Journal of Engineering Sciences, Assiut University, Vol. 39, No 3, pp.581 -605, May 2011

DEVELOPMENT OF COMPUTER VISION ALGORITHMS FOR MEASUREMENT AND INSPECTION OF EXTERNAL SCREW THREADS

E.S. Gadelmawla

Associate Professor, Mechanical Engineering Department, College of Engineering, Qassim University, Buridah 25415, Qassim, Saudi Arabia

(Received April 6, 2011 Accepted May 9, 2011)

The current methods of measuring screw threads are either time consuming or expensive. In addition, no single measurement method is available and capable of accurately measuring all screw thread features while significantly reducing the measurement time. This paper introduces a vision system for automatic measurement and inspection of most types of screw threads. Many image processing and computer vision algorithms have been developed to analyze the captured images of screw threads and perform the measurement and inspection processes. Most of the common screw thread features (18 features) could be measured by the introduced vision system. The system has been calibrated for both imperial and metric units and was verified by measuring a standard ISO metric thread plug gage and comparing the results of the measurements with the standard values. The results showed that the maximum difference between the standard and measured values was ±5.4 μm, which provide a good accuracy for measurement.

KEYWORDS: *Screw threads, Thread measurements, Thread inspection, Computer vision, Image processing.*

1. INTRODUCTION

Threaded pipe joints are commonly used in various industries such as petroleum, transport, mining, etc [1]. In addition, screw threads probably are among the most important of all machine elements [2]. Due to the basic design of a screw thread, which involves helical geometry, screw thread technology is complex [3]. Methods of measuring threads are as varied as the threads themselves. These methods include profile projectors, measuring microscopes, 3-wire units, screw thread micrometers and V-groove micrometers. On the other hand, methods of thread inspection range from lasers to fixed and variable limit gauging.

There are about 30 separate geometrical features and dimensional characteristics in the design and construction of screw threads [4]. The most rigorous standard in the United States inspects 11 major thread characteristics [5]. In addition, recent researches [6, 7] explored the effect of thread dimensional nonconformity on fastener performance and it was reported that dimensional nonconformity of 150% out of tolerance for pitch diameter yielded a 4% reduction in tensile strength. Therefore,

* *On leave from Prod. Eng. and Mech. Design Dept., Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt.*

the measurement and inspection of screw threads are very important to ensure the required degree of accuracy and conformity [8, 9].

Due to the importance of measuring screw thread features, research in the field of measuring screw thread is still underway. For example, Laczik [10] introduced a derivation of geometrical features of the screw surfaces with the exact mathematical forms. Fujun et al. [11] presented a method, which used optical techniques and image processing techniques to measure the physical dimension of tube thread and give further analysis. Hongyan and Guoxiong [12] introduced an online contact measuring system that can automatically detect large screw threads by using the neotype gauge head of threads. They used spectrum analysis to analyze the dynamic error and to obtain the data of the functional diameter. They reported that the neotype gauge head of thread is capable to get the error and the relative changes of the functional diameter of the screw thread. Tong et al. [13] introduced a system for automatic measurement of screw thread parameter based on the theory of laser measurement. The inspection and estimation of the screw thread contour were done using position sensitive device (PSD) to measure the coordinate data of the screw thread contour and using precise raster to measure the axial displacement of the precision worktable. Gadelmawla [14] developed a non-contact measurement system using a measurescope, personal computer and developed software, for measuring and inspecting most of the common types of screw threads.

The current methods of measuring screw threads are either time consuming or expensive. In addition, no single measurement method is available and capable of accurately measuring all screw features while significantly reducing the measurement time. Therefore, the measurement and inspection of screw threads has been emphasized by many researchers. On the other hand, vision systems have been recently used in many applications [15-18]. In addition, vision systems have been developed for quality control and started to be used as an objective measurement and evaluation systems [19].

The aim of this work is to develop a non contact measuring system, based on the computer vision techniques, for measuring and inspecting the most common features of most types of screw threads.

2. THE PROPOSED VISION SYSTEM

The proposed vision system comprises two main parts, hardware and a developed software. The hardware is used to capture images for screw threads to be measured and the software is used to analyze the captured images and perform the measurement and inspection processes.

System Setup

As shown in figure 1, the hardware consists of three items. The first item is the backlighting table (1), which is a lighting box with diffusing surface at its front, and it is used to produce a back lighting for the thread to be measured (2). The second item is a CCD color video camera (3) and a set of lenses with different focal lengths. The camera is carried by a camera holder (4) and it is connected to a PC computer (5) through a USB connection. Capturing software is provided with the CCD camera to acquire images with different resolutions up to 4.2 Mp (2048*2048 pixels).

The developed software (6), named MISTVision (Measurement and Inspection of Screw Threads by Vision), is fully written in-house using Microsoft Visual C++ as a 32-bit Windows application. It features many image processing and computer vision algorithms to measure and inspect screw thread features from captured images. Fig. 2 shows the main interface of the *MISTVision* software.

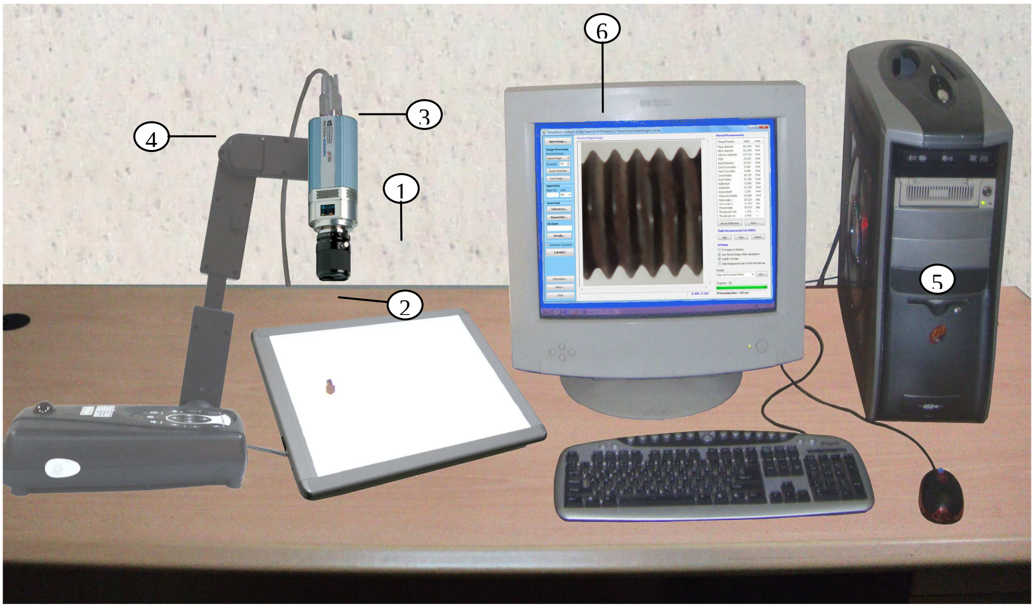


Fig 1: Photograph of the proposed vision system

2.1 Capturing Images

Before capturing images for screw threads to be measured, the vision system was adjusted to capture images that satisfy the following criteria:

1- The screw thread to be measured is set on the backlighting table so that its axis appears horizontally in the captured image as shown in Fig. 2.

2- The CDD camera is adjusted so that the image of the screw thread appears as maximum as possible in the capturing software window.

3- At least, three complete threads should appear in the captured image for both the upper and lower profiles of the screw thread.

