Development of a machine vision system for recording experimental data in the study of torsional stiffness by dynamic mechanical analysis

Development of a machine vision system for recording experimental data in the study of torsional stiffness by dynamic mechanical analysis

A S Gryaznov[1], S S Prugov[1], V A Plotnikov[2]

[1] Department of Technical Disciplines, Altai State Pedagogical University, 55, Molodezhnaya str., Barnaul, 656031, Russia
[2] Department of General and Experimental Physics, Altai State University, 61 Lenina ave., Barnaul, 656049, Russia

E-mail: gryaznov-as@yandex.ru, plotnikov@phys.asu.ru

Abstract. A machine vision system has been developed for recording data obtained on an experimental installation for studying relaxation transitions of polymers, as well as studying the dynamic shear modulus during phase transformations in alloys based on titanium nickelide. The computer vision system is based on simple algorithms implemented in the Java language in the IDE "Processing". The conditions for the experiment and the procedure for processing the experimental data obtained on the basis of the machine vision system are determined.

1. Introduction into technique measurements

The method of dynamic mechanical analysis (DMA) is well known and is a rather sensitive tool for studying relaxation transitions in polymers [1, 2]. The method can be successfully applied to study phase transitions in intermetallic compounds based on TiNi [3]. In the research installation of the laboratory of polymer physics at AltSPU, the DMA method is used to determine the dynamic stiffness G' (or dynamic shear modulus - C'), as well as the tangent of the angle of mechanical losses tgγ. Research can be carried out in a wide temperature range from -100 ° C to 350 ° C. The diagram of installation is shown in Figure 1.

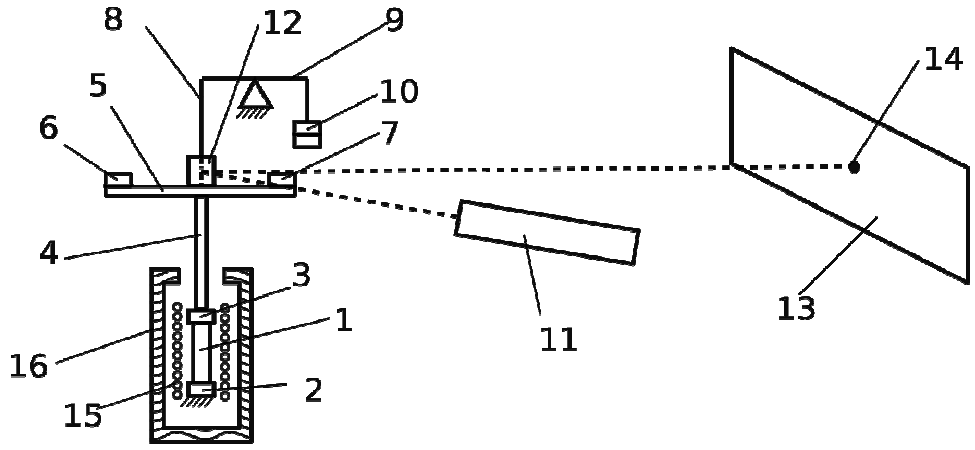

Figure 1. The schema of the experimental installation

Sample 1 (Fig. 1) is fixed with a clamp 2 to a fixed base. The upper end of the sample is fixed with a movable clamp 3 connected to the bar of the inertial part 5 with a steel rod 4. The moment of inertia is determined by the position of the weights 6 and 7 located on the bar of the inertial part. The entire system (3, 4, 5) is suspended on a torsion bar 8 (steel wire or plate with a high stiffness value) and balanced over the shoulder by 9 weights 10. The inertial part is set in motion by electromagnets and then the whole system performs damped oscillations for registration of which uses a laser 11 with a frequency of the red spectrum of radiation, located near the installation. The laser beam is reflected from the mirror 12 and is projected onto a remote white metal panel 13, on which a bright spot 14 is reflected. The sample itself is heated by an oven 15 in a heat chamber 16.
Earlier, to register mechanical vibrations of a pendulum, positional registration systems (Fig. 2) were developed, representing rulers with several sensors located on them.

