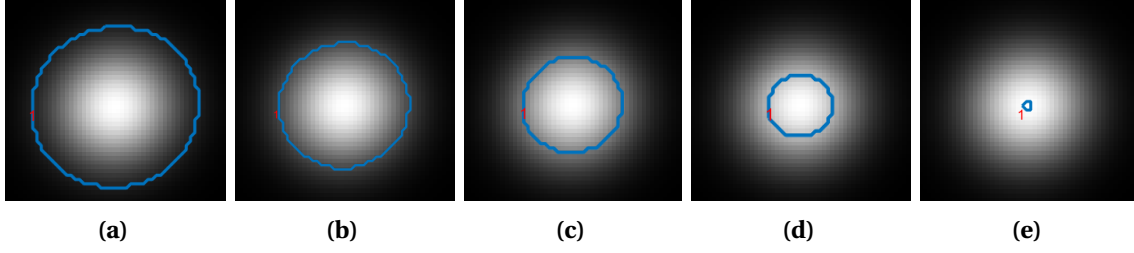


Figure 4.3: Detection of a single cluster when (a) $c = 0.001$, (b) $c = 0.1$, (c) $c = 1$, (d) $c = 10$ and (e) $c = 500$; with the meaningfulness values of (a) 1228.90, (b) 466.06, (c) 161.74, (d) 7.35, (e) 1.62. The blue circle and the red number indicate the detection of only one cluster.

final step is to find the maximal clusters. Starting with an empty set \mathbf{C} and the list of clusters \mathcal{C}_i ranked in decreasing order of meaningfulness, at each step, we increment i to select the next cluster \mathcal{C}_i , and, only if it is disjoint from any cluster already stored in \mathbf{C} , we add it to \mathbf{C} . In the end, \mathbf{C} represents the output of the algorithm which is the ranked list of detected clusters based on their \mathcal{S} value.

Let us finally provide some practical notes:

- Regarding the c parameter in Equation (1), it allows us to weight the spatial term with respect to the radiometric one in the distance definition. When $c \gtrsim 0$, the clusters would be spatially large and very dense including pixels j almost irrespective of their grey-level transformation value $f(y_j)$. Conversely, when $c \gg 0$, the clusters would be spatially scattered and very sparse only including pixels j with very low values $f(y_j)$. This behaviour is illustrated on Figure 4.3 which represents a 2D Gaussian function. In practice, c parameter is set based on image spatial and radiometric features. Based on performed experiences, we set $c = 0.1$ as default value (used in all our experiments).
- When computing Equation (4.2), intermediate values of $\binom{M}{k}$ can get very large and generate overflows. In cases with a small M (around 2000 or lower), we can deal with this by using Big number data types. Otherwise, instead we can approximate Equation (4.3) using the Hoeffding approximation like in [24]:

$$-\log(\text{NFA}_{\mathcal{M}}(M, k, \underline{V}_{\mathcal{C}}, \bar{V}_{\mathcal{C}})) \approx M \left[\frac{k}{M} \log \left(\frac{k}{M * \underline{V}_{\mathcal{C}}} \right) + \left(1 - \frac{k}{M} \right) \log \left(\frac{1 - k/M}{1 - \bar{V}_{\mathcal{C}}} \right) \right] \quad (4.4)$$

- Decreasing the quantisation level for the grey-level values (e.g. 128 levels instead of 256) can help improve the computational complexity of the algorithm by reducing the number of calculations in the 3D morphological operations.
- In practice, the parameter τ can also be set to higher values to reduce the number of points M (cf. Equation (4.2)). This will not affect the output as long

as the omitted points have colour difference values less than the minimum amount of difference perceivable (this value depends on the application, amount of noise present and the colour difference formula used). In all our experiments, we set $\tau = 3$ which allows us to focus on only 20% of the pixels (which is still much more important than the wear areas that represent only up to a few percents of the whole image).

Algorithm 2 Change detection between the current frame I and the reference frame I_0 .

```

1: Compute the colour difference map  $\Delta I$  between  $I$  and  $I_0$ 
2: for each pixel  $j$  in  $\Delta I$  do
3:    $\Delta I(j) = f(\Delta I(j))$ 
4: end for
5:  $\mathcal{P} \leftarrow$  3D points derived from pixels  $j$  such that  $\Delta I(j) < +\infty$ 
6:  $M \leftarrow |\mathcal{P}|$ 
7: for each pair of points  $i$  and  $j$  in  $\mathcal{P}^2$  do
8:   Compute  $D(i, j)$  according to Eq. (4.1)
9: end for
10: Compute the minimum spanning tree for the  $\mathcal{P}$  points based on  $D(i, j)$  so
    that each subtree in hierarchical representation stores in its root the distance
    between its two children.
11:  $V_{\mathcal{P}} \leftarrow$  volume of image cube
12: for each subtree  $T$  do
13:    $\mathcal{C} \leftarrow$  cluster of points in  $T$ 
14:    $\delta \leftarrow$  value stored in the root of  $T$ 
15:    $\delta' \leftarrow$  value in the parent node of  $T$ 
16:    $\underline{V}_{\mathcal{C}} \leftarrow [volume\ of\ dilate(\mathcal{C}, \delta/2)]/V_{\mathcal{P}}$ 
17:    $\bar{V}_{\mathcal{C}} \leftarrow [volume\ of\ dilate(\mathcal{C}, \delta')]/V_{\mathcal{P}}$ 
18:    $k \leftarrow$  the number of points in  $\mathcal{C}$ 
19:   Compute NFA value (up to scale owing to  $N_{test}$ ) according to Eq. (4.2) using
    values  $k$ ,  $M$ ,  $\underline{V}_{\mathcal{C}}$  and  $\bar{V}_{\mathcal{C}}$ 
20:    $\mathcal{S} \leftarrow -\log(\text{NFA})$ 
21: end for
22:  $\mathcal{J} \leftarrow$  list of indices of the clusters sorted according to  $\mathcal{S}$ 
23:  $\mathbf{C} \leftarrow \emptyset$ 
24: for each index  $j$  in  $\mathcal{J}$  do
25:    $\mathcal{C}_j \leftarrow j^{th}$  cluster according to  $\mathcal{J}$ 
26:   if  $\forall \mathcal{C}_l \in \mathbf{C}, \mathcal{C}_j \cap \mathcal{C}_l = \emptyset$  then
27:      $\mathbf{C} \leftarrow \mathbf{C} \cup \{\mathcal{C}_j\}$ 
28:   end if
29: end for
30:  $\mathbf{C}$  is the list of detected clusters

```

4.2 Robustness evaluation using simulated data

One of the beneficial features of the a-contrario framework is its robustness to noise. To evaluate our proposed algorithm against variable amounts of noise, we use simulated data in the next two experiments. This way, we can control the level of the noise present in the difference image and compare the result to an exact ground truth; two features that are not possible with real captured data.

To produce the simulated data, firstly, we created a binary map which serves as ground truth to distinguish the background and the foreground. The foreground region was hand drawn to represent two clusters with complex shapes. Then, the pixel values in both regions were randomly selected following two different heavy-tailed distributions. We chose a heavy-tailed distribution because a) it simulates the kind of noise present in the background that is more disruptive than Gaussian noise for instance and b) it allows us to illustrate that the naive model \mathcal{M} (Definition 6) does not need to be exact. Specifically, since the drawn values have to simulate colour difference values that are strictly non-negative, we focused on a Nakagami distribution in both cases (however any similar distribution with the adequate parameters can be used here). For the foreground, we use a Nakagami distribution plus a *shift* value which allows us to easily control the *mean* without changing the shape of the distribution. To represent realistic data with respect to our application, the background is a spread out distribution near zero and a tail with high values while the foreground, much less spread out, has a higher mean value than the background. Then, the aim is to detect the two clusters present in the image and to evaluate the result using the binary map as the ground truth.

For a quantitative evaluation of the proposed algorithm, we simulate data considering a large range of mean and standard deviation values for the foreground and the background. Specifically, we consider the two following experiments:

- Experiment 1: we increase progressively the spread of the background with respect to fixed foreground distribution parameters;
- Experiment 2: we vary the mean value of the foreground with respect to fixed background distribution parameters.

The Nakagami distribution has two parameters, denoted μ and ω , which control the *shape* and *spread* of the distribution, respectively. In the first experiment, we change the spread ω_0 of the background (from 2.0 to 11.0 by steps of 0.5; and then to 15 by steps of 1.0) while keeping the shape of the background μ_0 constant. The parameters of the foreground (ω_1 and μ_1) are also kept constant (Figure 4.4). From the *shape* and *spread* of the distribution, we can derive the mean of the background that is found to vary from 1.16 in the first step to 3.19 in the last. Since the mean of

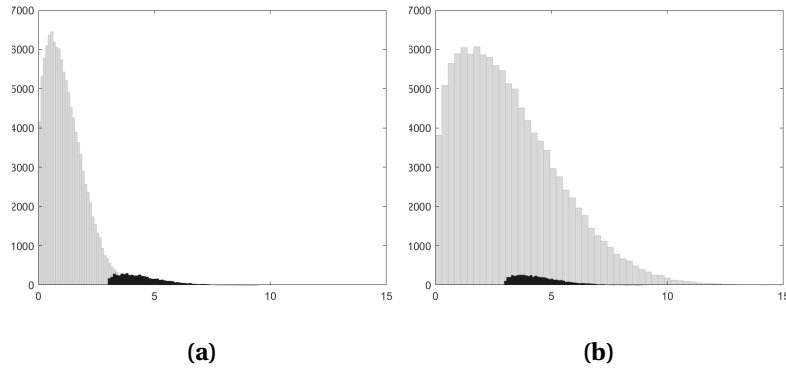


Figure 4.4: Experiment 1: a) histogram of the foreground (grey) and background (black) in the first step and b) the last step.

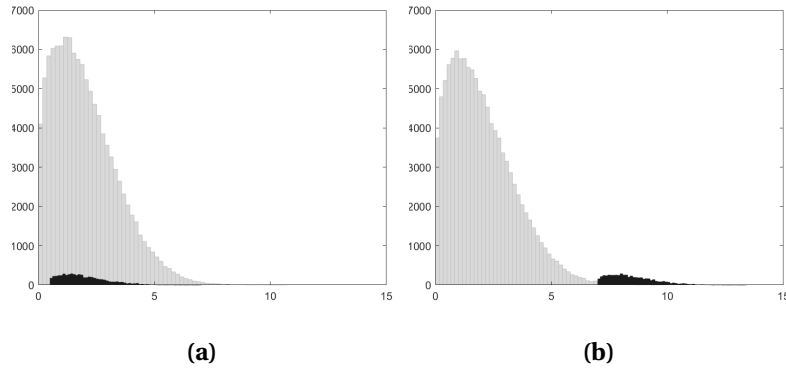


Figure 4.5: Experiment 2: a) histogram of the foreground (black) and background (grey) in the first step and b) the last step.

the foreground is constant at 4.42 we expect the detection to become progressively harder as the two means get closer.

In the second experiment, given constant parameters for both regions, the distribution of the foreground is progressively shifted towards higher values, which results in a gradual increase of the mean of the foreground from near zero to higher values. This allows us to evaluate the performance with respect to the overlapping between background and foreground distributions. Specifically, the background shape μ_0 is set to 0.6 and ω_0 to 6.0. This results in a mean of nearly 2.0 and a standard deviation of nearly 1.4. For the foreground, the mean will progressively increase from 1.92 to 8.42 in steps of 0.5 (Figure 4.5).

In both experiments, for each simulated image corresponding to a gradual change of parameters, we apply our algorithm to detect the clusters. Besides, to get statistically significant results, 10 image realisations are considered for each given set of distribution parameters. From the ground truth binary map, the detection results are evaluated in terms of F-score that is computed from *precision* and *recall* values as explained in Section 2.7: The evaluation results using the F-score are

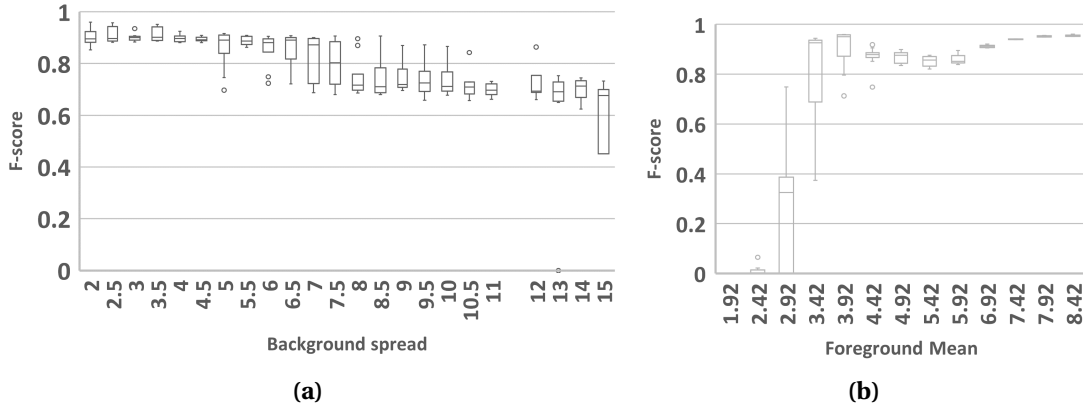


Figure 4.6: The F-score for the results of the algorithm with different (a) spread for the background (Experiment 1) and (b) mean for the foreground (Experiment 2).

