Suction system vapour velocity map estimation through SIFT-based algorithm

T. Tocci a, L. Capponi b, R. Marsili a, G. Rossi a, J. Pirisinu c

a Department of Engineering, University of Perugia, Via G. Duranti 93 - 06121 Perugia, Italy
b CISAS "G. Colombo", University of Padova, Via Venezia 15 - 35131 Padova, Italy
c Elica Tech Lab, Elica S.p.A, via E. Casoli 2 - 60044 Fabriano, Italy

E-mail: tommaso.tocci@outlook.it

Abstract. Measurement of velocity fields is a fundamental topic in fluid dynamics. Image-based analysis methods such as Particle Image Velocimetry or Laser Doppler Velocimetry are usually used. However, these techniques need complex instrumentation and particular test conditions. In this work, a computer vision-based approach is developed in order to obtain vapour velocity field map in effective, robust and economic way. Moreover, iterative filtering algorithm is applied to improve the results. The implemented method is tested on a suction system for domestic use, and the obtained velocity maps are validated by hot-wire anemometry, leading to totally comparable results, both in terms of profile and mean velocity. Uncertainty analysis shows acceptable results, considering the random nature of the phenomenon.

1. Introduction

In recent years, the use of non-contact image-based measurement techniques in different application fields has grown thanks to the technological advancement [1, 2]. In particular, in fluid dynamics, the experimental determinations of velocity fields through image-based methods is fundamental research topic. Usually, the measurement of velocity fields is performed by established methods, such as the Particle Image Velocimetry (PIV) [3, 4]. In the PIV, the fluid is seeded with tracer particles which, for sufficiently small particles, are assumed to faithfully follow the flow dynamics. The particles are illuminated by laser sheet so that particles are visible. The motion of the seeding particles is used to calculate speed and direction of the flow. In addition there are many techniques, such as the Laser Speckle Velocimetry (LSV) [5] and the Digital Particle Image Velocimetry (DPIV) [6]. Moreover, techniques based on different physical principles have been developed for these applications, such as the Laser Doppler Velocimetry (LDV) [7], which is using the doppler shift in a laser beam to measure the velocity in transparent or semi-transparent fluid flows. The measurement with laser Doppler anemometry is absolute and linear with velocity and requires in principle no pre-calibration. Furthermore, optical flow methods are commonly used for the velocity mapping [8, 9, 10, 11], even if issues like ambiguities, changing lighting conditions and occlusions limit the application of these methods.

The aim of this research is to develop a robust algorithm that is able to evaluate vapour velocity fields using less complex instrumentation and with fewer restrictions on test conditions, compared to most common techniques, such as PIV or LDV, which require a very expensive and complex instrumentation. Therefore, the Scale Invariant Feature Transform (SIFT) has been chosen as a starting point. It is a computer vision algorithm used to detect and describe images
local features [12, 13] and successfully applied in range images [14]. There are other feature
detection algorithms, such as Speeded Up Robust Feature (SURF) [15], which was created to
offer better computational performance but which does not offer precise and robust results such
as the SIFT [16]. In this study, a toolbox has been developed in Matlab environment. It is
based on the SIFT algorithm that is able to measure velocity fields. Moreover, Random Sam-
ple Consensus (RANSAC) filtering algorithm was used for improve the results. The proposed
method was tested on a suction system for domestic use produced by Elica S.p.A.. Nine differ-
ent conditions were analysed and the results were compared to an anemometric measurements,
obtained by using only the cold seeding particle.

2. Theoretical background
2.1. Scale Invariant Feature Transform
The Scale Invariant Feature Transform (SIFT) is a computer vision algorithm used to extract
a set of features from an image that uniquely describe it [12]. The method is invariant
from translations, rotations and projective transformations, and, partially, from changes in
illumination and in point of view shifts [17]. The algorithm identifies descriptors points that
are highly distinct, both in frequency and space domains, and, therefore, it finds correct
correspondences with high probability. Algorithm steps are now introduced in details.

