In this study, we developed a real-time vibration visualization system that can estimate and display vibration distributions at all frequencies in real time through parallel implementation of subpixel digital image correlation (DIC) computations with short-time Fourier transforms on a GPU-based high-speed vision platform. To help operators intuitively monitor high-speed motion, we introduced a two-step framework of high-speed video processing to obtain vibration distributions at hundreds of hertz and video conversion processing for the visualization of vibration distribution at dozens of hertz. The proposed system can estimate the full-field vibration displacements of 1920 × 1080 images in real time at 1000 fps and display their frequency responses in the range of 0–500 Hz on a computer at dozens of frames per second by accelerating phase-only DICs for full-field displacement measurement and video conversion. The effectiveness of this system for real-time vibration monitoring and visualization was demonstrated by conducting experiments on objects vibrating at dozens or hundreds of hertz.

**Keywords:** vibration monitoring, high-speed vision, digital image correlation, short-time Fourier transform

1. Introduction

Vibration monitoring is widely used in mechanical structure analysis [1–4] for operating machines with reciprocating and rotating components (e.g., engines, bearings, gear boxes, and motors) and structural health monitoring of civil infrastructures [5–7] (e.g., bridge condition analysis [8,9] and building structure monitoring [10–12]). Real-time vibration monitoring is critical for the continuous long-term monitoring of machines throughout their service life. Such monitored vibration data during operation can help operators identify early on damaged regions in structures, thereby reducing economic losses in machinery maintenance. Real-time and long-term monitoring of dilapidated buildings and bridges can help reduce accidents and unnecessary construction costs by extending their lifetime. In most cases, vibration monitoring is conducted by measuring the displacement, velocity, and acceleration using various types of contact sensors that need to be installed on the target structures to be observed [13–15]. These contact sensors can maintain the same motion with vibrating targets and provide accurate and robust vibration signals at their installation positions. These methods have two main limitations in vibration monitoring: (1) single-point measurement – multiple sensors need to be installed for a complex-shaped structure; however, dozens or hundreds of sensors require large installation times and increase system costs; and (2) uninstallable locations – there are many uninstallable locations for sensors in the cases of high-temperature, high-voltage, or small-scale structures operating at high speed.

Vision-based methods can address the aforementioned problems effectively. Cameras can remotely receive light signals reflected from the surfaces of vibrating structures as digital images, and the displacements at many measurement points can be calculated using computer vision algorithms [16, 17]. These methods can realize non-contact and full-field measurements without interfering with the targets to be observed. However, the limited image collection speed and high computational cost associated with image processing limit the sample frequency of the vibration signals in video-based monitoring. Conventional cameras working at 30 fps can only measure vibrations at frequencies below 15 Hz. In recent years, high-frame-rate (HFR) vision systems [18–20], which can capture and process images at hundreds or thousands of hertz, have been developed for the real-time tracking and recognition of fast-moving objects. Two main challenges still limit the realization of real-time full-field vibration monitoring. (1) Displacement measurements at thousands of points: faster computer vision algorithms are required for real-time full-field vibration monitoring as well as high-frame-rate capturing. (2) Real-time, human-readable monitoring: operators cannot directly observe vibration phenomena. High-speed phenomena are replayed offline in slow motion because they are too fast for the human eye to re-
solve.

In this study, we developed a real-time vibration visualization system that can estimate and display vibration distributions at all frequencies on a computer as real-time, human-readable data. The remainder of this paper is organized as follows. Section 2 summarizes the related studies on vibration sensing, digital image correlation, and high-speed vision. In Section 3, we introduce a two-step framework for high-speed video processing for digital image correlation (DIC)-based full-field vibration displacement estimation and short-time Fourier transform (STFT)-based vibration visualization. We also describe the proposed real-time vibration visualization system that can simultaneously estimate the vibration displacements of 1920 × 1080 images at 1000 fps and show their frequency responses at all frequencies in the range of 0–500 Hz in real time, including the system configuration, implemented algorithms, and specifications. To demonstrate its effectiveness, we present in Section 4 the real-time vibration monitoring results of (1) a free vibration test of a cymbal and (2) a forced vibration test of a steel box that vibrates at a wide range of frequencies.

2. Related Studies

2.1. Vibration Sensing

Current vibration sensing methods can be divided into (1) contact-sensor-based and (2) non-contact-sensor-based approaches. In contact-sensor-based methods, vibration sensors, such as strain gauges, velocimeters, and accelerometers, are directly installed on vibrating objects to be observed. Strain gauges [5, 13, 21] installed on vibrating objects capture strain signals caused by vibrations, which can then be easily converted into vibration displacements. Velocimeters [1, 14] output velocity signals that are linearly proportional to the velocities of target objects, based on the principle of electromagnetic induction. Accelerometers are the most popular sensors for contact sensor-based measurements [15, 22, 23]. Acceleration values become more sensitive in higher frequency ranges, and accelerometers are suitable for monitoring the vibrations of fast-vibrating objects with high precision and sensitivity. In recent years, microelectromechanical system accelerometers [24–26] have become popular for structural health monitoring. Most of these accelerometers are inexpensive and designed for real-time long-distance operations that utilize wireless sensor networks. These contact-type sensors can provide accurate and robust vibration signals; however, their installation is generally time-consuming, and their maintenance costs are high.

