UNIVERSITÀ DEGLI STUDI DI PADOVA

—

Dipartimento di Ingegneria Industriale

—

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCANICA

PERFORMANCE ANALYSIS AND COMPARISON OF VIRTUAL IMAGE CORRELATION ALGORITHMS APPLIED TO ADDITIVELY MANUFACTURED LATTICE STRUCTURE

Relatore: Ch.mo Prof. Ing. ENRICO SAVIO

Correlatore: Ch.mo Prof. Ing. NABIL ANWER

Correlatore: Dott. Ing. MARC-ANTOINE DE PASTRE

Laureando: FILIPPO MIOLI

Matricola: 2055146

ANNO ACCADEMICO 2022-2023

Contents

1 Introduction **3**

1.0.1 Aim of the work 6

2 DBACD method formulation **9**

2.1 Parametric Curves 9

2.1.1 Implicit form and parametric forms 9

2.1.2 Bézier Curves 10

2.1.3 B-Spline Curve 11

2.2 Score function minimization problem 13

2.2.1 Score function equation 13

2.2.2 Problem linearization 13

2.2.3 Matlab matrix notation 14

2.2.4 Problem code implementation 15

3 DBACD calibration and performance evaluation **19**

3.1 Influencing parameters 19

3.2 Best strategy selection 20

3.2.1 Different algorithm strategy proposals 20

3.2.2 Performance test 1 - Number of control points and algorithm strategy 22

3.2.3 Test 1 conclusions 26

3.3 Starting shape and position influence 29

3.4 Optimal input configuration - Taguchi method - Test 2 30

3.4.1 Test 2 results 31

3.4.2 Test 2 conclusions 33

3.5 Performance panoramic for both methods 35
3.5.1 Conclusions 36
3.6 CPPD filtering effect 36
3.6.1 Test setup 36
3.6.2 Results 37
3.7 Comparison between ISO50% + Gaussian filter and VIC methods 38
3.7.1 Comparison results 39
3.7.2 Consideration 40

4 Conclusions and future prospects 45
4.1 Performance: Weaknesses and Strengths 45
4.2 Method's applications 47
4.3 Future Works 49

A DBACD method algorithm 53

B Invert non square matrix 81

C Roundnes error 83

D Sintesi estesa in lingua italiana 85
D.1 Introduzione 85
D.1.1 Obiettivo del lavoro 88
D.2 Formulazioni 89
D.3 Calibrazione del metodo e valutazione delle prestazioni 90
D.3.1 Parametri influenti 90
D.3.2 Selezione della migliore strategia 91
D.3.3 Posizione della circonferenza di partenza 92
D.3.4 Configurazione ottimale dei parametri di input - Taguchi method 92
D.3.5 Panoramica delle prestazioni 94
D.3.6 Effetto filtrante della CPPD 94
D.4 Confronto tra il metodo ISO50% + filtro gaussiano e i metodi VIC 94
D.4.1 Risultati del confronto 95

D.4.2 Considerazioni 96
D.5 Conclusioni e prospettive future 96
D.6 Applicazioni del metodo 97
D.7 Lavori Futuri 99

Bibliography **101**

Chapter 1

Introduction

From its first introduction in the early 80's [1], **Additive Manufacturing** technologies have seen a rapid increase in interest from the industry in the last ten years. This, which is also a consequence of the **expiration** of the first patents, led also to an exponential rise in the number of academic research around this topic. Grasso et al. [2] in a review about in-situ monitoring of PBF (Powder Bed Fusion) processes showed how the number of studies per year about this specific AM topic has grown from less than **five** in 2010 to almost **fifty** in 2020. Initially introduced for **rapid prototyping**, today Additive Manufacturing is increasingly establishing itself as a proper manufacturing process for a specific type of product. In fact, thanks to the increase in the **final quality** of AM products, these technologies are becoming more and more used in the production of artefacts with **high geometric complexity** and low, if not null, **batch sizes** [1]. The biggest advantage of AM today is that the cost of fabrication for the same production system is directly related to the volume of material but almost unlinked to its complexity. **Lattice structures** (LS) are a clear consequence of this predisposition. This type of geometries, once called **porous materials** or **cellular structures**, are inspired by the lightweight and high-performance density material in nature, e.g. bones, wood and sponges [3]. LS are composed of an elementary cell geometry repeated multiple times to fill up the part volume. Remembering that **metrology requirements** are consequences of **functional conditions**, it will be immediate to realize that such radical differences between traditional processes (subtractive and casting) and AM parts imply also different **geometrical metrology tools and related standards** [4]. The need, for example, to measure complex internal shapes, non-accessible by contact measuring instruments, had led to the adoption of **Computer Tomography** (CT) as the main measuring system for AM artefacts[5]. It is common knowledge that defects occurring during the production process can significantly influence the **mechanical properties** life of artefacts, and this applies also to additively manufactured parts. Malekipour and El-Mounayri classified the defects in LPBF into four categories [6]. De Pastre et al. [7] performed a deep review of the process parameters that influence part defects and properties. Zhao et al. [8] found that the **fatigue cracks** of SLM AlSi12Mg alloy initiated from surface or subsurface **gas pores**, and its fatigue life mainly depended on the **pore size** and the **building direction**, more than the porosity fraction. Many researchers worked on **the identification of porosity** defects in AM process [8][9]. Geometrical defects, on the other hand, are also linked to the part functionality, especially when considering assembling and working conditions.

In advanced metrology, measurements are taken by capturing huge amounts of data (points cloud, images, voxels, etc.) and then by extracting information from them. One of the key steps that allows to go from unreadable data to measurement results is **contour identification**. In other words, it is the process that enables us to clearly distinguish the boundary between workpiece (material) and what surrounds it (air or another material).

In most measurements, **clouds of points** are collected from the surface, and several methods have nowadays been consolidated to fit geometric features into the collected data. With the advent of **Computer Tomography** and **Optical CMM** equipped with imaging sensors, the task is no more only to fit nominal geometries into points' clouds, but also to identify the transition values between workpiece and what surrounds it. Taking as an example CT, the results of the measuring procedure is a volume of **Voxels** (volumetric pixels) whose value between zero and one is an indication of the resistance that the material at that point provided to the **X-rays** that passed through it. Analyzing the spectrum of the value of the voxel, we will find two main peaks corresponding to the two different materials (e.g., metal and air) and a set of transition values in the middle. **ISO50% methodology**, as an example, sets as the **threshold** value between the two materials the one in the middle between the two peaks. Other approaches are available in the literature. The data output, obtained after this threshold methods application, is a set of discrete points. Lifton and Liu [10] proposed an **uncertainty evaluation** method for ISO50%. They highlighted how the measurement uncertainty due to the ISO50 surface determination method is larger for dimensions that are **sensitive** to surface determination such as internal radii, external radii and wall thicknesses, and **smaller** for threshold insensitive dimensions such as centre-to-centre distances. Furthermore, discrete point cloud information need later manipulations to extract measurement results as for traditional point clouds. These steps, described in ISO 17450-1:2011 [11] include **filtration** and **association**.

In this research, the performance of two promising **VIC** methods in shape defect identification on images coming from **additively manufactured lattice structures** was investigated. **Virtual Image Correlation** (VIC) is a boundary detection process based on creating a **virtual volume** and deforming it by minimizing a **score function** in order to best represent the **real volume** (physical). The output results of **VIC** methods are analytical information about the defects geometries of the part. The first method (**V2C**) developed by De Pastre et al. [12] [13] relies on a Virtual Volume correlation based on **modal decomposition** of the physical shape. The second method, which is the main focus of this research and has been specifically developed, employs a closed parametric curve (**B-spline**) to define the virtual image, which represents the contour of the section. Specifically, the test has been done on images obtained by slicing **CT scans** of AM lattice structures. Since the nominal shape of these sections should be a perfect circle (section of cylinders taken with planes perpendicular to its axis), the B-spline is initially defined by a closed regular polygon of **control points** (C.P.). These points are then **iteratively displaced** to deform the B-spline contour in a way that can minimize a score function Φ.

In literature, different researches have been conducted in the field of VIC. Réthoré et al [14] developed a B-spline Virtual Image Correlation method to detect and measure contour and silhouette from digital images applying consecutively sets of displacement **normal** to the starting shape. Jiang et al. [15][16] used a similar method to identify shapes on **medical images** (both in 2D and 3D). Since the displacement is always searched in the direction **normal** to the starting curve, **no complex** defect shapes could be analysed and detected (undercut). Furthermore, the influence of only a part of the input parameters has been tested. Subpixel uncertainty has been assessed in the detection of simple shapes with low presence and density of small local defect [14]. Since only a few mature methods have been developed and tested in the field of additive manufacturing and more specifically lattice structure, whose type of images are characterized by a high density of surface defects, **no uncertainty evaluation** has been made for high-resolution analysis yet. Even more, defects have often very complex shapes (Figure 1.1) not always detectable by simple models (e.g., normal displacement). Uncertainty evaluations have been performed by François [17] highlighting the sensibility of general Virtual Image Correlation methods to image **noise**, **discretization** and **blur.**

1.0.1 Aim of the work

The objective of this research, besides the development of a **Discrete B-spline Active Contour Deformation** algorithm (D.B.A.C.D), is to study how VIC methods perform in the evaluation of shapes coming from **Lattice Structure** sections. In this way, the hope is to provide provide a **perspective** of how these methods can be used within the **ISO/GPS standards scenario**. Specifically, after developing the b-spline VIC method (chapter 2), a calibration and performance evaluation will be carried out (chapter 3). In subsection 3.7.1 the **DBACD** and the **V2C** methods will be compared both visually and numerically through the identification of certain measures. Lastly, future application prospectives (section 4.2) and works (section D.7) will be proposed.

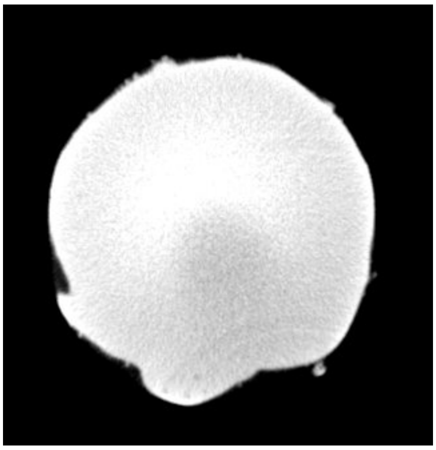


Figure 1.1: Section of a vertical strut with diam. 0.5 mm - Presence of an undercut defect on the bottom-left portion - Image 653

Chapter 2

DBACD method formulation

The first part of this chapter will describe some of the most commonly used methods for defining curves in the space. It will be shown how these are generated and managed. The information reported here is taken from [18].

2.1 Parametric Curves

2.1.1 Implicit form and parametric forms

Curves and surfaces are most commonly described in two forms: **implicit** and **parametric**. In the first form, the coordinates of the figure are the solution of an equation with 1, 2 or 3 unknowns. An example is the circle of unit radius centred in the origin which is described by the equation $f(x, y) = x^2 + y^2 - 1 = 0$. With the parametric form, each coordinate of each point is described by an **explicit function** varying with an independent parameter:

$$\mathbf{C}(u) = (x(u), y(u), z(u)) \qquad a \leq u \leq b \tag{2.1}$$

The circle of unit radius can be described by the following parametric functions

$$\begin{cases} x(u) = \cos(u) \\ y(u) = \sin(u) \end{cases} \qquad 0 \leq u \leq 2\pi \tag{2.2}$$

or from these:

$$\begin{cases} x(u) = \frac{1-t^2}{1+t^2} \\ y(u) = \frac{2t}{1+t^2} \\ y(u) = \frac{2t}{1+t^2} \\ y(u) = \frac{2t}{1+t^2} \end{cases} \qquad 0 \leq t \leq 1$$

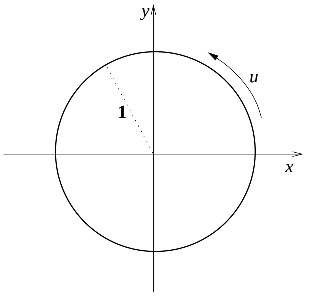


Figure 2.1: The circle of unit radius, drawn using the parametric form of the function 2.2

In an analogy between the motion of a particle and parametric representations, the independent variable u will represent time, defining the main direction of crossing.

It is difficult to say which, between implicit and parametric, is the better form of representation, as each has its advantages and disadvantages, depending on the application. Generally speaking, the explicit form is easier to use, especially with more dimension whereas defining curves in the implicit form requires calculating the intersection of two surfaces. In the course of this work, we will focus exclusively on parametric representations.

2.1.2 Bézier Curves

These curves are polynomial parametric forms: a Bézier curve of degree n is thus defined:

$$\mathbf{C}(u) = \sum_{i=0}^{n} B_{i,n}(u)\mathbf{P}_i \qquad 0 \leq u \leq 1 \tag{2.3}$$

where the basis functions (*blending functions*) $\{B_{i,n}(u)\}$ are the so-called Bernstein polynomials of degree n, given by

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!}u^i(1-u)^{n-i} \tag{2.4}$$

while the geometric coefficients $\{\mathbf{P}_i\}$ are called *control points*.

Bézier curves, differently from curves consisting of general polynomial functions, have geometric meanings in the parameters that define them. Indeed, the shape of the curve is approximated fairly well by the polygon formed by the control points $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n\}$ (it can be shown that the curve is contained entirely within the convex polygon formed by the control points).
Here are listed some of the properties of the B-spline:

1. Non-negativity: $B_{i,n}(u) \geq 0$ per ogni i, n e $0 \leq u \leq 1$;
2. Partition of the unit: $\sum_{i=0}^{n} B_{i,n}(u) = 1$ per ogni $0 \leq u \leq 1$;
3. Values at the extremes: $B_{0,n}(0) = B_{n,n}(1) = 1$;
4. $B_{i,n}(u)$ has only one maximum, placed at $u = i/n$, in the interval [0,1];
5. Recursive definition: $B_{i,n}(u) = (1-u)B_{i,n-1}(u) + uB_{i-1,n-1}(u)$; is then posed: $B_{i,n}(u) = 0$ if $i < 0$ or $i > n$;

In essence, the basis functions assign positive weights (propr.1), the sum of which gives the unit element (propr. 2), to each control point. Property 5 is important because it suggests a recursive algorithm to calculate, given a value of u, the values of the basis functions, and thus allow the representation of the curve.

2.1.3 B-Spline Curve

Due to the properties of the coefficients $B_{i,n}(u)$, as can be seen in the equation 2.4 it can be stated that to have a curve passing through n points, a polynomial of degree $(n-1)$ is required. This represents a major disadvantage for the Bézier curves just treated because it leads to inefficient algorithms and numerically unstable results. The solution to this is sums of polynomials or sums of rational polynomials.

The variability interval of the parameter $u \in [0,1]$ is subdivided into subintervals by (*breakpoints*), such as to have:

$$u_0 = 0 < u_1 < u_2 < \cdots < u_n = 1$$

Each of these subintervals becomes the domain of the corresponding polynomial, which then defines a segment of the entire curve. All segments are constructed in such a way that they join at the breakpoints with a certain degree of continuity. The various segments that form $\mathbf{C}$(u) are denoted by a subscript $C_i(u)$. All the polynomial forms seen above can be used to define $C_i(u)$; for example, by taking Bézier curves of degree 3, and dividing the domain of u into four intervals, obtaining $U = \{u_0, u_1, u_2, u_3$, we derive a curve with as many as 12 control points (four for each interval). If, however, we impose continuity of the same, we must posit $\mathbf{P}_1^3 = \mathbf{P}_2^0$ and likewise $\mathbf{P}_2^3 = \mathbf{P}_3^0$. Imposing continuity of the prime derivatives increases the conditions on the control points. Appropriate basis functions must therefore be sought, which also depend on the level of continuity required and a similar thing for the set of breakpoints.

After various considerations, we arrive at the following definitions; taken the vector $U = u_0, \ldots, u_m$, containing a non-decreasing sequence of real numbers, they are called *nodes* and U *vector of nodes*. The i-th function B-spline basis of degree p (and order $p+1$), expressed as $N_{i,p}(u)$, is defined in the following recursive way:

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{se } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{aligned} \tag{2.5}$$

With these basic functions, we can finally arrive at the definition of the B-spline curve of degree p:

$$\mathbf{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i \quad a \leq u \leq b \tag{2.6}$$

where $\{\mathbf{P}_i\}$ are the control points, and $\{N_{i,p}(u)\}$ are the B-spline basis functions of degree p, seen in D.2, defined on the vector of nodes ($m+1$ nodes):

$$U = \{\underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{b, \ldots, b}_{p+1}\}$$

We have that the degree p, the number of control points, $n+1$ and the number of nodes $m+1$ are thus related:

$$m = n + p + 1 \tag{2.7}$$

Since they consist of polynomial functions, it is relatively easy to calculate the derivatives, of any order

$$\mathbf{C}^{(k)}(u) = \sum_{i=0}^{n} N_{i,p}^{(k)}(u)\mathbf{P}_i \quad a \leq u \leq b \tag{2.8}$$

2.2 Score function minimization problem

In this second part will be reported all the equations and the mathematical steps used to implement the Discrete B-spline Active Contour Deformation algorithm.