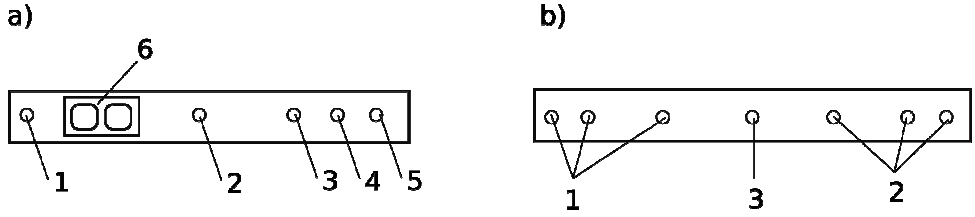

Figure 2. Optical registration systems: a - autonomous optical ruler, b - automated optical ruler.

In the operation of the first optical registration system (Fig. 2-a), sensors 1 and 3 are used, located relative to the middle sensor 4 at distances of 20 and 2 cm, respectively. Sensor 1 is designed to reset the count, which is displayed on indicator 6. Sensors 2 and 5 are additional and determine the direction of the beam. This arrangement of the sensors makes it possible to determine the logarithmic damping decrement based on the measured number of oscillations and time. With this method of measurements, it is necessary each time to check the location of the laser spot to the right near sensor 4, so that the amplitude ratio between sensors 1-4 and 3-4 is kept as 1:10. A more detailed description of the method of studying relaxation transitions in polymers can be found in the source [4].

Another measurement method is based on the following development (Fig. 2-b), in which groups of sensors are installed on the ruler - 1 and 2 located at a certain distance from the middle position 3. This ruler is developed on the basis of the AVR microcontroller and has an interface for matching with a personal computer. During the movement of the beam, the data of the position of the sensor on the ruler and the time of its actuation are transmitted to the computer. These data are approximated by the function of damping (1), the parameters of which (period T and damping coefficient) are calculated: G' (or C') and tgγ. In this case, the laser spot can be displaced from the midpoint (as a result of the change in the geometry of the sample under the influence of heat) by a certain amount without affecting the

measurements. Compared to the first, this measurement method is better suited for automating the operation of the installation:

$$x = x_0 + A \cdot \exp\left(-\frac{t}{\alpha}\right) \cdot \cos\left(\frac{2\pi t}{T} - \varphi_0\right). \quad (1)$$

However, both methods of recording experimental data have a drawback - correct measurements can be carried out at the number of vibrations n> 4, which in some cases is critical for the study, for example, of polymers based on wood materials at high temperatures. The way out of the situation may be the possibility of using a measuring ruler with a large number of sensors. The most convenient and universal way to solve this goal turned out to be the possibility of developing a registration system based on computer vision, the implementation of which is described below.

As a toolkit for developing a machine vision system, the IDE Processing was chosen, which allows you to develop a sketch program in the Java language. The resulting program can run on devices with a Java virtual machine (Windows, Linux, Mac OS), including the Android OS.

In this case, the system was tested on a personal computer with an AMD Ryzen 3700x processor (16 cores, base frequency - 3.6) and 16 GB of RAM under the Windows 10 operating system. A webcam was used as a capture device - Logitech C525 HD (1.2 MPix, resolution 1280x720, maximum frame rate - 30 Hz, autofocus). As a development tool, the IDE Processing of the current version 3.5.4 was downloaded from the official site [5].

To implement machine vision, the OpenCV library is usually used, which, among other things, can be interfaced with Processing [6]. However, in this case, simple programming techniques in the Java language (Processing) will be demonstrated.