Non-contact-sensor-based methods are mainly based on optical sensors. Laser Doppler vibrometers [27–29] can obtain vibration signals by calculating the Doppler shift of a laser beam reflected from a target vibrating object. They are highly accurate and sensitive for repeatable measurements in the frequency range of 0–300 kHz [5]. However, they are easily affected by speckle noise and involve cumbersome intermittent measurements. Vision-based solutions are popular for vibration measurements because of their easy installation and non-contact monitoring. Depth cameras, including infrared cameras [16, 30] and binocular cameras [17, 31] directly measure the actual distances between cameras and vibrating objects and convert their time-changing distance into vibration amplitudes. Typical cameras do not rely on distance calculation; instead, image displacements between adjacent frames are computed using various computer vision algorithms, such as template matching [32, 33], optical flow [34, 35], and digital image correlation [36, 37]. Video-based solutions require considerable memory for data storage and time-intensive data processing. Moreover, their frame rates are often limited to tens of frames per second, and most of them provide vibration monitoring in a low frequency range within tens of hertz.

High-speed vision systems [18–20] have been developed to measure vibrations in a high frequency range. Compared with standard video formats at low frame rates, high-speed vision systems can execute video processing for HFR images in real time at a high frame rate of hundreds or thousands of frames per second, enabling vibration measurements at a frequency of 500 Hz or higher. Real-time vibration monitoring using HFR images is limited by two challenges. (1) Simultaneous video processing in a small frame interval: frame intervals in HFR images are small, and acceleration of video processing is required for real-time execution during the frame interval. However, most current video analyses with computer vision algorithms require significantly longer computation times than their video-capturing times. (2) Real-time observation for operators: standard video rates of dozens of frames per second are designed for human eyes, whereas HFR images captured at hundreds or thousands of frames per second cannot be observed by human eyes without playing them in slow motion offline. Displaying HFR images in real time on a computer is not intuitive for operators to understand in detail the time-varying status of fast-vibrating objects.

2.2. Digital Image Correlation

DIC [38] is a well-known image-based measurement technique that can precisely estimate deformation displacements in images as a full-field distribution by calculating the similarities between digital images before and after deformation. Owing to its easy experimental setup and effective measurement, DIC has been widely applied in the field of experimental solid mechanics to quantitatively analyze the deformation displacements of materials [39], components [40], and structures [41]. DIC was proposed in 1980s to track the motion of a small aluminum specimen [42]. Many studies have been conducted to improve the performance of DIC by focusing on better similarity estimation criteria, such as sum of absolute differences [43], sum of squared difference [44], cross-correlation [45], and zero-mean normalized cross-correlation [46]. To improve the computational efficiency of DIC, Sutton et al. [47] proposed employing
the Newton–Raphson (NR) method with differential correlation to accelerate normalized cross correlation, and Chen [48] used a fast Fourier transform red (FFT) to estimate the similarity in the frequency domain and detected their peaks to determine displacements without pixel-by-pixel searching.

To improve the measurement accuracy of DIC, many methods have been proposed for subpixel image registration, such as iterative space-domain cross-correlation [49], gradient-based subpixel registration [50], and genetic algorithms [51]. Pan [46] demonstrated that the iterative NR algorithm achieves higher accuracy and better stability, and proposed the inverse-compositional Gauss–Newton algorithm [52] for more efficient subpixel registration. To improve the subpixel interpolation accuracy, Luu [53] employed B-spline interpolation with a family of recursive interpolation schemes, and many optimized versions of B-spline-based interpolation [54, 55] have been introduced for precise DIC computation. Several studies have aimed to reduce interpolation errors by introducing pre-processing techniques such as the random subset offset strategy [56], self-correlation scheme [57], and Gaussian pre-filtering [58]. However, most DIC algorithms cannot realize real-time visualization of full-field vibration displacements. This is because they focus on the accuracy of short-term videos, which are not designed for long-term and real-time applications. Heavy computation in DIC remains a challenging limitation in the development of real-time and long-term vibration monitoring.

2.3. High-Speed Vision

With the rapid development of computer and electronic device technologies, numerous high-speed vision systems have been developed and applied to various high-speed motion applications that cannot be directly observed by the naked eye, such as object tracking [59, 60], biomedical [61], robotic control [62], high-dynamic-range image capturing [63], and 3D measurement [64]. Wantanabe et al. [65] developed a high-speed vision system that can extract the moment features of 1024 objects in a 256 × 256 image at 955 fps using a parallel-implemented labelling algorithm on an field-programmable gate array (FPGA). To improve the allowed complexities of algorithms implemented on high-speed vision systems, Ishii et al. [59] developed a personal-computer (PC)-based high-speed vision system that can transfer HFR images and their processed image features in real time by organically linking three data-processing platforms: FPGA, CPU, and GPU. For real-time video processing at a high frame rate, various types of computer vision algorithms have been implemented on this PC-based high-speed vision system such as face tracking [66], multiboid object tracking [67], color-histogram-based tracking [68], and optical flow estimation [69].