2.2.1 Score function equation

The score function Φ defined in Equation D.3 consists of the square difference between the grey value of each pixel of the physical f and virtual g images. $\vec{x}$ is the planar coordinate of each pixel inside the region of interest ROI. A displacement field $\vec{d}$ is then inserted into the equation and will later be computed in a way to minimize Φ. The contour of the virtual image is defined as a closed B-spline Equation 2.6

$$\Phi = \iint_{ROI} [f(\vec{x}) - g(\vec{x} - \vec{d})]^2 d\Omega \tag{2.9}$$

2.2.2 Problem linearization

Thanks to the Equation 2.10 it is possible to define the deformation of the virtual contour as a separate parametric curve with the same basis function. D_i is defined as the displacement applied to the P_i control point. By inserting Equation 2.10 into Equation D.3 we obtain Equation 2.11:

$$C(u) + \Delta C(u) = \sum_{i=0}^{n} N_{i,p}(u)(P_i + D_i) = \sum_{i=0}^{n} N_{i,p}(u)P_i + \sum_{i=0}^{n} N_{i,p}(u)D_i \quad a \leq u \leq b \tag{2.10}$$

$$\Phi() = \iint_{ROI} [f(\vec{x}) - g(\vec{x} - \sum_{i=0}^{n} N_{i,p}(u)D_i)]^2 d\Omega \tag{2.11}$$

Equation 2.11 can be written also like in Equation 2.12 by replacing D_i with $D_i + \delta D_i$ where δD_i is the infinitesimal displacement applied to the P_i control point.

$$\Phi(D + \delta D) = \iint_{ROI} [f(\vec{x}) - g(\vec{x} - \sum_{i=0}^{n} N_{i,p}(u)(D_i + \delta D_i))]^2 d\Omega \tag{2.12}$$

Remembering that a function $f(x)$ can be linearized around a point a obtaining $f(x) = f(a) + \nabla f(a)(x - a) + o(||x - a||)$ is possible to write the linear Equation 2.13:

$$\Phi(D+\delta D) = \iint_{ROI} [f(\vec{x}) - g(\vec{x} - \sum_{i=0}^{n} N_{i,p}(u)D_i) - \nabla g(\vec{x} - \sum_{i=0}^{n} N_{i,p}(u)D_i)(\sum_{i=0}^{n} N_{i,p}(u)\delta D_i)]^2 d\Omega \tag{2.13}$$

2.2.3 Matlab matrix notation

The two images f g of dimension $p \times q$ are aligned on a vector of dimension $1 \times pq$. The deformation vector $[D_i]$ is a $[1 \times 2n]$ as also the $[\delta D_i]$ where n is the number of control points. Let's write again the Equation 2.13 in matrix notation without keeping track of the dimension relative to the image and the integration:

$$\Phi(\underset{1\times1}{[D] + [\delta D]}) = [[\underset{1\times1}{(f - g)}] - \underset{1\times2}{[\nabla g]}[\underset{1\times n}{[N_{i,p}(u)]}\underset{n\times2}{[\delta D_i]}]^T]^2 \tag{2.14}$$

Where:

$$(f - g) = f(\vec{x}) - g(\vec{x} - \sum_{i=0}^{n} N_{i,p}(u)D_i) \tag{2.15}$$

The function can rearranged to express the $[\delta D]$ matrix as a $[2n \times 1]$ vector :

$$\Phi(\underset{1\times1}{[D] + [\delta D]}) = [[\underset{1\times1}{(f - g)}] - \underset{1\times2n}{[[\nabla g_x][N_{i,p}(u)]\ [\nabla g_y][N_{i,p}(u)]}\underset{2n\times1}{[[\delta D_{x,i}]\ [\delta D_{y,i}]]} \tag{2.16}$$

By deriving the score function along $[\delta D]$ and setting it equal to zero, it is possible to search for local minima of the linearised function. Remembering that

$$\frac{\delta f(x)^2}{\delta x} = 2f(x) \cdot \frac{\delta f(x)}{\delta x}$$

we can obtain the Equation 2.17:

$$0 = [[\underset{1\times 1}{(f-g)}] - [\underset{1\times 2n}{\nabla g N_{i,p}(u)}][\underset{2n\times 1}{\delta D_i}]] \cdot \frac{\delta}{\delta[\delta D]}[[\underset{1\times 1}{(f-g)}] - [\underset{1\times 2n}{\nabla g N_{i,p}(u)}][\underset{2n\times 1}{\delta D_i}]] \tag{2.17}$$

Focusing on the solution of the second factor of the product on the right side of Equation 2.17 we can already divide the differential calculation between the two addends of the sum. The first addend is constant along the space of the variable $[\delta D_i]$ so it's null. For the second addend we know that the derivative of a product of a matrix follow this rule:

$$\frac{\delta[A][B]}{\delta[X]} = \frac{\delta[A]}{\delta[X]}[B] + [A]\frac{\delta[B]}{\delta[X]}$$

By some simple mathematical manipulation we can arrive at:

$$\frac{\delta}{\delta[\Delta D]}[[\underset{1\times 2n}{\nabla g N_{i,p}(u)}][\underset{2n\times 1}{\delta D_i}]] = [[\underset{1\times 2n}{\nabla g N_{i,p}(u)}] \underset{2n\times 2n}{[I]}\ ] \tag{2.18}$$

Substituting Equation 2.18 in Equation 2.17 we obtain:

$$0 = [[\underset{1\times 1}{(f-g)}] - [\underset{1\times 2n}{\nabla g N_{i,p}(u)}][\underset{2n\times 1}{\delta D_i}]]] \cdot [[\underset{1\times 2n}{\nabla g N_{i,p}(u)}] \underset{2n\times 2n}{[I]}\ ] \tag{2.19}$$

That can be rearranged in:

$$[\underset{1\times 1}{(f-g)}] \cdot [[\underset{1\times 2n}{\nabla g N_{i,p}(u)}] \underset{2n\times 2n}{[I]}\ ] = [[\underset{1\times 2n}{\nabla g N_{i,p}(u)}][\underset{2n\times 1}{\delta D_i}]] \cdot [[\underset{1\times 2n}{\nabla g N_{i,p}(u)}] \underset{2n\times 2n}{[I]}\ ] \tag{2.20}$$

$$\underset{2n\times 1}{[B]} = \underset{2n\times 2n}{[M]}\,\underset{2n\times 1}{[\delta D_i]} \tag{2.21}$$

2.2.4 Problem code implementation

The implementation as a Matlab code (Appendix A) of this linear system showed poor identification results. Different matrix inversion (**inv(M), pinv(M) and M\B** have been used producing the results in Figure 2.5. The goal was to identify a perfect circle starting from concentric and bigger circle b-spline.

This problem has been solved with the implementation of a slightly different linear system obtained by adding every i-th equation with the (n+i)-th equation with $1 \leq i \leq n$. This system can be written as in equations 2.22, 2.23, 2.24 and 2.25.

$$[B] = (f-g)(\nabla g_x + \nabla g_y)\underset{n\times 1}{[N_{i,p}(u)]} \tag{2.22}$$

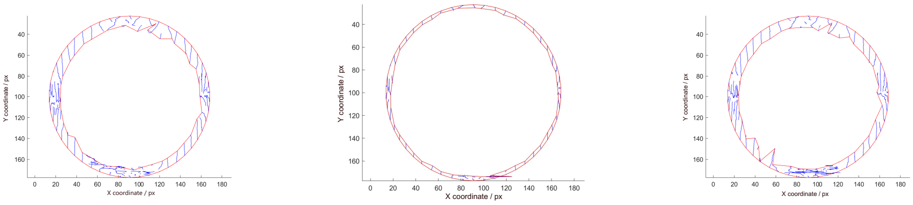


Figure 2.2: inv(M) matrix inversion function

Figure 2.3: M\B matrix inversion function

Figure 2.4: pinv(M) matrix inversion function

Figure 2.5: Control Points displacement using Equation 2.20 and different matrix inversion functions

$$\underset{n\times n}{[Mx] = [\nabla g_x(\nabla g_x + \nabla g_y)N_{i,p}(u)N_{j,p}(u)]} \tag{2.23}$$

$$\underset{n\times n}{[My] = [\nabla g_y(\nabla g_x + \nabla g_y)N_{i,p}(u)N_{j,p}(u)]} \tag{2.24}$$

$$\underset{n\times 2n}{[M] = [[Mx][My]]} \tag{2.25}$$

The two linear systems cannot be considered equivalent because the first one is composed of **2n** equations and the second one of **n** equations. A peculiar consequence of this problem formulation is that there is not a single solution of the linear equation system $[B] = [M][\Delta D]$. Indeed, for a **n** C.P. B-spline the $[M]$ matrix is a $[n \times 2n]$, the deformation vector $[\delta D]$ is a $[2n \times 1]$ and the $[B]$ vector is a $[n \times 1]$. For the **Rouché Capelli** theorem, since the maximum rank of $[M]$ is $n < 2n =$ number of elements in $[\delta D]$, the complete system will have $\infty^{(n-rk(M))}$ solutions. The reason for the malfunctioning of the formulation in Equation 2.20 is yet to be found. Henceforth, all experimentation has been carried out with the simplified formulation in equations 2.22, 2.23, 2.24 and 2.25. Given that the primary focus of this research is to compare standard procedures with the VIC methods, it is reasonable to evaluate every observation of the DBACD's performance from a general perspective, rather than intending to assess a fully developed method. It is believed that, for all the different parameters analyzed in chapter 3, the underlying influencing behaviour will remain consistent. Therefore, the evaluation will hold true in general terms, also applying to other potentially better-developed codes.

That being said, the minimization problem is carried out with an iterative process by solving the linear complete system $[B] = [M][D]$ which returns for every iteration a set of $\vec{x}$ and $\vec{y}$ displacement for every control point. This means that every C.P. has two D.F.O.s and can move in every direction without being forced to move normally to the starting circle. This allows as shown in Figure 2.6 to relocate C.P. in order to copy complex shapes like undercuts.

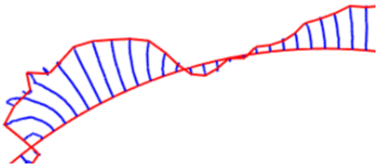


Figure 2.6: control points deformations path (blue) – first and last iteration control polygon (red)

To prevent control points from straying too far apart or reversing because they are too close together, two functions for insertion and deletion of C.P. were introduced. This is done by checking for every iteration the distance between every adjacent C.P. Taking as reference the starting distance at the creation of the control polygon, for points with a distance ratio $(dr = \frac{\text{points distance}}{\text{starting distance}})$ bigger than a threshold value u_d a control point situated in the middle between the two is added to the B-spline curve. Consequentially the base vector $[U]$ is also divided into m+1 intervals. C.P. with d_r smaller than another threshold value l_d will be replaced with a single point that corresponds to the middle point between them. Similarly to the insertion procedure, for the merging the base vector $[U]$ is also redivided into m-1 intervals.

Chapter 3

DBACD calibration and performance evaluation

In this chapter, the DBACD method will be tested to see how the different input parameters influence the detection performance. The main objective is to obtain a set of optimal values that will guarantee the best results without having to set the input parameters by user experience.

3.1 Influencing parameters

Several input parameters could affect the result of the contour identification:

1. B-spline degree.
2. Virtual image transition zone.
3. Image contrast.
4. Code implementation and solution of the linear system.
5. Control Points starting density.
6. Initial B-spline distance from the final contour.

Points 1) and 2) have already been discussed by other research. Rethore and Francois [14] found that the optimal value for the **transition zone** should lead to a transition ratio $R_g = r_v/r_f = 1$ where $rv =$ is the Virtual image transition zone and $rf =$ is the Physical image transition zone. For most of the tests carried out, they used a $p = 3$ degree B-spline, even though no big influence of these parameters on the final results has been seen.

3.2 Best strategy selection

A first experiment has been conducted to evaluate points 4), 5) and 6) influence in the **final results**. The experiment has been designed as a set of parameters combination. Four different **algorithm strategy** have been tested, each of them with different **starting number of control points**. The **Mean**, **Maximum**, **RMS Deviation** from ISO50% contour has been calculated and saved with the final **Score function** value and **Computational Time**. The contour identification algorithms have been tested with four **different images** coming from different sections of different struts. They have been chosen to well represent the **variability of conditions** that could occur.

3.2.1 Different algorithm strategy proposals

Here are listed the different algorithms:

- **noSmooth** is the algorithm developed on the mathematical model written above. The $[M]$ matrix is evaluated in every point resulting in very high computational times required.

- **noRealM** uses a simplified $[M]$ matrix with just the elements close to the diagonal to reduce computational time. For each $m_{i,j}$ element, the scalar product between $N_{i,p}$ and $N_{j,p}$ is performed. By the definition of basis function, we could state that each $N_{i,p}$ is non-null only inside the interval $[U_{i-p+1}, U_{i+p-1}]$; this means that the equation $[N_{i,p}] \cdot [N_{j,p}] \neq 0$ is true only when $|i-j| < p-1$. The result is a symmetrical matrix whose non-zero elements are all concentrated on a $2(p-1)+1$ width band. To accelerate the computational procedure, only the **non-zero** elements are calculated.

- **Smooth** implements an averaging function for the control points displacements in order to obtain smoother deformation in the initial stages. For each point deformation, the following rule is applied: $\Delta Dx_i = 1/4\Delta Dx_{i-1} + 1/2\Delta Dx_i + 1/4\Delta Dx_{i+1}$ The same rule is then applied for ΔDy_i. In Figure 3.1 is shown the C.P. displacements directly from the linear system solution (blue) and after the smoothing process (red)

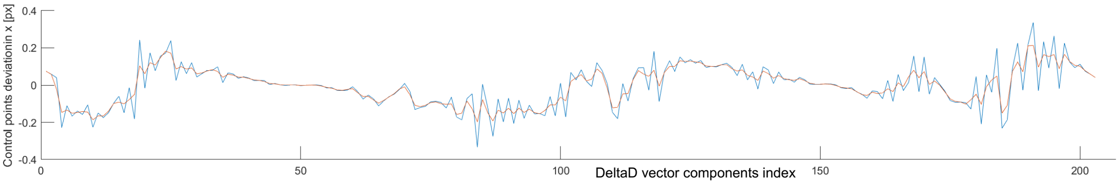


Figure 3.1: Control points displacements directly from the linear system solution (blue) and after the smoothing process (red)

- **Autoshape** implements an algorithm to perform a first coarse movement of the C.P. very close to the physical shape. It projects every control point in a **radial direction** on the **ISO50%** contour. The obtained result is a starting control polygon fitted on the ISO50% contour. This solution, although not the strictest applicable, turns out to be **very quick** to compute and **sufficiently effective**.

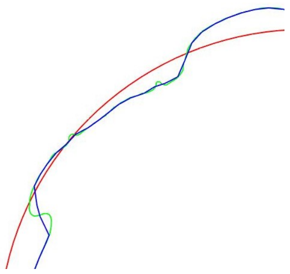


Figure 3.2: starting control polygon (red) – ISO50% contour (green) – autoshaped starting control polygon (blue)

The following paragraphs will show the obtained results. The number of **control points** is better expressed as the **control point perimetral density** (CPPD = $\frac{\text{n° of control points}}{\text{starting circumference perimeter}}$).

3.2.2 Performance test 1 - Number of control points and algorithm strategy

Image 653

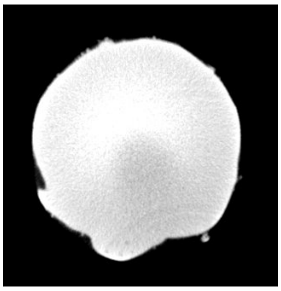


Figure 3.3: CT section of a strut - image 653

Image 653 shows an almost regular surface with few local defects. Medium contrast between the black and white regions. From the mean deviation value graph (Figure 3.4) is evident that the three **traditional methods** starting from the fitted circle (noRealM, noSmooth and Smooth) present a decreasing increasing trend with a **minimum point**. **Autoshape method**, on the contrary, shows a monotonous decreasing trend. Concerning Computational time, Autoshape is again the best-performing strategy, with noRealM being just a bit better than noSmooth and Smooth being the slowest. In Table 3.1 are reported the optimal value for CPPD.

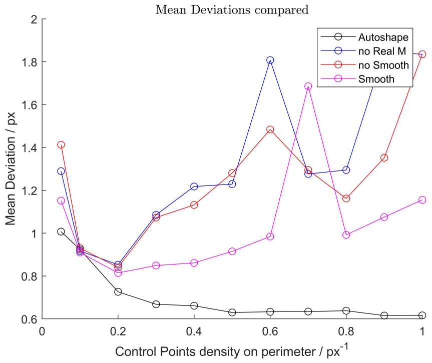


Figure 3.4: Least square deviations for im.653

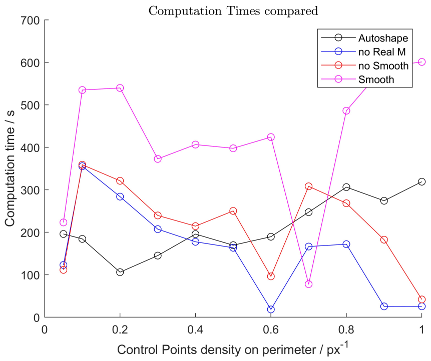


Figure 3.5: Computational times for im. 653

Table 3.1: Control points density value corresponding to optimal results - Image 653

	Optimal Density
noSmooth	0.2
noRealM	0.2
Smooth	0.2
Autoshape	0.5

Image 55

Figure 3.6: CT section of a strut - image 55

Image 55 has a very irregular shape rich in local defects. Good contrast between the two regions. The same behaviour of the previous case is also reported in this test. This time traditional **methods performances** are closer to **autoshape** until the minimum point is reached. Again, the optimal CDDP values are reported in Table 3.2.

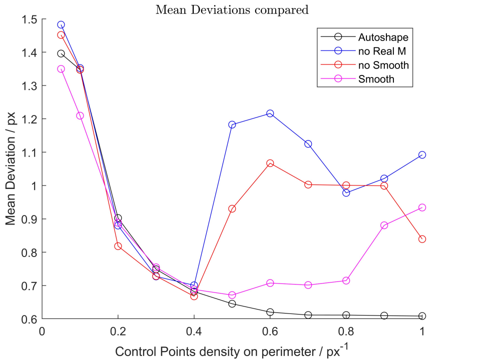


Figure 3.7: Least square deviations for im.55

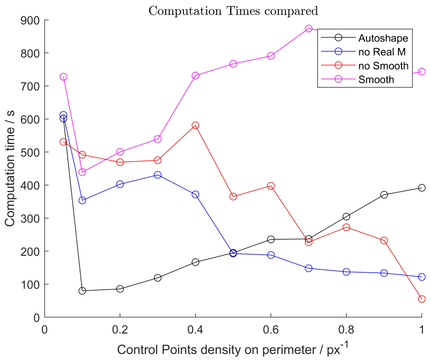


Figure 3.8: Computational times for im. 55

Table 3.2: Control points density value corresponding to optimal results - Image 55

	Optimal Density
noSmooth	0.4
noRealM	0.4
Smooth	0.5
Autoshape	0.7

Image 325

This image shows in almost all the perimeter a regular contour, except for the left side. The contrast between black and white regions is low. The trending behaviour for this test is even more extreme; only in the autoshape method the graph in Figure 3.10 shows a decreasing trend for the mean error. All the other three strategies, especially noSmooth and noRealM, almost fit in a monotone increasing curve.

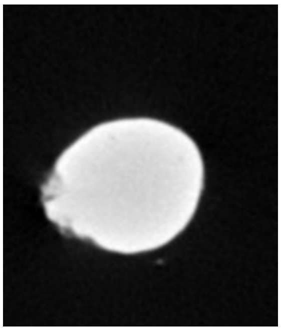


Figure 3.9: CT section of a strut - image 325

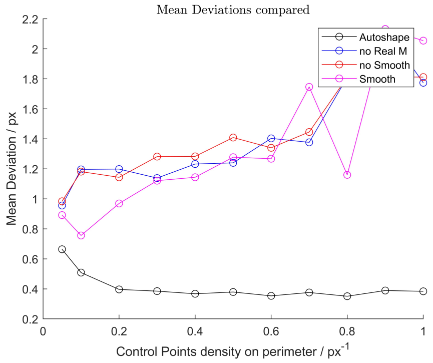


Figure 3.10: Least square deviations for im.325

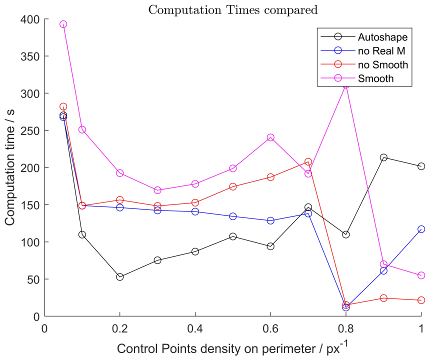


Figure 3.11: Computational times for im. 325

Image 11

This section has some local regular defects and a medium contrast level. Again, autoshape seems to be the only strategy able to increase the performance with the increase of the CPPD. The decreasing-increasing behaviour of the other three strategies is not really as appreciable as for Im.55 and Im.653, but still more than Im.55.

