Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 121 (2024) 85–90

www.elsevier.com/locate/procedia

11th CIRP Global Web Conference (CIRPe 2023)

Comparative Analysis of Image Processing Schemes to Register Motion from Video of Vibrating Cutting Tools

Varun Raizada[a], Harsh Singh Rajput[a], and Mohit Law[a]*

[a]*Machine Tool Dynamics Laboratory, Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India*

* *E-mail address:* mlaw@iitk.ac.in

Abstract

This paper reports on the efficacy of six different image processing schemes to properly resolve cutting tool motion from video. We compare the edge detection and tracking scheme with intensity- and phase-based optical flow-based schemes and with digital image correlation. We also introduce the use of two new schemes: one based on particle image velocimetry (PIV), and another based on the use of convolution neural networks (CNN). Methods are illustrated with video of two tools. Results are benchmarked with twice integrated accelerations. We find that the edge detection and tracking scheme, the PIV scheme, and the CNN scheme are robust.

Keywords: Vibrations, Measurement, Image processing, Modal analysis.

1. Introduction

Vision-based modal analysis of cutting tools involves estimating motion from video of vibrating tools. The central idea in the method is to treat every pixel as a virtual sensor and track its change of intensity across frames to correlate that change with the object's motion. The correlation is done using image processing schemes. The method is non-contact and full field and does not require sophisticated data acquisition systems. The method needs just a camera and image processing code running on computers. Due to these advantages, vision-based methods have found application across domains. These include monitoring of large civil infrastructure and turbines [1-4] that vibrate at low frequencies with large amplitudes, and vibration analysis of cutting tool systems that vibrate at high frequencies with small amplitudes [5-10].

Prior research has reported on the efficacy of different image processing schemes in resolving motion. For example, the edge detection [11] and tracking scheme was shown to work well for registering relatively large tool motion [5-8]. However, the scheme fairs poorly when motion is small and of the subpixel regime, as reported in [9-10]. Optical flow-based schemes [12-15] were shown to work even for small subpixel level tool motion [10]. The digital image correlation (DIC) scheme has also been used to register cutting tool motion [6]. However, since implementation in [6] correlated the tool's own features and not random speckle patterns sprayed on the tool, which is what is recommended [14], the scheme fared poorly in registering small motion [6].

Vision-based methods are fast-emerging as an alternative method for modal analysis of machine tool systems. As such, it would be prudent to characterize how different image processing schemes influence registered motion and modal parameters estimated from that motion. This is indeed the

10.1016/j.procir.2023.09.233

main aim of this paper: to compare the motion registration ability of six different image processing schemes.

The six schemes we compare are: edge detection, phase- and intensity-based optical flow, DIC, particle image velocimetry (PIV) and deep learning with convolution neural networks (CNN). Of these six, we show for the first time herein the capability PIV and CNN to register motion from video of tools. Characterizing the ability of PIV and CNN to register cutting tool vibratory motion and compare the performance of six different methods are the main new and modest technical contributions of this paper.

PIV is usually used to measure 2D velocity fields in macroscopic fluid flow by imaging the flow seeded with particles followed by cross-correlating groups of particles like in DIC to obtain average displacements. In our implementation we correlate the tool's own features across frames. We use PIVLab – which is an easy to use, GUI based MATLAB tool to analyze, validate, postprocess, visualize and simulate PIV data [16]. In our use of CNN, we leverage its ability to solve complex image processing tasks such as classification, segmentation, and recognition. In our implementation, we use CNN to learn spatial and temporal information from a pair of images to then estimate optical flow that describes motion [17-19]. Unlike classic image processing schemes, CNN learns what filters to use to estimate flow by training itself on a dataset. We implement the CNN algorithm in Python to learn motion from video. For the other four methods we write our own code in MATLAB.

Comparative analysis of the different image processing schemes to register motion and extract modal parameters from that motion is illustrated herein using two tools: an endmill and a grooving blade. Tools are excited with an impulse input, and the resulting vibrations are recorded using a high-speed camera. The experimental setups to acquire video are discussed in the second section of the paper.