Recently, high-speed vision systems have been used for vibration sensing. Considering an image sensor as a set of numerous optical sensors in which every pixel can measure time-varying brightness as a signal for time-series analysis, an HFR camera, in which HFR images can be used to observe human-invisible vibrations at the audio-frequency level, can provide full-field vibration signals sampled at its frame rate. Jiang et al. [70] achieved robust tracking of vibrating objects by executing pixel-level digital filters for HFR images and performed real-time tracking by processing 512 × 512 images at 1000 fps [71]. Shimasaki et al. [72] reported the pixel-level localization of flying honeybees with wing flapping at 180–240 Hz and estimated their trajectories by calculating the frequency responses of brightness signals at all pixels of 1024 × 1024 images at 500 fps. Similarly, HFR video-based tracking was conducted for flying multicopters with propellers at dozens of rotations per second by performing pixel-level STFTs in real time at 500 fps [73].

3. Real-Time Vibration Visualization System

3.1. Concept

Most high-speed vision systems are designed for real-time sensing of high-speed scenes, and they function as software vibration sensors to output scalar image features in real time. However, captured HFR images can only be displayed in slow motion to operators on an offline computer display; thus there is a demand for real-time vibration visualization of HFR images as intuitive full-field images for long-term vibration monitoring.

To solve this problem, we introduce a two-step framework for real-time vibration visualization that involves HFR video processing to convert human-visible images at dozens of frames per second from invisible HFR images in parallel with full-field vibration displacement measurements. Fig. 1 shows our framework for real-time vibration visualization. When input images of $M \times N$ pixels (frame number $k$) are obtained at time $k \tau$, denoted by $I(x, y, k\tau)$, the process flow is described below. $\tau$ and $f_0 = 1/\tau$ are the frame cycle time and frame rate, respectively.

(1) Full-field vibration displacement estimation

Input images of $M \times N$ pixels are converted to displacement images $A(i, j, k\tau)$ of $M' \times N'$ pixels:

$$A(i, j, k\tau) = \text{Displacement}(I(x, y, k\tau)) \ (k = 1, 2, \ldots), \quad (1)$$

where the $M'N'$ displacement sensors are virtually located on the input images, and they are operated at the same sampling time as the camera cycle time $\tau$. $A(i, j, k\tau) = (A_x(i, j, k\tau), A_y(i, j, k\tau))$ is composed of x- and y-displacement images with $M' \times N'$ pixels. Considering the computing resources for real-time execution, their resolutions are downconverted from $M \times N$ pixels.

(2) Conversion to temporal frequency response images

By performing STFTs of the displacement signals with $K$ frames at all pixels in the displacement images $A_{\tau}(i, j, t + k\tau)$ ($K' = 0, \ldots, K - 1$), the temporal frequency response (TFR) images are computed as follows:
Fig. 1. Two-step framework of real-time vibration visualization using HFR images.

\[
F(i,j,t) = (F_0(i,j,t), \ldots, F_{K-1}(i,j,t))
\]

\[
= \text{STFT}(A_d(i,j,t), \ldots, A_d(i,j,t+(K-1)\tau))
\]

(2)

where \(K\) determines the frequency resolution in STFT as \(\Delta f = 1/(K\tau)\). \(F_0(x,y,t)\) indicates the frequency-component image at a frequency of \(f_k' = k'\Delta f\) \((k' = 0, \ldots, K - 1)\). The displacement image to be processed \(A_d(i,j,t)\) is selected from the component images of \(A(i,j,t)\). The STFTs are performed at time \(t = lT\) \((l = 1, 2, \ldots)\), and \(T\) indicates the interval of the STFT computation. This corresponds to the frame-rate conversion for vibration visualization on a computer display. Owing to the symmetric frequency distribution in the FFT results, \(K/2\) TFR images corresponding to \(K/2\) frequency components are concatenated to generate a single image that presents all frequency information.

These TFR images, which are computed from the displacement images estimated at the camera frame rate, can indicate the vibration status as single-frame-based features. In this study, we computed the TFR images at all \(K/2\) frequency bands, and simultaneously visualized them so that they could be observed by the human eye at an interval \(T\) of tens of milliseconds, which is relatively longer than the camera cycle time \(\tau\).

3.2. GPU-Based High-Speed Vision Platform

For real-time vibration visualization, we used a GPU-based high-speed vision platform that can capture and process HFR images and display temporal frequency response images for full-field vibration displacements on a computer. It consisted of a high-speed CMOS camera (EoSens 2.0CXP2, Mikronot, Unterschleissheim, Germany) with a CoaXPress CXP-12 frame grabber (Coaxlink Quad CXP-12, Euresys, Seraing, Belgium) for HFR video capturing and a personal computer (PC) for HFR video processing accelerated by a GPU board.