Table 3.3: Control points density value corresponding to optimal results - Image 325

	Optimal Density
noSmooth	0.05
noRealM	0.05
Smooth	0.1
Autoshape	0.3

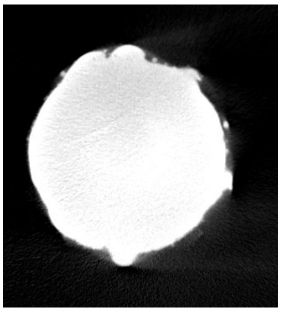


Figure 3.12: CT section of a strut - image 11

3.2.3 Test 1 conclusions

After conducting the experiment with the four different images (noise source) it was not possible to identify a single maximum value of control points density after whom there is no advantages in increasing the number of control point. CPPD optimal range have been reported in Table 3.5:

What is clear from this experiment is the general performance of the four algorithms:

Table 3.4: Control points density value corresponding to optimal results - Image 11

	Optimal Density
noSmooth	0.1
noRealM	0.2
Smooth	0.2
Autoshape	0.6

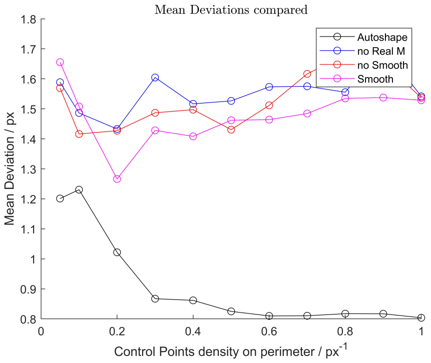


Figure 3.13: Least square deviations for im.11

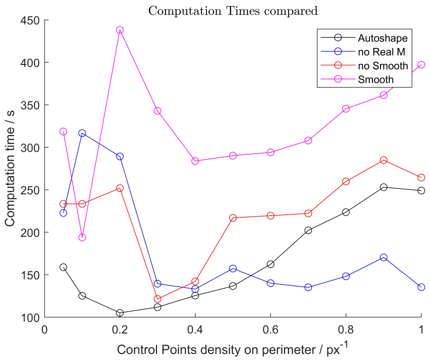


Figure 3.14: Computational times for im. 11

Table 3.5: Control points density value ranges corresponding to optimal results

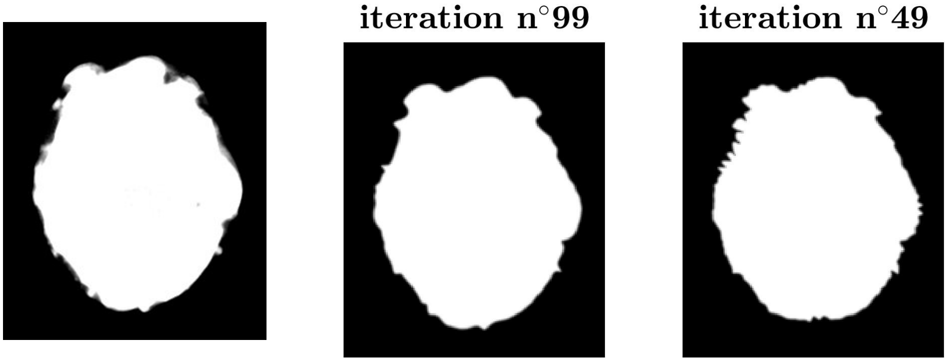
	Density correspondent to LSD minimum / stabilization
noSmooth	0.05-0.4
noRealM	0.05-0.4
Smooth	0.2-0.5
Autoshape	0.3-0.7

- **Autoshape** algorithm is the best-performing strategy both for final error and computational time. The disadvantage is that the starting position of the control points is with this algorithm less linked to the nominal profile and ***strictly dependent*** to another contour identification method. With is **monotone decreasing** trend it is almost insensible to variation in the CPPD.

- **Smooth** algorithm is the second-best performing strategy for accuracy but is way more time-consuming. Even though the mean deviation values with the optimal CPPD are comparable with the one coming from noSmooth and noRealM, the strategy seems to be **less sensible to variation** in the number of **C.P,** showing a **wider minimum plateau**.

- **noRealM** algorithm has small benefits on the computational performance but has no clear benefits on the accuracy of the results (sometimes it is better, sometimes it is worse).

Generally speaking, it seems that the lower the **image contrast**, the lower will be the optimal value for the **CPPD**. The decreasing-increasing behaviour of the three **traditional methods** starting from the fitted circle can be interpreted as a consequence of two different effects. The **decreasing trend** can be due to the **higher definition of the b-spline** contour capable of detecting smaller defects. The **increasing trend**, on the other hand, seems to be a consequence of the inability of the algorithm to detect shapes when starting too far away (bad **conditioning** of the problem). Indeed, with high CPPD and distances, the virtual contour moves toward the physical one creating **high-frequency vibrations** on his surface. This, in addition to causing problems during the iterative process (creations of loops), results in a final reconstructed section that does not match the real image profile in the microscale. An example is reported in Figure 3.18.



iteration n°99

iteration n°49

Figure 3.15: Image 55 real

Figure 3.16: Im. 55 reconstructed 0.4 CPPD

Figure 3.17: Im. 55 reconstructed 0.6 CPPD

Figure 3.18: Comparison between real section and reconstructed sections - Image 55 - noSmooth

It can be seen from this analysis that VIC methods are extremely sensitive to the distance between the initial and final shape. Therefore, from now on, future evaluations will be made for two methods: **autoshape** and **Smooth+noRealM**. Autoshape as more stable will provide important and immediate information on the sensitivity of the DBACD method. Remembering, however, that this strategy relies on already having an identified pseudo-contour, it is considered important to evaluate the sensitivity of the method also for those cases where autoshape

Table 3.6: Results of contour detection starting from different positions- Image 11

	Outer circle	mean circle	inner circle
time / s	767	628	1203
LSD / px	0.9753	0.8489	1.0116
Max deviation / px	4.8667	3.5480	4.9326
RMS / px	1.2662	1.0586	1.3409
Score function /	0.0920	0.0871	0.0931

cannot be used.

3.3 Starting shape and position influence

In the previous test, the initial shape used for contour identification was a circle with a centre correspondent to the **centroid** of the ISO50% contour and with a radius equal to the **mean distance** between the centre and each ISO50% point. This choice originates from the will to recreate some possible sources of uncertainty. Indeed, in the application of this method, it could be that the starting shape of the **virtual contour** will come from the nominal geometries, that could be slightly offset and/or with different sizes. Using the proper **Least square distance fitted circle** would have created an unlikely scenario. In this chapter, the **starting shape** definition of the control polygon will be investigated. In particular, the objective is to see if there is an actual **influence** in the algorithm performance when starting from the **maximum**, **least square** or **minimum distance** radius circle (always using the centroid as centre). For this test, the same image has been analyzed with the **noRealM** and the **Smoothness** tools starting from the three different initial circles with the **same CPPD** (0.4 point/pixel).

It can immediately be seen that starting from an **average value** of the radius of the initial circle, the performance of the DBACD method is **higher** for each selected output. This proves again that these types of algorithms are **sensitive to the starting distance**. In Figure 3.19 can be seen the typical detection error generated when starting from an initial shape far from the physical image.

Figure 3.19: Im.55 detected from the outer circle - noSmooth - error on the right side

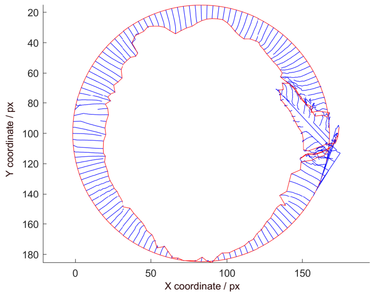


Figure 3.20: Control point displacement (blu)

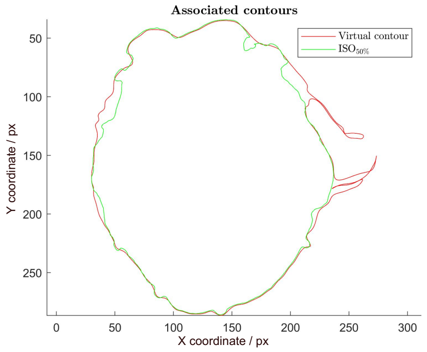
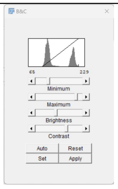
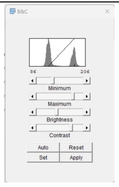
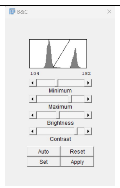


Figure 3.21: Final b-spline contour (red) - ISO50% contour (green)

Table 3.7: Influencing parameters levels

Parameter/Level	Low (1)	Medium (2)	High (3)
Trans. size ratio R_g (A) / px	0.5	1	2
B-spline degree p (B) /	2	3	4
C.P.P.D ρ (C) / $\frac{\text{point}}{\text{px}}$	Low range extreme	Middle range	High range extreme
Contrast c (D) /	 B&C 65 229 Minimum Maximum Brightness Contrast Auto Reset Set Apply	 B&C 86 206 Minimum Maximum Brightness Contrast Auto Reset Set Apply	 B&C 104 182 Minimum Maximum Brightness Contrast Auto Reset Set Apply

3.4 Optimal input configuration - Taguchi method - Test 2

For the next experiment, the **Taguchi Method** for the design of experiments will be used. The two most promising methods (Smooth and autoshape) will be tested with an **orthogonal array** of input combination. The process parameters taken into consideration are reported in Table 3.7.

An $L9(L3^4)$ orthogonal array (Table 3.8) will be used to conduct the experiment.

Table 3.8: Orthogonal table for tests design

Test n°	Input Levels			
	A	B	C	D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Table 3.9: Percentage deviation from the mean value for parameters levels - autoshape

Parameter/Level	Low (1)	Medium (2)	High (3)
Transition size ratio R (A) / %	-4,942	-2,401	7,343%
B-spline degree p (B) / %	3,250	-2,247	-1,003
C.P.P.D ρ (C) / %	7,622	-4,212	-3,410
Contrast c (D) / %	-3,244	2,601	0,643

3.4.1 Test 2 results

Autoshape

The Taguchi experiment has been run with two different images (Figure 3.12 and Figure 3.6). Results for each image have been **normalized** with the mean value of the nine tests. In Figure 3.22 can be seen the **optimal values** for each input parameter. To add more perspective to the analysis in Table 3.9 are reported the **percentage deviation** associated with the previous graphs.

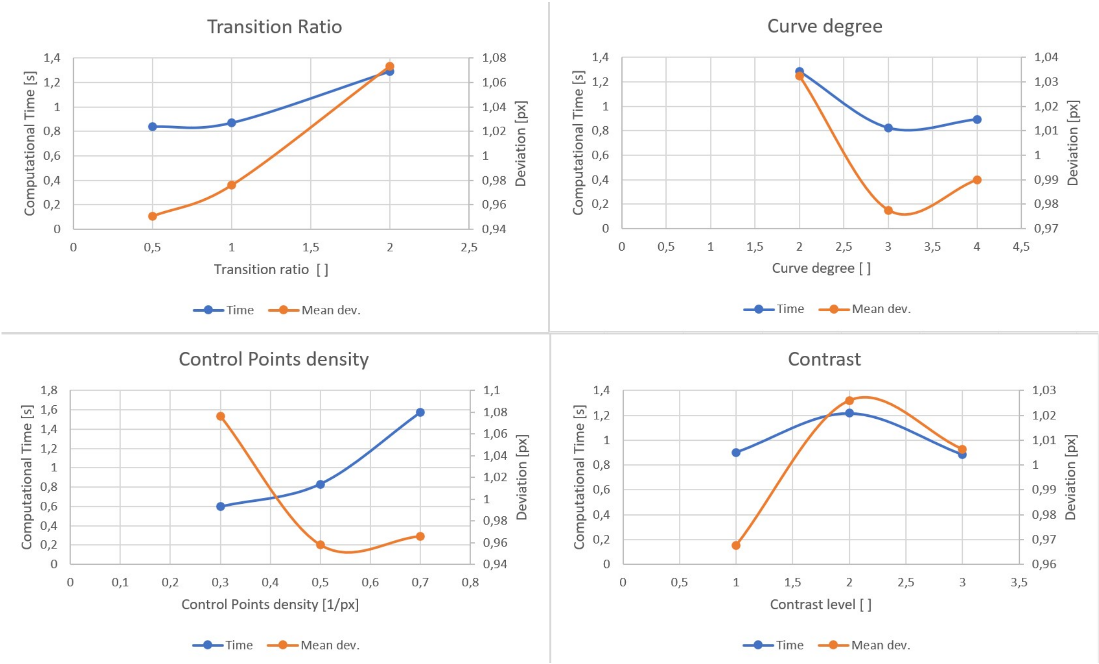


Figure 3.22: Mean deviation value vs. parameter level - autoshape method

Table 3.10: Percentage deviation from the mean value for parameters levels - smooth

Parameter/Level	Low (1)	Medium (2)	High (3)
Transition size ratio R (A) / %	-0,366	-0,464	0,830
B-spline degree p (B) / %	0,128	1,857	-1,984
C.P.P.D ρ (C) / %	1,046	-3,545	2,500
Contrast c (D) / %	5,237	-0,140	-5,097

Smooth + noRealM

In Figure 3.23 can be seen the **optimal values** for each input parameter. To add more perspective to the analysis in Table 3.9 are reported the **percentage deviation** associated with the previous graphs.

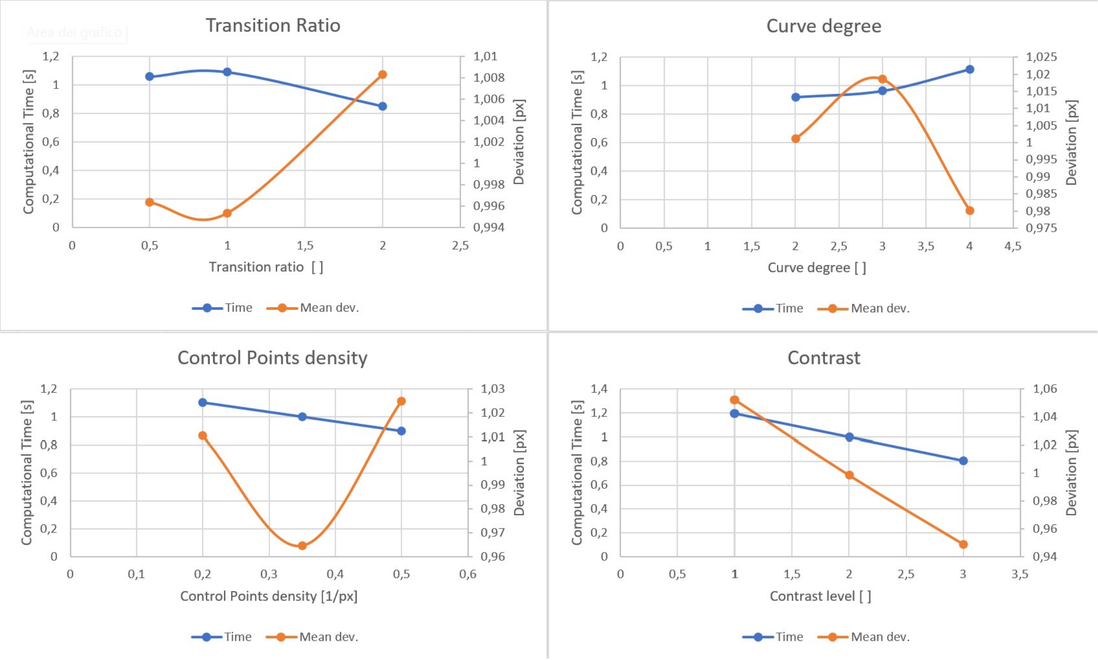


Figure 3.23: Mean deviation value vs. parameter level - smooth method

3.4.2 Test 2 conclusions

In Table 3.11 and Table 3.12 the deviation are expressed in absolute numbers. In this way is easier to appreciate the real contribution of measurement error relative to the parameters choices. **Smooth + noRealM** seems to be less sensible to **transition size ratio** but highly influenced by **CPPD** and **image contrast**. **Autoshape**, on the other hand, shows higher sensibility to **transition size ratio**, especially with high R_g values. This is attributable to the fact of having a deformation process that starts already with the final contour inside the transition zone. Furthermore, the influences of the **contrast level** also seem to be in opposition with the **Smooth + noRealM** method. Again this behaviour is to be expected given the nature of the two strategies. When performing this type of evaluation one must remember that the benchmark method is itself sensible to image contrast. To obtain a proper answer about the sensibility of this VIC method to image contrast, the Mean deviation value should be calculated with respect to a benchmark contour known with a lower uncertainty due to contrast. Instead, is difficult to interpret the different graphs for the **curve degree**. Even though in both methods p variation generates small errors, the fact that $p = 3$

Table 3.11: Deviation from the mean value for parameters levels - autoshape

Parameter/Level	Low (1)	Medium (2)	High (3)	Δ
Trans. size ratio R (A) / px	-0,038	-0,018	0,056	0,094
B-spline degree p (B) /	0,025	-0,017	-0,008	0,042
C.P.P.D ρ (C) / $\frac{\text{point}}{\text{px}}$	0,058	-0,032	-0,026	0,090
Contrast c (D) /	-0,025	0,020	0,005	0,045

Table 3.12: Deviation from the mean value for parameters levels - smooth

Parameter/Level	Low (1)	Medium (2)	High (3)	Δ
Trans. size ratio R / px (A)	-0,004	-0,005	0,010	0,015
B-spline degree p / (B)	0,001	0,022	-0,023	0,045
C.P.P.D ρ / $\frac{\text{point}}{\text{px}}$ (C)	0,012	-0,041	0,029	0,071
Contrast c / (D)	0,061	-0,002	-0,060	0,121

corresponds to a minimum for **autoshape** and to a maximum for **Smooth + noRealM** is still an unresolved issue. One could say that a higher curve degree means that the position of each point is influenced by a higher number of control points, maybe helping to reduce the vibrating behaviour seen when approaching the physical shape from high distances. Given its low influence on the method performance, no further investigation has been carried out and the optimal value has been kept $p = 3$. For what concern the image contrast, this test helped to highlight the different sensibility of the two strategies, but to perform a proper uncertainty evaluation it will be necessary to calculate the **expanded uncertainty**. Methods to to evaluate the ISO50% uncertainty can be explored in [5] [10]

3.5 Performance panoramic for both methods

As also stated by Jiang et al. [15] local minimum value for the score function can be used as a signal to increase the number of control points. Following this idea, an alternative strategy that could link the necessity of VIC methods to start close to the final contour and the will to have an independent method has been developed. With this strategy, the starting polygon for the b-spline is created by interpolation of a higher number of control points on the polygon resulting from a previous image correlation. Doing so, it will be possible to perform a first movement of a low number of **C.P.** towards the physical contour without suffering from low problem conditioning. After a first ”coarse” contour is identified, the CPPD could be gradually increased until an optimal result is obtained. To perform this was necessary to define the:

- initial points density.
- CPPD increasing step for refinement.
- final optimal control point density.

Taking 0.1 as the starting control points perimetral density, one test has been run by gradually increasing it by 0.1 $\frac{\text{points}}{\text{px}}$ until the value of 0.8. A second test has been conducted by direct interpolation of the 0.8 CPPD polygon into the 0.1 CPPD one. The value of 0.8 has been decided as being bigger than the higher extreme of the optimal value for **autoshape** method. For reference purposes, other two contour identifications have been conducted, both with a single step with CPPD= 0.8, one with smooth and the other with autoshape. In Figure 3.24 is reported the **score function value** as a function of the number of iterations. Remembering that the **computational time** is directly proportional to the **number of iteration**, the following observation could be made:

- autoshape method is visibly located in a higher efficiency region of the graph (bottom-left).
- The result of the identification performed with two steps of CPPD shows lower residual error compared to the one performed with eight steps (the

first and last values of CPPD are the same). Even more unexpected is the fact that its residual value is even better than the one obtained with autoshape.