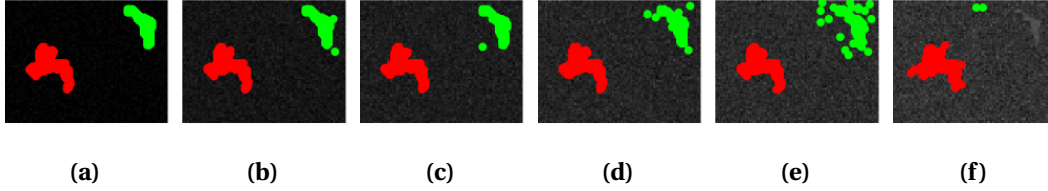


Figure 4.7: Experiment 1: the detected clusters from (a) to (f) in steps 1,8,12,17,21 and 27. (a) shows the perfect segmentation.

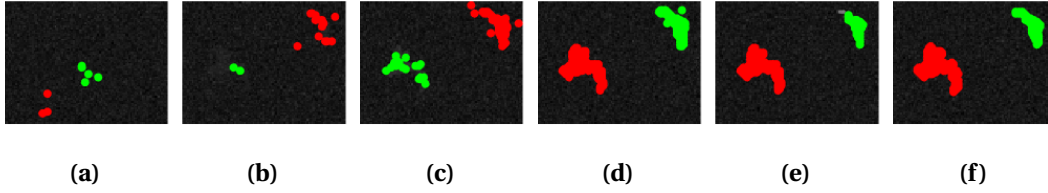


Figure 4.8: Experiment 2: the detected clusters from (a) to (f) in steps 1,2,3,6,9 and 14. (f) shows the perfect segmentation.

shown in Figure 4.6 for each experiment. In each step, the range observed during the repetitions, along with their average and outliers (if any) have been shown. According to these charts, Experiment 1 produces an F-score (on average) higher than 0.8 until step 12 for which the mean of the background is 2.24; and higher than 0.7 until step 20 for which the mean of the background is 2.85. In Experiment 2, the algorithm provides F-scores (in average) higher than 0.8 from step 4 for which the mean of the foreground is 3.42. Considering the fact that the background is chosen to not represent the naive model, thus providing a greater challenge, we find rather satisfying that the F-score plummets only in cases with extreme amount of high value pixels in the background.

Figures 4.7 and 4.8 show examples of the output of the algorithm for each experiment. Ideally, in each frame two separate clusters should be detected, namely one bigger to the left and one smaller. In addition, we expect that the red colour,

which indicates the most significant cluster, highlight the bigger one. In both cases, the algorithm shows resilience to the presence of the noise until the background and foreground become indistinguishable from each other.

These two experiments show the resilience of the algorithm to background noise and how well it can detect minute differences between the background and foreground. It is worth mentioning that these simulations aim at evaluating the proposed approach in a worst case scenario, since we expect actual data be less noisy and/or pre-processed by a noise removal process.

4.3 Performance on actual data

4.3.1 Evaluation regarding 3D clustering

Figure 4.9 shows the result of the proposed NFA-based clustering for four sample frames of WS01. In each frame the top 4 significant clusters have been indicated by their borders. The blue border shows the most meaningful cluster. As we can see, small noises change from frame to frame, big artefacts have a constant size and location, and the wear area grows over time. In all cases, small noises have been ignored and significant high change areas have been identified.

We have compared the clustering output of our algorithm with several other clustering methods, namely:

- *Agglomerative hierarchical clustering with complete linkage* [68];
- *Kmeans++* [5], an extension of classic K-means;
- *Robust spectral clustering* [112], a recent improvement on the classic spectral clustering;
- *Expectation Maximisation for Gaussian Mixture models (EM GM)* [61];
- *Clustering by fast search* [77], a novel approach based on density and distance (cf. 2.3);
- *GBKmeans* [73], a recent improvement on K-means;
- *Clustering by local gravitation* [97], clustering by considering each point as an object with mass and studying local forces among neighbours;
- *HDBSCAN* [14] (Hierarchical Density-Based Spatial Clustering with Application with Noise).

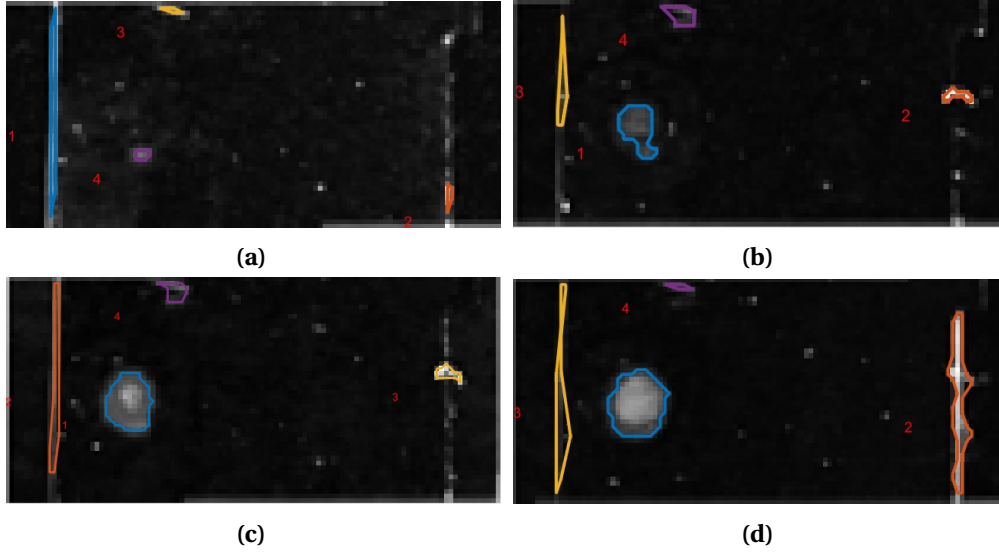


Figure 4.9: Clustering output from frames 3, 9, 15 and 20 of set WS01 using the proposed NFA clustering (Algorithm 2).

Chapter 2 has explained these methods in more details.

As we can see, the chosen algorithms belong to different families of clustering methods and include both established approaches and more recent works. Each of these algorithms has its own strengths and weaknesses such as their robustness to noise and their need to set the number of clusters as input. To make the comparison feasible we have set the number of clusters k equal to 4 whenever needed. Also, we have chosen manually the best cluster in the output to ensure to get an upper bound for performance metrics of the considered algorithm.

Figure 4.11 shows an example of the clustering output of each algorithm using the sequence WS01. The ideal clustering, in this example, would isolate the spherical group of points on the left (the wear) from the random noise and artefacts (the points on the left, right, and upper edge). As we can see, algorithms like *kmeans++* and *agglomerative-complete* fail to separate the wear area from the left border points. *Spectral clustering* is more successful in that regard, but fails to ignore some random noise around the wear region. The overall best result, for this example, comes from *HDBSCAN* and *local gravitation* algorithms, partly thanks to their ability to determine the best number of clusters.

Each sequence and each frame within it presents its own challenges for clustering; therefore, it is important to perform the comparison over every frame containing wear using a impartial quantitative metric. Considering every frame present in the three sequences WS01, WS02, and SV01, the precision, recall and F-score metrics (Table 2.2) have been computed for each clustering result. As an example, Figure 4.10 shows the F-score values for the sequence WS01 and each algorithm including our proposal (last column). The values are colour coded to

| Frame | agglomerative [complete] | Kmeans++ | Robust Spectral | EM GM | Fast Search | GBKmeans | Local gravitation | HDBSCAN | NFA |
|-------|-----------------------------|----------|--------------------|--------|----------------|----------|----------------------|---------|--------|
| 8 | 0.4813 | 0.2443 | 0.4583 | 0.3535 | 0.2436 | 0.2340 | 0.7559 | 0.4045 | 0.8615 |
| 9 | 0.3468 | 0.2717 | 0.3924 | 0.3525 | 0.2361 | 0.2083 | 0.5753 | 0.4802 | 0.8914 |
| 10 | 0.4249 | 0.4352 | 0.5182 | 0.4348 | 0.3101 | 0.3061 | 0.7453 | 0.5316 | 0.6963 |
| 11 | 0.4259 | 0.4858 | 0.7724 | 0.7484 | 0.4250 | 0.3443 | 0.7727 | 0.5737 | 0.8421 |
| 12 | 0.5650 | 0.5650 | 0.7206 | 0.7675 | 0.5423 | 0.4838 | 0.7707 | 0.8550 | 0.8433 |
| 13 | 0.6369 | 0.4990 | 0.6401 | 0.5692 | 0.5040 | 0.3700 | 0.5714 | 0.6705 | 0.8571 |
| 14 | 0.6202 | 0.5715 | 0.8002 | 0.6166 | 0.5189 | 0.4074 | 0.7732 | 0.5443 | 0.8889 |
| 15 | 0.6296 | 0.5661 | 0.8469 | 0.5660 | 0.5525 | 0.3914 | 0.6672 | 0.7323 | 0.8753 |
| 16 | 0.6192 | 0.5448 | 0.7970 | 0.5363 | 0.5544 | 0.4250 | 0.7596 | 0.5610 | 0.8791 |
| 17 | 0.5780 | 0.5590 | 0.8229 | 0.4350 | 0.3180 | 0.5003 | 0.5165 | 0.7952 | 0.8388 |
| 18 | 0.4953 | 0.4903 | 0.7630 | 0.6765 | 0.4016 | 0.3386 | 0.7159 | 0.5889 | 0.8444 |
| 19 | 0.4979 | 0.5569 | 0.7860 | 0.7479 | 0.4087 | 0.4101 | 0.6407 | 0.7658 | 0.8605 |
| 20 | 0.5929 | 0.5981 | 0.6727 | 0.5638 | 0.4794 | 0.4108 | 0.7566 | 0.6505 | 0.8616 |
| Avg | 0.5318 | 0.4914 | 0.6916 | 0.5668 | 0.4227 | 0.3715 | 0.6939 | 0.6272 | 0.8493 |
| Std | 0.0935 | 0.1129 | 0.1479 | 0.1437 | 0.1150 | 0.0858 | 0.0902 | 0.1319 | 0.0492 |

Figure 4.10: F-score values generated for each frame of the sequence WS01 using different algorithms.

show the best results in green and the worst in red. As we can see, *Kmeans++*, *GBKmeans*, *Fast search*, and *agglomerative* clustering consistently fail to produce acceptable clustering results. *Spectral clustering*, *EM GM*, *HDBSCAN* and *Local gravitation* produce good results for some frames but fail on others. Only *NFA* clustering produces acceptable and consistent results across all frames. This can also be seen by the high average F-score and low standard deviation.

To compare F-score values between sequences, Table 4.1 summarises the obtained results in terms of the average and standard deviation of F-score values. The *NFA* clustering performs well for all three sequences (high average F-score); and maintains that performance for each frame in the sequence (very low standard deviation).

Furthermore, it appears that the performance of each clustering method (except the proposed *NFA*-based one) varies from one sequence dataset to another and the best alternative to our algorithm is different for each sequence. This is due to the volatile nature of the noise and artefacts present in our data. Therefore, it is certainly difficult to choose, among previous works, a clustering method which consistently manages different types of noise.