The third section of the paper briefly overviews the workings of the six different image processing schemes of interest. These build on reports in [11-15], and their adaptation for cutting tools reported in [5-10]. We also discuss the merits of each of the methods of motion registration in Section 3. Since two of the six schemes are new, we discuss their implementation in greater detail. The fourth section of the paper compares motion and modal parameters extracted from that motion. For modal parameter extraction, we use the eigensystem realization algorithm (ERA) [20-21] that works well with impulse-type responses that result from the impulse-type excitation. Results are benchmarked against displacements and modal parameters obtained from twice integrated acceleration signals.

2. Experimental Setup to Record Video of Vibrating Tools

Fig. 1(a) shows an end mill cutter of 16 mm diameter and Fig. 1(b) shows a slender cantilevered grooving blade of width 7.5 mm. The tools were excited with a modal hammer with input energies of sufficient bandwidth to excite the modes of interest. The resulting impulse-like response for the end mill was recorded by a Chronos 2.1HD high-speed camera with an appropriate lens. For the blade, we used a OnePlus 7T smartphone camera. Each experiment was repeated at least twice. Since the cameras are setup perpendicular to tool motion only in-plane motion was recorded. The high-speed camera recorded at 2142 frames per second (fps) with a resolution of 1280 x 720 pixels to result in a per-pixel size of 36 μm. The smartphone camera recorded at 480 fps and with a frame resolution of 720 x 1280 pixels to result in a per-pixel size of 37.5 μm. We used a DC light source to illuminate the tools and used a white board to minimize the influence of background noise. Video recorded in the .mp4 format was transferred to MATLAB and Python for further processing. At first, each video was decomposed into its individual frames and all frames were converted to grayscale. Each frame for each video was then cropped around the edges of interest corresponding to the free ends of the tools. Only motion within this cropped region was tracked across frames using the different image processing schemes outlined next.

We simultaneously measured the response for both tools using an accelerometer mounted at the free ends of the tools. We acquired data at a sampling frequency of 51.2 kHz using National Instrument's (NI) data acquisition system (NI9234 + NI9171). Accelerations serve as the ground truth.

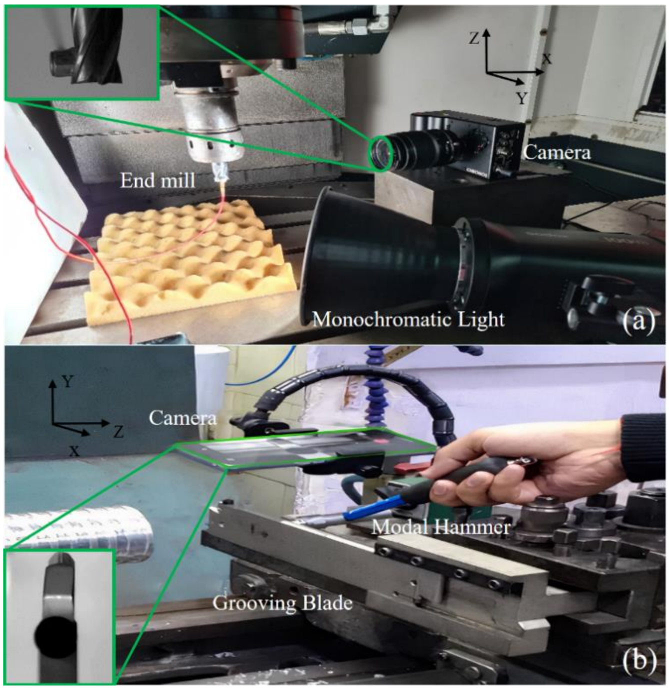

Fig. 1 (a) setup for recording an end mill cutter using a high-speed camera, (b) setup for recording a grooving blade using a smartphone camera.