- in Figure 3.26 and Figure 3.27 we could notice that the two identified contours, even if quite similar, differ in the identification of a dross on the right portion of the section. This behaviour will be discussed in section section 4.1.

3.5.1 Conclusions

The graph in Figure 3.24 is a perfect representation of the ideal use of each strategy. **Autoshape** should be, as stated multiple times, the benchmark methodology, with the fastest identification time and with optimal results. When it is not possible to pre-define the virtual initial contour, **Smooth + noRealM** with a two-step progressive analysis will give similar or even better results, but longer computational times. If the time required is also a critical limitation, **Smooth + noRealM** in a single step is the only remaining option, limited in the resolution of the b-spline contour.

3.6 CPPD filtering effect

3.6.1 Test setup

15 different artificial images have been generated thanks to the **V2C** mathematical model in order to represent surface vibration of one specific **frequency** $[\frac{\text{complete oscillation}}{\text{full revolution}}]$. The frequency represented are: $[2, 3, 4, 5, 6, 7, 8, 10, 20, 30, 40, 50, 60, 70, 80]$ **undulation per revolution** (upr). These images have been analysed with **DBACD** and the **CPPD** has been set to correspond at **80 control points**. It is expected that after the frequency of **40** upr, the **CPPD** value will not be high enough to match the vibrated geometry.

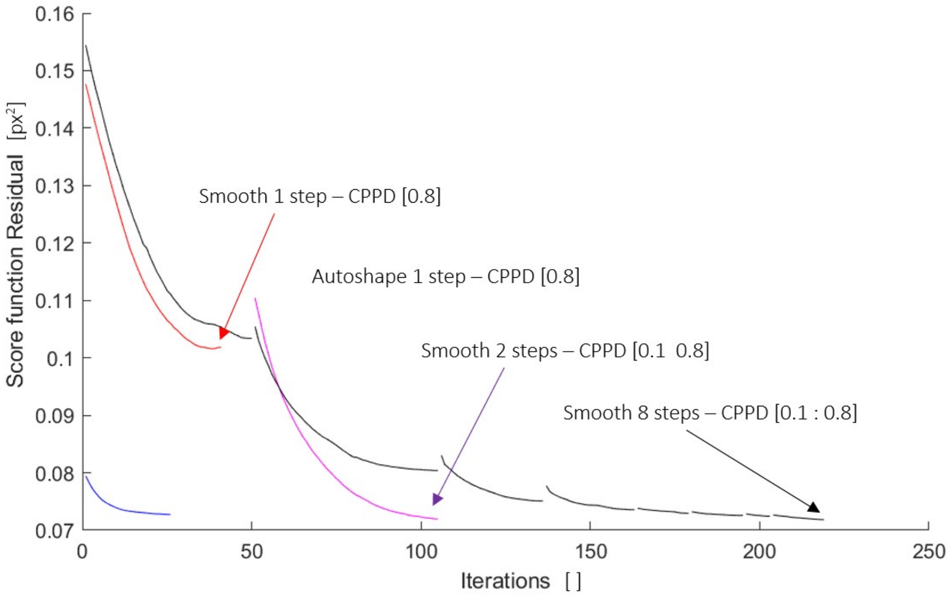


Figure 3.24: Residual score function value when progressively increasing the CPPD - image 11

3.6.2 Results

The result of this experiment confirmed what was expected: the ability of the **DBACD** method to identify specific frequency vibration decrease when approaching $\lambda_f = 2d$ and becomes null for shorter wavelength (λ_f wavelength associated to the f frequency). This is because to fit a b-spline in a sinusoidal oscillating curve, it is required to have at least one control point for every maximum and minimum of the sin function. In Figure 3.33 could be seen the transition over the cut-off value $\lambda_f = 40$. Over this value, the b-spline detected contour is well approximated to a perfect circle, as can be seen in Figure 3.32. The decreasing trend after $\lambda_f = 40$ in Figure 3.33 is because the average distance between the vibrant circle and the b-spline reconstructed one is proportional to the integral in Equation 3.1:

$$\int_0^{2\pi} |cos(nx)| = 4/n \tag{3.1}$$

which decreases with the increase of n.

It is so possible to state that the **DBACD** method is not able to identify defects with wavelengths shorter than $\lambda_f = 2d$ with d starting distance of the

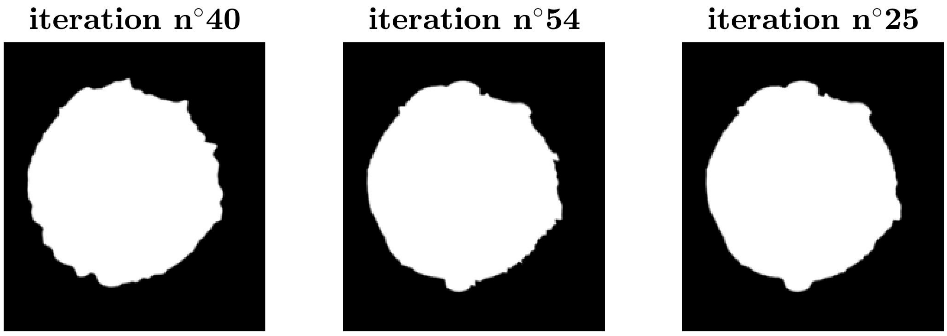


Figure 3.25: Image 11 reconstructed smooth in 1 step

Figure 3.26: Image 11 with reconstructed smooth in 2 step

Figure 3.27: Image 11 with reconstructed with autoshape in 1 step

Figure 3.28: Comparison between reconstructed sections - Image 11

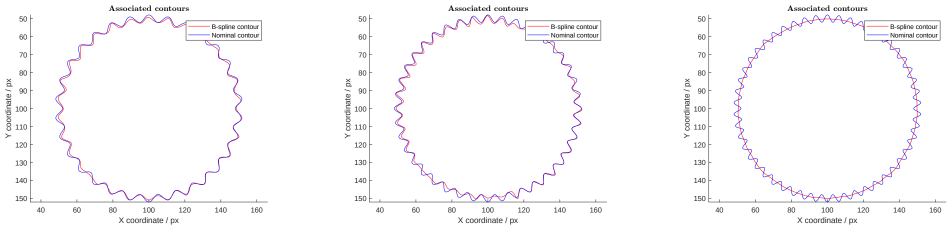


Figure 3.29: Frequency 30

Figure 3.30: Frequency 40

Figure 3.31: Frequency 50

Figure 3.32: Comparison between reconstructed frequency - 80 C.P.

control points. The **CPPD** can so be used as a **filtering parameters** for the method.

3.7 Comparison between ISO50% + Gaussian filter and VIC methods

A further step to understand the performance of these methods has been done with the following experiment. The contour of Figure 3.12 has been reconstructed with the three different methods:

- ISO50% + gaussian filter according to ISO 16610-21 [19] (Filter function)

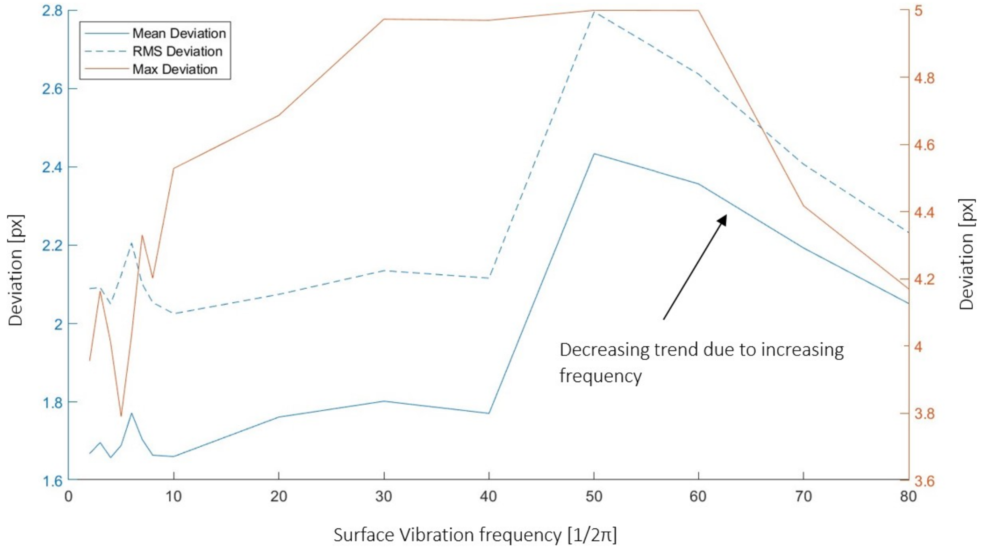


Figure 3.33: Mean, RMS and Max deviations for different vibration frequency - Smooth 80 C.P.

- DBACD method with autoshape algorithm and noSmooth. Curve degree three, $R_g = 0.5$.

- V2C method

The three contours are then compared to see if there is a relationship between the **resolution** input parameters and the filtering effect. This estimated relation is reported in Equation D.5

$$f_c = \frac{(n_m)}{2} - 3 = \frac{(n_c p)}{2} \tag{3.2}$$

with n_m net number of modes used for V2C and $n_c p$ number of control points used for DBACD. This comparison has been run with different values of f_c.

$$f_c = [5\ 10\ 15\ 20\ 25\ 30\ 35\ 40\ 45\ 50]$$

3.7.1 Comparison results

For each contour the **least square distance fitted circle**, the **maximum inscribed circle** and the **minimum circumscribed circle** are calculated following the definition from ISO17450-1 [11] and implemented in the functions: LSD circle, Minimum circumscribed Circle and Maximum inscribed circle. A roundness error function has been implemented in Appendix C. The resulting radius value and centre position have been plotted in the graphs in Table 3.13. With this information, we can state that:

- ISO50% + Gaussian filter, taken as reference, shows in all graphs great stability and clear trends.
- the relative errors between ISO50% and DBACD or V2C (Table 3.14) is generally higher with lower **cut-off filter** frequency. It is also appreciable that the relative errors for M.I. and M.C circles are generally higher than L.S.D.
- DBACD and V2C shows similar behaviour confirming the relationship in Equation D.5. Additionally, DBACD seems less stable at low f_c and more stable at high f_c compared to V2C.
- for what concerns the roundness evaluation, the relative deviation from the ISO50% reference value can arrive at peaks around **30%**, showing higher sensibility to different defect filtration properties at low f_c.

3.7.2 Consideration

After conducting this latest experiment, it can once again affirm the **effectiveness** of the DBACD and V2C methods in analyzing the overall shape of the section. However, one recurring issue lies in the sub-optimal detection of local defects due to local minima in the score function Φ. As the size of the smallest detectable defect increases with the lowering of the cut-off frequency, the impact of potential mis-detection on the final results becomes more significant.

This observation is evident in Figure 3.47, where the filtering results are compared. For instance, when using $f_c = 10$ upr, the DBACD method fails to represent a defect located in the top-right portion of the image (Figure 3.45). Likewise, the V2C method erroneously recognizes a small, partially melted volume near the edge as an actual part of the section (Figure 3.46 right side).

To overcome these challenges, further refinement and optimization of the detection algorithms will be essential in enhancing the accuracy and reliability of our analysis.

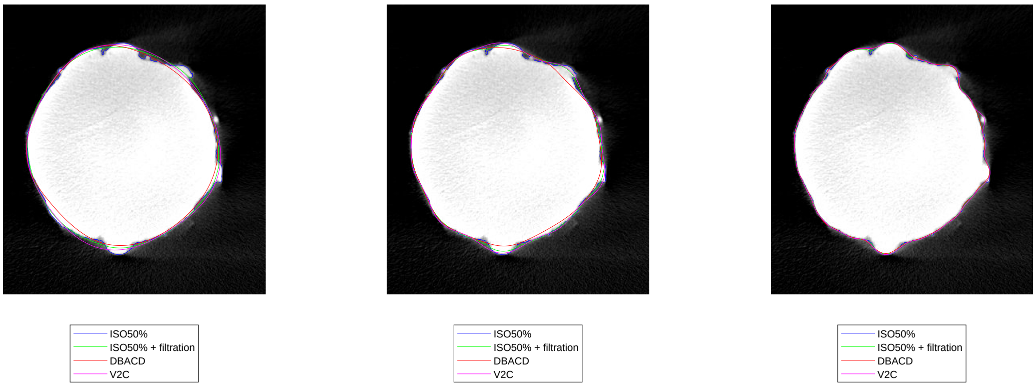


Figure 3.44: Image 11 reconstructed with $f_c = 5$ upr

Figure 3.45: Image 11 reconstructed with $f_c = 10$ upr

Figure 3.46: Image 11 reconstructed with $f_c = 30$ upr

Figure 3.47: Comparison between reconstructed sections - Image 11

Table 3.13: Radius value and center position at different f_c obtained with L.S.D., M.I. and M.C. circle from ISO50% + filt., DBACD and V2C contours

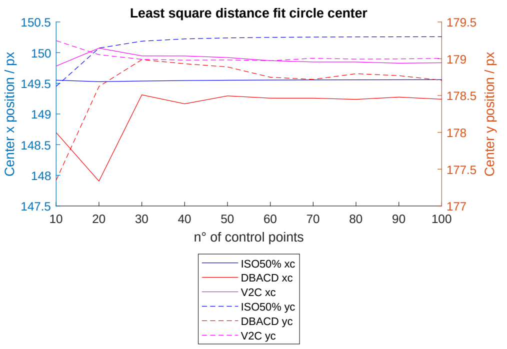


Figure 3.34: Center position for L.S.D fitted circle

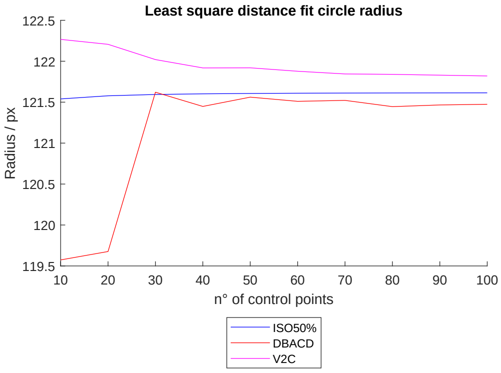


Figure 3.35: Radius value for L.S.D fitted circle

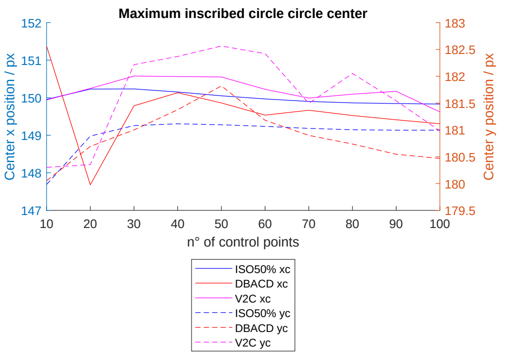


Figure 3.36: Center position for M.I. circle

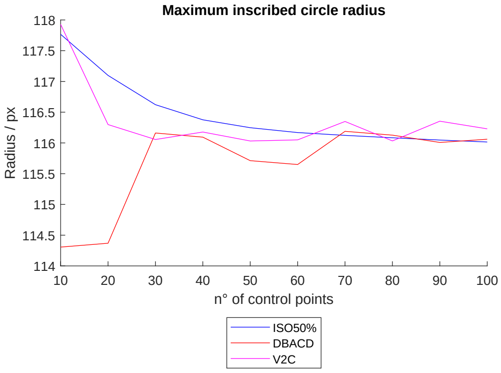


Figure 3.37: Radius value for M.I. circle

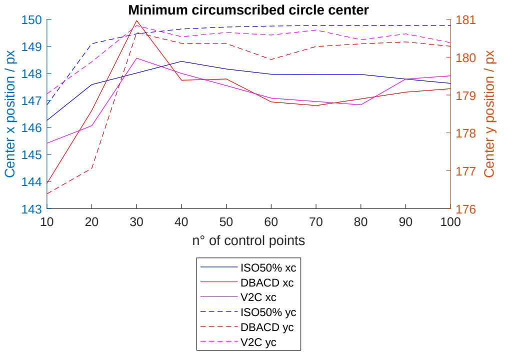


Figure 3.38: Center position for M.C. circle

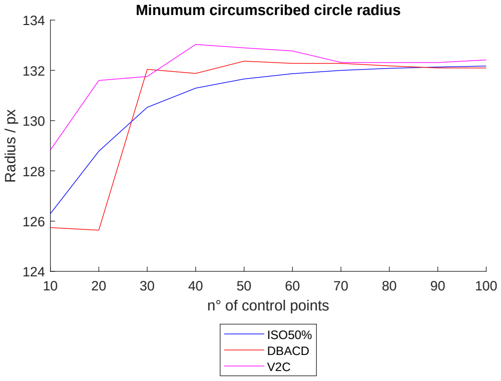


Figure 3.39: Radius value for M.C. circle

Table 3.14: Radius value and center position relative errors at different f_c obtained with L.S.D., M.I. and M.C. circle from DBACD and V2C contours using ISO50% + filt. as reference

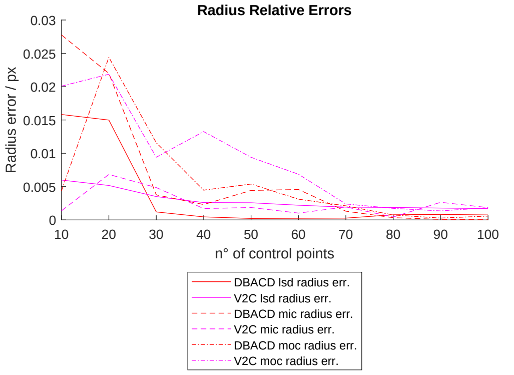


Figure 3.40: Radius value relative error

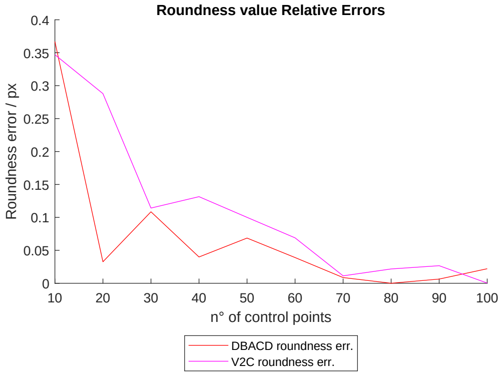


Figure 3.41: Roundness value relative error

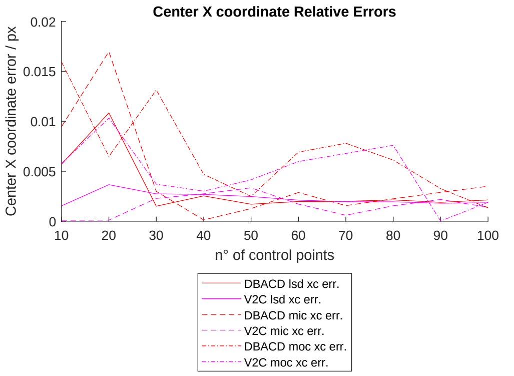


Figure 3.42: Center X coord. relative error

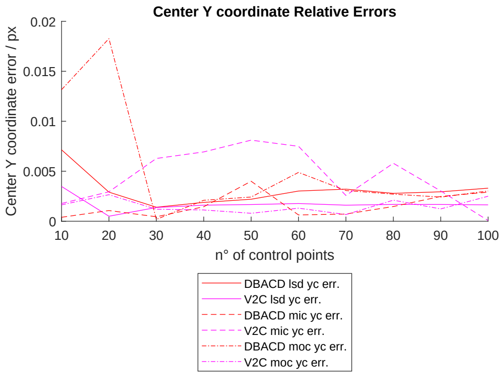


Figure 3.43: Center Y coord. relative error

Chapter 4

Conclusions and future prospects

In the first part of this chapter (4.1) will be summarized the results of the performance tests run on the DBACD method. With this information, in section 4.2 will be analysed possible use of this contour identification algorithm and the **V2C** method. Lastly, future possible works will be discussed in section D.7.