4.3.2 Comparison with 2D segmentation

An alternative to clustering is ΔI image segmentation (to detect the altered areas) followed by labelling of cluster components. Therefore, we also evaluate our

Table 4.1: Average and standard deviation of F-score values for different clustering algorithms on Seq. WS01, WS02 and SV01. Best results are in bold, second best results are underlined.

| Algorithm | WS01 | | WS02 | | SV01 | |
|--|---------------|--------|---------------|--------|---------------|--------|
| | Avg | Std | Avg | Std | Avg | Std |
| Agglomerative [complete] [68] | 0.5318 | 0.0935 | 0.6160 | 0.0848 | 0.6065 | 0.1023 |
| Kmeans++ [5] | 0.4914 | 0.1129 | <u>0.6769</u> | 0.0665 | 0.5930 | 0.0912 |
| Robust Spectral [112] | 0.6916 | 0.1479 | 0.6475 | 0.0665 | 0.6753 | 0.1177 |
| EM GM [61] | 0.5668 | 0.1437 | 0.6379 | 0.0798 | <u>0.7503</u> | 0.1157 |
| Fast Search [77] | 0.4227 | 0.1150 | 0.5273 | 0.0620 | 0.5828 | 0.0799 |
| GBKmeans [73] | 0.3715 | 0.0858 | 0.4870 | 0.0559 | 0.5760 | 0.1051 |
| Local Gravitation [97] | <u>0.6939</u> | 0.0902 | 0.6569 | 0.1213 | 0.6139 | 0.0936 |
| HDBSCAN [14] | 0.6272 | 0.1319 | 0.5968 | 0.0477 | 0.6418 | 0.0866 |
| NFA(Alg. 2) | 0.8493 | 0.0492 | 0.7634 | 0.0382 | 0.8018 | 0.0530 |













proposal against a two-step method: a binary segmentation, namely FRFCM [52], followed by a 2D clustering using HDBSCAN [14] (cf. Chapter 2).

Applied to our data, FRFCM provides rather good separation between background and foreground. Then, to spatially cluster the points produced from FRFCM we use HDBSCAN. This means we can evaluate our automatic process (regarding the number of clusters) with a direct comparison of resulting clusters from both methods. In addition, for each frame, the closest cluster to the ground truth is selected manually for the calculation of the performance metrics.

Tables 4.2, 4.3 and 4.4 show three sample frames from each dataset: their ground truth and output of each method. Also, for comparison with earlier attempts at wear detection on the same dataset, we included the results obtained from the process proposed in [27]. This solution was based on histogram quantisation and genetic algorithm, and was designed to minimise the false positive detection and to quickly give a rough estimation of the likely position of the altered region(s).

A qualitative analysis of the results from all three methods indicates that we have successfully dealt with background noise and artefacts from UV reflections in the majority of cases. For example, in the sequence SV01, reflections on the border of the violin are very close to the actual wear region. The NFA clustering has managed to avoid them (almost) completely, while FRFCM+HDBSCAN have grouped them together with the wear in a few cases. We have also improved the results with respect to [27], that, even if generally less prone to false detection

Table 4.2: Comparison between the proposed NFA clustering, Dondi et al. [27], FRFCM+HDBSCAN clustering and the ground truth for some sample frames from set WS01.

| No. | Ground truth | NFA clustering (Ours) | Dondi et al. [27] | FRFCM+HDBSCAN |
|--------------------|---|---|--|---|
| S ₁ :9 |  |  |  |  |
| S ₁ :15 |  |  |  |  |
| S ₁ :20 |  |  |  |  |

than FRFCM+HDBSCAN, is also less effective than NFA in properly identifying the boundaries of the altered regions. Inherently the NFA clustering allows for controlling the number of false alarms. This results in globally better wear detection (lower number of false positives).

To summarise both experiments, we compare the performance of two of the 3D clustering methods mentioned above (Local Gravitation [97] and robust spectral clustering [112]); FRFCM+HDBSCAN [14, 52] and our proposal (Algorithm 2). Figure 4.12 shows the precision/recall charts for sequences WS01, WS02 and SV01. In all three sets, the proposed NFA clustering has better precision while maintaining an acceptable recall in most cases. As we can see, the FRFCM+HDBSCAN method tends to have good recall values but with poor precision i.e high false positives. This is due to the fact that the binary segmentation step only filters out the low value noise present in the image. Therefore, in the clustering step, the high value artefacts are hard to separate from the actual wear. As a result, it is vital to consider the grey-level values at the same time as the spatial information.

Table 4.3: Comparison between the proposed NFA clustering, Dondi et al. [27], FRFCM+HDBSCAN clustering and the ground truth for some sample frames from set WS02.

























| No. | Ground truth | NFA clustering (Ours) | Dondi et al. [27] | FRFCM+HDBSCAN |
|-----------|---|---|--|---|
| $S_3 : 4$ |  |  |  |  |
| $S_3 : 7$ |  |  |  |  |
| $S_3 : 9$ |  |  |  |  |

Table 4.4: Comparison between the proposed NFA clustering, Dondi et al. [27], FRFCM+HDBSCAN clustering and the ground truth for some sample frames from set SV01.

| No. | Ground truth | NFA clustering (Ours) | Dondi et al. [27] | FRFCM+HDBSCAN |
|------------|---|---|--|---|
| $S_2 : 9$ |  |  |  |  |
| $S_2 : 15$ |  |  |  |  |
| $S_2 : 20$ |  |  |  |  |

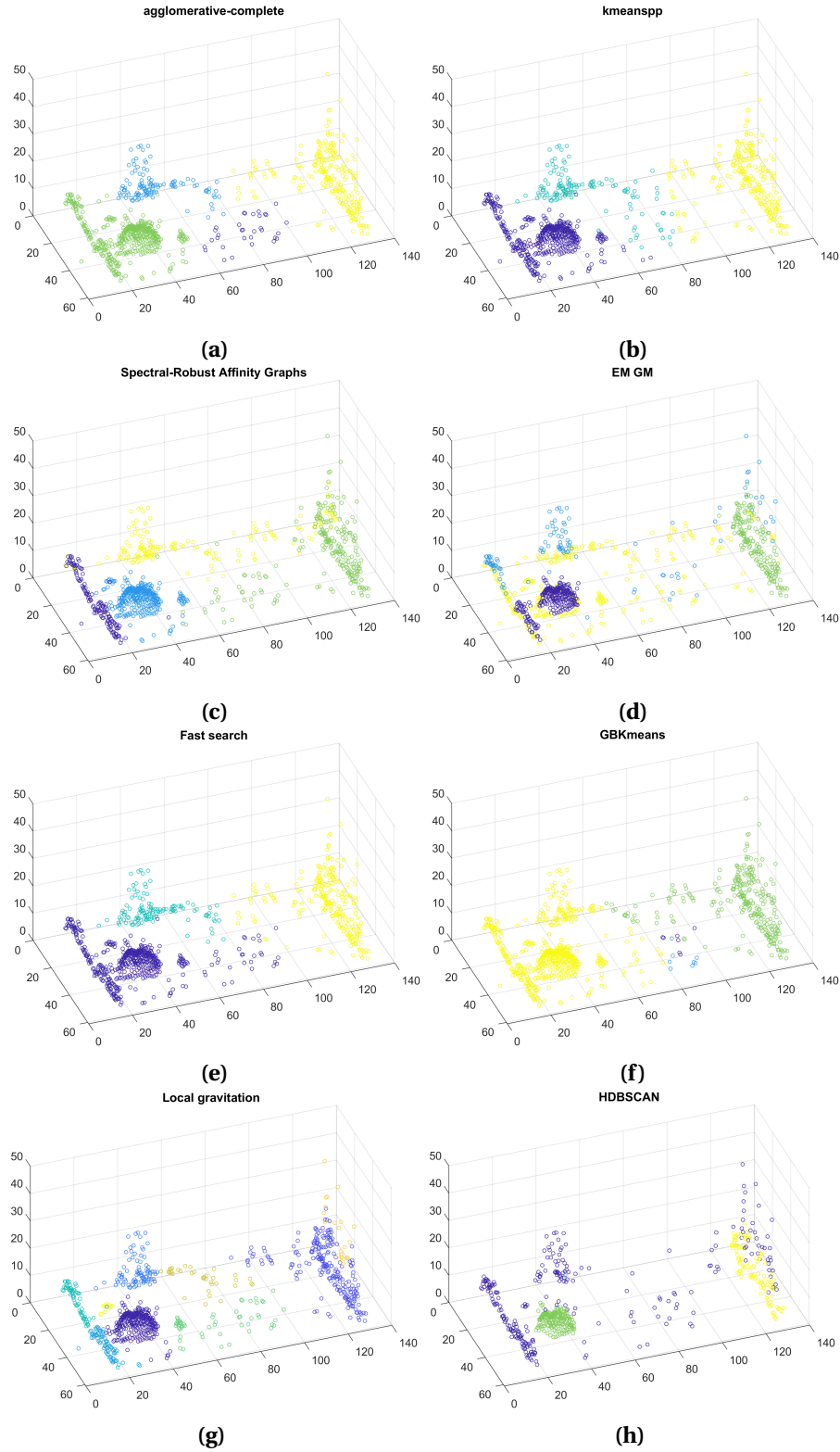
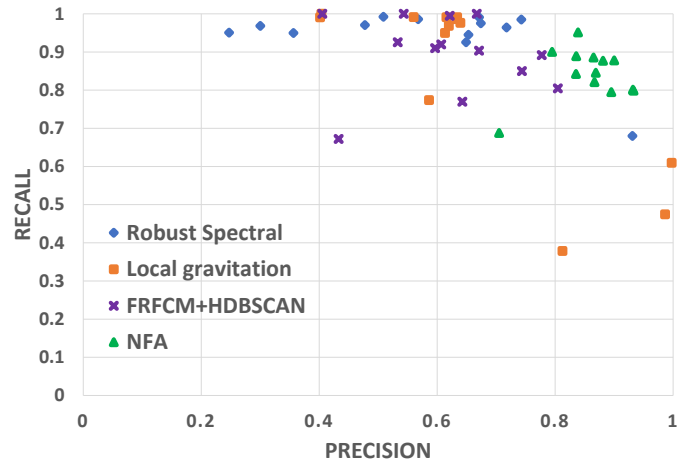
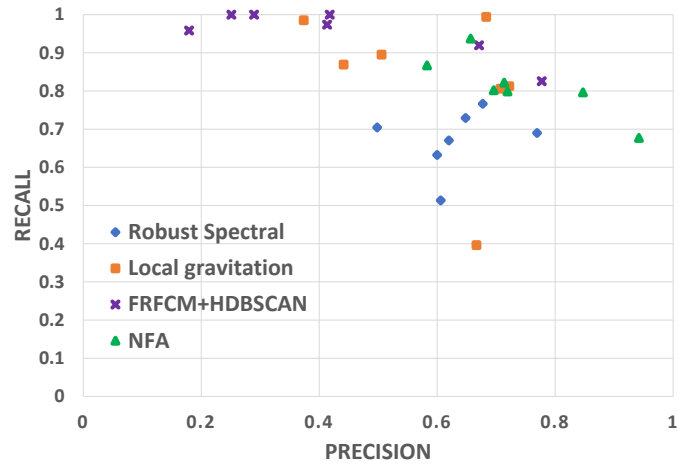


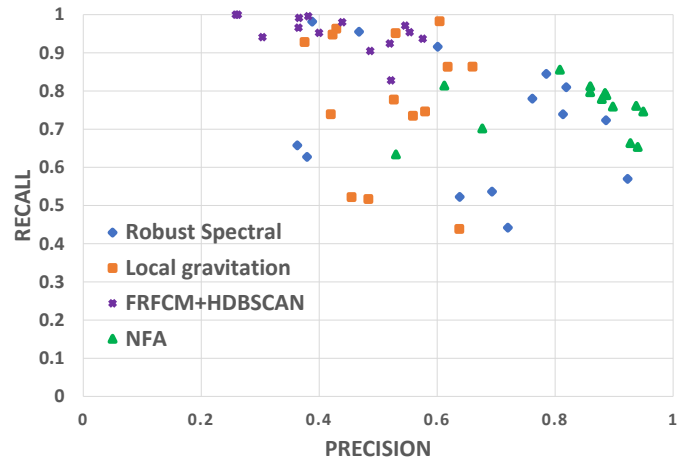
Figure 4.11: Clustering result of several chosen algorithms performed on the frame 12 of sequence WS01.



(a)



(b)



(c)

Figure 4.12: Precision-Recall plot for WS01 (a), WS02 (b) and SV01 (c). For a given algorithm (indicated by the colour), each point highlights the performance at a specific time-step of the sequence.

4.4 Conclusion

In this chapter, we proposed an algorithm for change detection between two images by clustering their grey-level difference map. The algorithm is based on the a-contrario framework and works with a single naive model which takes into account both spatial and spectral dimensions.

Two experiments were described using simulated difference maps in order to test the robustness of the method to different levels of noise. In addition, the algorithm was applied on several sequences of UVIFL images. The results of the clustering for each sequence were compared to several existing data clustering methods. Improvement was shown for both precision and recall.