2. Methodological aspects of computer program development

Capturing images from a camera in the Processing environment was carried out with a resolution of 640x360 through the **cam** object of the *Capture* class, which becomes available after installing the Video library and connecting it directly:

import **processing.video. *.**

The registration program interface (Figure 3) displays only a part of the captured area with a height defined by the constant *int* **H** = 50; and offset relative to the top - the global variable **y**; to do this, we use the **img** variable of the *PImage* class:

img.copy(**cam**, 0, **y**, **cam.width**, **H**, 0, 0, **width**, **H**).

The output of a part of the image placed in the **img** variable can be done using the functions: **image** or **set**. In the process of developing the program, it turned out that in unidentified cases, using the **image** function leads to the program freezing, and replacing it with set practically eliminates this problem, so we used:

set (**img**, 0, 0).

In the infinite loop function - **draw** (), the offset y is determined automatically when the *boolean* variable **ct** is **set** (see Listing 1).

Listing 1. Automatic laser spot search.

```
void draw() {
...
   if (ct) {
     int r=0;
     for (int yy=0; yy<height; yy++) {
       for (int x=0; x<width; x++) {
         int rj=round(red(cam.get(x, yy)));
         if (rj>r) {
           r=rj;
           y=yy – H/2;
         }
       } // end of for at x
     } // end of for at yy
     ct=false;
   }
…
} // end of draw()
```

Listing 2. Adjustment of the offset for the scanning area of the laser beam spot.

```
void mouseWheel(MouseEvent event) {
  y+=event.getCount();
}
```

```
void keyPressed() {
...
  if (keyCode==65) ct=true; // 'A'
  if (keyCode==38) y--; // Up
  if (keyCode==40) y++; // Down
…
}
```

Automatically finding the spot of the laser beam (Listing 1) scans the red component of the pixels in the captured image. In this case, the coordinate of the pixel with the brightest red component is written for the **y** variable. The spot of the laser beam (Fig. 3-2) should be in the middle of the image strip (Fig. 3-1) and fall into the processing area (Fig. 3-3), the half-width of which is determined by the constant *int* **D** = 5.

During the experiment, the offset **y** can be adjusted using the mouse wheel, keyboard arrows, or autosearch can be started with the "A" button by setting the value of the boolean variable **ct** = true (see Listing 2).

In the main processing loop (Listing 3), for each vertical line of image 1 at area 3 (Figure 3), the maximum value is calculated for the red component rm and is drawn from the bottom in the form of a column in red. This is how image 4 is formed. The column value is also written for array **L**. Then the array **L** is searched for values of the array equal to **rm** and the left and right boundaries of the laser spot are calculated - **lb** and **rb**, respectively. Borders **lb** and **rb** are drawn in the program interface in the form of area 5. Then the coordinate of the average spot value is calculated - **m**. The value of the spot width 6 at rest **W** is memorized once, which follows from the equality (**m** == **M**, where **M** is the previous value of the spot center coordinate).

Figure 4 shows a picture of a moving spot of a laser beam under conditions when artificial lighting was turned off in the laboratory. At the same time, the contrast of the image increases and the red spot passes much higher than the level in the area of minimum illumination. The glare from the fluorescent lamp also disappeared - 10 (Fig. 3). On the other hand, the frame rate per second dropped to 11, that is, more than doubled. And also the width of the moving spot has increased. These factors are negative in the experiment and therefore the best result can be obtained by choosing the correct artificial illumination of the video capture area.