4.1 Performance: Weaknesses and Strengths

The **Discrete B-spline Active Contour Detection** method tested in this document has shown controversial performances. When performed with the right input condition, the results could be compared to the **ISO50% contour**, especially after filtration. Even under the best conditions, though, the method has not always been able to identify the smallest notches due to a resolution limitation (Figure 4.1). As stated numerous times, it will be possible to mitigate this gap by increasing the **CPPD**, but the price to pay will be **lower robustness** and excessive long **computational time**. When the method cannot rely on previously identified contours, the sensitivity to other parameters like **image contrast** increases and the overall **performances decrease**. Furthermore, given the iterative nature of this problem, a new type of uncertainty source is introduced. Indeed, it has been seen multiple times (section 3.3, section 3.5 and subsection 3.7.2) that the starting position of the virtual contours could affect the identification process. Taking Figure 4.2 as an example, it is logical to understand

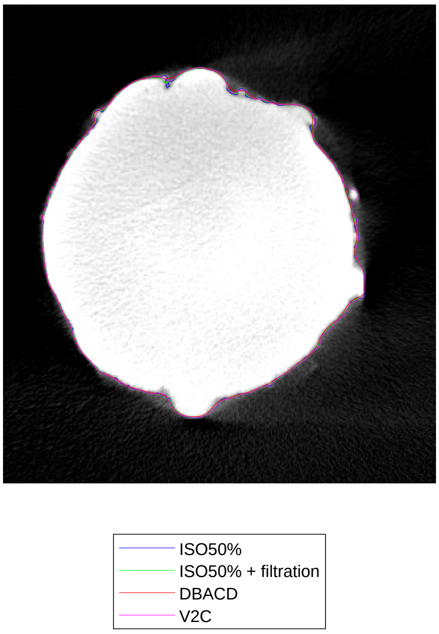


Figure 4.1: Image 11 - ISO50%, ISO50% + filtration, DBACD and V2C - $f_c = 50$ upr

that if the virtual contour deforms moving from left to right it will stop before encountering the partially melted defect. Instead, the contour coming from right to left will incorporate it. That is because the iterative process has information on the possible deformations only on the transition zone, while for the rest of the image, the gradient will be null. This implies that when coming from the left, the transition zone will at some point start to give information about the black region that separates the section from the defect, and this will be interpreted as the boundary. On the contrary, when approaching from the right, the white area of the defect will be recognised as the beginning of the material region, ignoring the presence of the black region, hence stopping the deformation there.

The biggest advantage of this method is of course the type of output data. Differently from the ISO50%, **DBACD** does not return a discrete set of points but a parametric curve with its equation. This results in a straight streamline of this method from the data collection and the **filtered and fitted geometry**. With an optimized, robust and automated method, the uncertainties and sources of errors coming from intermediate steps are avoided.

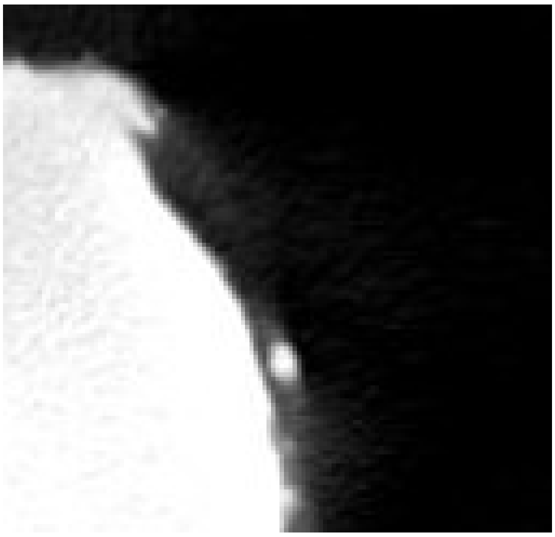


Figure 4.2: Image 11 - defect detail

4.2 Method's applications

The **DBACD** method proves to be highly useful and competitive for evaluating **waviness shape** since the resolution of the b-spline can be linked to the filtration process, with $\lambda_f = 2d$ cut-off wavelength and d representing the starting distance of control points (section 3.6). This implies that using the **DBACD** method for waviness evaluation can eliminate the intermediate steps typically required to obtain final shape information.

In standard procedures, data points collected during the measurement procedure (e.g., ISO50% coming from CT voxels) need to undergo **Gaussian filtering** to separate waviness from roughness information. Surface parameters are then calculated based on these filtered profiles and feature association could be performed. In most cases, a curve-fitting process is required to evaluate these results [Baier et al. [20]]. However, with the **DBACD** method, the filtered waviness profile is obtained as a result of the contour detection process. Moreover, profile information is collected under the b-spline curve equation. This makes the alternative method suitable for various evaluations and measurements pertaining to the **overall shape** of an object, rather than its surface roughness. Examples include evaluations of **circularity**, **cylindricity**, and **coaxiality**, which are particularly relevant in additive manufacturing lattice structure **process control**. Micro surface deviations from the nominal value, even though relevant for mechanical properties, are often attributed to **random errors** and some of those can be fixed by adjusting the process parameters. On the other hand, macro and form errors are typically associated with **systematic errors** and can be mitigated also with compensated geometries. The ability to separate these two types of errors directly, without the need for additional procedures, results in a robust and nearly automated process.

Additionally, thanks to the continuous definition of the identified contour, also **compensation strategies** can also benefit from this method. The displacement of each point from its nominal position can be expressed as the displacement of $2p-1$ control points. Thanks to Equation 2.10 the overall displacement field can be described as a b-spline curve with the same number of **C.P.**. A first coarse compensation strategy for form errors can be formulated in Equation 4.1:

$$C(u)+Comp\Delta C(u) = C(u)-s\times\Delta C(u) = \sum_{i=0}^{n} N_{i,p}(u)P_i - s\times\sum_{i=0}^{n} N_{i,p}(u)D_i \quad a \leq u \leq b \tag{4.1}$$

with s scaling factor. Similarly, the evaluation of the **average displacement** on different sections of a constant profile artefact can also be computed using the property in Equation 2.10. Indeed:

$$\begin{aligned} C(u) + \Delta C(u)_{mean} = AVG(C(u)_j + \Delta C(u)_j) &= C(u) + AVG(\sum_{i=0}^{n} N_{i,p}(u)D_{i,j}) \\ &= C(u) + \sum_{i=0}^{n} N_{i,p}(u)AVG(D_{i,j}) \quad a \leq u \leq b \end{aligned} \tag{4.2}$$

Analogue consideration could be carried out for **V2C**. Everything discussed so far regarding 2D geometries can be readily extended to **3D surfaces**. One significant advantage of b-splines is the ability to stack 2D curves from every section of an object and directly obtain the mathematical definition of the 3D surface (Equation 4.3).

$$\mathbf{C}(u) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,p}(v)\mathbf{P}_{i,j} \quad a \leq u \leq b \quad c \leq v \leq d \tag{4.3}$$

Where $P_{i,j}$ is the $i-th$ control point of the $j-th$ layer.

By incorporating the operations described in ISO 17450-1 2011 [11] (revision of ISO 14660), the DBACD method streamlines the process of obtaining the filtered and associated model. VIC methods with parametric surface definitions can also enhance **reverse engineering** processes, where reconstructing the mathematical definition of an artefact's surface model is required.

The best use for this methods can be summarized in all those applications in which the focus is not on identifying with high accuracy the specific defect, but rather to have an as comprehensive as possible general idea of the geometry of the part under examination. The fact that the final surfaces (in the volumetric analysis) are described as mathematical functions represents a potential benefit in the defect analysis. Taking as an example the reconstruction of a lattice structure artefact, VIC methods open the door to possibly really useful applications thanks to the possibility of extracting information directly from the surface parameters (mode amplitude or control point position) without having to perform specific association operations. When this type of operations are simple (e.g. finding the ovalization of a cylinder to compensate the nominal geometry [20]) standard procedures described in ISO 17450-1 2011 [11] still represent the benchmark strategy. But if we start to think about the idea of compensating the average systematic geometrical defect of inclined struts, whatever its shape is, the analyzed VIC methods could start to be competitive, especially for the fact of obtaining the final output without any intermediate step. The reconstructed surfaces could also be used in the mechanical performance analysis.

4.3 Future Works

First of all, some improvements in the DBACD method could be performed:

- Further study on the code implementation to enhance robustness.
- Insertion of a penalizing score function that accounts when **control points**

strive apart or get too close to each other. This could contribute to increase the process stability at high starting distances.

- implementation of a local contour refinement. By identifying the region with higher errors (e.g. local score function value), the CPPD could be locally increased to allow better detection of smaller defects.
- 3D surface reconstruction of a strut could be performed by piling up several sections contours.

After this, a practical case study could be performed to compare the three methods on the volumetric data in the field of geometry compensation. Furthermore, the correlation between the surface parameters and specific error sources could be searched. In addition, the issue of finding a benchmark reference measuring method could also be solved in future works