The next chapter tackles the problem of differentiating between static artefacts and wear regions by incorporating the time dimension and using more than one pair of images from the sequence.

Chapter 5

Analysing a multi-temporal image sequence

Contents

| | |
|--|--------------------|
| 5.1 From input data to 3D point cloud | 77 |
| 5.2 Clustering the 3D point cloud | 79 |
| 5.3 Changed area ranking | 81 |
| 5.4 Experiments and the benefits of the multitemporal aspect | 82 |
| 5.5 Comparative performance evaluation | 87 |
| 5.6 Conclusion | 90 |

Recalling the problem definition from Section 1.4, we divided our wear detection problem into two sub-problems: firstly, to detect changed areas between two image frames (whether or not those changed areas are actually wear or not); and secondly, to differentiate between growing wear regions and static artefacts. Until now, we have proposed a solution to the first sub-problem by detecting the meaningful clusters of points in the difference map of two image frames. The best way to tackle the second sub-problem would be to introduce the time dimension to our clustering algorithm. As mentioned in Section 1.4, the wear regions are assumed to persist once they appear. They also do not shrink in size and usually get bigger through time. This chapter proposes a clustering algorithm in three dimensions; i.e. image spatial dimensions + time.

5.1 From input data to 3D point cloud

The raw data are a series of K multi-temporal RGB images of varnished wooden samples and violins (see Section 1.3 for the dataset specifications); I_0 being

the reference image. After performing the necessary pre-processing steps (cf. Section 1.5), we will have a time series of K colour images (including the reference image) of spatial size $w \times h$ pixels; i.e. the input volume would be $w \times h \times K \times 3$.

Now, the optical monitoring of a varnished wooden surface is an ongoing process meaning that the number of available frames and the accuracy of the detection increases as the time goes on. At some point in time, the number of available frames becomes too large. To keep only the most relevant information, we maintain a sliding window of the last n frames with $n \leq K - 1$ (K being the number of frames, the maximal number of frames to compare to the reference one I_0 is $K - 1$).

Then, in order to reduce this data volume without harming the wear detection, we propose an extraction of the points likely to belong to wear, i.e., the points of interest. To extract these points, we could either rely on wear colour features (if learned for instance) or on change detection with respect to reference image I_0 . Due to the lack of sufficient labelled data and in order to be robust to the variability of the material, varnish and pigments present on the instruments, we focus on change detection approaches. That means we are only interested in new wear regions or the increase in area of the existing ones with respect to I_0 . Then, considering each pair of frames (I_t, I_0) , $t > 0$, we derive a binary image of the *points*, i.e. the pixels likely to belong to a wear region. This step can be done using the proposed approach outlined in Chapter 4. However, alternatives such as a simple fixed thresholding, clustering or binary segmentation can be considered depending on the contrast between the wear and normal areas. In any case, the result is a set of binary images denoted $\{B_t, t \in \llbracket 1, n \rrbracket\}$.

Considering the whole series $\{B_t, t \in \llbracket 1, n \rrbracket\}$, we create the 3D point cloud $\mathcal{P} \subset \mathbb{R}^3$ with two spatial dimensions defined by the image domain along with time as the third dimension:

$$\mathcal{P} = \left\{ (x, y, z), x \in \llbracket 1, w \rrbracket, y \in \llbracket 1, h \rrbracket, \frac{z}{c_t} \in \llbracket 1, n \rrbracket \right\}, \quad (5.1)$$

where $c_t \in \mathbb{R}_{>0}$ is the time coefficient that controls the importance of a distance in time compared to a distance in the image space. If $c_t = 1$, then we give the same importance to both. Then, the 3D volume of the cuboid from which \mathcal{P} was extracted is $V_{\mathcal{P}}$:

$$V_{\mathcal{P}} = w \times h \times (nc_t). \quad (5.2)$$

The derived 3D point cloud \mathcal{P} along with $V_{\mathcal{P}}$ are the input of the proposed clustering algorithm presented in the next section.

5.2 Clustering the 3D point cloud

Algorithm 3 details the clustering method that we propose to detect and rank the clusters of points within the point cloud. The ranking is based on the significance of each cluster both in space and time. This algorithm represents a methodological contribution, here applied to wear detection, but which could be adapted to the detection of any other “objects” of interest characterised by spatio-temporal consistency.

Algorithm 3 Detecting and ranking clusters; inputs: 3D point cloud \mathcal{P} and original 3D volume $V_{\mathcal{P}}$; output: list of detected clusters \mathbf{C}

```

1: for each pair of points  $j$  and  $j'$  in  $\mathcal{P}$  do
2:   Compute 3D distance  $D(j, j')$ 
3: end for
4: Compute the minimum spanning tree for the points in  $\mathcal{P}$  based on  $D(j, j')$  so
   that each subtree in hierarchical representation stores in its root the distance  $\delta$ 
   of the associated cluster
5:  $i \leftarrow 1$ 
6: for each subtree  $T$  do
7:    $\mathcal{C}_i \leftarrow$  cluster of points in  $T$ 
8:    $\delta \leftarrow$  value stored in the root of  $T$ 
9:    $\delta' \leftarrow$  value in the parent node of  $T$ 
10:   $\mathcal{V}(\mathcal{C}_i) \leftarrow$  dilation of  $\mathcal{C}_i$  with radius  $\delta/2$ 
11:   $\rho \leftarrow \delta' - \delta$ 
12:  Compute volumes  $V_{\mathcal{C}_i}$  and  $V_{\overline{\mathcal{C}_i}}$  according to Eq. (5.5)
13:   $S_i \leftarrow S(\mathcal{C}_i)$  computed using Eq. (5.6) and logarithm function
14:   $i \leftarrow i + 1$ 
15: end for
16:  $\mathcal{J} \leftarrow$  list of indices of the clusters sorted according to decreasing values of  $S_i$ 
17:  $\mathbf{C} \leftarrow \emptyset$ 
18: for each index  $j$  in  $\mathcal{J}$  do
19:    $\mathcal{C}_j \leftarrow j^{th}$  cluster according to  $\mathcal{J}$ 
20:   if  $\forall \mathcal{C}_l \in \mathbf{C}, \mathcal{C}_j \cap \mathcal{C}_l = \emptyset$  then
21:      $\mathbf{C} \leftarrow \mathbf{C} \cup \{\mathcal{C}_j\}$ 
22:   end if
23: end for
24: return  $\mathbf{C}$ 

```

The proposed approach relies on the a-contrario detection which permits evaluating the significance of any clusters based on distances between each pair of points [23] (cf. chapter 3.5.2). In this chapter, since Equation (5.1) has already integrated the desired balance between spatial and temporal aspect, we can simply consider the 3D Euclidean distance between the points of \mathcal{P} : For any pair of 3D points,

$$\forall ((x_a, y_a, z_a), (x_b, y_b, z_b)) \in \mathcal{P}^2, D(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}.$$

In Algorithm 3, the distances are computed in the first **for** loop.

Then, the clusters are defined based on the distance as follows: for a given distance δ , a cluster \mathcal{C} is a subset of \mathcal{P} such that for any point in \mathcal{C} there is another point in \mathcal{C} at a distance lower than δ and there is no other point in $\mathcal{P} \setminus \mathcal{C}$ at a distance lower than δ :

$$\left\{ \begin{array}{l} \forall p \in \mathcal{C}, \quad \exists p' \in \mathcal{C} \text{ such that } D(p, p') \leq \delta, \\ \forall p'' \in \mathcal{P} \setminus \mathcal{C}, \quad \nexists p' \in \mathcal{C} \text{ such that } D(p'', p') \leq \delta. \end{array} \right. \quad (5.3)$$

Note that with such a definition, not all subsets of \mathcal{P} are clusters. From now on, a “well-defined” cluster \mathcal{C} follows the previous definition, and among the range of values δ consistent with \mathcal{C} , we chose the minimum:

$$\delta = \max_{p \in \mathcal{C}} \min_{p' \in \mathcal{C} \setminus \{p\}} D(p, p'). \quad (5.4)$$

The measure of significance of such a well-defined cluster involves the computation of two normalised volumes (defined from 3D closed shapes in \mathbb{R}^3). The first one represents the volume associated to cluster \mathcal{C} . Since the points in \mathcal{C} are connected up to distance δ (defined in Equation 5.4), the dilation of every \mathcal{C} point with a radius $\delta/2$ provides a 3D connected component in \mathbb{R}^3 , called $\mathcal{V}(\mathcal{C})$. To ensure that $\mathcal{V}(\mathcal{C})$ is a closed area, we finally perform a morphological closing of this latter. The closing radius ρ is chosen as the maximum value which will not add any point to $\mathcal{V}(\mathcal{C})$, i.e. the difference between δ and the distance δ' to the closest point not in \mathcal{C} :

$$\begin{aligned} \delta' &= \min_{(p, p') \in \mathcal{C} \times (\mathcal{P} \setminus \mathcal{C})} D(p, p'), \\ \rho &= \delta' - \delta. \end{aligned}$$

Then, \mathcal{C} volume is approximated by the 3D volume of $\mathcal{V}(\mathcal{C})$ after closing.

The second volume of interest is the one of the points not in \mathcal{C} , i.e. the points in $\mathcal{P} \setminus \mathcal{C}$. It is evaluated as the complementary with respect to the initial 3D cuboid, of the dilation of $\mathcal{V}(\mathcal{C})$ with radius ρ , i.e. up to its closest point in $\mathcal{P} \setminus \mathcal{C}$, so that the empty space around \mathcal{C} does not belong to the volume of the points in $\mathcal{P} \setminus \mathcal{C}$. Then, the two volumes of interest are computed according to:

$$\left\{ \begin{array}{l} V_{\mathcal{C}} = \frac{1}{V_{\mathcal{P}}} \times [\text{volume of } \mathcal{V}(\mathcal{C}) \text{ after closing with radius } \rho], \\ V_{\mathcal{P} \setminus \mathcal{C}} = \frac{1}{V_{\mathcal{P}}} \times [\text{volume of } \mathcal{V}(\mathcal{C}) \text{ after dilation with radius } \rho]. \end{array} \right. \quad (5.5)$$

Then, for any well-defined cluster $\mathcal{C} \subseteq \mathcal{P}$ having $|\mathcal{C}|$ points among the total number of points $|\mathcal{P}|$, we are able to compute the Number of False Alarms (NFA) criterion as:

$$\text{NFA}(\mathcal{C}) \propto \sum_{j=|\mathcal{C}|}^{|\mathcal{P}|} \binom{|\mathcal{P}|}{j} V_{\mathcal{C}}^j (1 - V_{\mathcal{C}})^{|\mathcal{P}|-j}, \quad (5.6)$$

where $\binom{k}{j}$ denotes the binomial coefficient equal to $\frac{k!}{j!(k-j)!}$ with $n! = n \times (n-1) \times \dots \times 2 \times 1$. This formula comes from the assumption that in absence of any structure points are uniformly distributed [22] within a space that, in our case, is the 3D cuboid $w \times h \times (nc_t)$. Then the significance of a cluster \mathcal{C} is evaluated as $S(\mathcal{C}) = -\log(\text{NFA}(\mathcal{C}))$, i.e., the higher this significance value, the more likely the cluster is not a false alarm, but a real wear area characterised by points both spatially close and time-consistent. In Algorithm 3, the significance values of all well-defined clusters are computed in the second **for** loop.

The last part of Algorithm 3 (including the last **for** loop) aims at only keeping the most significant clusters that are not redundant, i.e., that correspond to disjoint subsets of points. Indeed, our objective is to get a partition of \mathcal{P} . Therefore, from all the derived clusters \mathcal{C}_i , only the ones not intersecting another well-defined cluster having a higher significance value are kept. This corresponds to keeping only the *maximal* clusters as defined in [21]. Then, the list of these *maximal* clusters sorted by their significance value is the output of the algorithm.

5.3 Changed area ranking

After running Algorithm 3 we get a sorted list of spatio-temporal clusters with the most significant cluster in first rank, the second most significant disjoint cluster in second rank, and so on.

Significance is higher for more compact or dense clusters of points which indicate individually a local change with respect to the reference frame. Based on the assumptions that wear areas are linked to colour change, and that compact or dense clusters emphasise time-persistent changes, we argue that the derived significance values provide a relevant ranking of the likelihood of the wear areas.

In the experiments presented in next section, from the ranked set **C** (output of Algorithm 3), we keep the three most meaningful clusters along with their respective rank that can be used in the evaluation. We also underline that, in contrast to clustering algorithms, providing such a ranking is a strength of the proposed method.