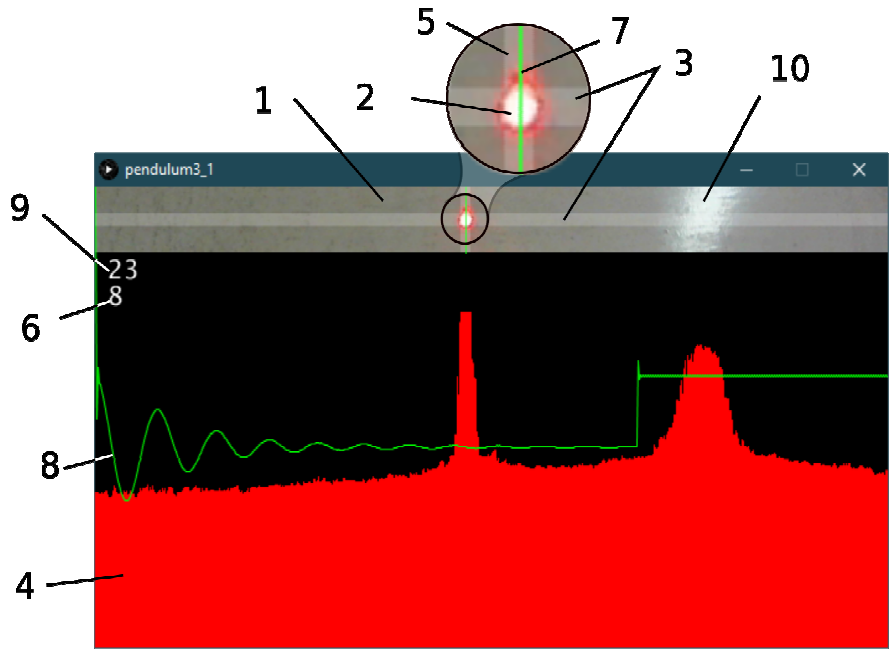

Figure 3. The interface of the program for recognizing the movement of the beam: 1 - video capture strip, 2 - laser beam reflection spot, 3 - processing strip, 4 - columns of the red component in the processing area, 5 - strip of the determined spot width, 6 - saved value of the laser spot width, 7 - determined center of the spot, 8 - dependence of the deviation of the spot coordinate from the frame number, 9 - frame rate per second, 10 - glare from the illumination lamp.

Listing 3. Main image processing loop.

```
void draw(){
...
   stroke(255, 0, 0); // set the color of line
   int r, rm=0;
   for (int i=0; i<width; i++) {
     r=0;
     for (int j=-D; j<D; j++) {
       int rj=round(red(get(i, H/2+j)));
       if (rj>r) r=rj;
     }
     L[i]=r;
     if (r>rm) rm=r;
// paint column of height r
     line(i, height, i, height-r);
   } // end of for at i
   int lb=0, rb=0;
   for (int i=0; i<width; i++) {
     if (L[i]==rm) {
       rb=i;
       if (lb==0) lb=i;
     }
   }
   int m=(rb+lb)/2;
   if ( (W==0) && (M==m) )
     W = (rb-lb);
   int x=lb+W/2;
   if (M<m) x=rb-W/2;
   M=m;
…
} // end of draw()
```

Figure 4 clearly demonstrates the blurring of the width of the reflection spot of the laser beam during registration, which directly depends on the speed of its movement even in the presence of good illumination. Therefore, the coordinate of the median spot value **m** cannot be taken for registration, since it will depend on the boundaries **lb** and **rb**. The true value of the coordinate of the center of the laser beam spot is at the spot front without taking into account its half-width at rest - **W** / 2. The direction of the front of the laser beam is determined by comparing the coordinates **m** and its previous value **M**.