3. Overview of the Image Processing Schemes of Interest

This section outlines the workings of the six image processing schemes of interest. We first discuss the edge detection and tracking scheme, followed by the two optical flow-based schemes. We then discuss DIC, PIV and CNN, respectively.

All pixels in a grayscale image act as a sensor with the ability to capture light intensity distribution $I(x, y)$. In the video of a vibrating tool, the intensities change in every frame to become a function of time, i.e., $I(x, y, t)$. We correlate the change in intensities to motion as discussed next.

3.1. Edge Detection and Tracking

The simplest technique to estimate motion from video involves using the Canny edge detector algorithm in every frame. It has already been shown that the Canny edge detector [11] outperforms other edge detectors in [5, 6]. Discussion herein is hence limited to Canny. Canny is a multi-stage algorithm that is robust in detecting strong edges and suppressing noise. The first step is the convolution of the image using a Gaussian kernel:

$$K(x,y) = \frac{1}{2\pi} exp\left(-\frac{(x-d-1)^2-(y-d-1)^2}{2\sigma^2}\right),$$
$$1 \le (x,y) \le (2d+1) \quad (1)$$

wherein σ represents the standard deviation and $(2d+1)$ x $(2d+1)$ is the dimension of the Gaussian kernel. After smoothening by use of a Gaussian kernel, the image gets convolved with the Sobel kernels which are of the form of:

$$G_x^S = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, G_y^S = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

for detection of edges. Finally, thresholds are used to bind the magnitude of detected edges between upper and lower bounds. Since edges of interest span several pixels, these are averaged in every frame to result in a scalar value for that frame. That value is stored in an array. The process is repeated for all frames to result in pixel-displacement time series data. For detailed implementation, please see [6].

This remains amongst the simpler techniques to register tool motion from video. The method however is sensitive to thresholds, to the size of the Gaussian blur, and is also not always a sub-pixel scheme.

3.2. Intensity-based Optical Flow

Intensity-based optical flow assumes brightness constancy [12, 13] i.e., the intensity of light for a point on the tool x_i, y_i that has moved slightly to another location $x_i + \Delta x, y_i + \Delta y$ in a short time Δt remains constant such that:

$$I(x_i, y_i, t) = I(x_i + \Delta x, y_i + \Delta y, t + \Delta t). \quad (3)$$

Using Taylor series expansion in Eq. (3) and ignoring higher order terms we can write:

$$\frac{dI}{dx}\Delta x|_{x=x_i} + \frac{dI}{dy}\Delta y|_{y=y_i} + \frac{dI}{dt}\Delta t|_t = 0, \quad (4)$$

wherein Δt represents the difference in the intensity of light at a point in two consecutive frames at some time t. Since our interest is only in displacements in the direction of intensity gradients, Eq. (4) gets converted to [14]:

$$|\nabla I|\Delta S(x_i, y_i, t) = I(x_i, y_i, t) - I(x_i, y_i, t + \Delta t) \quad (5)$$

wherein $|\nabla I|$ represents the magnitude of the intensity gradient and ΔS represents the displacement in direction of the intensity gradient. The displacement thus obtained is for tool motion within two consecutive frames of interest. For the complete time series data of pixel displacements, the procedure is repeated for all the subsequent frames. For detailed implementation, please see [6]. This method is more robust than the edge detection and tracking scheme. It is however sensitive to the size of the Gaussian blur, the choice of the kernel used to compute image intensity gradients, and to the method of averaging full field motion.

3.3. Phase-based Optical Flow

Phase-based optical flow is more robust, more accurate, and less sensitive than the intensity-based flow method and is also better suited for subpixel level motion registration. It convolves the image with a complex kernel to obtain local phase and amplitude information to decipher changes in the local phase to track motion over consecutive frames [15].