4- The left and right sides of the captured image should be opened, i.e., there is no background at the left or right sides of the image.

5- The size of the captured images will affect the accuracy of measurement, as it will be discussed in section 5-3; therefore, the vision system was adjusted to capture images of size 2048*2048 pixels to obtain good accuracy (5 μm/pixel).

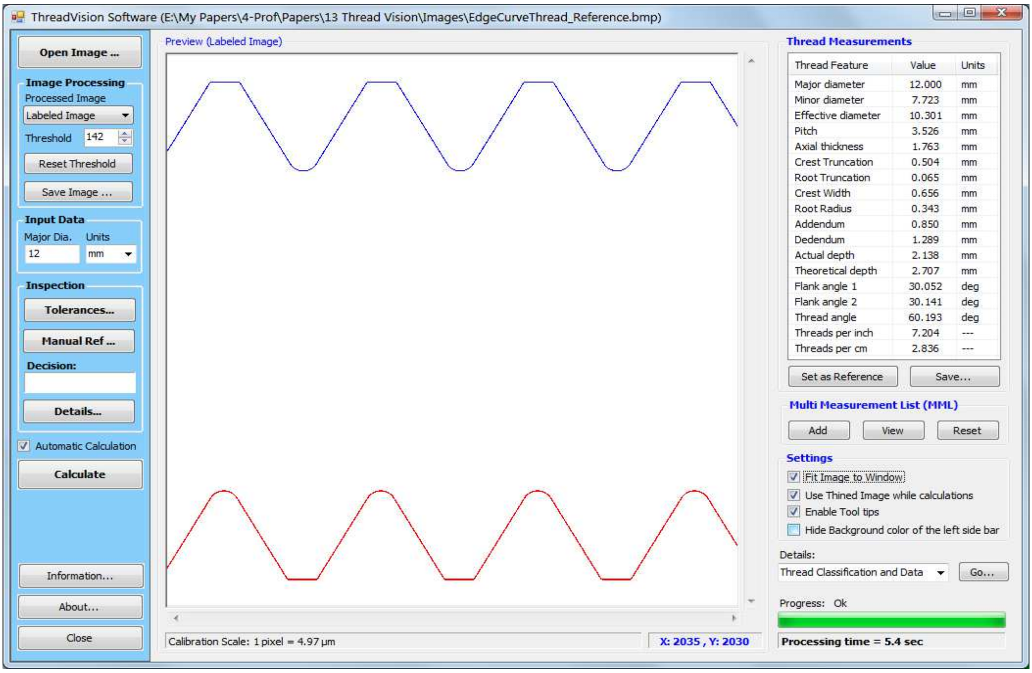


Fig. 2 The main interface of the MISTVision software

3. THE DEVELOPED ALGORITHMS

Several image processing and computer vision algorithms are applied to the captured image to perform the measurement and inspection processes. Figure 3 shows the procedures of applying these algorithms and the following sections explain them.

3.1 Image Processing Algorithms

The aim of the image processing algorithms is to extract the coordinates of the pixels that constitute the upper and lower profiles of the screw threads. These coordinates will be utilized by developed computer vision algorithms to measure the screw thread features. The image processing algorithms applied here include image segmentation, edge pixels detection (EPD), edge pixels thinning (EPT), edge pixel labeling (EPL), and edge pixels arrangement (EPA). The first three algorithms were explained in a previous work [20]. The next subsections describe both the edge pixel labeling (EPL) and edge pixels arrangement (EPA) algorithms.

3.1.1 Edge Pixels Labeling

The edge pixels labeling (EPL) algorithm creates a labeled image from the thinned image, created by the edge pixels thinning algorithm, to distinguish between the upper and lower profiles of the screw thread. In the labeled image, all pixels of the upper profile are assigned a unique color and all pixels of the lower profile are assigned a different color. The EPL algorithm works as follows:

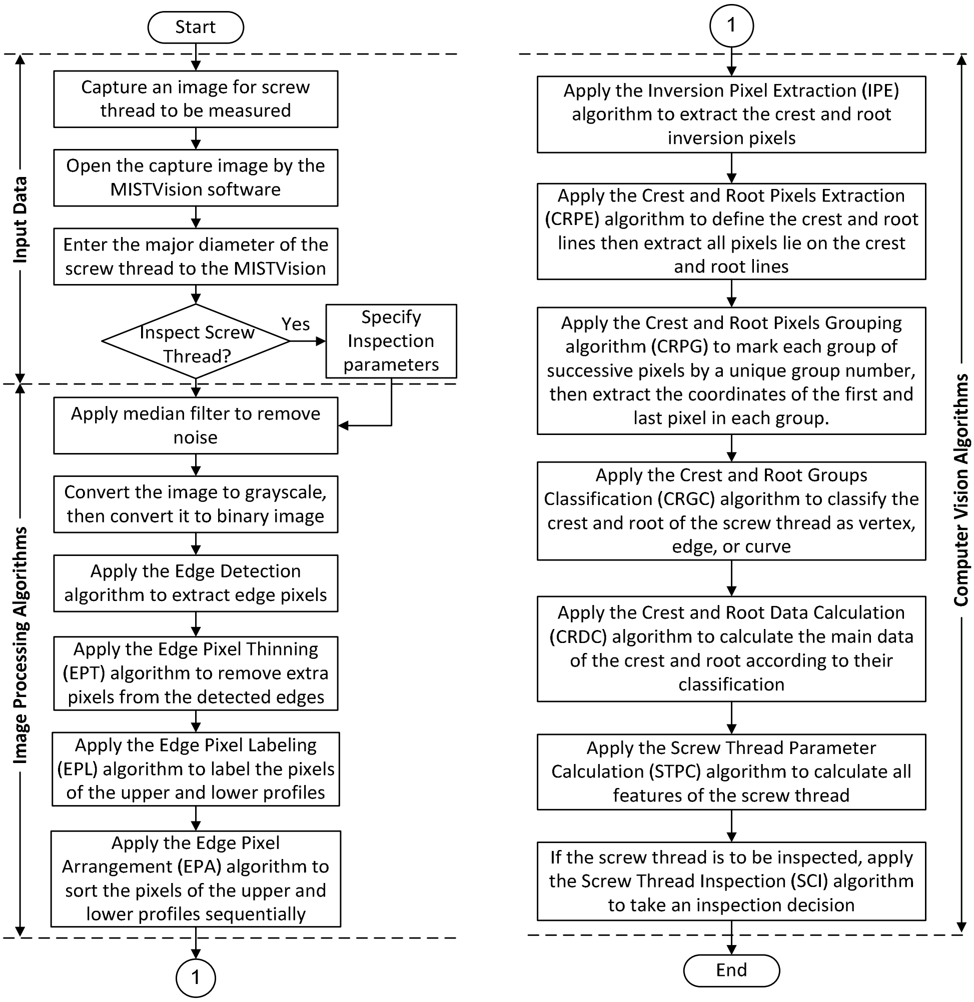


Fig. 3 Procedures of the measurement and inspection process by MISTVision software

1- Search the *y* coordinates of all pixels in the thinned image to find the minimum *y* coordinates (y_{min}). This value, of course, will be for a pixel lies on the upper profile of the screw thread, provided that the positive y coordinate of the images is going from top to bottom.
2. Similarly, search the *y* coordinates of all pixels in the thinned image to find the maximum y coordinate (y_{max}), which will be for a pixel lies on the lower profile of the screw thread.
3- Calculate the intermediate value of y coordinate (y_{mid}), which lies between the minimum and the maximum *y* coordinates, as follows:

$$y_{\mathrm{mid}} = y_{\mathrm{min}} + ((y_{\mathrm{max}} - y_{\mathrm{min}}) / 2) \tag{1}$$

4- Mark all pixels that have *y* coordinates greater than y_{mid} as upper profile pixels and mark all pixels that have *y* coordinates less that y_{mid} as lower profile pixels.