Individuation of the scale-space: Key-points are elements of interest to which every feature
refers, defined in a region called scale-space, described by \( L(x, y, \sigma) \) function from:

\[
L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)
\]

where \( * \) is the convolution operator, \( I(x, y) \) represents analysed frame and \( G(x, y, \sigma) \) is a
Gaussian operator, defined as:

\[
G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}
\]

To efficiently compute the invariant regions in different scales, the Difference of Gaussian (DoG)
is applied, as explained in Fig. 1. The DoG has the following expression:

\[
DoG(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)
\]

The DoG function is a good approximation of the Laplacian of normalized Gaussian, the proper
scale invariant [19], and it requests less computational effort if compared to other methods.
Key-point localization: Key-points are defined by local maximum or minimum of the DoG function. In order to identify the local extrema of the DoG, every point is compared with the eight adjacent points in the same scale and with the nine related points in the higher and lower scale, as shown in Fig. 1. The analysed point is indicated as local extrema if its value is higher or lower than the adjacent points. A threshold is also setted for the filtering step.

Orientation assignment: An orientation to each key-point is assigned in order to make the descriptor invariant to rotation. Orientation depends exclusively on the properties of image nearby the point itself. Comparison between assigned values is done and the maximum is then chosen as dominant orientation and it is used to rotate the image, to normalize its position and maintain invariance respect the rotation.

Key-point descriptor: In this step a descriptor invariant to change of illumination and perspective is determined. The descriptor is created by evaluating the magnitude in a region around the point of interest in order to avoid sudden changes in the descriptor due to small variations in the position of the window, but also to give less emphasis to the gradients further away from the centre of the descriptor. The final descriptor is built by dividing the sixteen pixels close to the key-point in 4x4 pixel sub-regions: for each region eight different orientations are identified: it means that each descriptor is a vector made up of 128 elements.

Key-points matching: The candidate point for the matching is identified by evaluating the descriptors of pairs of points extracted from different frames: those that are characterized by the lowest Euclidean distance are identified as homologous points. It is also necessary to use a method that allows to delete those points that do not match. The method consists of comparing the distance between the key-point and the nearest point with that between the key-point and the nearest second, establishing a minimum threshold value for accept the tie point.

2.2. Random Sample Consensus
Random Sample Consensus (RANSAC) is an non-deterministic iterative algorithm for the estimation of parameters of a mathematical model, starting from a set of input data containing a large percentage of outliers [20]. The RANSAC algorithm generates solutions using the minimum number of observations necessary to estimate the parameters of the model. In fact, while standard techniques use the greatest possible amount of data to obtain a solution and proceed only subsequently to delete the outliers, RANSAC uses a smaller data set to determine the model and proceeds to enlarge this set with the points of data consistent with the estimated model.

The parameters useful for this type of analysis are: Input data set; Outliers; Sample Set (MSS) (i.e., the minimum set $s$ of $k$ initial samples used to build the model $M$); Consensus Set (CS) (i.e., the set of initial data that is consistent with the model calculated through to the MSS; the CS therefore contains the data inliers that can be determined from the model). The number $N$ of minimum iterations to perform is:

$$N = \frac{\log(1-p)}{\log[1-(1-\epsilon)^s]}$$  \hspace{1cm} (4)

where $\epsilon$ is the probability of density of the outliers and $s$ is the number of points necessary to define a model. The size $T$ of a minimum CS can be deduct statistically simply as:

$$T = (1-\epsilon) n$$  \hspace{1cm} (5)

The algorithm is divided into five steps, that are:
1. **MSS choice:** the MSS is chosen randomly from the initial data set.

2. **Model hypothesis:** the model is calculated solely using the MSS points.

3. **Error estimation:** the consistency of the model is checked, i.e., the distance to each point of the initial set is estimated. Points that present an error below a threshold \( d \) are considered to be inliers.