In phase-based optical flow, image sequence gets processed with the help of an oriented Gabor filter to obtain information about the localized motion. Gabor filter is a complex filter that gives information about local phase and local amplitude in orientation θ for an image of intensity $I(x, y, t_0)$ at a time t_0 on convolution:

$$A_\theta(x, y, t_0)e^{i\varphi_\theta(x,y,t_0)} = (G_\theta + iH_\theta) * I(x, y, t_0) \quad (6)$$

wherein A represents the local amplitude and φ represents the local phase. G and H are a quadrature pair that differ in phase by 90° and represent a complex Gabor filter.

The phase-based technique assumes a constant phase whose evolution with time gives the displacement signal which can be written with the help of phase gradient as [15]:

$$\left[\frac{d\varphi_\theta(x,y,t)}{dx}, \frac{d\varphi_\theta(x,y,t)}{dy}, \frac{d\varphi_\theta(x,y,t)}{dt}\right] \cdot (u, v, 1) = 0 \quad (7)$$

wherein u and v are the velocities that can be integrated to get displacement in x and y directions at time t_0:

$$d_x(t_0) = -\left[\frac{d\varphi_0(x,y,t)}{dx}\right]^{-1} [\varphi_0(x, y, t_0) - \varphi_0(x, y, 0)] \quad (8)$$

and,

$$d_y(t_0) = -\left[\frac{d\varphi_{\pi/2}(x,y,t)}{dy}\right]^{-1} \left[\varphi_{\pi/2}(x, y, t_0) - \varphi_{\pi/2}(x, y, 0)\right] \quad (9)$$

These displacements correspond to tool motion within every frame. The above procedures are reiterated for every consecutive frame to get time series data of pixel displacements. For detailed implementation, please see [10].

3.4. Digital Image Correlation

DIC estimates motion by finding the maximum correlation between pixel subsets in the image of interest and the reference image [14]. In our case, we use the first frame as the reference image for our analysis. For the subset window in our reference frame, we choose the region around the tool edge as it is a region of high contrast which makes tracking easier. To deal with the noise we also convolve every frame with a Gaussian kernel. Motion of the tool is tracked frame to frame by correlating the subset windows between the frames and our initial reference frame. Correlation between the subset in the reference frame and the frame of interest is calculated using a cross-correlation function defined as [14]:

$$\gamma(u,v) = \frac{\sum x \sum y\left[f(x.y)-\bar{f}_{x,y}\right]\cdot[g(x-u.y-v)-\bar{g}]}{\left\{\sum x \sum y\left[f(x.y)-\bar{f}_{x,y}\right]^2 \sum x \sum y[g(x-u.y-v)-\bar{g}]^2\right\}^{0.5}} \quad (10)$$

wherein f represents the intensity of pixel at point x, y summed over x and y under the window of subset in reference frame. Similarly, g represents the intensity of pixel at point u, v summed over x and y under the window of subset in frame of interest. $\bar{f}$ and $\bar{g}$ represents average of the intensity matrices f and g respectively. Changes in position of tool relative to reference image can be calculated by maximizing the cross-correlation function in Eq. (10). Detailed implementation procedures are discussed in [6].

3.5. Particle Image Velocimetry

Particle image velocimetry estimates velocity by finding intensity peaks in the cross-correlation matrix of interrogation areas in two images in a similar manner to that of DIC. Like DIC, PIV too requires a speckle pattern to be sprayed on the object to be tracked to ensure a high degree of correlation is obtainable. However, since that is not possible, the challenges associated with DIC are applicable to PIV-based motion registration too. For motion extraction we use PIVLab – which is an open-source GUI-based MATLAB tool [16]. PIVLab is robust and easy to use and provides nice inbuilt features for image pre-processing and process parameter selection as compared to using our code for DIC. In contrast to our use of a direction correlation function for DIC, with PIVLab, we use its Fourier transform correlation with multiple passes to estimate motion. In the first pass we use a larger interrogation window followed by second pass with a smaller window. We also set a high threshold for the correlation. These procedures make motion estimation robust. Additional details on the use of PIVLab are available in [16].