After applying the EPL algorithm, the coordinates of the upper profile pixels are extracted and stored in 2-D array in the form of *UpperEdgePixels(x, y)*. Similarly, the coordinates of the lower profile pixels are extracted and stored in the form of *LowerEdgePixels(x, y)*.

3.1.2 Edge Pixels Arrangement

The pixels stored in both the *UpperEdgePixels* and *LowerEdgePixels* arrays, which were labeled by the EPL algorithm, are usually not arranged due to the scanning process of the edge detection algorithm discussed in [20]. To deal with these two arrays easily, the pixels in each array should be arranged sequentially according to their positions in the thread profiles. Hence, the Edge Pixel Arrangement (EPA) algorithm is designed to perform this process. For example, to arrange the pixels in the *UpperEdgePixels* array the algorithm works as follows:

1- Search all pixels in the *UpperEdgePixels* array to find the pixel that has the minimum x coordinate and mark it as the first pixel $P_1(x_1, y_1)$ in the array.

2- Use equation 2 to measure the distances between the first pixel and all remaining pixels in the *UpperEdgePixels* array.

$$d = \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2} \tag{2}$$

Where: x_1, y_1 are the coordinates of the first pixel and x_n, y_n are the coordinates of pixel number n in the *UpperEdgePixels* array.

3- Mark the pixel that has the minimum distance as the second pixel in the *UpperEdgePixels* array.

4- To find the third pixel, repeat steps 2 and 3 to measure the distances between the second pixel and the remaining pixels, then mark the pixel that has the minimum distance as the third pixel.

5- Repeat step 4 to arrange all remaining pixels according to their minimum distances from the previous pixel.

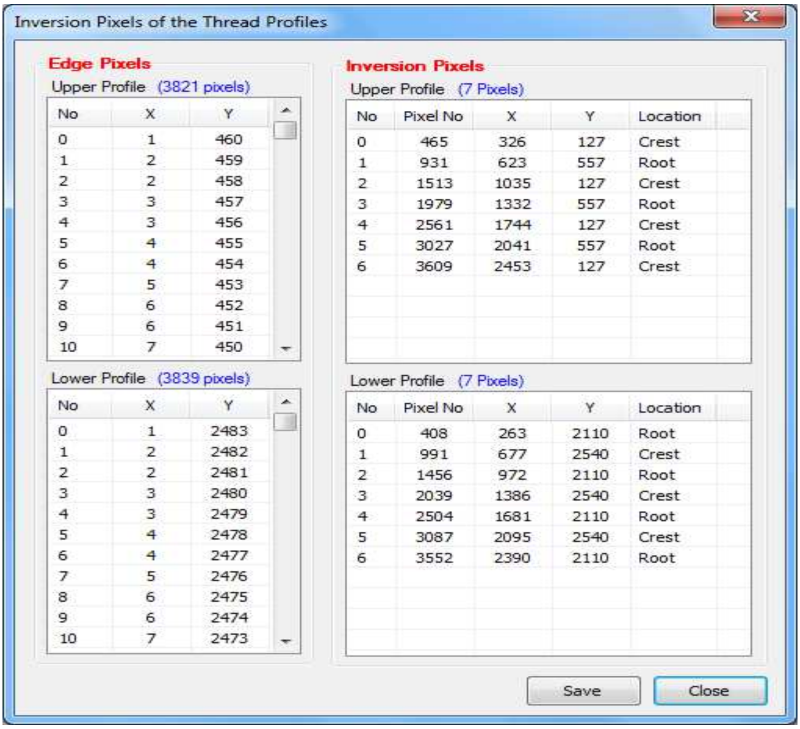
The result of the EPA algorithm is a list of pixels arranged sequentially from left to right. The two lists shown in the left side of Fig. 4 shows the extracted pixels of the upper and lower profiles after applying the EPA algorithm on the sample thread shown in Fig. 2.

3.2 Computer Vision Algorithms

Six computer vision algorithms were developed to calculate the screw thread features (18 features) and one algorithm was developed to inspect the screw threads. Once the coordinates of the upper and lower profiles are arranged sequentially by the EPL algorithm, the computer vision algorithms are applied to perform the measurement and inspection processes as discussed in the following subsections.

3.2.1 Inversion Pixels Extraction

For screw thread profiles, the inversion pixels are the pixels at which the profile changes its direction as shown in Fig. 5. For any pixel in the screw thread profile, the direction of the profile can be calculated by subtracting the y coordinate of that pixel from the y coordinates of both the previous and next pixels. If the two values have the same sign (positive or negative), then the pixel is considered an inversion pixel. The inversion pixels are useful to calculate the crest and root lines of the screw threads; therefore, the Inversion Pixels Extraction (IPE) algorithm has been developed to extract all inversion pixels from the edge pixels of the upper and lower profiles. For example, the IPE algorithm extracts the inversion pixels of the upper profile as follows:



Edge Pixels – Upper Profile (3821 pixels)

No	X	Y
0	1	460
1	2	459
2	2	458
3	3	457
4	3	456
5	4	455
6	4	454
7	5	453
8	6	452
9	6	451
10	7	450

Edge Pixels – Lower Profile (3839 pixels)

No	X	Y
0	1	2483
1	2	2482
2	2	2481
3	3	2480
4	3	2479
5	4	2478
6	4	2477
7	5	2476
8	6	2475
9	6	2474
10	7	2473

Inversion Pixels – Upper Profile (7 Pixels)

No	Pixel No	X	Y	Location
0	465	326	127	Crest
1	931	623	557	Root
2	1513	1035	127	Crest
3	1979	1332	557	Root
4	2561	1744	127	Crest
5	3027	2041	557	Root
6	3609	2453	127	Crest

Inversion Pixels – Lower Profile (7 Pixels)

No	Pixel No	X	Y	Location
0	408	263	2110	Root
1	991	677	2540	Crest
2	1456	972	2110	Root
3	2039	1386	2540	Crest
4	2504	1681	2110	Root
5	3087	2095	2540	Crest
6	3552	2390	2110	Root

Fig. 4 Extracted Coordinates of the edge pixels (left lists) and Inversion pixels (right lists) for the upper and lower profiles

1- Find the first two successive pixels which have different y coordinates, for example y_n, y_{n+1}.
2- Using the image coordinate system shown in Fig. 5, the initial direction (Dir_{Init}) of the profile can be calculated between these two pixels as follows:
   If $y_{n+1} < y_n$ then Dir_{Init} = Up, otherwise, Dir_{Init} = Down.
3- Consider the initial direction (Dir_{Init}) is the previous direction (Dir_{Prev}) for the next pixels, i.e., $Dir_{Prev} = Dir_{Init}$.
4- Loop through all remaining pixels in the *UpperEdgePixels* array and do the following for each pixel. Consider *i* the pixel number of the current pixel.
   a- Calculate the next direction (Dir_{Next}) of the profile by comparing the *y* coordinate of the next pixel (y_{i+1}) with the *y* coordinate of the current pixel (y_i) as follows:
   - If $y_{i+1} < y_i$ then Dir_{Next} = Up
   - Else If $y_{i+1} > y_i$ then Dir_{Next} = Down
   - Otherwise, skip.

b- If Dir_{Prev} and Dir_{Next} are the same then skip, otherwise, do the following:
- If Dir_{Next}= Down, then mark the current pixel as crest inversion pixel.
- If Dir_{Next}= Up, then mark the current pixel as root inversion pixel.
- Set Dir_{Prev} = Dir_{Next} then loop to the next pixel.