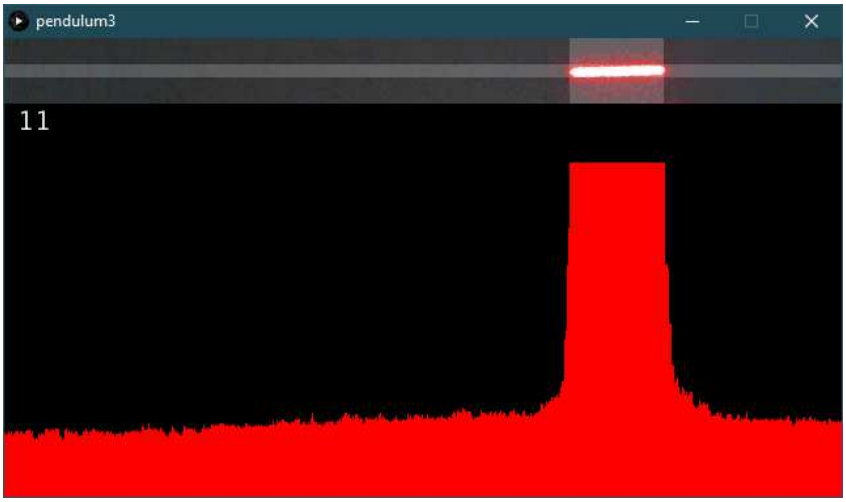

Figure 4. The process of image processing of a moving beam spot in the absence of artificial lighting.

Listing 4. Experimental data logging function.

```
void mesure(int x, int rm) {
// start recording
  if (rm<255) {
    for (int i=0; i<NM; i++)
      my[i]=0;
    mc=0; ly=0; ry=0;
    ms = true;
    st=millis();
  }
// point registration
  if (ms&&(mc<NM)) {
    my[mc]=x;
    mt[mc]=millis()-st;
    if (x>ry) ry=x;
    if (x<ly) ly=x;
    mc++;
  }
… // draw results
} // end mesure()
```

The true value of the coordinate of the moving laser beam is calculated (with the above) and is written to the variable **x** (see Listing 3). The **x** coordinate of the laser beam is displayed in the program interface by line 7 (Fig. 3).

The experiment data is registered in the **measure**() function (see Listing 4) by writing the **x** coordinate into the **my** array and the time of the experiment (from the moment of its beginning - **st**) into the **mt** array.

The minimum and maximum values of **ry** and **ly** are also calculated to scale the plot 8 in Figure 3 in dependence from registration of the point number.

3. Analysis of the results

Subsequently, the data were transferred to spreadsheets and approximated by the function of equation (1). Initial data and approximating dependence are shown in Figure 5.

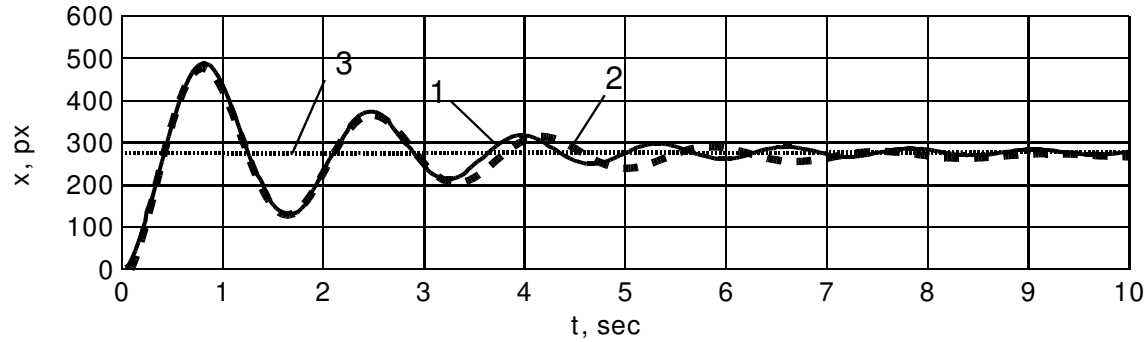

Figure 5. Dependence of the amplitude on time - 1, approximation - 2, resting line - 3.

Figure 5 shows the discrepancy between the approximation and the original data. Such a situation is always observed experimentally and is especially characteristic for small values of **n** (**n** <5). The systems of data registration presented in Figure 2 are not informative for these discrepancies and the situation is observed as a discrepancy between the period measured at the beginning of the experiment, in its middle and at the end. In this case, some averaged period is taken as the truth. The use of machine vision has made it possible to get a clearer picture. Based on the obtained dependence **x**(**t**), it is possible to select the points of intersection with the average value of $\mathbf{x_0}$ and select the amplitudes of the extrema **Amax** and plot these dependencies with time (Figure 6).