Appendix A

DBACD method algorithm

```
%% Discrete B-spline Active Contour Detection method

%Author: Filippo Mioli
%Reference code: Marc-Antoine de Pastre
%Universit  degli Studi di Padova in collaboration
   with ENS PARIS-SACLAY
%THESE LURPA 2023

%All rights reserved
%Any use is strictly forbidden without author's
   approval
%Contact: filippo.mioli@studenti.unipd.it
clear all
close all
clc

im=55; %image name (numeric)
name_s=int2str(im);
folder=['Input\images\']
name=[folder, char(im)]; %image name and folder
   position
```

```
np=80; %number of control points
p=3; %b-spline curve degree
R=1; %images transition zone ratio


sc=1; % number of iteration after smoothness removal
   before looking to divergence

figure (14)
merge=0; % flag in case of merged control points
tic; %starting timer
time=0;
smooth_removed=0;
Score_function=[];
addpath('x_ref');

%% graph visualization modes
showISO=0; % show pixel value histogram and ISO50%
   value
ISOfirst_iter=1; %use autoshape method
oldBS_first_iter=0; %use previous b-spline contour as
   starting contour
oldBS=[]; %previous b-spline contour
oldOffset=[0 0]; %previous b-spline offset
image_initial=0; %display and save initial virtual
   image
show_points_displacement=1; %creates the c.p.
   displacement graph
realM=0;  %realM --> 1 for real Matrix computation, 2
   for matrix computation only on non-zero elements
showM=0; %creates images showing M matrix structure
delta_smooth=0; %use simple smoothing algorithm
```

```
showD=0; %shows the results of implementing smoothness
   in deltaD
startingCircle=1,%startingCircle --> 0 outer, 1 mean, 2
    inner
%% Importation image tomo - define here the image you
   want to process

image = [name_s];
xlim1=0;
xlim2=100;
ylim1=0.06;
ylim2=0.25;
ylim3=-3.5;
ylim4=8;

z=im2double(imread([image,'.jpg']));
S_non_raff=z(:,:,1); %to keep only pixel value
   information

%% Constant initialization

discr_x_im=size(S_non_raff,1);  %image size in x
discr_y_im=size(S_non_raff,2);  %image size in y

taille_pixel = 1;

%image definition reduction
max_pixel_size=200;
refinement_degree = max(discr_x_im/max_pixel_size,
   discr_y_im/max_pixel_size);
refinement_degree = max(refinement_degree,1); %in case
   image is smaller than max available resolution
```

```
xi=[1:refinement_degree:discr_x_im];
yi=[1:refinement_degree:discr_y_im];
discr_x = size(xi,2);
discr_y = size(yi,2);

[Xq,Yq]=meshgrid(yi,xi); %coord pixels
[X,Y]=meshgrid(1:discr_y_im,1:discr_x_im); %coord
   pixels
S_raff=interp2(X,Y,S_non_raff,Xq,Yq);
S_raff(isnan(S_raff))=0; %image with reducted
   definition

n_s=num2str(np);

Rg=Calcul_mean_transition_zone(S_raff,0)*R; %transition
    zone in pixels

Rg_s=int2str(Rg);

%% finding ISO50 value
bins=linspace(0,1,100);
freq=zeros(1,length(bins));

for i=1:length(bins)-1
    freq(i)=length(find(S_non_raff>bins(i) & S_non_raff
       <=bins(i+1)));
end

air=bins(find(freq==max(freq(1:length(freq)/2))));
material=bins(find(freq==max(freq(length(freq)/2:length
   (freq)))));
ISOvalue=(material+air)/2;  %ISO50% treshold value
```

```
if showISO==1

figure (1), hold on
bar(bins,freq)
xline(air)
xline(material)
xline(ISOvalue)
title('\bf Image frequency graph','Interpreter','latex'
   );
exportgraphics(gcf,fullfile(['Output\',name,'_',n_s,'_'
   ,Rg_s,'_ISOvalue.jpg']),'Resolution',300)
close(1)

end

%% finding ISO50 contour

[Lines,Vertices,Objects]=isocontour(S_non_raff,ISOvalue
   ); %home-made iso-50 %is not the real ISO-50%
Contour_tomo=[Vertices(Objects{1},2),Vertices(Objects
   {1},1)];



%% finding center and radius
 if oldBS_first_iter==1 || oldBS_first_iter==2
    centroide=[mean(oldBS(:,2))+oldOffset(1) mean(oldBS
       (:,1))+oldOffset(2)];
    if startingCircle==0
       circle_radius=max(sqrt((centroide(1)-
          Contour_tomo(:,1)).^2+(centroide(2)-
          Contour_tomo(:,2)).^2))/refinement_degree
    elseif startingCircle==1
        circle_radius=mean(sqrt((centroide(1)-
           Contour_tomo(:,1)).^2+(centroide(2)-
           Contour_tomo(:,2)).^2))/refinement_degree
    elseif startingCircle==2
        circle_radius=min(sqrt((centroide(1)-
           Contour_tomo(:,1)).^2+(centroide(2)-
           Contour_tomo(:,2)).^2))/refinement_degree%
           -10
    end
else
    centroide=[mean(Contour_tomo(:,1)) mean(
       Contour_tomo(:,2))];
    if startingCircle==0
       circle_radius=max(sqrt((centroide(1)-
          Contour_tomo(:,1)).^2+(centroide(2)-
          Contour_tomo(:,2)).^2))/refinement_degree
    elseif startingCircle==1
        circle_radius=mean(sqrt((centroide(1)-
           Contour_tomo(:,1)).^2+(centroide(2)-
           Contour_tomo(:,2)).^2))/refinement_degree
    elseif startingCircle==2
        circle_radius=min(sqrt((centroide(1)-
           Contour_tomo(:,1)).^2+(centroide(2)-
           Contour_tomo(:,2)).^2))/refinement_degree%
           -10
    end
end

r=circle_radius/refinement_degree; % radius of the
   starting circle in the raffined image
n_p=np;
```

```
rho=n_p/(2*pi*r);  %Control points perimetral density



%% Circular B-spline generation
[BS,control_points,Nu,U,center,inve,dist]=
   generate_bspline(discr_x,discr_y,circle_radius,p,n_p
   ); %
delta_Dx=zeros(1,size(control_points,2)); %control
   point deviation in x
delta_Dy=delta_Dx;                        %control point
   deviation in y
%% Virtual image generation

offset=centroide/refinement_degree-[discr_y/2 discr_x
   /2];

figure(2); hold on; axis equal
plot((control_points(2,:)+offset(1))*refinement_degree
   ,(control_points(1,:)+offset(2))*refinement_degree,'
   r')  %,(handles.Contour_tomo(:,1)),(handles.
   Contour_tomo(:,2)),'g');
plot (Contour_tomo(:,1),Contour_tomo(:,2),'g')

if ISOfirst_iter==1  %autoshape

    theta_iso=atan2((Contour_tomo(:,2)-centroide(2))/
       refinement_degree,(Contour_tomo(:,1)-centroide
       (1))/refinement_degree);
    theta_cp=atan2(control_points(1,:)+offset(2)-
       centroide(2)/refinement_degree,control_points
       (2,:)+offset(1)-centroide(1)/refinement_degree);
    [theta_cp_sort, index]=sort(theta_cp([1:end-p]));
    theta_cp_sort=[theta_cp_sort pi];
    for i=1:size(theta_cp_sort,2)-1
        ind=find(theta_iso>theta_cp_sort(i) & theta_iso
           <theta_cp_sort(i+1));
        control_points(:,index(i))=[mean(Contour_tomo(
           ind,2));mean(Contour_tomo(ind,1))]/
           refinement_degree;
    end
    control_points(:,[end-p+1:end])=control_points
       (:,[1:p]); %to assure that the first p and last
       p c.p. coincide

    plot((control_points(2,:))*refinement_degree,(
       control_points(1,:))*refinement_degree,'b')
    [BS,center]=b_spline(control_points,Nu);
    offset=[0 0];

end

if oldBS_first_iter==1 %control points taken from old B
   -spline contour

    theta_iso=atan2((oldBS(:,1)-centroide(2))/
       refinement_degree,(oldBS(:,2)-centroide(1))/
       refinement_degree);
    theta_cp=atan2(control_points(1,:)+offset(2)-
       centroide(2)/refinement_degree,control_points
       (2,:)+offset(1)-centroide(1)/refinement_degree);
    [theta_cp_sort, index]=sort(theta_cp([1:end-p]));
    theta_cp_sort=[theta_cp_sort pi];
    for i=1:size(theta_cp_sort,2)-1
        ind=find(theta_iso>theta_cp_sort(i) & theta_iso
          <theta_cp_sort(i+1));
        control_points(:,index(i))=[mean(oldBS(ind,1));
          mean(oldBS(ind,2))]/refinement_degree;
    end
    control_points(:,[end-p+1:end])=control_points
      (:,[1:p]);

    plot((control_points(2,:))*refinement_degree,(
      control_points(1,:))*refinement_degree,'b')  %,(
      handles.Contour_tomo(:,1)),(handles.Contour_tomo
      (:,2)),'g');

    [BS,center]=b_spline(control_points,Nu);
    offset=[0 0];


end

if oldBS_first_iter==2  %control points taken from old
  B-spline control poligon
    oldPol=[];
    control_points=[]
    for i=1:size(oldBS,2)-1-p
        oldPol= [oldPol [linspace(oldBS(1,i)+oldOffset
          (2),oldBS(1,i+1)+oldOffset(2),100) ;
          linspace(oldBS(2,i)+ oldOffset(1) ,oldBS(2,i
          +1)+oldOffset(1),100)]];

    end
    nt=size(oldPol,2);
    gap=fix(nt/n_p);
    for i=1:n_p
```

```
        control_points(:,i)=oldPol(:,(i-1)*gap+1);
    end
    control_points=[control_points control_points(:,[1:
       p])];

    plot((control_points(2,:))*refinement_degree,(
       control_points(1,:))*refinement_degree,'b')  %,(
       handles.Contour_tomo(:,1)),(handles.Contour_tomo
       (:,2)),'g');

    [BS,center]=b_spline(control_points,Nu);
    offset=[0 0];


end

exportgraphics(gcf,fullfile(['Output\',name,'_',n_s,'_'
   ,Rg_s,'_ISO50_vs_Bspline_first_contour.jpg']),'
   Resolution',300)

[S,L,n_p,N,T,center2]=image_virtuelle(discr_x,discr_y,
   Rg,BS,center,inve,offset); %generate virtual image
S_initiale=S;

if image_initial==1

figure (3)
imshow(S);
exportgraphics(gcf,fullfile(['Output\',name,'_',n_s,'_'
   ,Rg_s,'_Virtual_Image_Iter1.jpg']),'Resolution',300)

figure (4)
```

```
imshow(S_raff);
exportgraphics(gcf,fullfile(['Output\',name,'_',n_s,'_'
   ,Rg_s,'_Physical_Image.jpg']),'Resolution',300)

figure (5)
imshow(S_raff-S);
exportgraphics(gcf,fullfile(['Output\',name,'_',n_s,'_'
   ,Rg_s,'_Physical-Virtual_Image_Iter1.jpg']),'
   Resolution',300)

close ([3:5])

end

if show_points_displacement==1
    figure (6), hold on, axis equal
    set(gca, 'YDir','reverse');
    %xlim ([24 34]), ylim ([126 135])
    plot (control_points(2,:),control_points(1,:),'-r',
       'DisplayName','Control Point')
end

%% Itereative process
N_iter=500;
K1=[];
Visu=[];
dyna_image=max(max(S_raff))-min(min(S_raff));
ecart=zeros(length(N_iter),1); %score function values

residu=sqrt(mean(mean((S-S_raff).^2)))/dyna_image;

Mx=zeros(size(Nu,1));
```

```
My=zeros(size(Nu,1));

merge=zeros(N_iter+2,1);
for k=1:N_iter

    % gradient calculation
    [Gy,Gx] = imgradientxy(S);
    G=sqrt(Gx.^2+Gy.^2);

    Gx2=reshape(Gx,[discr_x*discr_y,1]);   %Gx matrix
       transformed in vector
    Gy2=reshape(Gy,[discr_x*discr_y,1]);   %Gy matrix
       transformed in vector
    G2=Gx2+Gy2;

    PHI_diff=reshape(S-S_raff,[discr_x*discr_y,1]);
    PHI_diff(isnan(PHI_diff))=0;

    % resizing vectors after merging or adding control
       points
    if merge(k+1)==1
        basis_fun_matrix=zeros(size(Nu,1),size(S,1),
           size(S,2));
        basis_fun_vectors=zeros(discr_x*discr_y,size(
           basis_fun_matrix,1));
        B=zeros(1,size(basis_fun_matrix,1));
        Mx=zeros(size(basis_fun_matrix,1),size(
           basis_fun_matrix,1));
        My=zeros(size(basis_fun_matrix,1),size(
           basis_fun_matrix,1));
        M=zeros(size(basis_fun_matrix,1),2*size(
           basis_fun_matrix,1));
        delta_D=zeros(size(M,2),1);
        delta_Dx=zeros(1,size(M,2)/2);
        delta_Dy=zeros(1,size(M,2)/2);
    end
% to assign Nu values to each image point as the Nu
   value of the closest b-spline point
for i=1:size(S,1)  %size of S in X
    for j=1:size(S,2) %size of S in Y
        S_dis=sqrt((i-BS(:,1)).^2-(j-BS(:,2)).^2);
        ind_closest_point=find(S_dis==min(S_dis),1);
        basis_fun_matrix(:,i,j)=Nu(:,ind_closest_point)
           ;
    end
end

%B vector generation
for i=1:size(basis_fun_matrix,1)
    basis_fun_vectors(:,i)=reshape(basis_fun_matrix(i
       ,:,:),[discr_x*discr_y,1]);
    B(i)=sum(PHI_diff.*G2.*basis_fun_vectors(:,i));
end
B=B';



G2GX2=G2.*Gx2;
G2GY2=G2.*Gy2;

if realM==1
%% real M matrix
for i=1:size(basis_fun_matrix,1)
    for j=1:size(basis_fun_matrix,1)
        Mx(i,j)=sum(G2GX2.*basis_fun_vectors(:,i).*
           basis_fun_vectors(:,j));
        My(i,j)=sum(G2GY2.*basis_fun_vectors(:,i).*
           basis_fun_vectors(:,j));
        M(i,2*j-1)=Mx(i,j);
        M(i,2*j)=My(i,j);
    end
end

%% real M matrix without zeros
elseif realM==2
l=size(basis_fun_matrix,1);
for i=1:l
    bfv_sq=basis_fun_vectors(:,i).*basis_fun_vectors(:,
       i);
    Mx(i,i)=sum(G2GX2.*bfv_sq);
    My(i,i)=sum(G2GY2.*bfv_sq);
    M(i,2*i-1)=Mx(i,i);
    M(i,2*i)=My(i,i);
    for j=1:p-1
        jj=min(i+j,l);
        bfv_sq=basis_fun_vectors(:,i).*
           basis_fun_vectors(:,jj);
        Mx(i,jj)=sum(G2GX2.*bfv_sq);
        Mx(jj,i)=Mx(i,jj);
        My(i,jj)=sum(G2GY2.*bfv_sq);
        My(jj,i)=My(i,jj);
        M(i,2*jj-1)=Mx(i,jj);
        M(i,2*jj)=My(i,jj);
        M(jj,2*i-1)=Mx(jj,i);
        M(jj,2*i)=My(jj,i);
    end
```

```
end
end
%%
Minv=invertNonSquareMatrix(M);

if showM==1
figure (20)
imshow (M)
figure (21)
imshow (Minv)
figure (24)
imshow (abs(M))
figure (25)
imshow (abs(Minv))
end

delta_D=Minv*B;  %control points displacement values

B=B';
for j=1:length(delta_D)/2
    delta_Dx(j)=delta_D(2*j-1);
    delta_Dy(j)=delta_D(2*j);
end

% imposing same displacement for overlapping points
delta_Dx(end-p+1:end)=(delta_Dx(1:p)+delta_Dx(end-p+1:
   end))/2;
delta_Dx(1:p)=delta_Dx(end-p+1:end);
delta_Dy(end-p+1:end)=(delta_Dy(1:p)+delta_Dy(end-p+1:
   end))/2;
delta_Dy(1:p)=delta_Dy(end-p+1:end);
```

```
deltaD=sqrt(delta_Dx.^2+delta_Dy.^2);
averageD(k)=mean(deltaD);

%% plotting D
if showD==1
close (14)
figure (14), hold on
subplot (3,1,1)
ylabel (["Control points deviation [px]"])
plot(deltaD)
%yline (averageD)
subplot (3,1,2), hold on
ylabel (["Control points deviationin x [px]"])
plot(delta_Dx)
subplot (3,1,3), hold on
ylabel (["Control points deviation in y [px]"])
plot(delta_Dy)
end
%% smoothing D
if delta_smooth==1

for i=2:length(delta_Dx)-1
    delta_Dx(i)=delta_Dx(i-1)/4+delta_Dx(i)/2+delta_Dx(
       i+1)/4;
    delta_Dy(i)=delta_Dy(i-1)/4+delta_Dy(i)/2+delta_Dy(
       i+1)/4;
end
if showD==1
    subplot (3,1,2)

    plot(delta_Dx)
    subplot (3,1,3)
    plot(delta_Dy)
end
delta_Dy(1)=delta_Dy(end-p+1);
delta_Dx(1)=delta_Dx(end-p+1);
delta_Dx(end)=delta_Dx(p);
delta_Dy(end)=delta_Dy(p);



end

%% plotting control points displacement

[control_points_new,Nu_new,merge(k+2),U_new]=
   control_points_displacement(delta_Dx,delta_Dy,
   control_points,p,dist,U,Nu); %displacing control
   points

if show_points_displacement==1
    figure (6)

    for q=1:size(control_points,2)
        plot ([control_points(2,q) control_points(2,q)+
           delta_Dy(q)*1],[control_points(1,q)
           control_points(1,q)+delta_Dx(q)*1],'-b','
           DisplayName','Control Point')
    end

end

[BS_new,center_new]=b_spline(control_points_new,Nu_new)
   ;
```

```
[S_new,L,n_p,N,T,S_center]=image_virtuelle(discr_x,
   discr_y,Rg,BS_new,center_new,inve,offset);

% saving score function values
    Visu=1;
    K1=[K1,k];
    if smooth_removed>0
         ecart(k)=residu/1.5;
         Score_function(k)=residu/1.5;
         smooth_removed=smooth_removed-1;
    else
        ecart(k)=residu;
        Score_function(k)=residu;
    end




    %% Stop criteria

    residu_new=sqrt(mean(mean((S_new-S_raff).^2)))/
       dyna_image
    norm_delta=0;
    for l=1:length(delta_D)
        norm_delta=norm_delta+delta_D(l)^2;
    end
    norm_delta=sqrt(norm_delta);


    %small residual
    if abs(residu_new-residu)<1e-5
        if delta_smooth==1
            delta_smooth=0;
            fprintf('stop for convergence. Curve
               smoothing removed');
            smooth_removed=sc;
        else
            figure(7);
            imshow(S);
            title({'\bf Correlated virtual volume:';['\
               bf iteration n$^{\circ}$',num2str(k)]},
               'Fontsize',16,'interpreter','latex');
            exportgraphics(gcf,fullfile(['Output\',name
               ,'_',n_s,'_',Rg_s,'_iteration',num2str(k
               ),'.jpg']),'Resolution',300)

            Score_function(k+1)=residu_new; %extracting
                Least square deviation
            S=S_new;
            BS=BS_new;
            control_points=control_points_new;
            Nu=Nu_new;
            centre=center_new;
            U=U_new;
            time=toc;
            close (7)
            fprintf('stop for convergence');
            break
        end
    %% case of divergence
    elseif (residu_new-residu)>0
```

```
if delta_smooth==1
    delta_smooth=0;
    fprintf('stop for divergence. Curve
       smoothing removed');
    smooth_removed=sc;
else
    loop=1;
    cont=0;

    while loop==1
        %% checking if there is any loop in the
            polygon
        int=0;
        loop=0;
        s=100; %number of points per segment
        toll=(dist/3)/s;
        clear segments
        segments(1,:)=linspace(control_points
           (1,1),control_points(1,2),s);
        segments(2,:)=linspace(control_points
           (2,1),control_points(2,2),s);
        for i=2:size(control_points,2)-p-1
            segments(1,1:size(segments,2)+s)=[
               segments(1,1:end) linspace(
               control_points(1,i),
               control_points(1,i+1),s)];
            segments(2,1:size(segments,2))=[
               segments(2,1:end-s) linspace(
               control_points(2,i),
               control_points(2,i+1),s)];
    end
    figure (8), hold on
```

```
plot(segments(2,:),segments(1,:),'.r')
j=1;
for i=1:s
        int=0;
        int=find(sqrt((segments(1,(j-1)*s+i
           )-segments(1,(j+1)*s:end-s)).^2
           + (segments(2,(j-1)*s+i)-
           segments(2,(j+1)*s:end-s)).^2)
           <10*toll);
        if int ~= 0
            loop=1;
            cont=cont+1;
            merge(k+2)=1;
            fprintf(['there is a loop in
               index ',num2str((j-1)*s+i),'
                and ',num2str((j+1)*s+int
               -1)]);

            l=fix((int(1)-1)/s);
            first_loop_point=j;
            second_loop_point=j+3+l;

            plot(segments(2,(j-1)*s+i),
               segments(1,(j-1)*s+i),'ob')
            plot(control_points(2,[
               first_loop_point
               second_loop_point]),
               control_points(1,[
               first_loop_point
               second_loop_point]),'om')

            %% erasing loop
            [control_points,U,Nu]=
               loop_erased(first_loop_point
               ,second_loop_point,
               control_points,U,p);
            plot(control_points(2,:),
               control_points(1,:),'-k')
            break


        end


end
if int == 0

for j=2:size(control_points,2)-p-2
    for i=1:s
        int=0;
        int=find(sqrt((segments(1,(j-1)*s+i
           )-segments(1,(j+1)*s:end)).^2 +
           (segments(2,(j-1)*s+i)-segments
           (2,(j+1)*s:end)).^2)<10*toll);

        if int ~= 0
            loop=1;
            cont=cont+1;
            merge(k+2)=1;
            fprintf(['there is a loop in
               index ',num2str((j-1)*s+i),'
                and ',num2str((j+1)*s+int
               -1)]);
            l=fix((int(1)-1)/s);
```

```
                first_loop_point=j;
                second_loop_point=j+3+1;

                plot(segments(2,(j-1)*s+i),
                   segments(1,(j-1)*s+i),'ob')
                plot(control_points(2,[
                   first_loop_point
                   second_loop_point]),
                   control_points(1,[
                   first_loop_point
                   second_loop_point]),'om')

                %% erasing loop
                [control_points,U,Nu]=
                   loop_erased(first_loop_point
                   ,second_loop_point,
                   control_points,U,p);
                plot(control_points(2,:),
                   control_points(1,:),'-k')
                break
            end
        end
        if int ~= 0
            clear segments
            close (8)
            break
        end

    end
    end

end
```

```
    if cont==0

        exportgraphics(gcf,fullfile(['Output\',name
           ,'_',n_s,'_',Rg_s,'_Loop_Control.jpg']),
           'Resolution',300)
        close (8)

        figure(7);
        imshow(S);
        title({'\bf Correlated virtual volume:';['\
           bf iteration n$^{\circ}$',num2str(k)]},
           'Fontsize',16,'interpreter','latex');
        exportgraphics(gcf,fullfile(['Output\',name
           ,'_',n_s,'_',Rg_s,'_iteration',num2str(k
           ),'.jpg']),'Resolution',300)
        time=toc;

        Score_function(k+1)=residu; %extracting
           Least square deviation
        close (7)
        fprintf('stop for divergence');
        break
    end

end

%% very smaal displacement
elseif norm_delta<0.01

    figure(7);
    imshow(S);
```

```
        title({'\bf Correlated virtual volume:';['\bf
           iteration n$^{\circ}$',num2str(k)]}, '
           Fontsize',16,'interpreter','latex');
        exportgraphics(gcf,fullfile(['Output\',name,'_'
           ,n_s,'_',Rg_s,'_iteration',num2str(k),'.jpg'
           ]),'Resolution',300)

        time=toc;

        Score_function(k+1)=residu_new; %extracting
           Least square deviation
        S=S_new;
        BS=BS_new;
        control_points=control_points_new;
        Nu=Nu_new;
        centre=center_new;
        U=U_new;
        close (7)
        fprintf('stop for to small displacement');
        break
    else
        S=S_new;
        BS=BS_new;
        control_points=control_points_new;
        Nu=Nu_new;
        centre=center_new;
        U=U_new;
    end

    if smooth_removed>0
        residu=residu_new*1.5;
    else
```

```
            residu=residu_new;
        end




%% final plots

if show_points_displacement==1
    figure (6)
    plot (control_points(2,:),control_points(1,:),'-r',
       'DisplayName','Control Point')
    exportgraphics(gcf,fullfile(['Output\',name,'_',n_s
       ,'_',Rg_s,'_Control_points_displacement.jpg']),'
       Resolution',300)
    exportgraphics(gcf,fullfile(['Output\',name,'_',n_s
       ,'_',Rg_s,'_Control_points_displacement.pdf']),'
       Resolution',300)
    close (6)
end

figure (9), hold on
legend show
plot ([1:k],averageD,'DisplayName',n_s)
xlabel('Iterations');
ylabel('Average points displacement');
exportgraphics(gcf,fullfile(['Output\',name,'_',n_s,'_'
   ,Rg_s,' p average displacement.jpg']),'Resolution'
   ,300)

figure (11), hold on
legend show
plot ([1:k+1],[ecart residu_new],'DisplayName',n_s)
```

```
xlabel('Iterations');
ylabel('Residual');
exportgraphics(gcf,fullfile(['Output\',name,'_',n_s,'_'
   ,Rg_s,' p residu.jpg']),'Resolution',300)
%close (9)

figure (12), hold on
legend show
xlabel('Iterations');
yyaxis left
plot ([1:k+1],[ecart residu_new],'DisplayName',n_s)
ylabel('Residual');
yyaxis right
plot ([1:k],averageD,'DisplayName',n_s)
ylabel('Average Displacement');

exportgraphics(gcf,fullfile(['Output\',name,'_',n_s,'_'
   ,Rg_s,' p residual and displacement.jpg']),'
   Resolution',300)
close (12)

    %% offsetting BS
    BS(:,2)=BS(:,2)+offset(1);
    BS(:,1)=BS(:,1)+offset(2);
    control_points(1,:)=control_points(1,:)+offset(2);
    control_points(2,:)=control_points(2,:)+offset(1);
    offset=[0 0];
    %% Computing deviation between ISO50% and V2C

    [ecart_moy_iso50,ecart_max_iso50,ecart_RMS_ISO50]=
       comparaison_contour(BS(:,[1 2])*
       refinement_degree,Contour_tomo,taille_pixel,
       discr_x_im,discr_y_im,discr_x,discr_y);
    figure (10), hold on
    plot((BS(:,2)+offset(1))*refinement_degree,(BS(:,1)
       +offset(2))*refinement_degree,'r')  %,(handles.
       Contour_tomo(:,1)),(handles.Contour_tomo(:,2)),'
       g');
    plot (Contour_tomo(:,1),Contour_tomo(:,2),'g')
    legend('Virtual contour','ISO$_{50\%}$','
       Interpreter','latex');
    set(gca, 'YDir','reverse');
    axis equal
    title('\bf Associated contours','Interpreter','
       latex');

    exportgraphics(gcf,fullfile(['Output\',name,'_',n_s
       ,'_',Rg_s,'_ISO50 vs. BS.pdf']),'Resolution'
       ,300)
    exportgraphics(gcf,fullfile(['Output\',name,'_',n_s
       ,'_',Rg_s,'_ISO50 vs. BS.jpg']),'Resolution'
       ,300)
    close (10)
    close (2)



    BS_fullsize=BS*refinement_degree;


    n_f=size(control_points,2);
figure (13)
imshow (abs(S-S_raff))
```

Appendix B

Invert non square matrix

```
function inverted_matrix = invertNonSquareMatrix(matrix)
    [m, n] = size(matrix);

    if m == n
        fprintf('The matrix is square and can be inverted using "inv".\n');
        inverted_matrix = inv(matrix);
        return;
    end

    if rank(matrix) < min(m, n)
        fprintf('The matrix has no full rank and cannot be inverted.\n');
        inverted_matrix = pinv(matrix);
        return;
    end

    lambda = 0.1; % parametro di regolarizzazione
    inverted_matrix = (matrix' * matrix + lambda * eye(n)) \ matrix';
```

```
end

% To not use the pseudo-inverse, the inversion of a non
    -square matrix is not always possible, as it
    requires the matrix to have maximum rank. However,
    there are approximate methods that can be used to
    approximate the inverse of a non-square matrix.
% One such method is the truncated Moore-Penrose
    inversion (Tikhonov regularisation), which involves
    introducing a regularisation term to handle non-
    invertible matrices.
```