3.6. Deep Learning using Convolution Neural Networks

To extract small and subpixel level motion from video, we use the iterative residual refinement (IRR) scheme [19] which is an improvement over the PWC-Net scheme [20], which in turn builds on standard FlowNet scheme [21]. IRR provides a scheme based on weight sharing that can be implemented on existing optical flow models. It reduces the number of parameters and improves the accuracy. The scheme has been designed according to simple and well-established principles of encoder, warping, use of a cost volume and a decoder.

Implementation of the scheme proceeds as follows. First, a pair of images is passed into the encoder which is a 6-level convolution pyramid. At each level, filters are applied on the images for feature extraction. The filters, otherwise called kernels with size $k \times k$ form the learnable unit of the neural network. k is a hyperparameter that must be given as an input by the user while designing the model architecture. A convolution operation over a $n \times n$ sized input by a $k \times k$ sized kernel includes sliding the kernel over $k \times k$ subspace of input and performing a summation of the dot product of kernel and local subspace to give a single cell in the output as:

$$G(x,y) = \sum_{i=-k/2}^{k/2} \sum_{j=-k/2}^{k/2} W(i,j)F(x+i,y+j) \quad (11)$$

wherein G is the output matrix, F is the input image matrix consisting of entries corresponding to pixel intensities, W is the kernel matrix of size $k \times k$ and x and y are coordinates of point output in matrix G as shown schematically in Fig. 2. In our model we use kernels of size 3×3 pixels. This size along with the number of kernels to be used are predetermined.

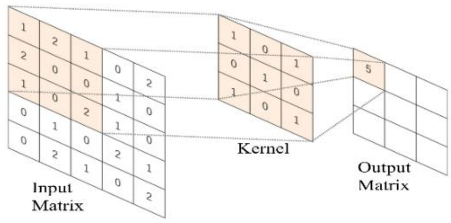

Fig. 2. Schematic of a convolution operation.

Output from the sixth level of the encoder (level L^1) is passed directly to the decoder which is also a 6-level deconvolution pyramid for optical flow estimation which is then used in next level. A schematic for these operations is shown in Fig. 3. In our implementation, the input image is of a size of 101 x 101 x 3 pixels. This is encoded to result in an image size of 4 x 4 x 196 pixels, which is again decoded to result in its original size. The optical flow thus obtained is then upsampled using bilinear interpolation. This upsampled flow becomes the current optical flow for that level to warp the features of the second image using bilinear interpolation.

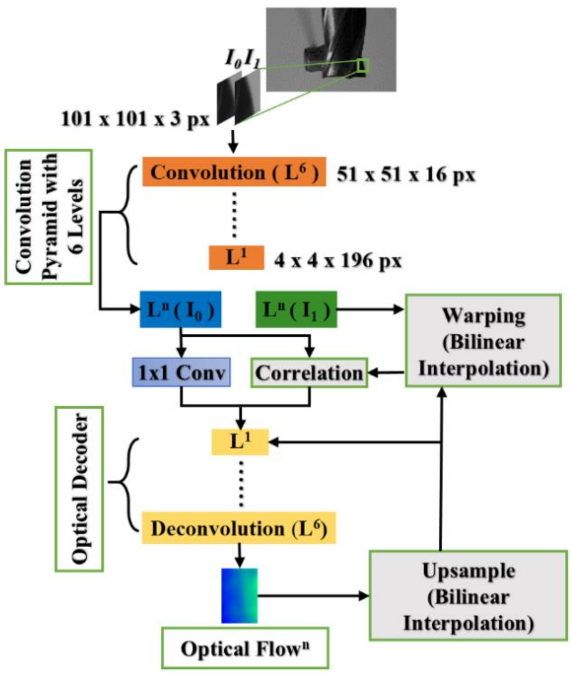

Fig. 3. CNN Architecture for estimating motion from optical flow.