The camera head had a 1920 × 1080-pixel CMOS image sensor (19.2 × 10.8 mm²), with a pixel size of 10 × 10 μm². It could capture 8-bit gray images of 1920 × 1080 pixels at 2220 fps and transfer them to a PC using a CXP-12 frame grabber. We used a PC with the following specifications: ASUSTek WS C422 PRO/SE main board, Intel Core CPU i9-11700K @ 3.60 GHz, 10 cores, 128-GB memory, and Windows 10 Professional 64-bit OS (Microsoft, Redmond, WA, US). To accelerate HFR video processing, a GPU board (GeForce RTX 3090, NVIDIA, Santa Clara, CA, US) was installed on the PC.

3.3. Implemented Algorithm

To execute a two-step framework for vibration visualization in real time at a high frame rate, we implemented a parallelized algorithm for DIC-based full-field displacement measurement and STFT-based vibration visualization by considering parallel grid management in GPU-based high-speed vision.

3.3.1. Parallel Grid Management on GPU

To parallelize the DIC computation for full-field displacement measurement, we assign grid management for creating multiple blocks to be processed in parallel on the GPU-based high-speed vision platform. Given input images \(I(x,y,t)\) of \(M \times N\) pixels, the four parameters for grid management were block size \((b_x, b_y)\) and the block step \((s_x, s_y)\). They determined the accuracy and density of the full-field displacement measurements. The block and step sizes determined the \(M' \times N'\) resolution of the full-field displacement measurement as follows:

\[
\begin{align*}
N' & = \left\lfloor \frac{N - b_x + s_x}{s_x} \right\rfloor \\
M' & = \left\lfloor \frac{M - b_y + s_y}{s_y} \right\rfloor
\end{align*}
\]

(3)

where \(\lfloor x \rfloor\) denotes the largest integer that is less than or equal to \(x\). The following \(M'N'\) sub-images \(I_{ij}(x,y,t)\) of \(b_x \times b_y\) pixels are parallel-processed on the GPU for acceleration:

\[
I_{ij}(x,y,t) = I(x + is_x, y + js_y, t) \\
(i = 0, \ldots, M' - 1, \quad j = 0, \ldots, N' - 1).
\]

(4)

3.3.2. DIC-Based Displacement Measurement

For full-field displacement measurements, we parallelized the phase-only correlation (POC) [74] algorithm, which can utilize efficient peak detection in the frequency domain for subpixel image displacements. The displacements between the two sub-images \(I_{ij}(x,y,t)\) and \(I_{ij}(x,y,t')\) were calculated as follows. In the following steps, \(t\) and \(t'\) indicate the times at the current frame and the reference frame in the DIC computation, respectively.
(1) 2-D FFTs of the input and reference sub-images

The sub-images of $b_x \times b_y$ pixels were converted into the frequency domain as follows:

$$G_{ij}(u,v,t) = \mathcal{F} \{ I_{ij}(x,y,t) \cdot Q(x,y) \}, \quad \ldots \quad (5)$$

$$G_{ij}(u,v,t_R) = \mathcal{F} \{ I_{ij}(x,y,t_R) \cdot Q(x,y) \}, \quad \ldots \quad (6)$$

where $\mathcal{F} \cdot$ indicates the 2-D FFT function. $Q(x,y)$ is a binary mask image that indicates the pixels to be processed in the DIC computation.

(2) Cross-power spectrum computation

The phase correlation distribution in the frequency domain was computed using the following cross-power spectrum:

$$R_{ij}(u,v,t) = \frac{G_{ij}(u,v,t) \cdot G_{ij}^* (u,v,t_R)}{|G_{ij}(u,v,t) \cdot G_{ij}^* (u,v,t_R)|}, \quad \ldots \quad (7)$$

where $G_{ij}^*$ is the complex conjugate of $G_{ij}$.

(3) Inverse 2-D FFT for peak detection

The cross-power spectrum $R_{ij}(u,v,t)$ was transformed into the spatial domain, as follows:

$$r_{ij}(x,y,t) = \mathcal{F}^{-1} \{ R_{ij}(u,v,t) \}, \quad \ldots \quad \ldots \quad \ldots \quad (8)$$

where $\mathcal{F}^{-1} \cdot$ indicates the inverse 2-D FFT function.

The displacement vector $\Delta \mathbf{x}_{ij}(t) = (\Delta x_{ij}(t), \Delta y_{ij}(t))$ at time $t$ was obtained with integer pixel precision by determining the maximum peak value of $r_{ij}(x,y,t)$, as follows:

$$\Delta \mathbf{x}_{ij}(t) = (\Delta x_{ij}(t), \Delta y_{ij}(t)) = \arg \max_{x,y} r_{ij}(x,y,t). \quad (9)$$