Finally, considering the temporal evolution of the clusters, we can further refine the results and identify more reliably the wear region(s). For example, a small but constantly growing cluster is more likely to be a worn-out area with respect to a large

but stable cluster.

5.4 Experiments and the benefits of the multitemporal aspect

For each frame in the sequences WS01, SV01 and SV02, we generate a binary change image with respect to the reference frame of the respective sequence applying the method described in Chapter 4. By accumulating the change detection results of all frames, we create a 3D point cloud for each sequence (Figure 5.1). The coefficient c_t is set to 2, in order to give slightly more importance to spatial proximity over time proximity, to penalise the clusters which disappear in some frames and to keep the size of the representation domain small enough for a better performance. The time domain is the upward axis in all visualisations. From this figure it clearly appears that there is a considerable amount of noise present in all point clouds. In addition, there are artefacts present in the left and right sides of all sequences, which are related to the UV reflections from the the wooden sample.

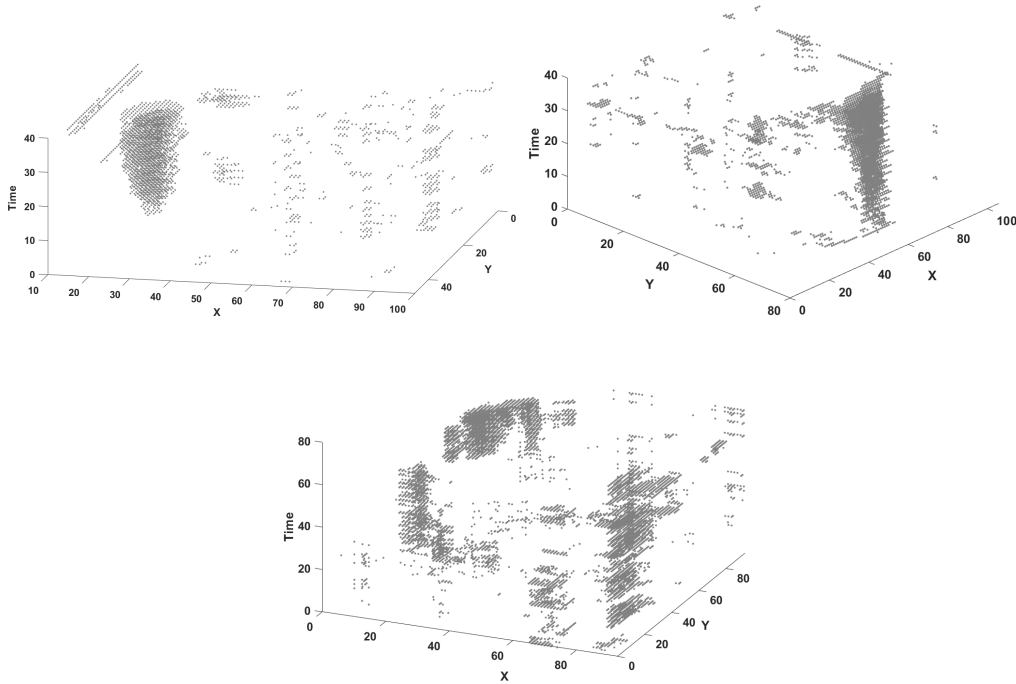


Figure 5.1: Point clouds \mathcal{P} derived from the sequences WS01 (left), SV01 (middle) and SV02 (right). X and Y axes are in pixels, while the time dimension (Z axis) depends on the factor c_t (in these experiments $c_t = 2$).

To simulate a practical optical monitoring process and to analyse different detection results in any given point in time, we start the experiment with the first

two frames for each sequence; then, we run the algorithm repeatedly adding one more frame each time until we reach n frames. From then on, we keep only the last n frames; basically, adding a new frame and removing the oldest one in each step.

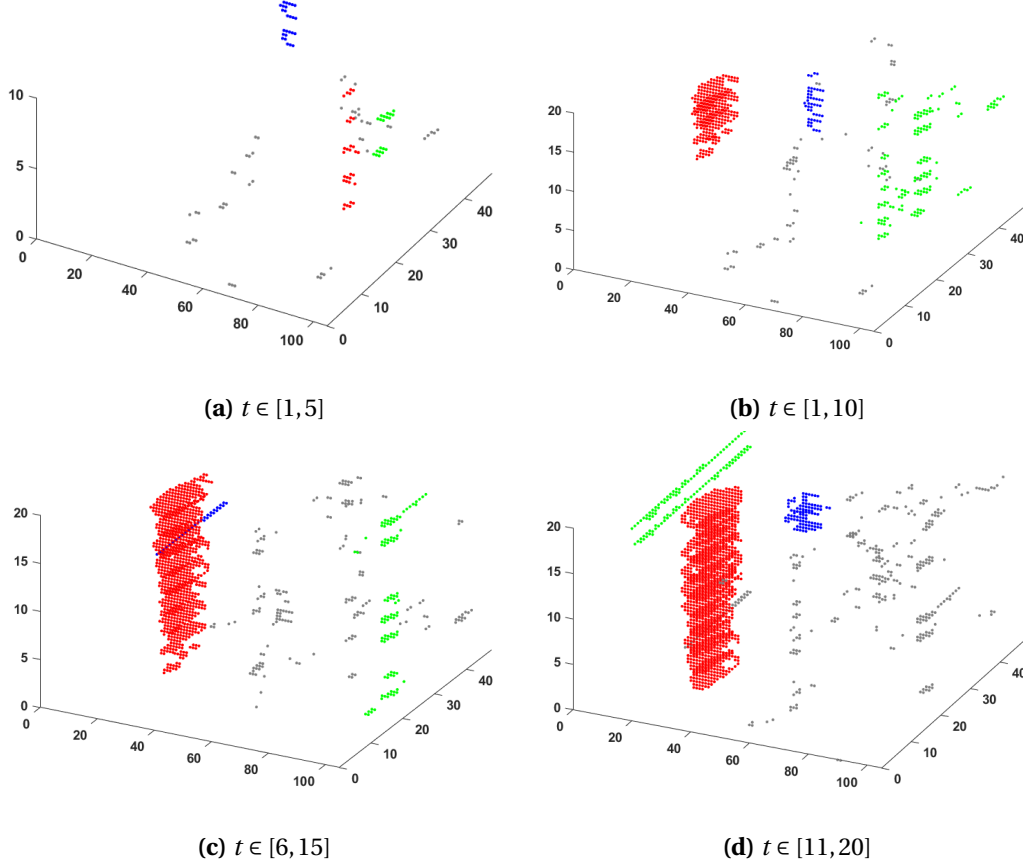


Figure 5.2: Evolution of the detection by using the later time frames t in the sequence WS01. Colour code gives the rank according to significance: red first, green second, blue third. The time domain is the upward axis.

Each run outputs a list of clusters sorted according to their meaningfulness value. Since each run uses the frames situated later in time, the rank of each cluster within the whole set of clusters and its associated meaningfulness value evolve depending on the nature of the cluster. In general, assuming that a wear region expands over time, our expectation is that a wear cluster starts in lower ranks and steadily rises to the first rank. Similarly, we expect that the significance of a wear region increases over time. Conversely, the meaningfulness value for the noise and artefact clusters should remain nearly constant or change randomly. Another difference between the wear region and the rest is that it should be present in every frame after its first appearance. For other non-wear areas, it is possible that they divide into two or more clusters along the time axis. In that case, for time evolution analysis, we only take into account the one with the highest meaningfulness.

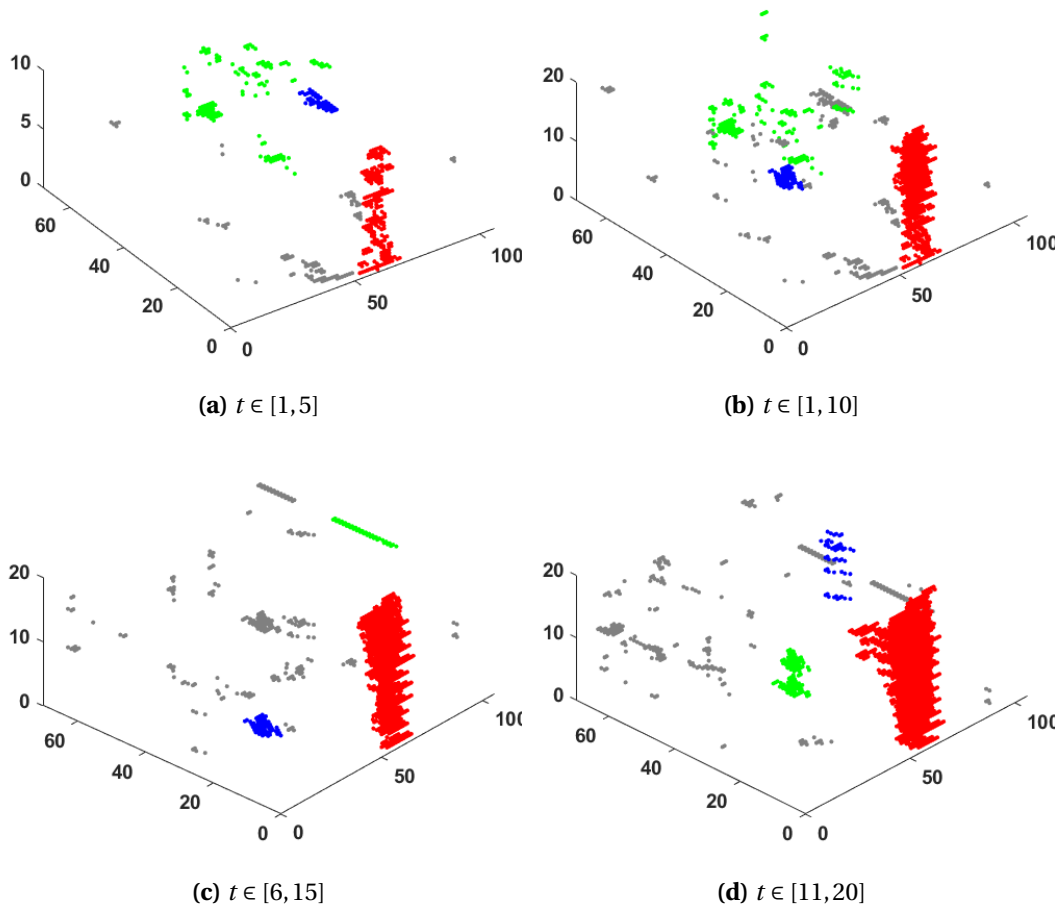


Figure 5.3: Evolution of the detection by using the later time frames t in the sequence SV01. Same conventions as Figure 5.2.

Figures 5.2, 5.3 and 5.4 illustrate the detection results for each sequence through time, i.e., based on the current last frame used as input. In each case, the top three clusters have been shown. It is noticeable that, firstly, only the clusters which are consistent have been detected. Minor artefacts and small noises have been ignored, a fact which is a very desirable behaviour when multi-temporal information is available. Secondly, the region of interest which is the wear area has been chosen by the end of the sequence as the most significant cluster.