4. **CS selection:** if the number of inliers of the CS relative to the estimated model exceeds a certain value and the total error is below a threshold, this model is considered good as an approximation of all the initial data.

5. **Repetition the hypothesis and verification:** the previous steps are repeated enough times to guarantee that the probability of not having outliers in the chosen CS is greater than the threshold value. After iterations, the model which has obtained the best ratio inliers/outliers and magnitude of the error is chosen as *best model*. If no model hypothesis is enough accurate, the algorithm does not return any model.

### 3. Materials and methods

In this Section the experimental analysis is described: the test bench and the measurement chain are introduced and the experimental setup is presented in details.

#### 3.1. Algorithm development

In this research, the frame acquired at the instant \( t_i \) is called the *PRE* frame and the one extracted at the time \( t_i + \Delta t \) is called *POST* frame, where the interval \( \Delta t \) is defined from the acquisition frame rate. The developed algorithm allows to use the SIFT to determine the displacements in pixel described by the key-points, identified between a PRE frame and the POST frame. The system is temporally and spatially calibrated, which allows to evaluate the speed in \([\text{m/s}]\). A vectorial map of the velocity of these key-points is thus created. Repeating this operation for all the frames of the video and superimposing the vector maps between each pair of PRE and POST frames, a complete velocity field is built using all the key-points identified for each couple of frame. The algorithm described below is divided into two sequential phases: matching phase and vector filtering and map building phase.

#### 3.1.1. Matching:

Each frame is subdivided into nine overlapped *Regions of Interest* (ROI), with a dimension of 1/4 of the original frame, as shown in Fig. 2. The matching phase begins from the first frames pair, where PRE = \( t_i \) and POST = \( t_i + \Delta t \). The first iteration looks for matching between the ROI 1 of the PRE frame and all the neighbouring ROIs and itself in the POST frame, e.g., the ROI 1, 2, 4, 5. The second iteration looks for matching between the ROI 2 of the PRE frame and all the neighbouring ROIs and itself in the POST frame, *i.e.*, the ROI 1, 2, 3, 4, 5, 6. The iteration is repeated 9 times, one for each ROI of the PRE frame. However, in different ROIs, being overlapped, the same matching can be observed. For this reason, these duplicates are deleted. In the next step, the second frames pair is analysed (PRE = \( t_i + \Delta t \), POST = \( t_i + 2\Delta t \)). The process is carried on for each pair of frames extracted from the video. As the pairs of frames are processed, a matching matrix is filled out: it contains the x-y coordinates [pixel] of the key-points of the PRE and POST frames, the displacement and velocity values of the points, as shown in Tab. 1. To clarify the matching process, the iteration scheme is shown in Fig. 3. The setting parameters of the matching phases are the *key-point threshold* and the *matching threshold*. The first one is used in order to avoid non-stable key-points, so they are selected only if the extremes of the Gaussian function are higher than a threshold value. The second parameter acts when a descriptor \( D_i \) (of PRE frame) is matched with a descriptor \( D_{i+1} \) (of the POST frame) only if the Euclidean distance between \( D_i \) and \( D_{i+1} \) multiplied by the threshold value is not greater than the distance between \( D_i \) and all the other descriptors.
Figure 2. Subdivision in ROI of the analysed frame

Table 1. Matching matrix

| # Matching | X_{PRE} | Y_{PRE} | X_{POST} | Y_{POST} | u[m] | v[m] |
|------------|---------|---------|----------|----------|------|------|
| 1          | x_{11}  | y_{11}  | x_{21}   | y_{21}   | u_1  | v_1  |
| 2          | x_{12}  | y_{12}  | x_{22}   | y_{22}   | u_2  | v_2  |
| ...        | ...     | ...     | ...      | ...      | ...  | ...  |
| n          | x_{1n}  | y_{1n}  | x_{2n}   | y_{2n}   | u_n  | v_n  |

Results of this phase are two different maps, one with the key-point displacements, exposed in Fig. 4, and a table containing the matching matrix.