The warped features from the second image combined with features of the first image are then used to construct a cost volume using a correlation function. This then becomes the input to our optical decoder along with first image and current optical flow from previous level resulting in a more refined optical flow. The original PWC-Net incrementally updated the estimation across the pyramid levels with individual decoders for each level. The IRR scheme substitutes the multiple decoders with only one shared decoder that iteratively refines the output over all the pyramid levels. A variable 1×1 convolution layer is used on the first image at each level to ensure a single input size to our common decoder. For detailed implementation, please refer to [19].

One of the drawbacks of CNNs are that they need lots of labelled data to learn relationships. Generating enough video data of tools vibrating with small motion is difficult in laboratory settings. We instead estimate small tool motion using a publicly available dataset called FlyingThings3D [19]. The dataset consists of 39000 image pairs of everyday objects flying along randomized 3D trajectories. We test on other publicly available data and against the ground truth. We use an end point error estimate to decide on the number of epochs as a stopping criterion. Besides needing large data for training, the method is expected to be robust and capable of predicting small subpixel level motion.

4. Comparative Analysis of Motion and Modal Parameters

We compare response estimated using the six different image processing schemes with twice integrated accelerations. Fig. 4 shows results for the end mill and Fig. 5 for the grooving blade. To quantify differences, we estimate a normalized root mean square error (NRMSE) between the ground truth data and the response obtained the different image processing schemes. Since these two data sets are sampled at different rates, to estimate the NRMSE, we upsample motion estimated from video to the ground truth data acquisition rate using spline interpolation.

From Fig. 4 and the NRMSE shown therein, it is evident that response obtained with the DIC method is not as well resolved as the other five methods. The PIV method fares best in this case and is closely followed by the other four methods. Though PIV uses the same principle as DIC, lower NMRSE with it is thought to be due PIVLab handling image noise better than our own implementation of the DIC method. Moreover, the results are likely also better because PIVLab automatically suggests the interrogation window size and allows us to select multiple interrogation windows in multiple passes. This is different than our implementation of DIC in which we select only one subset size in two frames to be correlated. What is furthermore evident is that for all methods, motion of the order of ±54 μm that is averaged over the first four cycles is easily resolved even when the pixel size was 36 μm. However, since there is damping in the system, the response decays and becomes of magnitudes lesser than the pixel size. And, since not all methods are good at subpixel level motion registration, the methods that are, fare better.

From Fig. 5 it is observable the CNN-based method fares best in this case followed by the edge and the PIV schemes, respectively. The DIC method fares better in this case than in the case of the end mill, and this is thought to be due the blade vibrating with larger magnitudes than the end mill, and due to the blade having different features and having been acquired with different parameters. The response obtained with the intensity-based optical flow method has the largest NRMSE, followed by the phase-based optical flow method.

These difference between results for the end mill and the blade suggest that the registered motion is sensitive to the magnitude of motion, to the image processing scheme being used, to tool features, and to image acquisition parameters. However since the edge detection scheme, the PIV scheme, and the CNN scheme give consistently low NMRSE, it may be concluded that these schemes are more robust.

Since the ultimate objective of registering motion from video is to extract modal parameters from that video, we extract parameters using the ERA, details of which are outlined in [20, 21]. Parameters extracted for both tools along with ground truths for both cases are listed in Table 1. Since we do not measure the input force, this is a case of output-only modal analysis, and we do not estimate the eigenvector and hence do not report it in Table 1.