The IPE algorithm uses the same procedures to extract the inversion pixels of the lower profile with one exception in step 4-b. For the lower profile, if Dir_{Next} = Down, then the current pixel is marked as root inversion pixel instead of crest inversion pixel. Similarly, if Dir_{Next} = Up, then the current pixel is marked as crest inversion pixel instead of root inversion pixel. The two lists shown in the right side of Fig. 4 shows the inversion pixels extracted by the IPE algorithm for the upper and lower profiles.

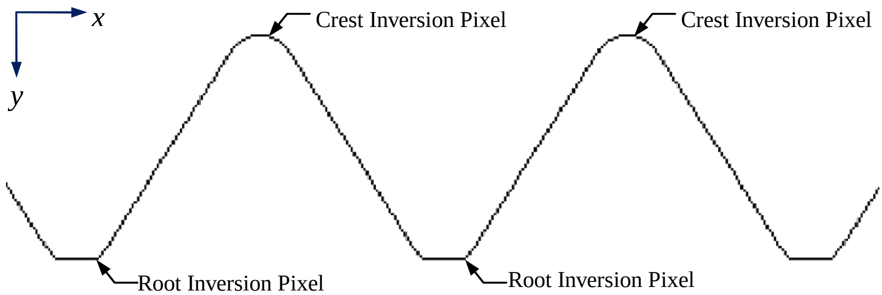


Fig. 5 Inversion pixels at the crest and root of the upper profile (enlarged image)

3.2.2 Crest and Root Pixels Extraction

The Crest and Root Pixels Extraction (CRPE) algorithm has been developed to extract all pixels lie on the crest and root lines of the screw thread profiles. Based on the geometry of the screw thread profile shown in Fig. 6, the crest lines are the two lines pass through the crest inversion pixels of the upper and lower profiles. Similarly, the root lines are the two lines pass through the root inversion pixels of the upper and lower profiles.

As shown in Fig. 4, the upper right list represents the crest and root inversion pixels of the upper profile. From this list, the crest line of the upper profile can be defined as the line connects between the first crest inversion pixel (pixel 465) and the last crest inversion pixel (pixel 3609). Similarly, the root line of the upper profile can be defined as the line connects between the first root inversion pixel (pixel 931) and the last root inversion pixel (pixel 3027). Similarly, the crest and root lines of the lower profiler can be defined. Table 1 shows the extracted data for the crest and root lines of the upper and lower profiles.

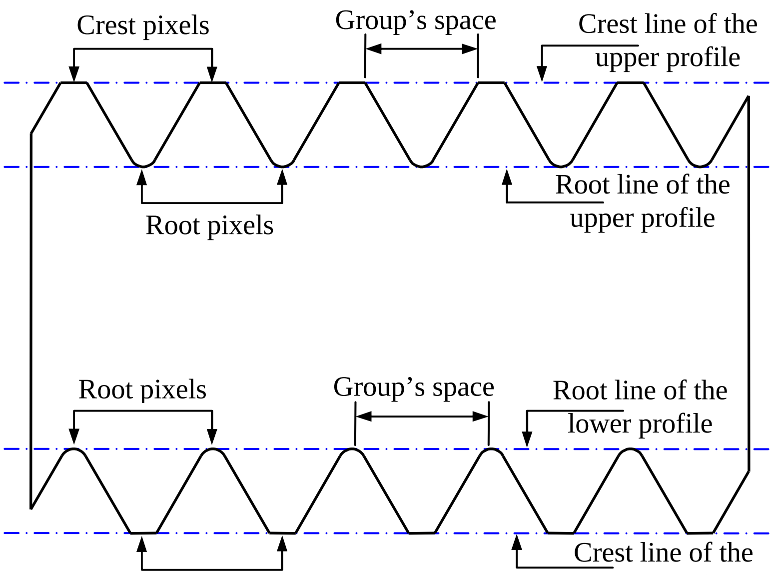


Fig. 6 Using crest and root lines to extract crest and root pixels

Table 1: Extracted data for the crest and root lines of the upper and lower profiles

Profile	Line type	First pixel			Last pixel		
		Pixel No.	X1	Y1	Pixel No.	X2	Y2
Upper	Crest line	465	326	127	3609	2453	127
	Root line	931	623	557	3027	2041	557
Lower	Crest line	991	677	2540	3087	2095	2540
	Root line	408	263	2110	2552	2390	2110

Any pixel can be classified as crest/root pixel if the perpendicular distance (*pd*) between the pixel and the crest/root line is nearly equal to zero, i.e. $1> pd >-1$. For example, the CRPE algorithm extracts the crest pixels of the upper profile by calculating *pd* between each pixel in the *UpperEdgePixel*s array and the crest line of the upper profile. All pixels satisfy the condition of $1> pd >-1$ are marked as crest pixel for the upper profile. The value of *pd* can be calculated using equation 3:

$$Pd = \frac{(x_2 - x_1)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \tag{3}$$

Where: *pd* is the perpendicular distance, x_1, y_1 are the coordinates of the first pixel (p_1) of the crest line and x_2,y_2 are the coordinates of the last pixel (p_2), x_3,y_3 are the coordinates of the pixel at which *pd* is calculated (p_3).

For Cartesian coordinates, negative values of *pd* means that the pixel p_3 lies to the left of the crest line (looking from the first pixel to the second pixel), positive values means that the pixel p_3 lies to the right of the crest line, and zero values means that the pixel p_3 lies on the crest line. For the image coordinates shown in Fig. 5, the opposite is true, i.e., negative values of *pd* means that the pixel p_3 lies to the right of the crest line and positive values means that the pixel p_3 lies to the left.

The crest and root pixels of the upper profile can be extracted by applying the following steps:

1- For each pixel in the upper profile, calculate *pd* between the pixel and the crest line.
2- If $pd < 1$ and $pd > -1$, then mark the pixel as crest pixel.
3- For each pixel in the upper profile, calculate *pd* between the pixel and the root line.
4- If $pd < 1$ and $pd > -1$, then mark the pixel as root pixel.

The crest and root pixels of the lower profile can be extracted by the same way. Fig. 7 shows the extracted crest and root pixels for the upper and lower profiles. The number of crest pixels for the upper and lower profiles is greater than the number of root pixels because the crest of the investigated thread (Fig. 2) is a straight edge while the root is a curve.