3.1.2. Vector filtering and map building: The matching phase generates the vector map with a percentage of wrong vectors, both in modulus and direction. The origin of matching errors is due to the nature of the SIFT algorithm, which searches for similarities between two geometric markers with a given tolerance defined by the threshold values. Analysing a gaseous fluid is more complex than a solid body, thus in order to reduce the number of false matches, it is better to increase the frame rate. By acquiring at higher frame rate, abrupt vapour shape changes are avoided and the correct key-point matching is guaranteed. False matches cannot be deleted upstream, so, filtering is necessary in order to discard the majority of failed vectors and build a velocity field as precise as possible. Starting from the matching map Fig. 4, each operation is carried out in a floating window. It improves the robustness and consistency of the filtering. The filtering operates on three different levels:

- **Preliminary filtering:** After the definition of an expected speed range, vectors that do not fit in are filtered, as shown in Fig. 5-(a). Subsequently, the mean direction of the vectors within the floating window is determined: each vector with a direction that differs from the mean value direction is then filtered Fig. 5-(b).
- **RANSAC filtering:** this filtering phase is based on RANSAC algorithm [20]. The process...
Figure 3. Iteration processing for matching phase

involves 100 principal iterations for each floating window. A new matrix is created, called the *ransac matrix*, which contains all the key-points identified within the window. The process of each iteration is here outlined:

1. Four key-points are randomly identified within the *ransac matrix*. The displacement of these key-points is used to determine a projective transformation matrix that transforms their position on the PRE frame into the position of the same points in the POST frame.

2. The projective transformation matrix is exploited to transform the PRE coordinates of the points present in the *ransac matrix*, obtaining a new matrix that contains the POST coordinates determined by geometric transformation.
3. The algorithm determines which of the points is an outlier. In order to obtain this result, the difference between the actual POST and the projected POST position is determined. If the distance between these two points, expressed in mm, is greater than the setted maximum threshold value, the point is marked, adding a $+1$ to its outliers counter.

For each iteration it adds a $+1$ to the outlier counter at each point that does not check the inlier condition. At the end of iterations, the function automatically determines the mean value of the outlier counter for each key-point. Each point which has a counter value greater...
than the average value of all the counters multiplied by a filter gain value is rejected by the key-point matrix, as shown in Fig. 5-(c).

- **Finishing filtering**: for each window the standard deviation $\sigma$ with respect to the average speed value is identified, and the values that deviate from a $2\sigma$ confidence interval are discarded, as shown in Fig. 5-(d).

Finally, the algorithm autonomously creates a 25 pixel mesh grid that is used to determine an interpolated velocity field. In order to have a map as complete as possible, it is therefore necessary to have the highest and most uniform number of real vectors.

![Images of filtering methods](a)(b)(c)(d)

**Figure 5.** Comparison of filtering. Modulus filtering (a) Direction filtering (b) RANSAC filtering (c) Standard deviation filtering (d)