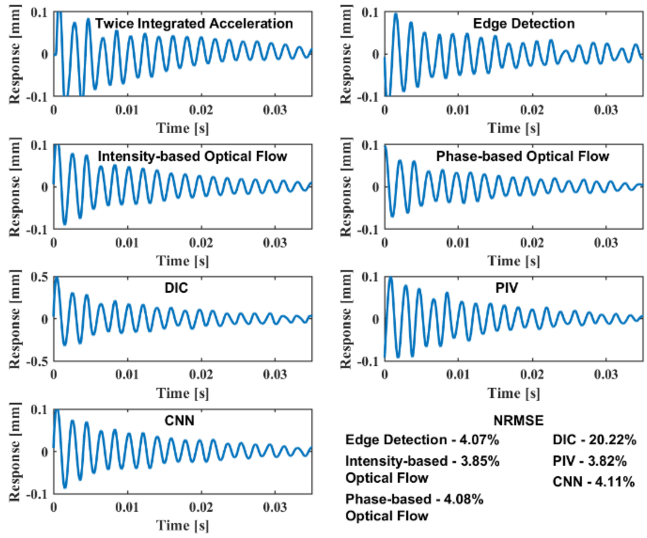

Fig. 4. End mill's motion estimated from the six image registration schemes being benchmarked with ground truth in terms of the NMRSE.

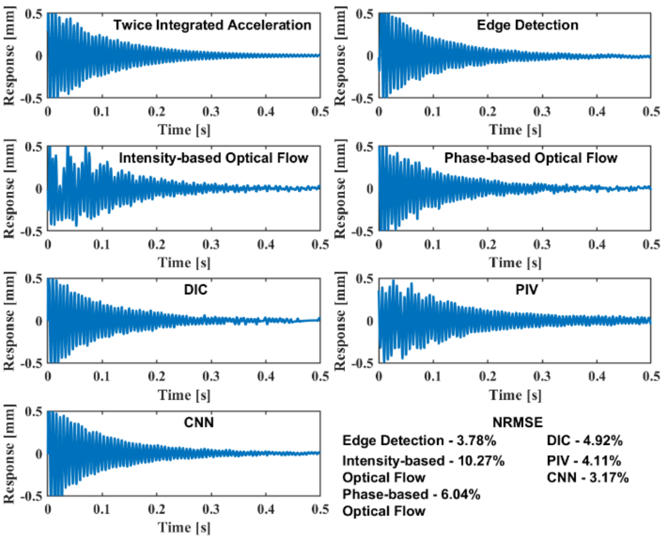

Fig. 5. Grooving blade's motion estimated from the six image registration schemes being benchmarked with ground truth in terms of the NMRSE.

From Table 1 it is evident that the vision-based schemes can estimate the natural frequencies correctly. However, there are differences in the estimated damping ratios (ζ). Since correct damping is a function of the response being captured accurately, those schemes with larger NMRSE also exhibit higher errors in damping estimates.

Table 1. Modal parameters comparison for the endmill and the blade

	Modal parameters			
Method	End mill		Grooving blade	
	f [Hz]	ζ [%]	f [Hz]	ζ [%]
Twice integrated accelerations	523	1.80	151	1.08
Edge Detection	524	1.91	150	0.98
Intensity-based Optical Flow	524	1.96	151	0.62
Phase-based Optical Flow	524	2.05	150	1.03
DIC	523	2.12	150	0.93
PIV	524	2.30	151	0.60
CNN	523	1.88	150	1.00

5. Conclusion

This paper compared the performance of six different image processing schemes for their ability to register motion of vibrating cutting tools from video. Such comparative analysis was the main and new technical contribution made in this paper. The six methods investigated included the edge detection scheme, the intensity- and phase-based optical flow schemes, the DIC scheme, and the PIV scheme, and a CNN-based scheme. Of these six schemes, the use of PIV and CNN to estimate motion is being reported for the first time in this paper. Methods were demonstrated with video of a vibrating end mill and a grooving blade. The features present on the tool itself were used to extract motion from the videos. Twice-integrated measured tool point accelerations were used to benchmark the estimated motion.

In cases involving motion larger than the pixel size, all six methods exhibited effective performance. However, when dealing with small motion, our observations indicate that the edge detection method, the PIV method, and the CNN method consistently yielded minimal errors. Conversely, the optical flow-based method and the DIC method displayed greater sensitivity to variations in tool features and image acquisition parameters. While the natural frequency estimation from different methods is accurate, it is worth noting that errors in damping estimation correlate positively with errors in motion estimation. The findings presented in this study can contribute to the informed selection of appropriate methods for vision-based modal analysis of cutting tool systems.