For a more in-depth analysis, we can follow the meaningfulness evolution in time of both the wear and the artefacts for each sequence. Observing Figure 5.5a, we can infer that in sequence WS01 the meaningfulness of the most significant noise cluster (the orange line) remains very low and does not change throughout the experiment. This is expected because the noise regions are fairly small and do not grow over time. The cluster indicating the wear region (the blue line) starts from frame seven when the wear appears and consistently increases over time. This is

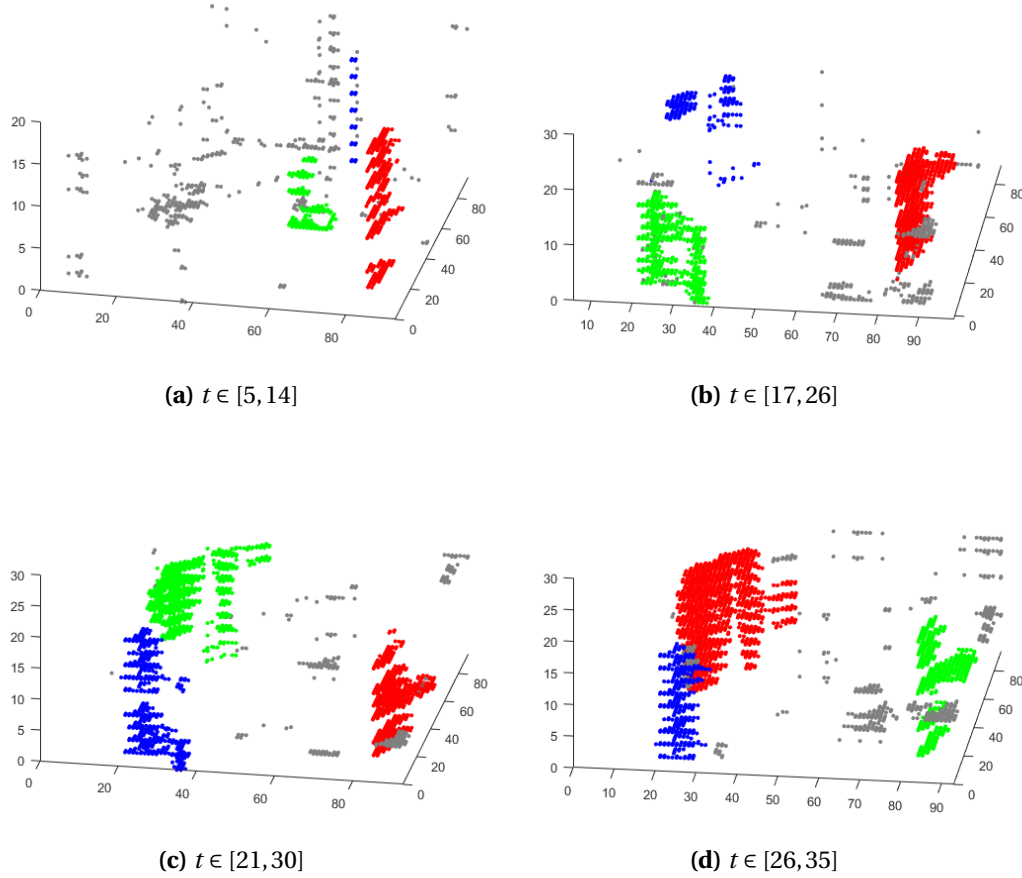


Figure 5.4: Evolution of the detection by using the later time frames t in the sequence SV02. Same conventions as Figure 5.2.

in line with our expectation about an “ideal” wear that keeps growing. In practice, the wear may not expand over multiple acquisitions which will appear as a plateau. However, it will never shrink or disappear so the values should remain increasing overall.

In the same manner, we can interpret Figure 5.5b for sequence SV01. The wear cluster (the blue line) appears from the beginning and always has the highest meaningfulness. This latter increases in each step as the wear area grows through time. This fast growing is coherent with the characteristics of the sequence, that shows the worsening of an already present worn-out region. The two distinct artefact clusters (the grey and orange lines) present in the sequence have meaningfulness values which change randomly from one step to another and sometimes disappear altogether. This is again in line with our expectation of a typical artefact region.

Finally, Figure 5.5c shows the results for sequence SV02, the most complex one. In the first 10 frames we have two artefact clusters which appear to have increasing

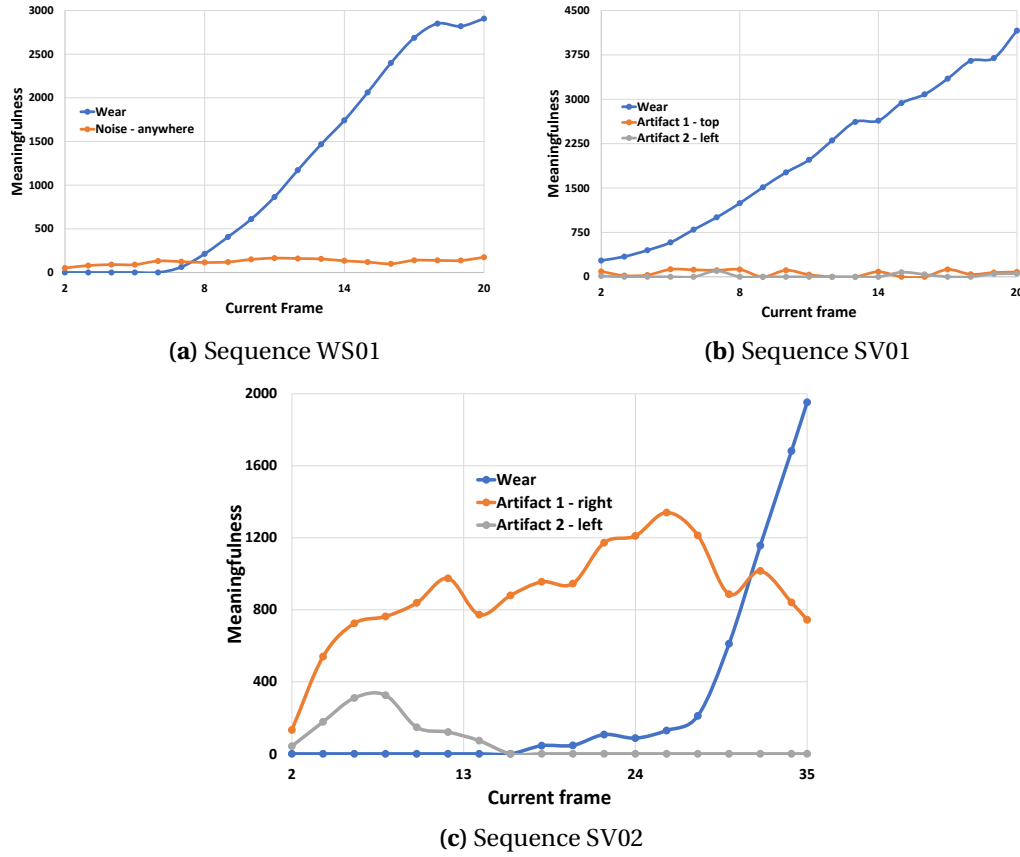


Figure 5.5: The evolution of the meaningfulness value of sample clusters in different runs of the algorithm for sequence WS01 (a), SV01 (b) and SV02 (c). In each run, at most the last 10 frames have been used.

meaningfulness. This can lead to a false positive detection until we have enough frames to distinguish their actual behaviour. Artefact 2 (the grey line) is proven to be an artefact after frame 6. The same applies for artefact 1 (the orange line), after frame 12. On the other hand, the wear cluster (the blue line) has a meaningfulness value which continues to increase from its appearance around frame 20 until the end. More importantly, by the time the wear appears, we have already dismissed the other two clusters as artefacts; therefore, automatically, it becomes the sole candidate wear region.

Overall, the study of evolution of meaningfulness values helps us to quickly identify the wear region as the correct area of interest even if it is smaller than the artefacts present on the surface, in full agreement with the preventive conservation principles. Comparing to the method presented in Chapter 4 which makes use of single change maps, the current analysis based on the temporal dynamics of the changes is able to pinpoint more reliably the emergence of a wear, also reducing the false positive detection. Figure 5.6 illustrates the improvements made by the multi-temporal analysis with respect to the single change map detection.

Comparing the results achievable by using only two frames at a time (second row), with the outcomes of this chapter's proposal (third row), we can notice how the multi-temporal analysis clearly improves the change detection by removing all the artefacts which do not grow over time and/or are not consistently present in every frame.

5.5 Comparative performance evaluation

In this section, we have made a quantitative comparison between our proposed multi-temporal 3D clustering algorithm and several other clustering methods, namely: Agglomerative hierarchical clustering with complete linkage [68], Kmeans++ [5], robust spectral clustering [112], Expectation Maximisation for Gaussian Mixture models (EM GM) [61], GBKmeans [73] and clustering by local gravitation [97]. These methods have been described in depth in Chapter 2.

All methods have been applied on the point cloud \mathcal{P} (Equation 5.1) generated from the input binary images. If the algorithm requires the total number of clusters as input, we set it to 4. Each method would cluster the point cloud \mathcal{P} into several clusters. Since none of the considered clustering approaches have a built-in feature for ranking the output clusters, and to perform a fair comparison among the algorithms, we evaluated every cluster they produce with our metric and selected the most performing one. As metric for the comparison, we use the F-score (F_1 indicator) described in Section 2.7. To compute the precision and recall, the obtained results of each algorithm have been compared with the ground truth of each sequence (Figure 5.6).

For each sequence, after computing the F-score for every frame containing any wear, we calculated the average and the standard deviation. The results are summarised in Table 5.1. As we can see, our proposal performs better than all the other state-of-the-art solutions in all three sequences, by having the highest average F-score while maintaining a low standard deviation. It should also be noticed that the second-best algorithm is different for each sequence, meaning that there is no alternative method with consistent performance in all the considered conditions.

Although this comparison is performed at pixel level, at object level the benefit of our approach will be even more visible. Indeed F-score aims to measure the method's ability to detect *all* the wear pixels, but it does not directly reflect the algorithm's capability to early detect worn-out regions. In fact, although it is important to be able to identify as much wear pixels as possible, in a monitoring process we can afford to lose or mislabel some boundary pixels, if we properly spot the correct position of the wear clusters and avoid the noise. Thus, at pixel level we prefer a high Precision (to guarantee noise avoidance), while at cluster level a high

Table 5.1: The average and standard deviation of F-score values for the 3D clustering of each sequence using the proposed algorithm as well as six other clustering methods. First and second-best results are highlighted in green and light green respectively.

| Algorithm | WS01 | | SV01 | | SV02 | |
|------------------------------|-------|-------|-------|-------|-------|-------|
| | avg | std | avg | std | avg | std |
| Agglomerative[complete] [68] | 0.657 | 0.088 | 0.707 | 0.128 | 0.820 | 0.124 |
| Kmeans++ [5] | 0.694 | 0.128 | 0.709 | 0.129 | 0.727 | 0.273 |
| Robust Spectral [112] | 0.699 | 0.108 | 0.716 | 0.076 | 0.564 | 0.270 |
| EM GM [61] | 0.756 | 0.126 | 0.810 | 0.083 | 0.630 | 0.263 |
| GBKmeans [73] | 0.630 | 0.116 | 0.585 | 0.144 | 0.646 | 0.193 |
| Local gravitation [97] | 0.773 | 0.119 | 0.796 | 0.081 | 0.703 | 0.137 |
| NFA (Ours) | 0.784 | 0.119 | 0.812 | 0.113 | 0.858 | 0.130 |

Recall (to not miss any wear cluster). This is highlighted in Figure 5.5 and 5.6, where we can see that in all three sequences the proposed algorithm was able to properly identify the wear position, and to distinguish it from any noise cluster(s), in a few frames from its appearance.

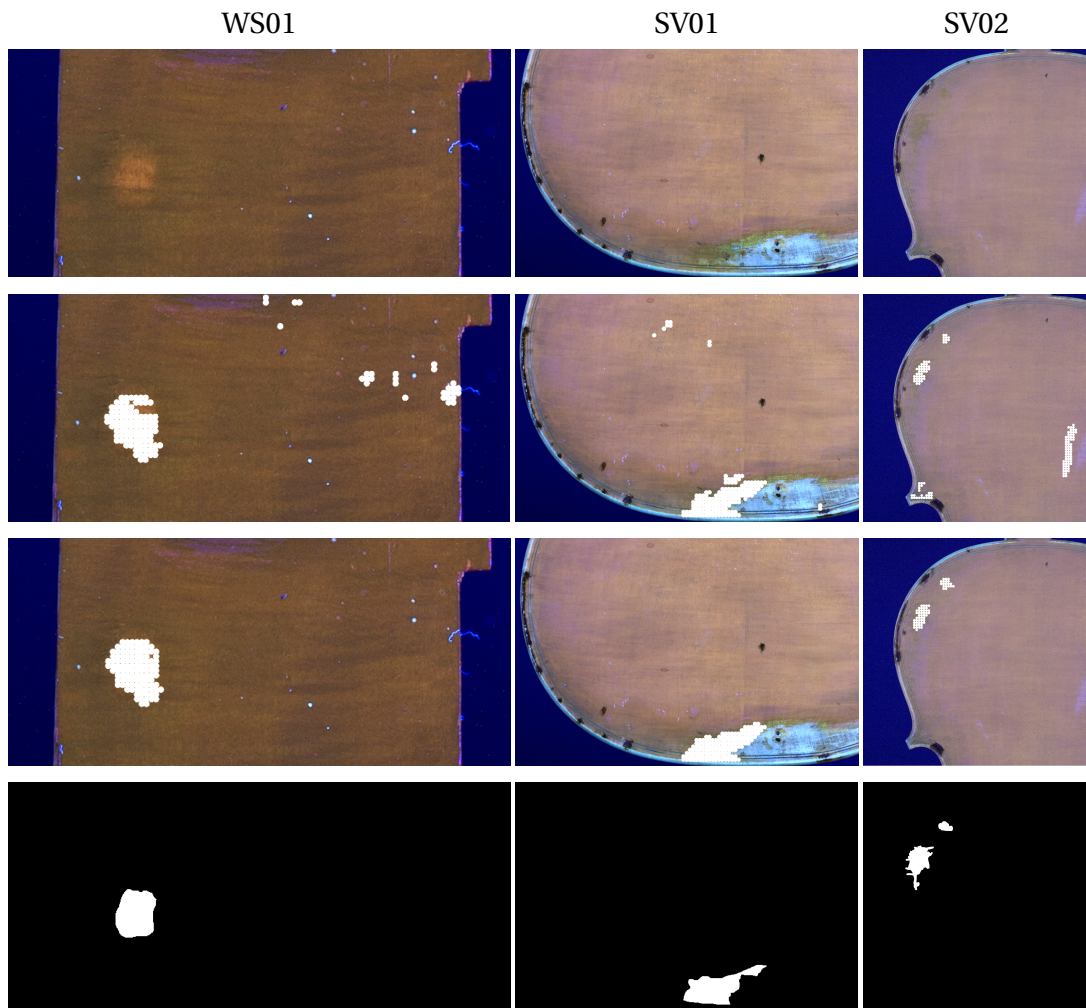


Figure 5.6: Comparison, on sample frames of each sequence, between a wear detection performed considering only two frames at a time and the multi-temporal approach: first row, original UVIFL image; second row, detected clusters as in Chapter 4; third row, detected clusters applying the multi-temporal analysis; fourth row, ground truth showing the actual worn-out regions.