(4) Estimation of the subpixel displacement

To estimate the displacement vector with higher accuracy, subpixel-level peak detection was conducted by computing the weighted centroid values of the correlation values $r_{ij}(x,y,t)$ in the neighborhood $N(\Delta \mathbf{x}_{ij}(t))$ as follows:

$$\Delta \mathbf{x}_{ij}(t) = \frac{\sum_{(x,y) \in N(\Delta \mathbf{x}_{ij}(t))} x \cdot r_{ij}(x,y,t)}{\sum_{(x,y) \in N(\Delta \mathbf{x}_{ij}(t))} r_{ij}(x,y,t)}, \quad \ldots \quad \ldots \quad \ldots \quad (10)$$

$$\Delta \mathbf{y}_{ij}(t) = \frac{\sum_{(x,y) \in N(\Delta \mathbf{x}_{ij}(t))} y \cdot r_{ij}(x,y,t)}{\sum_{(x,y) \in N(\Delta \mathbf{x}_{ij}(t))} r_{ij}(x,y,t)}, \quad \ldots \quad \ldots \quad \ldots \quad (11)$$

where the displacement vector $\Delta \mathbf{x}_{ij}(t) = (\Delta x_{ij}(t), \Delta y_{ij}(t))$ corresponds to the displacement image of $M' \times N'$ pixels in the DIC-based displacement measurement. The neighborhood $N(x)$ indicates the $P \times P$-pixel neighborhood of $x$, where $P$ is an odd positive integer. $A_d(i,j,t)$ to be processed in the STFT-based vibration visualization was set to the $x$- or $y$-displacement component based on the vibration characteristics:

$$A_d(i,j,t) = \Delta x_{ij}(t) \quad \text{or} \quad \Delta y_{ij}(t). \quad \ldots \quad \ldots \quad \ldots \quad (12)$$

Subprocesses (1)–(4) were simultaneously executed for $M'N'$ blocks in parallel on a GPU at time $t = kt$ every time an input image of $M \times N$ pixels was captured.

3.3.3. STFT-Based Vibration Visualization

As described in Eq. (2) in Subsection 3.1, full-field displacement images of $M' \times N'$ pixels at $K$ consecutive frames, $A_d(i,j,kt) \ (k = 0, \ldots, K - 1)$ were simultaneously processed using STFTs, and their TFR images $F(i,j,t) = (F_0(i,j,t), \ldots, F_{K-1}(i,j,t))$ were obtained with a frequency resolution of $\Delta f = 1/(K \tau)$ as the vibration visualization results. These operations were executed at intervals of $T$, which is relatively longer than the camera cycle time, $\tau$. The displacement images computed in the DIC-based displacement measurement were transferred from the GPU to the CPU memory at intervals of $\tau$, and the calculated displacement results at consecutive $K$ frames were transferred from the CPU memory to the GPU for STFT computation. Finally, the TFR images processed on the GPU were transferred back to the CPU for vibration visualization every time the STFT computation starts at intervals of $T$.

3.4. Execution Time

To verify the acceleration of the video-based vibration visualization when implementing the algorithm described in the previous subsection, we evaluated the execution times for (1) DIC-based displacement measurement and (2) STFT-based vibration visualization on the GPU-based high-speed vision platform. We accelerated the algorithm by implementing it on a GPU board (GeForce RTX 3090) with the C++ language and CUDA Toolkit 11.4 using Microsoft Visual Studio Community 2017.

Table 1 lists the execution time of the DIC-based displacement measurement for 8-bit 1920 × 1080 images ($M = 1920$, $N = 1080$) when the block size $b = \{b_x = b_y\}$ and step $s = \{s_x = s_y\}$ were set to be $b = 64$, 128, and 256, and $s = b, b/2, \text{and} b/4$, respectively. The $M' \times N'$ resolution in the full-field displacement measurement was determined using Eq. (3). The DIC computation was accelerated by parallelizing it to use the global memory on the GPU for $M' \times N'$ threads, corresponding to the block operation of $b \times b$ pixels. The image and processed data transfer times between the CPU and GPU are not negligible, and the execution time listed in Table 1 involved the time required to transfer (a) two 8-bit unsigned-char input images of 1920 × 1080 pixels from the CPU to the GPU and (b) the full-field 32-bit-floating point displacement vectors of $M' \times N'$ blocks from the GPU to the CPU. The execution time of the DIC computation was similar when the block size was equal to the block step: 0.720 ms ($b = s = 64$), 0.947 ms ($b = s = 128$), and 1.203 ms ($b = s = 256$). When the block size was larger than the block step, the execution time increased: 8.725 ms ($s = 32$), 2.677 ms ($s = 64$), and 0.947 ms ($s = 128$), with $b = 128$.

Table 2 lists the execution times with $K = 128, 256, 512$, and 1024 for 32-bit-floating point $M' \times N'$ images when the
Table 1. Execution time of DIC-based displacement measurement (unit: ms).