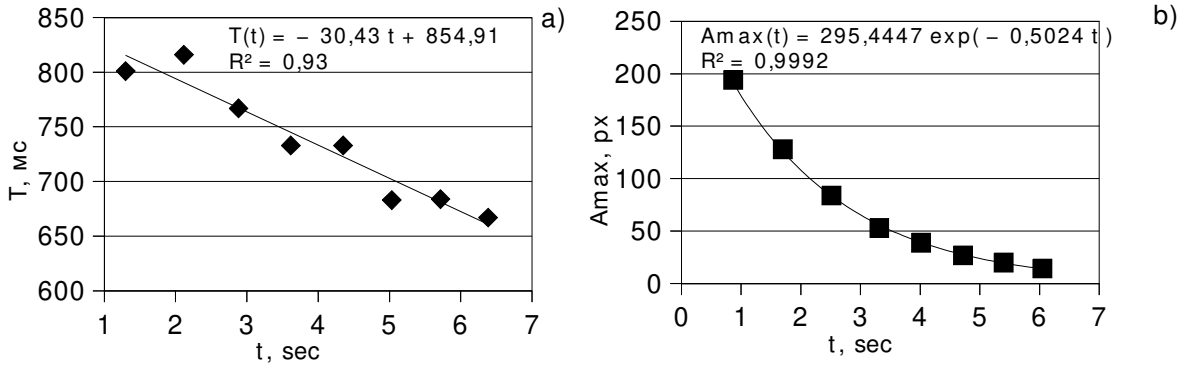

Figure 6. Dependence of the oscillation period T - (a) and the amplitude Amax - (b) on the experiment time t.

The approximation of the **T**(**t**) graph with a high probability ($\mathbf{R^2}$ = 0.93) indicates linear regression in which the period decreases by 30 ms every second. Therefore, in this case, it is advisable to choose $\mathbf{T_0}$ as a constant value of the period based on regression analysis from equation: **T**(**t**) = - **k** * **t** + $\mathbf{T_0}$. The dependence **Amax**(**t**) is a strict exponential ($\mathbf{R^2}$ = 0.9992), regardless of whether we take into account the period compression or not.

4. Conclusions

Thus, a sketch program of machine vision in the Java language in the Processing environment has been developed, which allows obtaining the original dependence of the amplitude of oscillations of a torsion pendulum on time. It was found that for small oscillations, the function of the dependence of the amplitude on the oscillation time differs from the classical function (1). The data obtained make it possible to calculate with high accuracy the period and the dampening exponent of oscillations at small oscillations of the sample-torsion system, which could not be resolved by the registration

systems used earlier. It can be assumed that the developed machine vision system will poorly cope with damped oscillations with a short period of oscillations due to the inertia of the response of the recording video system (strongly smeared spot), for example, in the study of intermetallic compounds based on titanium nickelide and registration systems based on optical rulers will be more effective. It can be assumed that this machine vision system can be made universal when using video cameras with a higher frame rate per second processing, for example, 120 fps.

References

[1] Malkin A Ya, Askadsky A A and Kovriga V.V. 1978 *Methods for measuring the mechanical properties of polymers* (Moscow: Chemistry) p 336 (in Russian).
[2] Perepechko I I 1978 *Introduction to the physics of polymers* (Moscow: Chemistry) p 312 (in Russian).
[3] Plotnikov V A and Gryaznov A S 2002 *Low-frequency method of studying the phase transformation in alloys of type* $Ti_{50}Ni_{40}CU_{10}$ (Barnaul: Composite-02) (in Russian).
[4] Betenkov F M, Gryaznov A S, Nasonov A D, Novichikhina T I 2015 *Laboratory work on polymer physics: workshop* (Barnaul: AltGPU) p 40 (in Russian).
[5] Processing – URL: https://processing.org/
[6] Chung B WC 2017 *Pro Processing for Images and Computer Vision with OpenCV* / (APress. Paperback, eBook) p 301. ISBN: 978-1-4842-2774-9.