5.6 Conclusion

In this chapter, we presented the final part of our wear detection algorithm. A 3D clustering method was described which works over the spatial and temporal dimension of an image sequence. The benefits of using the temporal information was demonstrated by comparing the results to those of Chapter 4. Once again, quantitative comparisons between our algorithm and several other clustering methods showed improvements for both precision and recall.

Conclusion and future work

Summary

In this study, we tackled the problem of optical monitoring for historic musical instruments, and more specifically violins. These instruments are susceptible to surface varnish wear because of their constant use by musicians. Chemical methods and spectroscopy can be used for wear detection but optical monitoring provides a faster preliminary check. Scarcity of annotated data, prevalent presence of noise and lack of a-priori knowledge about the shape and number of wear regions make this study challenging and at the same time necessary.

Chapter 1 gave a brief introduction to the field of preventive conservation and iterated over our data pre-processing steps. Then, we presented a brief survey on existing clustering methods (and their strengths and weaknesses) in Chapter 2. Chapter 3 contained some theoretical background information on the a-contrario framework and its connection to Gestalt psychology.

In Chapter 4, we proposed a probabilistic algorithm to detect clusters of change between two temporally different images of the same scene. Our proposal is based on an a-contrario framework and performs the clustering process directly on the grey-level difference image, while dealing with the background noise and artefacts. Simulated test cases generated stochastically were used to test extensively the behaviour and limitations of the method, and showed flexibility to background noise and the ability to detect minute differences. Moreover, comparisons with recent clustering methods show meaningful improvements in precision and recall while providing the benefit of an inherent ranking criterion for the resulting clusters.

In Chapter 5, we introduced our a-contrario 3D clustering method for detecting new alterations on varnished surfaces starting from a multi-temporal series of images. This algorithm works based on a single naive model describing both the spatial and temporal information. Once again, tests conducted on UVIFL image sequences showed a good performance comparing to the other state-of-the-art clustering algorithms.

Contributions

Both of our algorithms provide the following advantages:

- We propose a process which is free from parameters characterising the changed regions (shape, number, position, growth pattern, etc.).
- In contrast to other statistical learning-based and deep learning methods, our algorithms, rooted in the Gestalt theory, do not need precise annotations, and can properly work even with few data. These are both important properties in the Cultural Heritage field, where a limited availability of data and a lack of proper annotations are common.
- Our approach can be used in preventive conservation as a fast, preliminary examination of the surface of a violin able to identify the most likely altered areas. Thus, a verification using more precise but slower techniques (like spectroscopic analyses) will be done only on the detected areas, reducing the time needed for completing the monitoring procedures.
- Finally, even though we focused on the case of historical violins, our methods can be adapted to work on other kinds of relics with few modifications, which will be needed mostly in the preliminary processing of the input images.

Future works

For future studies, a vital task is to perform a long-term (more than one or two years) monitoring process considering real historical violins which are played weekly. Beside creating a valuable dataset for the community, this will allow the researchers to validate any wear detection algorithm on real wear patterns. In addition, a long-term image acquisition plan may create more challenging conditions for the wear detection algorithm. For example, different people capturing the photos at different times may result in more human errors or equipment may deteriorate after months of use. This task will be possible as soon as the concert season restarts at the same frequency as during the pre-pandemic period.

Regarding the time complexity of both proposed algorithms, the current version can benefit from parallel implementation and alternative methods for faster 3D dilation operations.

Finally, every computational algorithm needs a user friendly interface to be beneficial to the experts in the problem domain. In our case, the method should be usable by conservation experts who may or may not be familiar with computer vision applications.

Bibliography

- [1] Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459. [31](#)
- [2] Akinlar, C. and Topal, C. (2011). Edlines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13):1633 – 1642. [48](#)
- [3] Aldea, E. and Le Hégarat-Masclé, S. (2015). Robust crack detection for unmanned aerial vehicles inspection in an a-contrario decision framework. *Journal of Electronic Imaging*, 24(6):061119–061119. [48](#), [50](#)
- [4] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60. [26](#)
- [5] Arthur, D. and Vassilvitskii, S. (2006). k-means++: The advantages of careful seeding. Technical report, Stanford. [24](#), [68](#), [71](#), [87](#), [88](#)
- [6] Attneave, F. (1954). Some informational aspects of visual perception. *Psychological review*, 61(3):183. [46](#)
- [7] Ball, G. H. and Hall, D. J. (1967). A clustering technique for summarizing multivariate data. *Behavioral science*, 12(2):153–155. [24](#)
- [8] Bitossi, G., Giorgi, R., Mauro, M., Salvadori, B., and Dei, L. (2005). Spectroscopic techniques in cultural heritage conservation: A survey. *Applied Spectroscopy Reviews*, 40(3):187–228. [9](#)
- [9] Bradley, S. (2005). Preventive conservation research and practice at the British museum. *Journal of the American Institute for Conservation*, 44(3):159–173. [1](#), [5](#)
- [10] Brandmair, B. and Greiner, P. S. (2010). *Stradivari Varnish: Scientific Analysis of His Finishing Technique on Selected Instruments*. Serving Audio. [9](#)
- [11] Bucur, V. (2016). *Handbook of materials for string musical instruments*. Springer. [6](#)

- [12] Cai, W., Chen, S., and Zhang, D. (2007). Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. *Pattern recognition*, 40(3):825–838. [30](#)
- [13] Campello, R. J., Kröger, P., Sander, J., and Zimek, A. (2020). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2):e1343. [xiii](#), [26](#), [27](#)
- [14] Campello, R. J., Moulavi, D., Zimek, A., and Sander, J. (2015). Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):1–51. [26](#), [68](#), [71](#), [72](#)
- [15] Chen, S. and Zhang, D. (2004). Robust image segmentation using fcm with spatial constraints based on new kernel-induced distance measure. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1907–1916. [30](#)
- [16] Clarke, F. J., McDonald, R., and Rigg, B. (1984). Modification to the jpc79 colour-difference formula. *Journal of the Society of Dyers and Colourists*, 100(4):128–132. [19](#)
- [17] Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(5):603–619. [26](#)
- [18] CSN EN 16242 (2011). Conservation of cultural heritage - Procedures and instruments for measuring humidity in the air and moisture exchanges between air and cultural property. Standard, European Standard. [6](#)
- [19] de Vos, B. D., Berendsen, F. F., Viergever, M. A., Staring, M., and Išgum, I. (2017). End-to-end unsupervised deformable image registration with a convolutional neural network. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 204–212. Springer. [17](#)
- [20] Desolneux, A., Moisan, L., and Morel, J.-M. (2000). Meaningful alignments. *International journal of computer vision*, 40(1):7–23. [46](#), [47](#), [48](#)
- [21] Desolneux, A., Moisan, L., and Morel, J.-M. (2003). A grouping principle and four applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):508–513. [3](#), [47](#), [49](#), [51](#), [52](#), [59](#), [61](#), [81](#)
- [22] Desolneux, A., Moisan, L., and Morel, J.-M. (2004). Gestalt theory and computer vision. In *Seeing, Thinking and Knowing*, pages 71–101. Springer. [81](#)

- [23] Desolneux, A., Moisan, L., and Morel, J.-M. (2007). *From gestalt theory to image analysis: a probabilistic approach*, volume 34. Springer Science & Business Media. [xiv](#), [41](#), [42](#), [43](#), [44](#), [45](#), [46](#), [52](#), [58](#), [79](#)
- [24] Dibos, F., Koepfler, G., and Pelletier, S. (2009). Adapted windows detection of moving objects in video scenes. *SIAM Journal on Imaging Sciences*, 2(1):1–19. [62](#)
- [25] Dibos, F., Pelletier, S., and Koepfler, G. (2005). Real-time segmentation of moving objects in a video sequence by a contrario detection. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–1065. [51](#)
- [26] Dondi, P., Lombardi, L., Invernizzi, C., Rovetta, T., Malagodi, M., and Licchelli, M. (2017). Automatic analysis of UV-induced fluorescence imagery of historical violins. *Journal on Computing and Cultural Heritage*, 10(2):12:1–12:13. [9](#)
- [27] Dondi, P., Lombardi, L., Malagodi, M., and Licchelli, M. (2019). Segmentation of multi-temporal UV-induced fluorescence images of historical violins. In *New Trends in Image Analysis and Processing – ICIAP 2019*, volume 11808 of *Lecture Notes in Computer Science*, pages 81–91. Springer International Publishing. [xvii](#), [10](#), [71](#), [72](#), [73](#)
- [28] Dondi, P., Lombardi, L., Malagodi, M., Licchelli, M., Rovetta, T., and Invernizzi, C. (2015). An interactive tool for speed up the analysis of UV images of Stradivari violins. In *New Trends in Image Analysis and Processing - ICIAP 2015 Workshops*, volume 9281 of *Lecture Notes in Computer Science*, pages 103–110. Springer International Publishing. [11](#)
- [29] Dondi, P., Lombardi, L., Rocca, I., Malagodi, M., and Licchelli, M. (2018). Multimodal workflow for the creation of interactive presentations of 360 spin images of historical violins. *Multimedia Tools and Applications*, 77(21):28309–28332. [11](#)
- [30] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231. [26](#)
- [31] Everitt, B. and Hothorn, T. (2011). *An introduction to applied multivariate analysis with R*. Springer Science & Business Media. [23](#)
- [32] Fichera, G. V., Albano, M., Fiocco, G., Invernizzi, C., Licchelli, M., Malagodi, M., and Rovetta, T. (2018). Innovative monitoring plan for the preventive conservation of historical musical instruments. *Studies in Conservation*, 63(sup1):351–354. [xiii](#), [1](#), [7](#), [8](#)

- [33] Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190. [33](#)
- [34] Fiocco, G., Invernizzi, C., Grassi, S., Davit, P., Albano, M., Rovetta, T., Stani, C., Vaccari, L., Malagodi, M., Licchelli, M., et al. (2021). Reflection ftir spectroscopy for the study of historical bowed string instruments: Invasive and non-invasive approaches. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 245:118926. [6](#)
- [35] Flenner, A. and Hewer, G. (2011). A helmholtz principle approach to parameter free change detection and coherent motion using exchangeable random variables. *SIAM Journal on Imaging Sciences*, 4(1):243–276. [49](#)
- [36] Forgey, E. (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classification. *Biometrics*, 21(3):768–769. [24](#)
- [37] Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *science*, 315(5814):972–976. [25](#), [26](#)
- [38] Ghasedi Dizaji, K., Herandi, A., Deng, C., Cai, W., and Huang, H. (2017). Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*, pages 5736–5745. [34](#)
- [39] Ghedini, N., Ozga, I., Bonazza, A., Dilillo, M., Cachier, H., and Sabbioni, C. (2011). Atmospheric aerosol monitoring as a strategy for the preventive conservation of urban monumental heritage: The florence baptistry. *Atmospheric Environment*, 45(33):5979 – 5987. [1](#)
- [40] Gong, M., Zhou, Z., and Ma, J. (2011). Change detection in synthetic aperture radar images based on image fusion and fuzzy clustering. *IEEE Transactions on Image Processing*, 21(4):2141–2151. [30](#)
- [41] Grosjean, B. and Moisan, L. (2009). A-contrario detectability of spots in textured backgrounds. *Journal of Mathematical Imaging and Vision*, 33(3):313. [47](#), [48](#)
- [42] Hsu, C.-C. and Lin, C.-W. (2017). Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. *IEEE Transactions on Multimedia*, 20(2):421–429. [35](#)
- [43] Huang, P., Huang, Y., Wang, W., and Wang, L. (2014). Deep embedding network for clustering. In *2014 22nd International conference on pattern recognition*, pages 1532–1537. IEEE. [34](#)