3.2.3 Crest and Root Pixels Grouping

The aim of the Crest and Root Pixels Grouping (CRPG) algorithm is to extract useful information about the screw thread from the crest and root pixels extracted by the CRPE algorithm. This can be done by applying the following two steps:

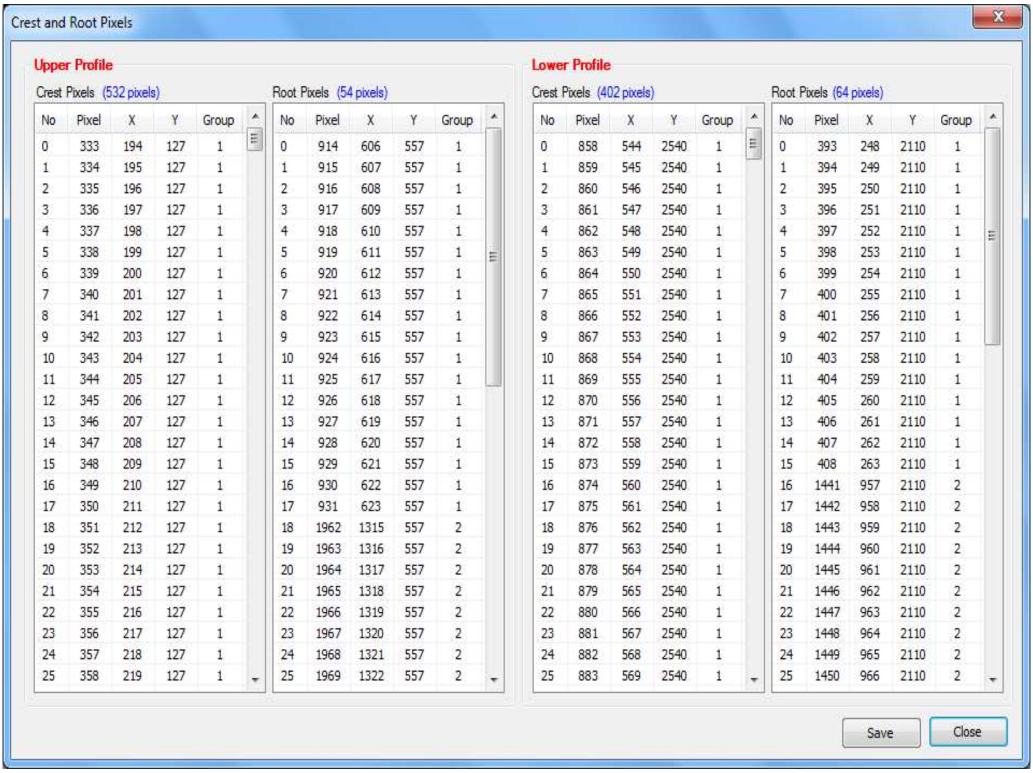
1- Assigning each group of successive pixels a unique group number.
2- Extracting useful information from the identified groups.

3.2.4 Assigning each group of successive pixels a unique group number

In this step, each pixel is assigned a unique group number based on the distance between the pixel and the next pixel. As shown in Fig. 6, the distance between each two successive pixels in the same group will be equal to 1 pixel and should not be greater than 1.414 ($\sqrt{2}$). The space between each two groups of pixels is represented by the distance between the last pixel in one group and the first pixel in the next group. This space will always be greater than 2 pixels. Considering this, the CRPG algorithm can group the crest pixels of the upper profile, for example, as follows:

1- Consider the first pixel is the current pixel and assign it a group number of 1.
2- Calculate the distance between the current pixel and the next pixel.
3- If the calculated distance is less than or equal to 2 pixels, then assign the same group number to the next pixel, otherwise, increase the group number by 1 and assign the new group number to the next pixel.
4- Move to the next pixel and consider it the current pixel.
5- Repeat steps 2 to 4 for all crest pixels of the upper profile.

The root pixels of the upper profile as well as the crest and root pixels of the lower profile can be assigned a unique group number using the same way. As shown in the last column of each list in Fig. 7, each pixel is assigned a unique group number by this algorithm.


Crest and Root Pixels

Upper Profile

Crest Pixels (532 pixels)

No	Pixel	X	Y	Group
0	333	194	127	1
1	334	195	127	1
2	335	196	127	1
3	336	197	127	1
4	337	198	127	1
5	338	199	127	1
6	339	200	127	1
7	340	201	127	1
8	341	202	127	1
9	342	203	127	1
10	343	204	127	1
11	344	205	127	1
12	345	206	127	1
13	346	207	127	1
14	347	208	127	1
15	348	209	127	1
16	349	210	127	1
17	350	211	127	1
18	351	212	127	1
19	352	213	127	1
20	353	214	127	1
21	354	215	127	1
22	355	216	127	1
23	356	217	127	1
24	357	218	127	1
25	358	219	127	1

Root Pixels (54 pixels)

No	Pixel	X	Y	Group
0	914	606	557	1
1	915	607	557	1
2	916	608	557	1
3	917	609	557	1
4	918	610	557	1
5	919	611	557	1
6	920	612	557	1
7	921	613	557	1
8	922	614	557	1
9	923	615	557	1
10	924	616	557	1
11	925	617	557	1
12	926	618	557	1
13	927	619	557	1
14	928	620	557	1
15	929	621	557	1
16	930	622	557	1
17	931	623	557	1
18	1962	1315	557	2
19	1963	1316	557	2
20	1964	1317	557	2
21	1965	1318	557	2
22	1966	1319	557	2
23	1967	1320	557	2
24	1968	1321	557	2
25	1969	1322	557	2

Lower Profile

Crest Pixels (402 pixels)

No	Pixel	X	Y	Group
0	858	544	2540	1
1	859	545	2540	1
2	860	546	2540	1
3	861	547	2540	1
4	862	548	2540	1
5	863	549	2540	1
6	864	550	2540	1
7	865	551	2540	1
8	866	552	2540	1
9	867	553	2540	1
10	868	554	2540	1
11	869	555	2540	1
12	870	556	2540	1
13	871	557	2540	1
14	872	558	2540	1
15	873	559	2540	1
16	874	560	2540	1
17	875	561	2540	1
18	876	562	2540	1
19	877	563	2540	1
20	878	564	2540	1
21	879	565	2540	1
22	880	566	2540	1
23	881	567	2540	1
24	882	568	2540	1
25	883	569	2540	1

Root Pixels (64 pixels)

No	Pixel	X	Y	Group
0	393	248	2110	1
1	394	249	2110	1
2	395	250	2110	1
3	396	251	2110	1
4	397	252	2110	1
5	398	253	2110	1
6	399	254	2110	1
7	400	255	2110	1
8	401	256	2110	1
9	402	257	2110	1
10	403	258	2110	1
11	404	259	2110	1
12	405	260	2110	1
13	406	261	2110	1
14	407	262	2110	1
15	408	263	2110	1
16	1441	957	2110	2
17	1442	958	2110	2
18	1443	959	2110	2
19	1444	960	2110	2
20	1445	961	2110	2
21	1446	962	2110	2
22	1447	963	2110	2
23	1448	964	2110	2
24	1449	965	2110	2
25	1450	966	2110	2