### 3.2. Test Bench

The test bench used in the experimental analysis consists of steam production, suction, laser sheet illumination, seeding and motion system as well as signal acquisition camera. Test bench components are illustrated in Fig. 6. The fluid analysed in this experiment is water vapour, generated and aspirated by a domestic inductive plane. The steam is generated by boiling water inside a pot, placed above the plane, and it is aspirated by a kitchen hood, centrally incorporated in the inductive plane itself. The lighting system (Sanctity Laser Green DPPS Laser - 532 nm and 2000 mW) consists of a Nd:YAG green laser generator, which produces a laser sheet with
a 90° opening. The inseminating fluid used is a glycol-water mixture, produced by a proper
generator (Stage Line FM). A tracer is also used [21], due to the reflections and permanence
properties better than water vapour. The seeding product is inoculated directly on the analysis
plane using a perforated copper tube, adjustable in height. In order to be able to handle laser
sheet and insemination system together, the entire system is mounted on a slide (IGUS 670 mm)
driven by a stepper motor (SANYO DENKI Step Syn) controlled by a proprietary driver and
an Arduino board. The acquisition system consists of a CCD vision camera (Teledyne DALSA
Genie Nano M2450) capable of recording greyscale video with a resolution of 2448 × 2048 pixels
at 90 fps and a 2/3 lens (Chiopt FA1202A) with 1.4 focal aperture. In order to optimize
the video acquisition, the system exploits the scattering angle of 45° between the camera and the
laser sheet orthogonal plane, as shown in Fig. 7-(a), according to to Mie theory [2]. The system
is calibrated, using an algorithm that uses a check board and projective transformation. Thus,
every frame acquired from the actual position of the camera is transformed as if it were acquired
frontally.

3.3. Experimental setup
The algorithm has been tested in different conditions of use, using different pot sizes and fan
speed of the suction system. Two types of pots have been used and their characteristics are
shown in Tab. 2. A set of 400 frames is acquired for each testing condition, shown in Tab. 3.

| Table 2. Pot setup | Diameter [mm] | Height [mm] |
|--------------------|---------------|-------------|
| High Pot           | 180           | 80          |
| Low Pot            | 280           | 20          |

The acquisition plane and the positioning of the pots are defined trying to limit shadows and
interferences: it means that with one pot, the acquisition system is setted as shown in Fig. 7-(b),
while with two pots, this procedure is not applicable as the inner pot would produce a shadow,
so the plane is then positioned parallel to the laser sheet Fig. 7-(c).
Figure 7. Positioning scheme of the vision camera (a) One pot setup (b) Two pot Setup (C)

### Table 3. Experimental setup

| Test Number | Number of Pot | Type of Pot | Aspiration Speed |
|-------------|---------------|-------------|------------------|
| 1           | 1             | High        | 3                |
| 2           | 1             | High        | 6                |
| 3           | 1             | High        | 9                |
| 4           | 1             | Low         | 3                |
| 5           | 1             | Low         | 6                |
| 6           | 1             | Low         | 9                |
| 7           | 2             | High        | 3                |
| 8           | 2             | High        | 6                |
| 9           | 2             | High        | 9                |

### 3.4. Anemometric measurement

In order to validate the results obtained through image analysis, hot-wire anemometry has been used. Measurement has been performed sequentially in a grid of acquisition points and they
are repeated for different speeds of the suction system (i.e., 3-6-9). At each measurement point the anemometer signal has acquired for 60 seconds. Speed values are determined by temporal averaging and using interpolation methods, the speed profile has been obtained, as shown in the Fig. 8.

4. Results
To simplify the reading of the data, due to the large number of tests performed, the results shown in this section refer to test number 2, described in Tab. 3. The interpolated velocity field map has been extracted from the filtering algorithm and it is shown in Fig. 9. In order to verify the developed method and validate the speed maps, a comparison between the velocity profiles obtained with SIFT-based algorithm and hot-wire anemometer has been performed. The comparison has been made for two profiles: profile 1 at 40 mm from the suction plane and profile 2 at 100 mm. The profile 2 analysis is shown in Fig. 10. The fluctuation of the velocity obtained with the image analysis is due to the turbulence caused by convective phenomena. The comparison with the anemometer has been done in term of average velocity values. Then analysing these parameters, it can be clearly seen that the profiles are almost in acceptable agreement.