Appendix C

Roundnes error

```
function [R1, R2, Xc, Yc, Err]=roundness(x, y, lsd_xc,
   lsd_yc)
a=50;
xc=[lsd_xc-a : lsd_xc+a];
yc=[lsd_yc-a : lsd_yc+a];
for i=1:size(xc,2)
    for j=1:sizeyesc,2)
        r1(i,j)=max(sqrt((x-xc(i)).^2+(y-yc(j)).^2));
        r2(i,j)=min(sqrt((x-xc(i)).^2+(y-yc(j)).^2));
        err(i,j)=r1(i,j)-r2(i,j);
    end
end

[indx, indy]=find(err==min(min(err)));

R1=r1(indx, indy);
R2=r2(indx, indy);
Xc=xc(indx);
Yc=yc(indy);
Err=err(indx, indy);
```

Appendix D

Sintesi estesa in lingua italiana

D.1 Introduzione

Dagli anni '80, le tecnologie di **Manifattura Additiva** [1] hanno suscitato un crescente interesse nell'industria. Questo, come conseguenza della **scadenza** dei primi brevetti, ha portato negli ultimi dieci anni ad un aumento esponenziale delle ricerche accademiche su questo argomento. Grasso et al. [2] in una revisione sui processi di fusione a letto di polvere (Powder Bed Fusion - PBF) mostrano come il numero di studi all'anno su questo specifico argomento sia cresciuto da meno di **cinque** nel 2010 a quasi **cinquanta** nel 2020. Inizialmente introdotta per il **prototipaggio rapido**, oggi la Manifattura Additiva sta sempre più affermandosi come un processo di produzione adeguato per un tipo specifico di prodotto. Infatti, grazie all'aumento della **qualità finale** dei prodotti AM, queste tecnologie sono sempre più utilizzate nella produzione di manufatti con **elevata complessità geometrica** e lotti di produzione ridotti, se non nulli [1]. Il maggior vantaggio dell'AM oggi è che il costo di fabbricazione per lo stesso sistema di produzione è direttamente correlato al volume di materiale, ma quasi svincolato dalla sua complessità. Le **strutture reticolari** (LS) sono una chiara conseguenza di questa predisposizione. Questo tipo di geometrie, un tempo chiamate **materiali porosi** o **strutture cellulari**, si ispirano a materiali leggeri ad alta densità di prestazioni presenti in natura, come ossa, legno e spugne [3]. Le LS sono composte da una geometria a cella elementare ripetuta più volte per riempire il volume della parte. Ricordando che le **esigenze metrologiche** sono conseguenza delle **condizioni funzionali**, sarà immediato comprendere che tali differenze radicali tra i processi tradizionali (sottrattivi e di fusione) e le parti AM implicano anche l'utilizzo di **strumenti di misura e relative normative** diversi [4]. La necessità, ad esempio, di misurare forme interne complesse, non accessibili con strumenti di misura a contatto, ha portato all'adozione della **Tomografia Computerizzata** (CT) come il principale sistema di misurazione per i manufatti AM [5]. È noto che i difetti che si verificano durante il processo di produzione possono influenzare significativamente la **durata e le proprietà meccaniche** dei manufatti, e questo si applica anche alle parti prodotte con tecnologia additiva. Malekipour e El-Mounayri hanno classificato i difetti in LPBF in quattro categorie [6]. De Pastre et al. [7] hanno eseguito una revisione approfondita delle influenze dei parametri di processo sui difetti e le proprietà delle parti. Zhao et al. [8] hanno scoperto che le **cricche da fatica** nella lega AlSi12Mg SLM hanno origine da pori gassosi in superficie o immediatamente sotto la superficie, e la vita a fatica dipende principalmente dalla **dimensione dei pori** e dalla **direzione di costruzione**, più che dalla frazione di porosità. Molte ricerche si sono concentrate sull'**identificazione dei difetti di porosità** nel processo AM [8][9]. Anche i difetti geometrici sono legati alla funzionalità della parte, specialmente in fase di assemblaggio e in condizioni operative.

Nella metrologia avanzata, le misurazioni vengono effettuate catturando grandi quantità di dati (nuvole di punti, immagini, voxel, ecc.) e quindi estraendo informazioni da essi. Uno dei passaggi chiave che consente di passare da dati illeggibili a risultati di misurazione è l'**identificazione del contorno**. In altre parole, è il processo che ci consente di distinguere chiaramente il confine tra la parte da misurare (materiale) e ciò che la circonda (aria o altro materiale).

Nella maggior parte delle misurazioni, vengono raccolti **insiemi di punti** dalla superficie e attualmente esistono diversi metodi consolidati per allineare le forme geometriche nominali ai dati raccolti. Con l'avvento della **Tomografia Computerizzata** e delle **CMM Ottiche** dotate di sensori di immagine, il compito non è più solo quello di adattare geometrie nominali ai punti, ma anche di identificare i valori di transizione tra la parte e ciò che la circonda. Prendendo ad esempio la TC, i risultati della procedura di misurazione sono un insieme di **Voxel** (pixel volumetrici), il cui valore tra zero e uno indica la resistenza che il materiale in quel punto ha opposto ai **raggi X** che lo hanno attraversato. Analizzando lo spettro dei valori del voxel, troveremo due picchi principali corrispondenti ai due diversi materiali (ad esempio, metallo e aria) e un insieme di valori di transizione nel mezzo. La metodologia **ISO50%**, ad esempio, imposta come valore di **soglia** tra i due materiali quello a metà tra i due picchi. Altri approcci sono presenti in letteratura. L'output dei dati, ottenuto dopo l'applicazione di questi metodi di soglia, è un insieme di punti discreti. Lifton e Liu hanno proposto un metodo di **valutazione dell'incertezza** per ISO50%. Hanno evidenziato come l'incertezza di misura dovuta al metodo di determinazione della superficie ISO50% sia maggiore per dimensioni che sono **sensibili** alla determinazione della superficie, come raggi interni, raggi esterni e spessori delle pareti, e **minore** per dimensioni insensibili alla soglia, come distanze centro-centro. Inoltre, le informazioni sulla nuvola di punti discreti richiedono manipolazioni successive per estrarre risultati di misurazione come per le nuvole di punti tradizionali. Questi passaggi, descritti in ISO 17450-1:2011, includono **filtraggio** e **associazione**.

In questa ricerca è stata esaminata la performance di due promettenti metodi **VIC** nell'identificazione dei difetti di forma su immagini provenienti da **strutture reticolari fabbricate mediante manifattura additiva**. La **Virtual Image Correlation** (VIC) è un processo di rilevamento dei contorni basato sulla creazione di un **volume virtuale** e sulla sua deformazione attraverso la minimizzazione di una **funzione obbiettivo** al fine di rappresentare al meglio il **volume reale** (fisico). I risultati ottenuti dai metodi **VIC** forniscono informazioni analitiche sulla geometria dei difetti della parte. Il primo metodo (**V2C**) sviluppato da De Pastre et al. si basa su una correlazione del Volume Virtuale basata sulla **decomposizione modale** della forma fisica. Il secondo metodo, che è l'obiettivo principale di questa ricerca ed è stato sviluppato specificamente per essa, utilizza una curva parametrica chiusa (**B-spline**) per definire l'immagine virtuale, che rappresenta il contorno della sezione. In particolare, il test è stato eseguito su immagini ottenute sezionando scansioni da **CT** di strutture reticolari AM. Poiché la forma nominale di queste sezioni dovrebbe essere un cerchio perfetto (sezione di cilindri presi con piani perpendicolari al loro asse), la B-spline è inizialmente definita da un poligono regolare chiuso di **punti di controllo** (C.P.). Questi punti vengono quindi **mossi in modo iterativo** per deformare il contorno della B-spline in modo da minimizzare la finzione obbiettivo Φ.

Nella letteratura sono stati condotti diversi studi nel campo del VIC. Réthoré et al. hanno sviluppato un metodo di ”B-spline Virtual Image Correlation” per rilevare e misurare contorni e silhouette da immagini digitali applicando successivamente set di spostamenti **normali** alla forma iniziale. Jiang et al. hanno utilizzato un metodo simile per identificare forme in immagini **mediche** (sia in 2D che in 3D). Poiché lo spostamento è sempre cercato nella direzione **normale** alla curva iniziale, **forme complesse** dei difetti non possono essere analizzate e rilevate (sotto-squadro). Inoltre, è stata testata l'influenza solo di una parte dei parametri di input. Incertezza inferiore alla dimensione del pixel è stata valutata nella rilevazione di forme semplici con bassa presenza e densità di piccoli difetti locali. Poiché solo pochi metodi consolidati sono stati sviluppati e testati nel campo della manifattura additiva e, in particolare, delle strutture reticolari, il cui tipo di immagini è caratterizzato da un'alta densità di difetti superficiali, finora **non è stata effettuata alcuna valutazione dell'incertezza** per l'analisi ad alta risoluzione. Inoltre, i difetti spesso hanno forme molto complesse (Figure 1.1), non sempre rilevabili da modelli semplici (ad esempio, spostamento normale). L'incertezza dei VIC è stata valutata da François evidenziando la sensibilità dei metodi generali di Virtual Image Correlation al **rumore** dell'immagine, alla **discretizzazione** e alla **sfocatura**.

D.1.1 Obiettivo del lavoro

L'obiettivo di questa ricerca, oltre allo sviluppo di un algoritmo di **Discrete B-spline Active Contour Deformation** (D.B.A.C.D), è studiare come i metodi VIC si comportano nell'analisi delle forme provenienti da sezioni di **Strutture Reticolari**. In questo modo, si spera di fornire una **prospettiva** su come questi metodi possano essere utilizzati all'interno del **contesto degli standard ISO/GPS**. In particolare, dopo aver sviluppato il metodo VIC basato su B-spline (chapter 2), verrà eseguita una calibrazione e una valutazione delle prestazioni (chapter 3).

Nella sezione subsection 3.7.1 i metodi **DBACD** e **V2C** verranno confrontati sia visivamente che numericamente attraverso l'identificazione di determinate misure. Infine, saranno proposte prospettive future di applicazione (section 4.2) e ulteriori sviluppi (section D.7).

D.2 Formulazioni

Una curva generica di tipo b-spine può essere espressa dall'equazione:

$$\mathbf{C}^{(k)}(u) = \sum_{i=0}^{n} N_{i,p}^{(k)}(u)\mathbf{P}_i \quad a \leq u \leq b \tag{D.1}$$

dove P_i è l'i-esimo **punto di controllo** e $N_{i,p}$ la i-esima **funzione di base** di grado p definita:

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{se } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{aligned} \tag{D.2}$$

Il contorno da deformare al fine di identificare la sezione proveniente da CT viene definito da un poligono regolare composto da **n** punti di controllo. Questi punti vengono deformati iterativamente al fine di ottenere una forma che meglio corrisponda con la sezione reale. Al fine di ottenere ciò, un immagine virtuale viene generata a partire dalla curva parametrica associando ad i pixel contenuti al suo interno valore massimo (1 che corrisponde al colore bianco) e minimo a quelli al suo esterno (0 ovvero nero). Ai pixel lungo la curva, vengono assegnati valori intermedi per creare una zona di transizione. La funzione obbiettivo Φ viene definita come la somma dei quadrati della differenza di valore di ogni pixel tra l'immagine fisica e quella virtuale:

$$\Phi = \iint_{ROI} [f(\vec{x}) - g(\vec{x} - \vec{d})]^2 d\Omega \tag{D.3}$$

Linearizzando la funzione obbiettivo e andando a ricercarne il valore di minimo al variare della posizione dei punti di controllo, si ottiene un sistema lineare del tipo:

$$\underset{2n\times 1}{[B]} = \underset{2n\times 2n}{[M]} \underset{2n\times 1}{[\delta D_i]} \tag{D.4}$$

La cui incognita è il vettore $[\delta D_i]$ il cui elemento i-esimo corrisponde allo spostamento che l'i-esimo punto di controllo dovrebbe effettuare per minimizzare Φ. Essendo il sistema lineare basato sulla linearizzazione della finzione obbiettivo, la soluzione viene ricercata attraverso un processo iterativo in cui i punti vengono deformati da un susseguirsi di spostamenti fino a raggiungere una convergenza del valore di Φ.

D.3 Calibrazione del metodo e valutazione delle prestazioni

In questo capitolo, il metodo DBACD sarà testato per valutare come diversi parametri di input influenzino le prestazioni di rilevamento. L'obiettivo principale è ottenere un insieme di valori ottimali che garantiscano i migliori risultati senza dover impostare i parametri di input tramite l'esperienza dell'utente.

D.3.1 Parametri influenti

Diversi parametri di input potrebbero influenzare il risultato dell'identificazione del contorno:

1. Grado della B-spline.
2. Zona di transizione dell'immagine virtuale.
3. Contrasto dell'immagine.
4. Implementazione del codice e soluzione del sistema lineare.
5. Densità iniziale dei punti di controllo.
6. Distanza iniziale della B-spline dal contorno finale.

I punti 1) e 2) sono già stati discussi da altre ricerche. Rethore e Francois [14] hanno scoperto che il valore ottimale per la **zona di transizione** dell'immagine dovrebbe portare a un rapporto di transizione $R_g = r_v/r_f = 1$ dove r_v è la zona di transizione dell'immagine virtuale e r_f è la zona di transizione dell'immagine fisica. Per la maggior parte dei test effettuati, hanno utilizzato una B-spline di grado $p = 3$, anche se non è stata riscontrata una grande influenza di questo parametro sui risultati finali.

D.3.2 Selezione della migliore strategia

Un primo esperimento è stato condotto per valutare l'influenza dei punti 4), 5) e 6) nei **risultati finali**. L'esperimento è stato progettato come una serie di combinazioni di parametri. Sono state testate quattro diverse **strategie di algoritmo**, ognuna con un diverso **numero iniziale di punti di controllo**. La **deviazione Media**, quella **Massima**, e la **Deviazione RMS** dal contorno ISO50% sono stati calcolati e salvati insieme al valore finale della **funzione obiettivo** e al **Tempo di Calcolo**. L'algoritmo di identificazione del contorno è stato testato con quattro **immagini diverse** provenienti da diverse sezioni di diversi elementi reticolari. Sono state scelte per rappresentare bene la **variabilità delle condizioni** che potrebbero verificarsi.

Ecco elencati i diversi algoritmi:

- **noSmooth** è l'algoritmo sviluppato sul modello matematico descritto in precedenza. La matrice $[M]$ viene valutata in ogni punto, risultando in tempi computazionali molto elevati.

- **noRealM** utilizza una matrice $[M]$ semplificata con solo gli elementi vicini alla diagonale per ridurre il tempo computazionale. Per ogni elemento $m_{i,j}$, viene effettuato il prodotto scalare tra $N_{i,p}$ e $N_{j,p}$. Dalla definizione della funzione di base, possiamo affermare che ogni $N_{i,p}$ è diverso da zero solo all'interno dell'intervallo $[U_{i-p+1}, U_{i+p-1}]$; ciò significa che l'equazione $[N_{i,p}] \cdot [N_{j,p}] \neq 0$ è vera solo quando $|i - j| < p - 1$. Il risultato è una matrice simmetrica i cui elementi diversi da zero sono tutti concentrati su una banda di larghezza $2(p-1)+1$. Per accelerare la procedura computazionale, vengono calcolati solo gli elementi **non nulli**.

- **Smooth** implementa una funzione di media per lo spostamento dei punti di controllo al fine di ottenere una deformazione più uniforme nelle fasi iniziali. Per ogni deformazione dei punti viene applicata la seguente regola: $\Delta Dx_i = 1/4 \Delta Dx_{i-1} + 1/2 \Delta Dx_i + 1/4 \Delta Dx_{i+1}$ La stessa regola viene quindi applicata a ΔDy_i. In Figure 3.1 vengono mostrati gli spostamenti dei punti di controllo direttamente dalla soluzione del sistema lineare (blu) e dopo il processo di smoothing (rosso).

- **Autoshape** implementa un algoritmo per eseguire un primo movimento approssimativo dei punti di controllo molto vicino alla forma fisica. Proietta ogni punto di controllo in **direzione radiale** sul contorno **ISO50%**. Il risultato ottenuto è un poligono di controllo iniziale adattato al contorno **ISO50%**. Questa soluzione, sebbene non sia la più rigorosa applicabile, risulta essere **molto veloce** da calcolare e **sufficientemente efficace**.

Nei paragrafi seguenti il numero di **punti di controllo** è meglio espresso come **densità perimetrale di punti di controllo** ($\text{CPPD} = \frac{\text{n° di punti di controllo}}{\text{perimetro della circonferenza iniziale}}$).

Il risultato dei valori ottimali di CPPD per le quattro diverse strategie è riassunto nella Table 3.5.

D.3.3 Posizione della circonferenza di partenza

Successivamente un secondo test ha evidenziato come le indagini performate a partire da circonferenze più vicine al contorno finale diano risultati migliori. In altre parole, l'aumento di distanza media dei punti dalla configurazione iniziale a quella finale influisce negativamente sulle prestazioni del metodo (section 3.3).

D.3.4 Configurazione ottimale dei parametri di input - Taguchi method

Successivamente attraverso il metodo Taguchi per il design di esperimenti si è ricercata la configurazione ottima di input per i seguenti parametri: "CPPD, contrasto dell'immagine fisica, grado della curva parametrica e rapporto delle zone di transizione" su tre diversi livelli:"basso, medio e alto". Le configurazioni sono riportate in Table 3.7 e Table 3.8. L'esperimento è stato condotto sulle due strategie più promettenti ovvero **autoshape** e **Smooth+noRealM**. In Table 3.9 e Table 3.10 si può vedere l'influenza relativa dei diversi parametri sulla deviazione media finale misurata tra il contorno b-spline e quello ottenuto con ISO50%. Nella Table 3.11 e nella Table 3.12, le deviazioni sono espresse in valori assoluti. In questo modo è più facile apprezzare il reale contributo dell'errore di misura rispetto alla scelta dei parametri. **Smooth + noRealM** sembra essere meno sensibile al **rapporto delle zone di transizione** ma altamente influenzato dalla **CPPD** e dal **contrasto dell'immagine**. Dall'altra parte, **Autoshape** mostra una maggiore sensibilità al **rapporto delle zone di transizione**, specialmente con valori elevati di R_g. Ciò è attribuibile al fatto di avere un processo di deformazione che inizia già con il contorno finale all'interno della zona di transizione. Inoltre, anche le influenze del **livello di contrasto** sembrano essere in opposizione con il metodo **Smooth + noRealM**. Anche questo comportamento è da aspettarsi data la natura delle due strategie.