<table>
<thead>
<tr>
<th>Block size / block step</th>
<th>64 / 64</th>
<th>64 / 32</th>
<th>128 / 128</th>
<th>128 / 64</th>
<th>128 / 32</th>
<th>256 / 256</th>
<th>256 / 128</th>
<th>256 / 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy from CPU to GPU</td>
<td>0.180</td>
<td>0.180</td>
<td>0.180</td>
<td>0.180</td>
<td>0.180</td>
<td>0.180</td>
<td>0.180</td>
<td>0.180</td>
</tr>
<tr>
<td>(1) 2-D FFT</td>
<td>0.232</td>
<td>0.555</td>
<td>0.361</td>
<td>1.270</td>
<td>4.446</td>
<td>0.441</td>
<td>1.848</td>
<td>6.380</td>
</tr>
<tr>
<td>(2) Cross power spectrum</td>
<td>0.060</td>
<td>0.226</td>
<td>0.059</td>
<td>0.207</td>
<td>0.793</td>
<td>0.056</td>
<td>0.187</td>
<td>0.650</td>
</tr>
<tr>
<td>(3) Inverse 2-D FFT</td>
<td>0.199</td>
<td>0.590</td>
<td>0.304</td>
<td>0.957</td>
<td>3.235</td>
<td>0.457</td>
<td>2.026</td>
<td>6.392</td>
</tr>
<tr>
<td>(4) Subpixel estimation</td>
<td>0.009</td>
<td>0.010</td>
<td>0.008</td>
<td>0.009</td>
<td>0.009</td>
<td>0.008</td>
<td>0.008</td>
<td>0.008</td>
</tr>
<tr>
<td>Copy from GPU to CPU</td>
<td>0.040</td>
<td>0.038</td>
<td>0.035</td>
<td>0.054</td>
<td>0.062</td>
<td>0.061</td>
<td>0.051</td>
<td>0.069</td>
</tr>
<tr>
<td>Total time</td>
<td>0.720</td>
<td>1.599</td>
<td>0.947</td>
<td>2.677</td>
<td>8.725</td>
<td>1.203</td>
<td>4.300</td>
<td>13.679</td>
</tr>
</tbody>
</table>

Table 2. Execution time of STFT-based vibration visualization when the block size \( b = 64 \) (units: ms).

<table>
<thead>
<tr>
<th>( K = 128 )</th>
<th>( K = 256 )</th>
<th>( K = 512 )</th>
<th>( K = 1024 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s = 32 )</td>
<td>0.750</td>
<td>0.800</td>
<td>1.459</td>
</tr>
<tr>
<td>( s = 64 )</td>
<td>0.265</td>
<td>0.571</td>
<td>0.725</td>
</tr>
<tr>
<td>( s = 128 )</td>
<td>0.171</td>
<td>0.174</td>
<td>0.343</td>
</tr>
<tr>
<td>( s = 256 )</td>
<td>0.121</td>
<td>0.123</td>
<td>0.154</td>
</tr>
</tbody>
</table>

block steps \( s \) were 32, 64, 128, and 256. The execution time involved the data transfer time for the displacement vectors of \( M' \times N' \) blocks at \( K \) consecutive frames, and \( K/2 \) computed 8-bit-unsigned-char TFR images of \( M' \times N' \) pixels between the CPU and GPU, at the assumed block size \( b = 64 \). The STFT computation was accelerated by parallelizing it to use global memory on the GPU for the \( M' \times N' \) threads, corresponding to \( M'N' \) one-dimensional STFTs. The execution time of the STFT computation increased as \( K \) increased, and decreased as \( s \) increased. The execution times were 0.75 ms (\( s = 32 \)) and 0.121 ms (\( s = 256 \)) when \( K = 128 \), and 3.162 ms (\( s = 32 \)) and 0.193 ms (\( s = 256 \)) when \( K = 1024 \). These values were relatively smaller than the execution time of the DIC computation.

In parallel with the STFT-based vibration visualization, the DIC-based displacement measurement can be simultaneously operated without significantly increasing its execution time by implementing it as multithreaded processes on a GPU-based high-speed vision platform. This is because the STFT computation does not need to be executed frequently for vibration visualization to the human eye, compared with the DIC computation at the camera frame rate \( \tau \). We confirmed that the 1920 \( \times \) 1080 input images were captured and DIC-processed in real time at 1000 fps (\( \tau = 1 \) ms), and their STFT-based full-field vibration was visualized with \( K = 128, 256, \) and 512 at 31.25 fps (\( T = 32 \) ms) when the block and step sizes were set to the same value, such as \( b = s = 64 \) or 128.

4. Experiments

4.1. Free Vibration of a Cymbal

To verify the effectiveness of our vibration visualization system, we show the experimental results of a metal cymbal with a diameter of 45.7 cm freely vibrated after knocking it with a wooden stick. Fig. 2 shows (a) a photo of the experimental setup and (b) a photo of the cymbal to be observed. The cymbal was vertically installed on a metal stand at a distance of 4.0 m from the camera head with an 85 mm lens. The cymbal was painted in a black and white speckled pattern. A total of 1920 \( \times \) 1080 images (\( M = 1920, N = 1080 \)) were captured and processed at 1000 fps (\( \tau = 1 \) ms).
block and step sizes in the DIC computation were set to \( b_x = b_y = 64 \) and \( s_x = s_y = 64 \), respectively. Corresponding to the \( 1920 \times 1080 \) image, full-field displacement vectors of \( 30 \times 16 \) blocks \((M' = 30, N' = 16)\) were computed using DIC in real time at 1000 fps. In the experiment, the DIC computation between \( I_{ij}(x,y,t) \) and \( I_{ij}(x,y,t-\tau) \), which were the sub-images at the current and previous frames \((\tau = t - \tau)\), respectively, was executed to obtain full-field velocity vectors. A mask image \( Q(x,y) \) was set to cover only the cymbal region in the images. The vertical velocity components were processed using STFTs with \( K = 256 \) at intervals of \( T = 32 \) ms; 128 TFR images in the range of 0–500 Hz with a frequency resolution of 3.9 Hz were displayed on a computer for vibration visualization in real time at 31.25 fps.