Acknowledgements

This work was supported by the Government of India's Science and Engineering Research Board's Core Research Grant number CRG/2020/001010.

References

[1] Feng D, Feng MQ. Computer vision for SHM of civil infrastructure: From dynamic response measurement to damage detection–A review. Engineering Structures. 2018;156:105-17.

[2] Xu Y, Brownjohn JM. Review of machine-vision based methodologies for displacement measurement in civil structures. Journal of Civil Structural Health Monitoring. 2018;8:91-110.

[3] Spencer Jr BF, Hoskere V, Narazaki Y. Advances in computer vision-based civil infrastructure inspection and monitoring. Engineering. 2019;5(2):199-222.

[4] Baqersad J, Poozesh P, Niezrecki C, Avitabile P. Photogrammetry and optical methods in structural dynamics–A review. Mechanical Systems and Signal Processing. 2017;86:17-34.

[5] Law M, Gupta P, Mukhopadhyay S. Modal analysis of machine tools using visual vibrometry and output-only methods. CIRP Annals. 2020;69(1):357-60.

[6] Gupta P, Rajput HS, Law M. Vision-based modal analysis of cutting tools. CIRP Journal of Manufacturing Science and Technology. 2021;32:91-107.

[7] Gupta P, Law M. Evaluating tool point dynamics using smartphone-based visual vibrometry. Procedia CIRP. 2021;101:250-3.

[8] Law M, Lambora R, Mukhopadhyay S. Modal parameter recovery from temporally aliased video recordings of cutting tools. CIRP Annals. 2022;71(1):329-32.

[9] Lambora R, Law M, Mukhopadyay S. Recovering cutting tool modal parameters from fractionally uncorrelated and potentially aliased signals. CIRP Journal of Manufacturing Science and Technology. 2022;38:414-26.

[10] Nuhman PA, Singh A, Lambora R, Law M. Methods to estimate subpixel level small motion from video of vibrating cutting tools. CIRP Journal of Manufacturing Science and Technology. 2022;39:175-84.

[11] Canny J. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence. 1986;679-98.

[12] Horn BK, Schunck BG. Determining optical flow. Artificial intelligence. 1981;17(1-3):185-203.

[13] Lucas BD, Kanade T. An iterative image registration technique with an application to stereo vision. 7th international joint conference on Artificial intelligence. 1981; 674-679.

[14] Sutton MA, Orteu JJ, Schreier H. Image correlation for shape, motion and deformation measurements: basic concepts, theory and applications. Springer Science & Business Media; 2009.

[15] Fleet DJ, Jepson AD. Computation of component image velocity from local phase information. International journal of computer vision. 1990;5(1):77-104.

[16] Thielicke W, Stamhuis E. PIVlab–towards user-friendly, affordable and accurate digital particle image velocimetry in MATLAB. Journal of open research software. 2014;2(1).

[17] Hur J, Roth S. Iterative residual refinement for joint optical flow and occlusion estimation. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019; pp. 5754-5763.

[18] Sun D, Yang X, Liu MY, Kautz J. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. InProceedings of the IEEE conference on computer vision and pattern recognition. 2018;pp. 8934-8943.

[19] Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Van Der Smagt P, Cremers D, Brox T. Flownet: Learning optical flow with convolutional networks. InProceedings of the IEEE international conference on computer vision. 2015; pp. 2758-2766.

[20] Juang JN, Pappa RS. An eigensystem realization algorithm for modal parameter identification and model reduction. Journal of guidance, control, and dynamics. 1985;8(5):620-7.

[21] Gupta P, Law M, Mukhopadhyay S., Evaluating tool point dynamics using output-only modal analysis with mass-change methods. CIRP Journal of Manufacturing Science and Technology. 2020;31:251-64.