Save Close

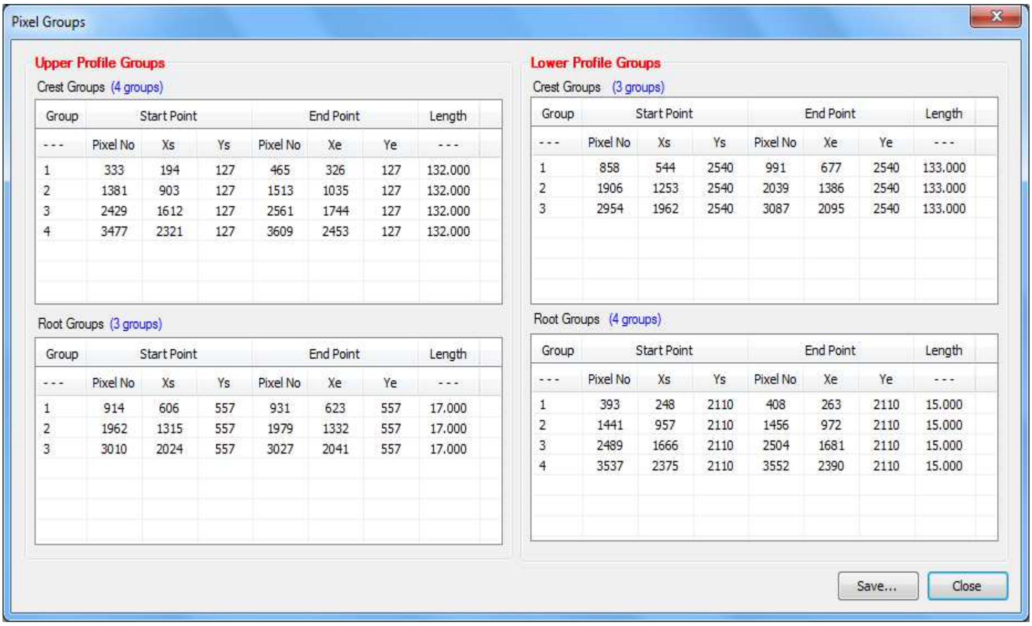
Fig. 7 Extracted crest and root pixels for the upper and lower profiles

3.2.4.1 Extracting useful information from the identified groups

After assigning each pixel a unique group number, each group of successive pixels can be considered as a short line. Therefore, useful information such as the start, the end, and the length of each short line can be calculated from each group as shown in Fig. 8. For each group, the coordinates of the start point are the coordinates the first pixel in the group. Similarly, the coordinates of the end point are the coordinates of the last pixel in the group. The length of each short line is the distance between the start and the end points of each group.

3.2.5 Crest and Root Groups Classification

Based on the screw thread types, the root and crest can be classified as vertex, edge, or curve. As a result, the shape of the screw threads can be classified into the five groups shown in Fig. 9 and Fig. 10-a. Based on this classification, the Crest and Root Groups Classification (CRGC) algorithm uses the information extracted by the CRPG algorithm to classify the crest and root of the screw thread. By referring to Fig. 9, the CRGC algorithm can classify the crest and root of the upper profile, for example, as follows:



Pixel Groups

Upper Profile Groups

Crest Groups (4 groups)

Group	Start Point			End Point			Length
---	Pixel No	Xs	Ys	Pixel No	Xe	Ye	---
1	333	194	127	465	326	127	132.000
2	1381	903	127	1513	1035	127	132.000
3	2429	1612	127	2561	1744	127	132.000
4	3477	2321	127	3609	2453	127	132.000

Root Groups (3 groups)

Group	Start Point			End Point			Length
---	Pixel No	Xs	Ys	Pixel No	Xe	Ye	---
1	914	606	557	931	623	557	17.000
2	1962	1315	557	1979	1332	557	17.000
3	3010	2024	557	3027	2041	557	17.000

Lower Profile Groups

Crest Groups (3 groups)

Group	Start Point			End Point			Length
---	Pixel No	Xs	Ys	Pixel No	Xe	Ye	---
1	858	544	2540	991	677	2540	133.000
2	1906	1253	2540	2039	1386	2540	133.000
3	2954	1962	2540	3087	2095	2540	133.000

Root Groups (4 groups)

Group	Start Point			End Point			Length
---	Pixel No	Xs	Ys	Pixel No	Xe	Ye	---
1	393	248	2110	408	263	2110	15.000
2	1441	957	2110	1456	972	2110	15.000
3	2489	1666	2110	2504	1681	2110	15.000
4	3537	2375	2110	3552	2390	2110	15.000

Save... Close

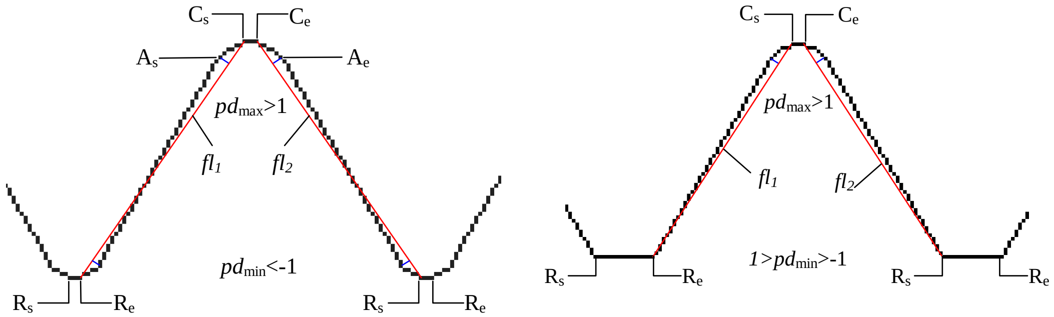
Fig. 8 Results of the Crest and Root Pixels Grouping (CRPG) algorithm

1- Assume a flank line fl_1 connects the end pixel of a root group (R_e) and the start pixel of the next crest group (C_s).
2- For all pixels lie between the two pixels (R_e, C_s), use equation 3 to calculate the minimum perpendicular distance (pd_{min}) and the maximum perpendicular distance (pd_{max}) between each pixel and the flank line fl_1.
3- For the four cases shown in Fig. 9, use the calculated values of pd_{min} and pd_{max} to classify the crest and root shape as shown in table 2. It should be noticed here that the calculated values shown in the table are based on the image coordinate system, not Cartesian coordinate system.
4- As shown in Fig. 10, use the lengths of the crest and root groups (C_s, C_e / R_s, R_e), which were calculated by the CRPG algorithm, to distinguish between vertex and edge shapes. If the group length is less than 2 pixels, then classify the crest/root as vertex, otherwise, classify the crest/root as edge.

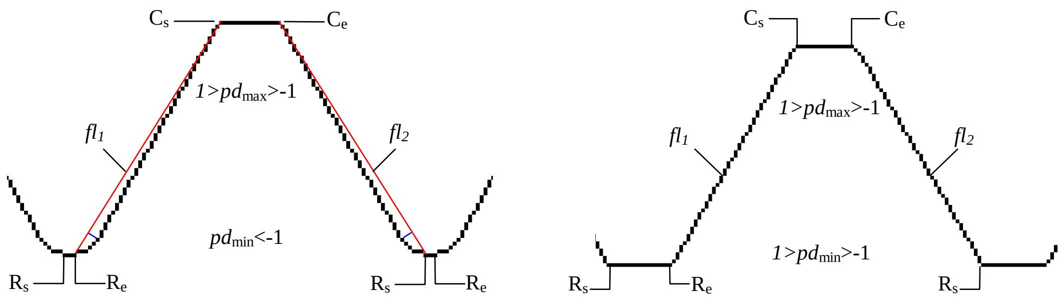
After applying the CRGC algorithm, the crest and root of the screw thread are classified as shown in the upper section in Fig. 11.