Figure 3 shows the estimated velocity vectors at intervals of 0.8 s for \( t = 0.0–3.2 \) s. The cymbal was knocked by a wooden stick at \( t = 0.6 \) s, and the velocity vectors were magnified so that a 1-pixel-length corresponded to \( 2.0 \times 10^{-3} \) pixel/s. Figure 4 shows (a) the 17 measurement points on the cymbal, and (b) the vertical velocity components for \( t = 1.0–2.0 \) s. Figure 5 shows their frequency amplitudes in the range of 0–500 Hz. As shown in Fig. 4(b), the vertical velocity components at \( p_1, p_2, \ldots, p_8 \) around the edge of the cymbal vibrated more strongly than those at \( p_9, p_{10}, \ldots, p_{16} \) located on the inner circle, and those at \( p_{17} \) around the center of the cymbal vibrated weakly with the smallest amplitude. Corresponding to the natural vibration of the cymbal, nonspatially uniform velocity vectors were time-varying, as shown in Fig. 4(b), whereas the peak frequencies in the frequency amplitudes were similar at the measurement points shown in Fig. 5.

The data shown in Fig. 6 were obtained for the vibration visualization data to be monitored on a computer display in real time at 31.25 fps. Figure 6 shows (a) an image map of the 128 TFR images in the range of 0–500 Hz with their average frequency amplitude at \( t = 1.1 \) s, (b) TFR images at the peak frequencies of 42.9, 89.8, 148.4, and 210.9 Hz at \( t = 1.1 \) s, (c) the vertical velocity components at \( p_1 \) for \( t = 0.0–30.0 \) s, and (d) the spectrogram of the average frequency amplitude for \( t = 0.0–30.0 \) s. The colormaps in (b) and (d) indicate the frequency amplitudes. These peak frequencies correspond to the natural frequencies of the cymbal. The amplitudes of the TFR images correspond to various mode shapes. As shown in Fig. 6(d), frequency amplitudes at these peak frequencies were observed after knocking the cymbal at \( t = 0.6 \) s, whereas the frequency amplitude remained constant for a longer time as the peak frequency decreased. This is because the damping ratios of the vibration components in the low-frequency range were larger than those in the high-frequency range.

4.2. Forced Vibration of a Steel Box

Next, we show the experimental results of a steel box forcibly vibrated on a vibration-testing machine. Figure 7 shows (a) the photo of experimental setting and (b) the photo of steel box to be observed. A vibration-testing machine (D-Master APD-200FCD, Asahi Seisakusyo, Hino, Japan) excited the steel box in the horizontal direction...
by manually adjusting its sweeping frequency using a sine wave in the range of 5–200 Hz for 272 s. A steel box (35 × 24 × 24 cm) was painted in a black and white speckled pattern and fixed on the vibration testing machine using a belt. A camera head with an 85 mm lens was installed 3.5 m in front of the steel box so that its 35 × 24 cm² front surface could be observed in the images. With this installation, a 1920 × 1080 image and a pixel corresponded to 19.2 × 10.8 mm² and 10 × 10 μm², respectively. In the experiment, 1920 × 1080 images were captured and processed at 500 fps (τ = 2 ms), and the block and step sizes in the DIC computation were set to \( b_x = b_y = 128 \) and \( s_x = s_y = 128 \), respectively. With the DIC computation between the sub-images at the current and previous frames, full-field velocity vectors of 15 × 8 blocks (\( M' = 15, N' = 8 \)) were computed in real time at 500 fps. A mask image \( Q(x,y) \) was set to cover only the region of the steel box in the image. The horizontal velocity components were processed using STFTs with \( K = 512 \) at intervals of \( T = 50 \) ms. A total of 256 TFR images in the range of 0–250 Hz with a frequency resolution of 0.98 Hz were displayed in real time at 20 fps.

Figure 8 shows the estimated velocity vectors at \( t = 20.0, 50.0, 100.0, \) and \( 200.0 \) s. The velocity vectors were magnified so that a 1-pixel-length corresponded to \( 2.0 \times 10^{-3} \) pixel/s. Fig. 9 shows the (a) 12 measured points on the steel box and (b) horizontal velocity components for \( t = 30.0–31.0 \) s, when the metal box was vibrating at one of its resonant frequency of 28.3 Hz. Fig. 10 shows their frequency amplitudes in the range of 0–250 Hz. This shows that the metal box did not vi-

---

**Fig. 5.** Frequency amplitudes at 17 measurement points on the cymbal.
Real-Time Vibration Visualization Using High-Speed Vision

Fig. 7. Experimental setup for a steel box with forced vibration.

Fig. 8. Estimated velocities in a steel-box experiment.