- [44] Invernizzi, C., Fichera, G. V., Licchelli, M., and Malagodi, M. (2018). A non-invasive stratigraphic study by reflection FT-IR spectroscopy and UV-induced fluorescence technique: The case of historical violins. *Microchemical Journal*, 138:273 – 281. [9](#)
- [45] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc. [21](#)
- [46] Janssens, K. and Van Grieken, R. (2004). *Non-destructive micro analysis of cultural heritage materials*, volume 42. Elsevier. [9](#)
- [47] Károly, A. I., Fullér, R., and Galambos, P. (2018). Unsupervised clustering for deep learning: A tutorial survey. *Acta Polytechnica Hungarica*, 15(8):29–53. [36](#)
- [48] Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons. [24](#)
- [49] Kaya, I. E., Pehlivanlı, A. Ç., Sekizkardeş, E. G., and Ibrikci, T. (2017). Pca based clustering for brain tumor segmentation of t1w mri images. *Computer methods and programs in biomedicine*, 140:19–28. [31](#)
- [50] Kramer, R., Schellen, H., and Van Schijndel, A. (2016). Impact of ashrae’s museum climate classes on energy consumption and indoor climate fluctuations: Full-scale measurements in museum hermitage amsterdam. *Energy and Buildings*, 130:286–294. [6](#)
- [51] Le Hégarat-Masclé, S., Aldea, E., and Vandoni, J. (2019). Efficient evaluation of the number of false alarm criterion. *EURASIP Journal on Image and Video Processing*, 2019(1):35. [51](#)
- [52] Lei, T., Jia, X., Zhang, Y., He, L., Meng, H., and Nandi, A. K. (2018). Significantly fast and robust fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering. *IEEE Transactions on Fuzzy Systems*, 26(5):3027–3041. [30](#), [71](#), [72](#)
- [53] Lisani, J. L. and Morel, J.-M. (2003). Detection of major changes in satellite images. In *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, volume 1, pages I–941. IEEE. [49](#)
- [54] Liu, G., Gousseau, Y., and Tupin, F. (2019). A contrario comparison of local descriptors for change detection in very high spatial resolution satellite images of urban areas. *IEEE Transactions on Geoscience and Remote Sensing*. [48](#)
- [55] Lowe, D. G. (1985). Perceptual organization and visual recognition. [43](#), [46](#), [47](#)

- [56] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110. [17](#)
- [57] Lucchi, E. (2018). Review of preventive conservation in museum buildings. *Journal of Cultural Heritage*, 29:180 – 193. [1](#)
- [58] Luo, M. R., Cui, G., and Rigg, B. (2001). The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 26(5):340–350. [xiii](#), [18](#), [19](#)
- [59] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA. [24](#)
- [60] McDonald, R. and Smith, K. (1995). Cie94-a new colour-difference formula. *Journal of the Society of Dyers and Colourists*, 111(12):376–379. [19](#)
- [61] McLachlan, G. J. and Basford, K. E. (1988). *Mixture models: Inference and applications to clustering*, volume 38. M. Dekker New York. [68](#), [71](#), [87](#), [88](#)
- [62] Michaelsen, E. (2016). Self-organizing maps and gestalt organization as components of an advanced system for remotely sensed data: An example with thermal hyper-spectra. *Pattern Recognition Letters*, 83:169 – 177. *Advances in Pattern Recognition in Remote Sensing*. [49](#)
- [63] Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., and Long, J. (2018). A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514. [xiii](#), [30](#), [34](#), [35](#), [36](#), [37](#)
- [64] Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60. [29](#)
- [65] Mou, L., Bruzzone, L., and Zhu, X. X. (2018). Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):924–935. [18](#)
- [66] Moulon, P., Monasse, P., and Marlet, R. (2012). Adaptive structure from motion with a contrario model estimation. In *Asian Conference on Computer Vision*, pages 257–270. Springer. [49](#)

- [67] Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2):181–201. [33](#)
- [68] Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*. [68](#), [71](#), [87](#), [88](#)
- [69] Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14. [31](#), [32](#)
- [70] Ortiz, R. and Ortiz, P. (2016). Vulnerability index: A new approach for preventive conservation of monuments. *International Journal of Architectural Heritage*, 10(8):1078–1100. [1](#), [6](#)
- [71] Pavan, M. and Pelillo, M. (2006). Dominant sets and pairwise clustering. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):167–172. [32](#)
- [72] Premachandran, V. and Kakarala, R. (2013). Consensus of k-nns for robust neighborhood selection on graph-based manifolds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1594–1601. [32](#)
- [73] Rezaee, M. J., Eshkevari, M., Saberi, M., and Hussain, O. (2021). Gbk-means clustering algorithm: An improvement to the k-means algorithm based on the bargaining game. *Knowledge-Based Systems*, 213:106672. [68](#), [71](#), [87](#), [88](#)
- [74] Rezaei, A., Aldea, E., Dondi, P., Malagodi, M., and Le Hégarat-Masclé, S. (2019). Detecting alterations in historical violins with optical monitoring. In *Proceedings of the 14th International Conference on Quality Control by Artificial Vision (QCAV)*, volume 11172, pages 1117210–1 – 1117210–8. [49](#)
- [75] Rezaei, A., Le Hégarat-Masclé, S., Aldea, E., Dondi, P., and Malagodi, M. (2022). A-contrario framework for detection of alterations in varnished surfaces. *Journal of Visual Communication and Image Representation*, 83:103357. [56](#)
- [76] Robin, A., Moisan, L., and Le Hégarat-Masclé, S. (2010). An a-contrario approach for subpixel change detection in satellite imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1977–1993. [49](#), [50](#)
- [77] Rodriguez, A. and Laio, A. (2014). Clustering by fast search and find of density peaks. *science*, 344(6191):1492–1496. [27](#), [68](#), [71](#)
- [78] Rousseau, F., Faisan, S., Heitz, F., Armspach, J.-P., Chevalier, Y., Blanc, F., de Seze, J., and Rumbach, L. (2007). An a contrario approach for change detection in 3D multimodal images: application to multiple sclerosis in mri. In *2007 29th*

- Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2069–2072. IEEE. [49](#)
- [79] Rovetta, T., Invernizzi, C., Fiocco, G., Albano, M., Licchelli, M., Gulmini, M., Alf, G., Rombola, A., and Malagodi, M. (2019). The case of Antonio Stradivari 1718 ex-San Lorenzo violin: History, restorations and conservation perspectives. *Journal of Archaeological Science: Reports*, 23:443–450. [1](#), [7](#)
- [80] Rovetta, T., Invernizzi, C., Licchelli, M., Cacciatori, F., and Malagodi, M. (2018). The elemental composition of Stradivari’s musical instruments: new results through non-invasive EDXRF analysis. *X-Ray Spectrometry*, 47(2):159–170. [9](#)
- [81] Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319. [33](#)
- [82] Shah, S. A. and Koltun, V. (2018). Deep continuous clustering. *arXiv preprint arXiv:1803.01449*. [34](#)
- [83] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905. [32](#)
- [84] Soille, P. (2013). *Morphological image analysis: principles and applications*. Springer Science & Business Media. [60](#)
- [85] Springenberg, J. T. (2015). Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*. [36](#)
- [86] Stovel, H. (1998). *Risk preparedness: a management manual for world cultural heritage*. ICCROM. [5](#)
- [87] Stuart, B. H. (2007). *Analytical techniques in materials conservation*. John Wiley & Sons. [9](#)
- [88] Suganya, R. and Shanthi, R. (2012). Fuzzy c-means algorithm-a review. *International Journal of Scientific and Research Publications*, 2(11):1. [29](#), [30](#)
- [89] Thomson, G. et al. (1968). *Contributions to the London Conference on museum climatology, 18-23 September 1967*. International Institute for Conservation of Historic and Artistic Works. [5](#)
- [90] Torr, P. H. and Zisserman, A. (2000). Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138–156. [17](#)

- [91] Veit, T., Cao, F., and Bouthemy, P. (2006). An a contrario decision framework for region-based motion detection. *International journal of computer vision*, 68(2):163–178. [48](#)
- [92] Veit, T., Cao, F., and Bouthemy, P. (2007). Space-time a contrario clustering for detecting coherent motions. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 33–39. IEEE. [48](#)
- [93] Vincent, L. (1993). Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE transactions on image processing*, 2(2):176–201. [30](#)
- [94] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416. [31](#)
- [95] Wang, J., Chang, S.-F., Zhou, X., and Wong, S. T. (2008). Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE. [32](#)
- [96] Wang, Q., Zhang, X., Chen, G., Dai, F., Gong, Y., and Zhu, K. (2018). Change detection based on faster r-cnn for high-resolution remote sensing images. *Remote sensing letters*, 9(10):923–932. [18](#)
- [97] Wang, Z., Yu, Z., Chen, C. P., You, J., Gu, T., Wong, H.-S., and Zhang, J. (2017). Clustering by local gravitation. *IEEE transactions on cybernetics*, 48(5):1383–1396. [68](#), [71](#), [72](#), [87](#), [88](#)
- [98] Widynski, N. and Mignotte, M. (2011). A contrario edge detection with edgelets. In *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 421–426. IEEE. [48](#)
- [99] Wirilander, H. (2012). Preventive conservation: A key method to ensure cultural heritage s authenticity and integrity in preservation process. *E-conservation Magazine*, 6(24):165–176. [5](#)
- [100] Witkin, A. P. and Tenenbaum, J. M. (1983). On the role of structure in vision. In *Human and machine vision*, pages 481–543. Elsevier. [47](#)
- [101] Wu, G., Kim, M., Wang, Q., Munsell, B. C., and Shen, D. (2016). Scalable high-performance image registration framework by unsupervised deep feature representations learning. *IEEE Transactions on Biomedical Engineering*, 63(7):1505–1516. [17](#)

- [102] Wu, Z.-d., Xie, W.-x., and Yu, J.-p. (2003). Fuzzy c-means clustering algorithm based on kernel method. In *Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003*, pages 49–54. IEEE. [33](#)
- [103] Xiang, T. and Gong, S. (2008). Spectral clustering with eigenvector selection. *Pattern Recognition*, 41(3):1012–1029. [32](#)
- [104] Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR. [35](#)
- [105] Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193. [xvii](#), [23](#), [24](#), [26](#), [27](#), [31](#), [32](#), [33](#), [38](#)
- [106] Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678. [21](#), [22](#), [23](#), [24](#), [28](#), [29](#), [33](#)
- [107] Yan, Y., Ren, J., Sun, G., Zhao, H., Han, J., Li, X., Marshall, S., and Zhan, J. (2018). Unsupervised image saliency detection with gestalt-laws guided optimization and visual attention based refinement. *Pattern Recognition*, 79:65 – 78. [49](#)
- [108] Yang, B., Fu, X., Sidiropoulos, N. D., and Hong, M. (2017). Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR. [34](#)
- [109] Yang, J., Parikh, D., and Batra, D. (2016). Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5147–5156. [35](#)
- [110] Zhang, C., Yue, P., Tapete, D., Jiang, L., Shangguan, B., Huang, L., and Liu, G. (2020). A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:183–200. [18](#)
- [111] Zhang, D.-Q. and Chen, S.-C. (2003). Kernel-based fuzzy and possibilistic c-means clustering. In *Proceedings of the International Conference Artificial Neural Network*, volume 122, pages 122–125. Citeseer. [33](#)
- [112] Zhu, X., Change Loy, C., and Gong, S. (2014). Constructing robust affinity graphs for spectral clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1450–1457. [32](#), [68](#), [71](#), [72](#), [87](#), [88](#)

- [113] Zhuang, X., Huang, Y., Palaniappan, K., and Zhao, Y. (1996). Gaussian mixture density modeling, decomposition, and applications. *IEEE Transactions on Image Processing*, 5(9):1293–1302. [28](#)