Figure 8. Interpolated anemometric velocity map for velocity 6
Figure 9. Interpolated Velocity Field - Test 2 (High Pot - Aspiration Speed 6)

Figure 10. Test 2 - Comparison between velocity profiles extracted in the distant plane (82.5 mm from suction plane)
4.1. Uncertainty analysis
For a further validation of the algorithm, an uncertainty analysis on the measurement has been performed. In order to estimate the uncertainty value, five measurements were carried out on the same test with a fixed set-up condition. For completeness, also for uncertainty analysis setup of test number 2 has been chosen. The profile 1 velocity values are extracted and reported in Tab. 4. Subsequently, in order to have a denser sample of measurements, a grid of spatial coordinates has been defined in x [pixels]. However, where a value is missing, a speed value obtained by interpolation with spline starting from the known speed values is inserted. The cumulative map is then built with the complete velocity fluctuation for each measurement, as shown in Fig. 11. The uncertainty of measurement [22] has been performed by:

\[
\bar{q} = \frac{1}{n} \sum_{k=1}^{n} q_k
\]

\[
s^2(q_k) = \frac{1}{n-1} \sum_{k=1}^{n} (q_k - \bar{q})^2
\]

\[
s(q_k) = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} (q_k - \bar{q})^2}
\]

\[
i(\bar{q}) = \frac{s(q_k)}{\sqrt{n}}
\]

where \( n \) is the number of measurements, \( q \) is the measured value, \( s^2(q_k) \) is the experiential variance, \( s(q_k) \) is the standard deviation and \( i(\bar{q}) \) is the uncertainty value. In order to obtain a unique value of uncertainty that characterizes the entire measurement process, the uncertainty values have been mediated, resulting in the average uncertainty, expressed in percentage points. It results in:

\[
i(\bar{v}) = 4.05\%
\]

5. Conclusions
In the suction systems industrial field it is important to carry out products quality checks through velocity fields. Classical image-based analysis, such as PIV, allow to reach high level of details and precision on velocity measurement. However, in this industrial application, they are
not required, due to expensive and complex equipment. For this reason, the aim of this work is to develop an easy-to-use and low cost image-based analysis tool. In particular, computer vision were used to measure vapour velocity fields. Scale Invariant Feature Transform and Random Sample Consensus has been chosen as working base. The algorithms were tested on a suction system for domestic usage. After a software development in Matlab environment, experimental analysis has been carried out. Videos were acquired in different operating conditions, in order to simulate a complete use of the suction system. Velocity fields were measured and compared to those obtained from hot-wire anemometry. This comparison leads to acceptable results: in fact, outcomes given by anemometry and by SIFT-based algorithm are comparable, both in terms of profile and mean velocity. Moreover, an uncertainty analysis was performed. Results have shown relatively low uncertainty, considering the hard operating conditions, in terms of turbulence and random nature of the phenomenon.

References

[1] G. Allevi, L. Capponi, P. Castellini, P. Chiariotti, F. Docchio, F. Freni, R. Marsili, M. Martarelli, R. Montanini, and S. Pasinetti, “Investigating additive manufactured lattice structures: a multi-instrument approach,” IEEE Transactions on Instrumentation and Measurement, 2019.

[2] M. Raffel, C. E. Willert, F. Scarano, C. J. Kähler, S. T. Wereley, and J. Kompenhans, Particle image velocimetry: a practical guide. Springer, 2018.

[3] L. Adrian, R. J. Adrian, and J. Westerweel, Particle image velocimetry. No. 30, Cambridge University Press, 2011.

[4] P. Korczyk, S. P. Malinowski, and T. A. Kowalewski, “Mixing of cloud and clear air in centimeter scales observed in laboratory by means of particle image velocimetry,” Atmospheric research, vol. 82, no. 1-2, pp. 173–182, 2006.

[5] R. Meynart, “Instantaneous velocity field measurements in unsteady gas flow by speckle velocimetry,” Applied Optics, vol. 22, no. 4, pp. 535–540, 1983.
[6] C. E. Willert and M. Gharib, “Digital particle image velocimetry,” *Experiments in fluids*, vol. 10, no. 4, pp. 181–193, 1991.