Fig. 9. Estimated velocities at 12 measurement points on a steel box vibrating at 28.3 Hz.

brate uniformly under the resonant frequency. Although all points were vibrating at the same peak frequency of 28.3 Hz, the amplitudes of $p_1, p_2, p_3$ distributed on the top region were larger than those of $p_{10}, p_{11}, p_{12}$ distributed on the bottom region.

Figure 11 shows the real-time visualized vibration data at three resonant frequencies of 28.3 Hz ($t = 32.3$ s), 45.9 Hz ($t = 54.5$ s), and 137.7 Hz ($t = 177.1$ s). The figure shows the (a) image maps of the 256 TFR images in the range of 0–250 Hz at the three moments and their average frequency amplitudes, (b) horizontal components at measurement point $p_2$ for $t = 0.0–272.0$ s, and (c) spectrogram of the average frequency amplitude at $t = 0.0–272.0$ s. When the sweep frequency increased over time, the amplitude of the horizontal component varied and the peak frequency increased, as illustrated in Figs. 11(a) and (b). At the resonant frequencies when the steel box vibrated with a loud sound, the amplitudes of the TFR images at these frequencies corresponded to the various mode shapes shown in Fig. 11(a). The amplitudes were strong around its upper, middle-lower, and upper-edge sections in the 28.3 Hz-, 45.9 Hz-, and 137.7 Hz-TFR images, respectively.

We confirmed that all data in Fig. 11, as shown in Fig. 6, were monitored in real time on a computer display, and the resonant frequencies of the steel box and its mode-shape-like velocity amplitudes were intuitively visualized as vibration-feature data by using an image map of all frequency amplitudes. This can provide timely and effective information for structural analysis and mechanical defect detection to operators.
Fig. 10. Frequency responses at 12 measurement points on a steel box vibrating at 28.3 Hz.

5. Conclusion

In this study, we realized real-time HFR-video-based vibration visualization in which operators can intuitively see the frequency responses of full-field vibration displacements at all frequencies as visible data at tens of frames per second on a computer display. Using parallel-accelerating DIC computations with STFTs on a GPU-based high-speed vision platform, full-field vibration displacements of $1920 \times 1080$ images captured at 1000 fps were simultaneously visualized as TFR images in the range of 0–500 Hz. To verify its effectiveness, real-time monitoring experiments were conducted using a (1) free vibrating cymbal and (2) forced vibrating steel box. These experimental results demonstrate that our vibration visualization system can estimate full-field vibration responses at different levels of frequencies, which are extremely fast for human eyes to see. And operators can directly observe them as visible image data by displaying them at 31.25 fps on a computer.

This study focused on the real-time visualization of full-field vibration and utilized local information for DIC calculation, with the accuracy depending heavily on the texture information of measured targets. For future work, we plan to improve the proposed system to generate more accurate velocity fields by merging the results obtained from multiple DIC calculations. Moreover, we aim to assess more practical applications of the system, such as bridge vibration monitoring and aided analysis for structural designs.

Fig. 11. Vibration visualization in steel-box experiment.
References:


Wang, F. et al.


Real-Time Vibration Visualization Using High-Speed Vision

Name: Kohei Shimasaki
Affiliation: Specially Appointed Assistant Professor, Digital Monozukuri (Manufacturing), Education, and Research Center, Hiroshima University
Address: 3-10-32 Kagamiyama, Higashi-hiroshima, Hiroshima 739-0046, Japan
Brief Biographical History: 2016 Received B.E. degree from Hiroshima University 2018 Received M.E. degree from Hiroshima University 2019- Specially Appointed Assistant Professor, Digital Monozukuri (Manufacturing), Education, and Research Center, Hiroshima University 2020 Received Ph.D. degree from Hiroshima University
Membership in Academic Societies: • The Iron and Steel Institute of Japan (ISIJ) • Institute of Electrical and Electronics Engineers (IEEE) Robotics and Automation Society (RAS) • The Japan Society of Mechanical Engineers (JSME)

Name: Idaku Ishii
Affiliation: Smart Robotics Laboratory, Graduate School of Advanced Science and Engineering, Hiroshima University
Address: 1-4-1 Kagamiyama, Higashi-hiroshima, Hiroshima 739-8527, Japan
Brief Biographical History: 1992 Received B.E. degree from The University of Tokyo 1994 Received M.E. degree from The University of Tokyo 1996-2000 Assistant Professor, The University of Tokyo 2000 Received Ph.D. degree from The University of Tokyo 2000-2002 Lecturer, Tokyo University of Agriculture and Technology 2002-2003 Associate Professor, Tokyo University of Agriculture and Technology 2003-2007 Associate Professor, Hiroshima University 2007- Professor, Hiroshima University
Main Works: • Hyper human robotics • Vision and sensing • Mechatronics and manipulation • Hyper human applications
Membership in Academic Societies: • Institute of Electrical and Electronics Engineers (IEEE) • The Robotics Society of Japan (RSJ) • The Japan Society of Mechanical Engineers (JSME) • The Society of Instruments and Control Engineers (SICE)