Table 2: Classification of the crest and root shapes according to the minimum and maximum distances between flank pixels and the flank line

Thread Type	pd_{min}	pd_{max}	Crest	Root
(a)	$pd_{min} < -1$	$pd_{max} > 1$	Curve	Curve
(b)	$1 > pd_{min} > -1$	$pd_{max} > 1$	Curve	Vertex/Edge
(c)	$pd_{min} < -1$	$1 > pd_{max} > -1$	Vertex/Edge	Curve
(d)	$1 > pd_{min} > -1$	$1 > pd_{max} > -1$	Vertex/Edge	Vertex/Edge

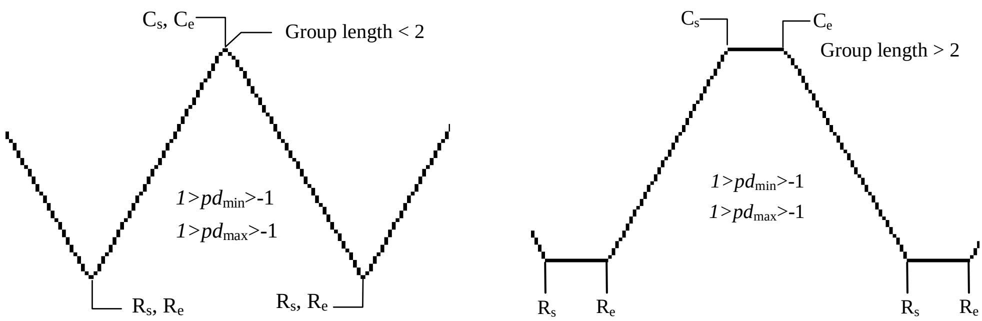


(a) Thread with curved crest and root (b) Thread with curved crest and straight root



(c) Thread with curved root and straight crest (d) Thread with straight root and crest

Fig. 9: Classification of crest and root as straight and curve (magnified images) (C_s: Crest start, C_e: Crest end, R_s: Root start, R_e: Root end, A_s: Arc start, A_e: Arc end, pd_{min}: Minimum perpendicular distance, pd_{max}: Maximum perpendicular distance)

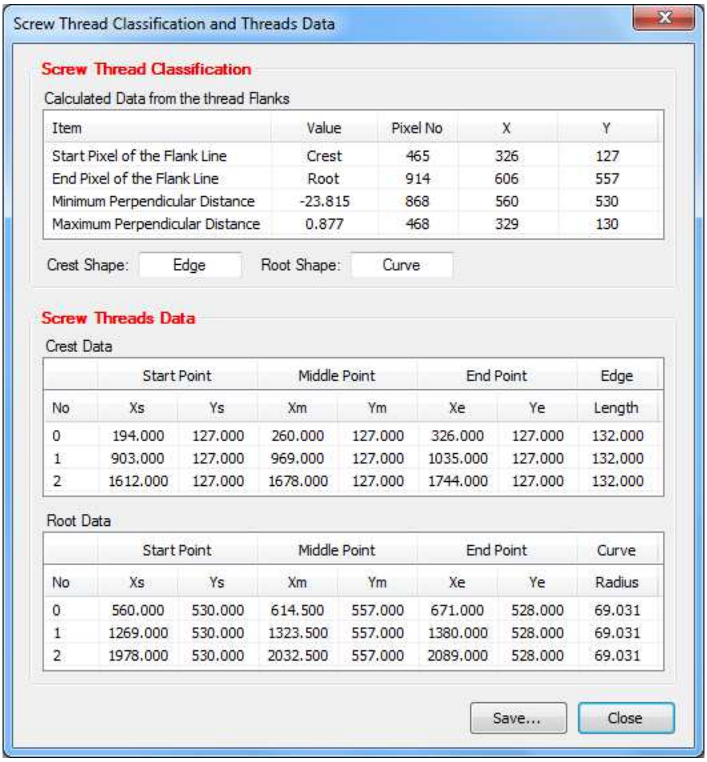


(a) Thread with sharp crest and root (b) Thread with straight crest and root

Fig. 10: Classification of screw threads having straight and vertex crests and roots

3.2.6 Crest and Root Data Calculation (CRDC)

After classifying the crest and root of the screw thread, the Crest and Root Data Calculation (CRDC) algorithm is used to calculate the main data of the screw thread. This data include the crest/root width for straight features and the crest/root radius for curved features. By referring to Fig. 9, the CRDC algorithm calculates the main data of the screw thread as follows:



Screw Thread Classification and Threads Data

Screw Thread Classification

Calculated Data from the thread Flanks

Item	Value	Pixel No	X	Y
Start Pixel of the Flank Line	Crest	465	326	127
End Pixel of the Flank Line	Root	914	606	557
Minimum Perpendicular Distance	-23.815	868	560	530
Maximum Perpendicular Distance	0.877	468	329	130

Crest Shape: Edge Root Shape: Curve

Screw Threads Data

Crest Data

	Start Point		Middle Point		End Point		Edge
No	Xs	Ys	Xm	Ym	Xe	Ye	Length
0	194.000	127.000	260.000	127.000	326.000	127.000	132.000
1	903.000	127.000	969.000	127.000	1035.000	127.000	132.000
2	1612.000	127.000	1678.000	127.000	1744.000	127.000	132.000

Root Data

	Start Point		Middle Point		End Point		Curve
No	Xs	Ys	Xm	Ym	Xe	Ye	Radius
0	560.000	530.000	614.500	557.000	671.000	528.000	69.031
1	1269.000	530.000	1323.500	557.000	1380.000	528.000	69.031
2	1978.000	530.000	2032.500	557.000	2089.000	528.000	69.031

Save... Close

Fig. 11 Classification of the crest and root of the screw thread (upper section) and calculating the main data of the screw thread (lower section)

1- If the crest/root is classified as a vertex, the start and end pixels will be the same and their coordinates can be extracted from the pixel groups shown in Fig. 8. Accordingly, the coordinates of the middle pixel will be the same as the first and end pixels.
2- If the crest/root is classified as an edge, the start and end pixels will be different and their coordinates can be extracted from the pixel groups shown in Fig. 8. The middle pixel is calculated as the point between the start and end pixels.
3- If the crest/root is classified as a curve, three points are required to identify the arc. As shown in Fig. 9-a, the start and end points of the crest arc (A_s, A_e) can be defined as the two pixels which produce the maximum perpendicular distances (pd_{max}) from the two flank lines (fl_1, fl_2). The second point of the arc can be calculated as the middle point between the start and end pixels.
   Once the three points of the arc are defined, the center and radius of the arc can be calculated. If (x_1,y_1), (x_2,y_2) and (x_3,y_3) are the coordinates of the start, middle, and end point of the arc, respectively, then the coordinates of the center point can be calculated using equations 4 and 5 and the radius can be calculated using equation 6:

$$c_x = \frac{a(cc'+bb') - b(dd'+aa')}{2(ce - be')} \tag{4}$$

$$c_y = \frac{c(dd'+aa') - d(cc'+bb')}{2(ce-be')} \tag{5}$$

$$R = \sqrt{(x_i - c_x)^2 + (y_i - c_y)^2} \tag{6}$$

Where:

c_x *and* c_y are the x and y coordinates of the center point of the arc.

$a = y_3 - y_1, \quad a' = y_3 + y_1$

$b = y_2 - y_1, \quad b' = y_2 + y_1$

$c = x_2 - x_1, \quad c' = x_2 + x_1$

$d = x_3 - x_1, \quad d' = x_3 + x_1$

$e = y_3 - y_2, \quad e' = x_3 - x_2$

x_i and y_i are the x and y coordinates of the start point of the arc.

After applying the CRDC algorithm, the main data of the screw thread is calculated as shown in the lower section in Fig.11.