[7] F. Durst, “Principles of laser Doppler anemometers,” in *In Von Karman Inst. of Fluid Dyn. Meas. and Predictions of Complex Turbulent Flows, Vol. 111 p (SEE N81-15263 06-34)*, vol. 1, 1980.

[8] D. B. Bung and D. Valero, “Optical flow estimation in aerated flows,” *Journal of Hydraulic Research*, vol. 54, no. 5, pp. 575–580, 2016.

[9] D. B. Bung and D. Valero Huerta, “FlowCV-An open-source toolbox for computer vision applications in turbulent flows,” in *Proceedings of the 37th IAHR World Congress*, 2017.

[10] G. Allevi, L. Casacanditella, L. Capponi, R. Marsili, and G. Rossi, “Census Transform Based Optical Flow for Motion Detection during Different Sinusoidal Brightness Variations,” in *Journal of Physics: Conference Series*, vol. 1149, p. 12032, IOP Publishing, 2018.

[11] S. Yoneyama and G. Murasawa, “Digital image correlation,” *Experimental mechanics*, 2009.

[12] D. G. Lowe, “Object recognition from local scale-invariant features,” in *iccv*, p. 1150, Ieee, 1999.

[13] T. Lindeberg, “Scale invariant feature transform,” 2012.

[14] M. Moretti, G. Gambucci, R. K. Leach, and N. Senin, “Assessment of surface topography modifications through feature-based registration of areal topography data,” *Surface Topography: Metrology and Properties*, vol. 7, p. 025003, may 2019.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*, pp. 404–417, Springer, 2006.

[16] L. Juan and L. Gwon, “A comparison of sift, pca-sift and surf,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 3, pp. 169–176, 2007.

[17] R. Szliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[18] G. Wang, B. Rister, and J. R. Cavallaro. “Workload analysis and efficient OpenCL-based implementation of SIFT algorithm on a smartphone,” in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 759–762, IEEE, 2013.

[19] T. Lindeberg, “Scale-space theory: A basic tool for analyzing structures at different scales,” *Journal of applied statistics*, vol. 21, no. 1-2, pp. 225–270, 1994.

[20] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] A. Melling, “Tracer particles and seeding for particle image velocimetry,” *Measurement Science and Technology*, vol. 8, no. 12, p. 1406, 1997.

[22] H. W. Coleman and W. G. Steele, *Experimentation, validation, and uncertainty analysis for engineers*. John Wiley & Sons, 2018.

[23] E. Speranzini, R. Marsili, M. Moretti, and G. Rossi, “Image Analysis Technique for Material Behavior Evaluation in Civil Structures,” *Materials*, vol. 10, p. 770, jul 2017.

[24] F. Bianconi, S. Catalucci, M. Filippucci, R. Marsili, M. Moretti, G. Rossi, and E. Speranzini, “Comparison between two non-contact techniques for art digitalization,” in *Journal of Physics: Conference Series*, vol. 882, p. 012005, IOP Publishing, aug 2017.

[25] M. Becchetti, R. Flori, R. Marsili, and G. L. Rossi, “Measurement of stress and strain by a thermocamera,” in *Proceedings of the SEM Annual Conference*, 2009.

[26] G. Allevi, M. Cibeca, R. Fioretti, R. Marsili, R. Montanini, and G. Rossi, “Qualification of additively manufactured aerospace brackets: A comparison between thermoelastic stress analysis and theoretical results,” *Measurement*, vol. 126, pp. 252–258, oct 2018.

[27] R. Marsili, G. Rossi, and E. Speranzini, “Fibre Bragg Gratings for the Monitoring of Wooden Structures,” *Materials*, vol. 11, p. 7, dec 2017.

[28] E. Cardelli, M. Cibeca, A. Faba, R. Marsili, M. Pompei, and G. Rossi, “Magnetic sensors for motion measurement of avionic ball screws,” *AIP Advances*, vol. 7, p. 056639, may 2017.