Quando si effettua questo tipo di valutazione, è importante ricordare che il metodo di riferimento è anch'esso sensibile al contrasto dell'immagine. Per ottenere una risposta adeguata sulla sensibilità di questo metodo VIC al contrasto dell'immagine, il valore della deviazione media dovrebbe essere calcolato rispetto a un contorno di riferimento noto con una minore incertezza dovuta al contrasto. Inoltre, è difficile interpretare il diverso grafico per il **grado della curva**. Anche se in entrambi i metodi la variazione di p genera piccoli errori, il fatto che $p = 3$ corrisponda a un minimo per **Autoshape** e a un massimo per **Smooth + noRealM** è ancora una questione irrisolta. Si potrebbe dire che un grado di curva più elevato significhi che la posizione di ciascun punto è influenzata da un maggior numero di punti di controllo, forse contribuendo a ridurre il comportamento vibratorio osservato quando si avvicina la forma fisica da distanze elevate. Data la scarsa influenza sulle prestazioni del metodo, non è stata condotta ulteriore indagine e il valore ottimale è stato mantenuto a $p = 3$.

Per quanto riguarda il contrasto dell'immagine, questo test ha contribuito a evidenziare la diversa sensibilità delle due strategie, ma per eseguire una corretta valutazione dell'incertezza sarà necessario calcolare l'**incertezza espansa**. Metodi per valutare l'incertezza ISO50% possono essere esplorati in [5] [10].

D.3.5 Panoramica delle prestazioni

Nel grafico in Figure 3.24 si può vedere una panoramica delle prestazioni delle due strategie. Con il test in (section 3.5) si è potuto evidenziare ancora una volta la superiorità prestazionale del metodo **autoshape**, ma anche che risultati similari, sebbene se con tempi di calcolo maggiore, possano essere ottenuti anche con la strategia **Smooth+noRealM** grazie ad una pre-identificazione svolta con un basso valore di CPPD per ottenere un contorno approssimativo su cui basare la ricerca del contorno ad alta densità di punti di controllo. Questa soluzione potrebbe tornare utile nelle casistiche in cui non fosse possibile basarsi su contorni identificati attraverso altre metodologie.

D.3.6 Effetto filtrante della CPPD

In questa sezione si è potuto dimostrare che il metodo **DBACD** non è in grado di identificare difetti con lunghezza d'onda inferiore a $\lambda_f = 2d$, con d distanza iniziale dei punti di controllo. La **CPPD** può quindi essere utilizzata come **parametro di filtraggio** per il metodo.

D.4 Confronto tra il metodo ISO50% + filtro gaussiano e i metodi VIC

Un ulteriore passo per comprendere le prestazioni di questi metodi è stato compiuto con il seguente esperimento. Il contorno della Figure 3.12 è stato ricostruito con i tre diversi metodi:

- Metodo ISO50% + filtro gaussiano secondo la norma ISO 16610-21 [19] (Funzione di filtro)

- Metodo DBACD con algoritmo autoshape e noSmooth. Grado della curva $\rho = 3$, $R_g = 0.5$.

- Metodo V2C

I tre contorni sono quindi stati confrontati per vedere se c'è una relazione tra i parametri di input di **risoluzione** e l'effetto di filtraggio. Questa relazione stimata è riportata in Equation D.5

$$f_c = \frac{(n_m)}{2} - 3 = \frac{(n_c p)}{2} \tag{D.5}$$

con n_m numero netto di modi utilizzati per V2C e $n_c p$ numero di punti di controllo utilizzati per DBACD. Questo confronto è stato eseguito con diversi valori di f_c.

$$f_c = [5\ 10\ 15\ 20\ 25\ 30\ 35\ 40\ 45\ 50]$$

D.4.1 Risultati del confronto

Per ciascun contorno, il **cerchio di riferimento con la distanza quadrata minima dei punti (LSD)**, il **cerchio massimo inscritto** e il **cerchio minimo circoscritto** vengono calcolati seguendo la definizione della norma ISO17450-1 [11] e implementata nelle funzioni: LSD circle, Minimum circumscribed Circle e Maximum inscribed circle. Una funzione di errore di circolarità è stata implementata in Appendix C. I valori risultanti del raggio e della posizione del centro sono stati rappresentati nei grafici in Table 3.13. Con queste informazioni possiamo affermare che:

- Il metodo ISO50% + filtro gaussiano, considerato come riferimento, mostra in tutti i grafici grande stabilità e chiare tendenze.

- Gli errori relativi tra ISO50% e DBACD o V2C (Table 3.14) sono generalmente più elevati con una frequenza di **taglio del filtro** inferiore. È anche apprezzabile che gli errori relativi per i cerchi M.I. e M.C siano generalmente più elevati rispetto a L.S.D.

- DBACD e V2C mostrano un comportamento simile, confermando la relazione in Equation D.5. Inoltre, DBACD sembra meno stabile a basso f_c e più stabile ad alto f_c rispetto a V2C.

- Per quanto riguarda la valutazione della circolarità, la deviazione relativa rispetto al valore di riferimento ISO50% può raggiungere picchi intorno al

30%, mostrando maggiore sensibilità alle diverse proprietà di filtraggio dei difetti a basso f_c.

D.4.2 Considerazioni

Dopo aver condotto questo ultimo esperimento, si può ancora una volta affermare l'**efficacia** dei metodi DBACD e V2C nell'analisi complessiva della forma della sezione. Tuttavia, una problematica ricorrente risiede nella rilevazione subottimale dei difetti locali dovuta ai minimi locali nella funzione obbiettivo Φ. Man mano che la dimensione del difetto più piccolo rilevabile aumenta con la riduzione della frequenza di taglio, l'impatto di una possibile non-rilevazione diventa più significativo sui risultati finali.

Questa osservazione è evidente nella Figure 3.47, dove vengono confrontati i risultati del filtraggio. Ad esempio, quando si utilizza $f_c = 10$ upr, il metodo DBACD non riesce a rappresentare un difetto situato nella parte in alto a destra dell'immagine (Figure 3.45). Allo stesso modo, il metodo V2C riconosce erroneamente un piccolo volume parzialmente fuso vicino al bordo come parte effettiva della sezione (Figure 3.46 lato destro).

Per superare queste sfide, sarà essenziale affinare ulteriormente e ottimizzare gli algoritmi di rilevamento per migliorare l'accuratezza e l'affidabilità delle nostre analisi.

D.5 Conclusioni e prospettive future

Il metodo di **Discrete B-spline Active Contour Detection** testato in questo documento ha mostrato prestazioni controverse. Quando eseguito con le giuste condizioni di input, i risultati potrebbero essere confrontati con il **contorno ISO50%**, specialmente dopo il filtraggio. Anche nelle migliori condizioni, però, il metodo non è sempre stato in grado di identificare i difetti più piccoli a causa di una limitazione di risoluzione (Figure 4.1). Come già detto molte volte, sarebbe possibile mitigare questo gap aumentando il **CPPD**, ma il prezzo da pagare sarebbe una **minor robustezza** e **tempi di calcolo** eccessivamente lunghi. Quando il metodo non può fare affidamento su contorni precedentemente identificati, la sensibilità ad altri parametri come il **contrasto dell'immagine** aumenta e le **prestazioni complessive diminuiscono**. Inoltre, data la natura iterativa di questo problema, viene introdotta una nuova fonte di incertezza. Infatti, è stato osservato più volte (section 3.3, section 3.5 e subsection 3.7.2) che la posizione iniziale dei contorni virtuali potrebbe influenzare il processo di identificazione. Prendendo ad esempio Figure 4.2, è logico comprendere che se il contorno virtuale si deforma spostandosi da sinistra a destra si fermerà prima di incontrare il difetto parzialmente fuso. Invece, il contorno proveniente da destra a sinistra lo incorporerà. Questo perché il processo iterativo ha informazioni sulle possibili deformazioni solo nella zona di transizione, mentre per il resto dell'immagine il gradiente sarà nullo. Ciò implica che quando si avvicina da sinistra, la zona di transizione inizierà a fornire informazioni sulla regione nera che separa la sezione dal difetto, e ciò sarà interpretato come il confine. Al contrario, avvicinandosi da destra, l'area bianca del difetto verrà riconosciuta come l'inizio della regione del materiale, ignorando la presenza della zona nera e quindi fermando la deformazione lì.

Il più grande vantaggio di questo metodo è ovviamente il tipo di dati di output. Diversamente dall'ISO50%, il metodo **DBACD** non restituisce un insieme discreto di punti ma una curva parametrica con la sua equazione. Ciò si traduce in una processo diretto che partendo dalla raccolta dati arriva senza passaggi intermedi alla geometria **filtrata**. Con un metodo ottimizzato, robusto e automatizzato, si evitano le incertezze e le fonti di errore derivanti dai passaggi intermedi.

D.6 Applicazioni del metodo

Il metodo **DBACD** si rivela estremamente utile e competitivo per valutare i **macro-difetti di forma**, poiché la risoluzione della b-spline può essere collegata al processo di filtraggio, con lunghezza d'onda di taglio $\lambda_f = 2d$ e d che rappresenta la distanza iniziale dei punti di controllo (section 3.6). Ciò implica che utilizzando il metodo **DBACD** per la valutazione dell'ondulazione si possono eliminare i passaggi intermedi normalmente richiesti per ottenere le informazioni sulla forma finale.

Nei procedimenti standard, i punti dati raccolti durante la procedura di misurazione (ad esempio, con l'ISO50% applicato su voxel CT) devono essere sottoposti a **filtraggio** per separare le informazioni sull'ondulazione da quelle sulla rugosità superficiale. I parametri di superficie vengono poi calcolati basandosi su questi profili filtrati e le geometrie che vi vengono associate. Nella maggior parte dei casi, è necessario un processo di adattamento della curva per valutare questi risultati [Baier et al. [20]]. Tuttavia, con il metodo **DBACD**, il profilo di ondulazione filtrato viene ottenuto come risultato del processo di rilevamento del contorno. Inoltre, le informazioni sul profilo vengono raccolte sotto l'equazione della curva b-spline. Ciò rende il metodo alternativo adatto a varie valutazioni e misurazioni relative alla **forma generale** di un oggetto, piuttosto che alla sua ruvidezza superficiale. Gli esempi includono valutazioni di **circolarità**, **cilindricità** e **coassialità**, che sono particolarmente rilevanti nel **controllo del processo** delle strutture reticolari. Deviazioni micro-superficiali dal valore nominale, sebbene rilevanti per le proprietà meccaniche, sono spesso attribuite a **errori casuali** e alcuni di essi possono essere corretti regolando i parametri di processo. D'altra parte, i macro difetti e i difetti di forma sono tipicamente associati a **errori sistematici** e possono essere mitigati anche con geometrie compensate. La capacità di separare direttamente questi due tipi di errori, senza la necessità di procedure aggiuntive, porta a un processo robusto e quasi automatizzato.

Inoltre, grazie alla definizione continua del contorno identificato, anche le **strategie di compensazione** possono beneficiare di questo metodo. Lo spostamento di ogni punto dalla sua posizione nominale può essere espresso come lo spostamento di $2p - 1$ punti di controllo. Grazie a Equation 2.10, il campo di spostamento complessivo può essere descritto come una curva b-spline con lo stesso numero di **C.P.**. Una prima strategia di compensazione grossolana per gli errori di forma può essere formulata nell'Equation 4.1.

Analoghe considerazioni potrebbero essere fatte anche per **V2C**. Tutto ciò che è stato discusso finora riguardo alle geometrie 2D può essere facilmente esteso alle **superfici 3D**. Un vantaggio significativo delle b-spline è la capacità di impilare curve 2D da ogni sezione di un oggetto e ottenere direttamente la definizione matematica della superficie 3D (Equation 4.3).

Incorporando le operazioni descritte in ISO 17450-1 2011 [11] (revisione di ISO 14660), il metodo DBACD semplifica il processo di ottenere il modello filtrato e associato. I metodi VIC con definizioni parametriche delle superfici possono anche migliorare i processi di **reverse engineering**, in cui è richiesta la ricostruzione matematica del modello della superficie di un manufatto.
Il miglior utilizzo di questo metodo può essere riassunto in tutte quelle applicazioni in cui l'obiettivo non è identificare con alta precisione un difetto specifico, ma piuttosto avere un'idea generale il più completa possibile della geometria della parte in esame. Il fatto che le superfici finali (nell'analisi volumetrica) siano descritte come funzioni matematiche rappresenta un potenziale vantaggio nell'analisi dei difetti. Prendendo come esempio la ricostruzione di un manufatto a struttura reticolare, i metodi VIC aprono la porta a possibili applicazioni davvero utili grazie alla possibilità di estrarre informazioni direttamente dai parametri superficiali (ampiezza dei modi di vibrare o posizione dei punti di controllo) senza dover effettuare operazioni di associazione specifiche. Quando questo tipo di operazioni sono semplici (ad esempio, trovare l'ovalizzazione di un cilindro per compensare la geometria nominale [20]), le procedure standard descritte in ISO 17450-1 2011 [11] rappresentano ancora la strategia di riferimento. Ma se iniziamo a pensare all'idea di compensare un difetto geometrico sistematico medio presente su strutture stampate con asse inclinato rispetto alla verticale, qualunque sia la sua forma, i metodi VIC analizzati potrebbero iniziare a essere competitivi, soprattutto per il fatto di ottenere l'output finale (geometria compensata) senza alcun passaggio intermedio. Le superfici ricostruite potrebbero anche essere utilizzate nell'analisi delle prestazioni meccaniche.

D.7 Lavori Futuri

Innanzitutto, potrebbero essere effettuati alcuni miglioramenti al metodo DBACD:

- Ulteriori studi sull'implementazione del codice per aumentarne la robustezza.

- Introduzione di una funzione di punteggio penalizzante che tenga conto quando i **punti di controllo** si allontanano troppo o si avvicinano troppo l'uno all'altro. Questo potrebbe contribuire ad aumentare la stabilità del processo a distanze iniziali elevate.

- Implementazione di un raffinamento locale del contorno. Identificando la regione con errori più elevati (ad esempio, valore locale della funzione obiettivo), la CPPD potrebbe essere aumentata localmente per consentire una migliore rilevazione di difetti più piccoli.

- La ricostruzione della superficie 3D di un cilindro potrebbe essere effettuata impilando i contorni di diverse sezioni.

Successivamente, potrebbe essere eseguito uno caso di studio pratico per confrontare i tre metodi sui dati volumetrici nel campo della compensazione geometrica. Inoltre, potrebbe essere cercata una correlazione tra i parametri di superficie e specifiche fonti di errore.

Inoltre, la questione di trovare un metodo di misurazione di riferimento (benchmark), meno sensibile a fattori come il contrasto, potrebbe essere risolta in lavori futuri.

Bibliography

[1] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, *Additive Manufacturing Technologies*. Cham: Springer International Publishing, 2021. [Online]. Available: https://link.springer.com/10.1007/978-3-030-56127-7

[2] M. Grasso, A. Remani, A. Dickins, B. M. Colosimo, and R. K. Leach, "In-situ measurement and monitoring methods for metal powder bed fusion: an updated review," *Measurement Science and Technology*, vol. 32, no. 11, p. 112001, Nov. 2021. [Online]. Available: https://iopscience.iop.org/article/10.1088/1361-6501/ac0b6b

[3] L. J. Gibson, "Cellular Solids," *MRS Bulletin*, vol. 28, no. 4, pp. 270–274, Apr. 2003. [Online]. Available: http://link.springer.com/10.1557/mrs2003.79

[4] R. Leach, D. Bourell, S. Carmignato, A. Donmez, N. Senin, and W. Dewulf, "Geometrical metrology for metal additive manufacturing," *CIRP Annals*, vol. 68, no. 2, pp. 677–700, 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0007850619301611

[5] F. Zanini, M. Sorgato, E. Savio, and S. Carmignato, "Dimensional verification of metal additively manufactured lattice structures by X-ray computed tomography: Use of a newly developed calibrated artefact to achieve metrological traceability," *Additive Manufacturing*, vol. 47, p. 102229, Nov. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214860421003894

[6] E. Malekipour and H. El-Mounayri, "Common defects and contributing parameters in powder bed fusion AM process and their classification for

online monitoring and control: a review," *The International Journal of Advanced Manufacturing Technology*, vol. 95, no. 1-4, pp. 527–550, Mar. 2018. [Online]. Available: http://link.springer.com/10.1007/s00170-017-1172-6

[7] M.-A. De Pastre, Y. Quinsat, and C. Lartigue, "Effects of additive manufacturing processes on part defects and properties: a classification review," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 16, no. 4, pp. 1471–1496, Dec. 2022. [Online]. Available: https://link.springer.com/10.1007/s12008-022-00839-8

[8] J. Zhao, M. Easton, M. Qian, M. Leary, and M. Brandt, "Effect of building direction on porosity and fatigue life of selective laser melted AlSi12Mg alloy," *Materials Science and Engineering: A*, vol. 729, pp. 76–85, June 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921509318306890

[9] A. Du Plessis, I. Yadroitsava, and I. Yadroitsev, "Effects of defects on mechanical properties in metal additive manufacturing: A review focusing on X-ray tomography insights," *Materials & Design*, vol. 187, p. 108385, Feb. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0264127519308238

[10] J. Lifton and T. Liu, "Evaluation of the standard measurement uncertainty due to the ISO50 surface determination method for dimensional computed tomography," *Precision Engineering*, vol. 61, pp. 82–92, Jan. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S014163591930159X

[11] *Geometrical product specifications (GPS). General concepts Model for geometrical specification and verification*, Confirmed ed., 2012, oCLC: 9521555236.

[12] M.-A. De Pastre, Y. Quinsat, and C. Lartigue, "Shape defect analysis from volumetric data - Application to lattice struts in additive manufacturing," *Precision Engineering*, vol. 76, pp. 12–28, July 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S014163592200040X

[13] M.-A. De Pastre and Y. Quinsat, “Virtual volume correlation of lattice structures: From volumetric data to geometrical and dimensional defects identification,” *Additive Manufacturing*, vol. 61, p. 103347, Jan. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214860422007369

[14] J. Réthoré and M. François, “Curve and boundaries measurement using B-splines and virtual images,” *Optics and Lasers in Engineering*, vol. 52, pp. 145–155, Jan. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0143816613002042

[15] Z. Jiang, J.-F. Witz, P. Lecomte-Grosbras, J. Dequidt, C. Duriez, M. Cosson, S. Cotin, and M. Brieu, “B-spline Based Multi-organ Detection in Magnetic Resonance Imaging: B-spline Based Multi-organ Detection in MRI,” *Strain*, vol. 51, no. 3, pp. 235–247, June 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/str.12136

[16] Z. Jiang, O. Mayeur, J. Witz, P. Lecomte-Grosbras, J. Dequidt, M. Cosson, C. Duriez, and M. Brieu, “Virtual image correlation of magnetic resonance images for 3D geometric modelling of pelvic organs,” *Strain*, vol. 55, no. 3, p. e12305, June 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/str.12305

[17] M. L. M. François, “Uncertainty of the virtual image correlation method,” *International Journal for Numerical Methods in Engineering*, vol. 123, no. 18, pp. 4367–4390, Sept. 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nme.7037

[18] L. Piegl, M. Andreolli, and W. Tiller, *The NURBS Book*. Germany: Springer, 1996.

[19] *Geometrical product specifications (GPS). Filtration Linear profile filters: Gaussian filters*, Revision underway ed., 2012, oCLC: 9521556471.

[20] M. Baier, M. Sinico, A. Witvrouw, W. Dewulf, and S. Carmignato, “A novel tomographic characterisation approach for sag and dross defects in

metal additively manufactured channels," *Additive Manufacturing*, vol. 39, p. 101892, Mar. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214860421000579