3.2.7 Screw Thread Features Calculation

The Screw Thread Features Calculation (STFC) algorithm is used to calculate the screw thread features based on the main data extracted by the CRDC algorithm. The main important features are calculated as follows:

3.2.7.1 Calculation of the Major and Minor diameters

By referring to Fig 6, the major diameter can be calculated by calculating the perpendicular distance (pd) between any pixel in the crest pixel groups of the upper profile and the crest line of the lower profile. To obtain accurate measurements, three calculations are performed for three pixels in different groups, then the average of the three calculations is taken as the major diameter. Similarly, the minor diameter can be calculated by calculating pd between any pixel in the root pixel groups of the upper profile and the root line of the lower profile.

3.2.7.2 Calculation of the theoretical depth and crest truncation

Referring to Fig. 12, the theoretical depth (h_1) of the screw thread can be calculated as follows:

1- If P_1 and P_2 are the two end points of the flank line fl_1 and P_3 and P_4 are two end points of the flank line fl_2, then the intersection point (P_5) of the two flank lines can be calculated using equations 7 and 8.

$$x_{\text{int}} = P1_x + r * (P2_x - P1_x) \tag{7}$$

$$y_{\text{int}} = P1_y + r * (P2_y - P1_y) \tag{8}$$

Where:

x_{int} is the x coordinate of the intersection point,

y_{int} is the y coordinate of the intersection point,

$$r = \frac{(P1_y - P3_y)(P4_x - P3_x) - (P1_x - P3_x)(P4_y - P3_y)}{(P2_x - P1_x)(P4_y - P3_y) - (P2_y - P1_y)(P4_x - P3_x)} \tag{9}$$

If the denominator in equation 9 is equal to zero, then the two lines are parallel.

2- Similarly, the intersection points P_6 and P_7 can be calculated between the two flank lines fl_3 and $fl4$ and the two flank lines fl_2 and fl_3 respectively.
3- The theoretical depth (h_1) can be calculated as the perpendicular distance between the point P_7 and the line connecting P_5 and P_6. This distance can be calculated using equation 3.
4- The crest truncation (h_2) can be calculated as the perpendicular distance between any pixel in the crest groups of the upper profile and the line connecting P_5 and P_6. Equation 3 is also used to calculate this perpendicular distance.

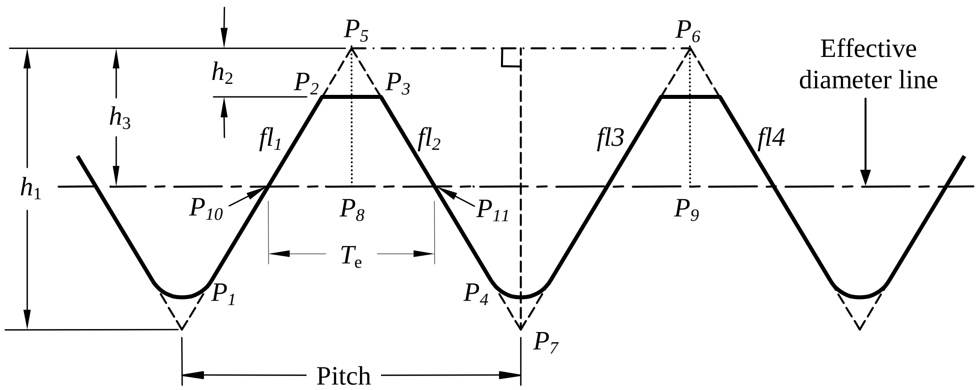


Fig. 12 calculating the effective diameter and other thread features

3.2.7.3 Calculation of the pitch and effective diameter

By referring to Fig. 12, the pitch can be calculated as the distance between P_5 and P_6. The effective diameter line is defined as the line which divides the screw thread so that the widths of the threads and the widths of the spaces are equal on a perfect thread. According to this definition, the distance h_3 is equal to half of the theoretical depth, i.e. $h_3 = 0.5h_1$. Hence, the effective diameter ($D_{effective}$) of the screw thread can be calculated as follows:

$$D_{effective} = D_{major} - 2(h_3 - h_2),$$

$$D_{effective} = D_{major} - 2(0.5h_1 - h_2) \tag{10}$$

Where: D_{major} is the major diameter, h_1 is the theoretical depth, h_2 is the crest truncation.

3.2.7.4 Calculation of the axial thickness

By referring to Fig. 12, the axial thickness (T_e) is the width of the screw thread measured at the effective line. To calculate the axial thickness, the effective diameter line should be defined by any two end points, for example, P_8 and P_9. Based on the screw thread geometry, the effective diameter line is parallel to the line $P_5 P_6$ and far from it with a distance equal to h_3. Therefore, the two end points (P_8 and P_9) of the effective diameter line can be calculated as follows:

$$P_8.x = P_5.x + h_3 * Cos(\theta) \quad (11)$$

$$P_8.y = P_5.y + h_3 * Sin(\theta) \quad (12)$$

$$P_9.x = P_6.x + h_3 * Cos(\theta) \quad (13)$$

$$P_9.y = P_6.y + h_3 * Sin(\theta) \quad (14)$$

Where: $P_8.x$ and $P_8.y$ are the x and y coordinates of P_8, $P_9.x$ and $P_9.y$ are the x and y coordinates of P_9, h_3 equal to the half of the theoretical depth, θ is the angle of inclination of the line $P_5 P_6$. For horizontal lines, θ is equal to zero.

After defining the effective diameter line, the axial thickness can be calculated as follows:

1- Use equations 7, 8, and 9 to calculate the intersection point (P_{10}) between the effective diameter line and the flank line fl_1.
2- Similarly, calculate the intersection point (P_{11}) between the effective diameter line and the flank line fl_2.
3- Calculate the axial thickness as the distance between P_{10} and P_{11}.

3.2.7.5 Calculation of the other features of the screw thread

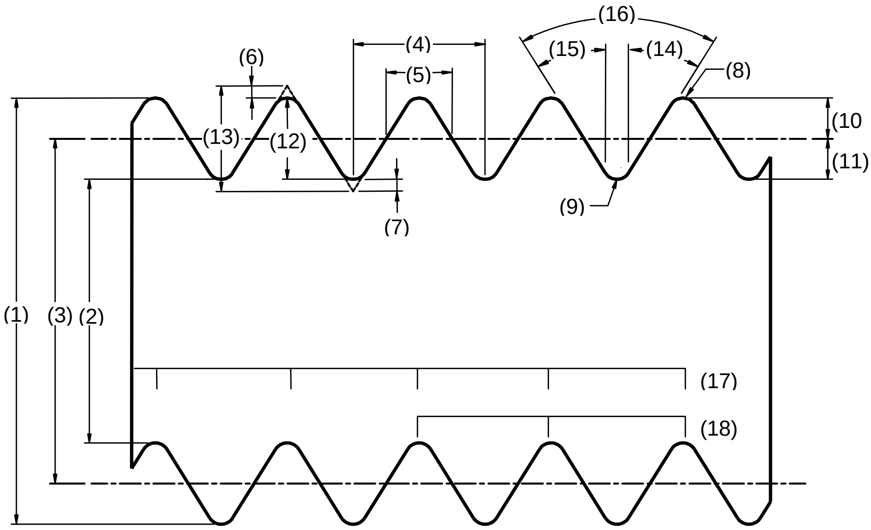
The rest of the screw thread features can be calculated using similar ways according to the basic geometry shown in Fig. 13. For example, the Addendum (10) can be calculated from the upper profile as the perpendicular distance between any pixel in the crest groups and the effective diameter line. Similarly, the Dedendum (11) can be calculated as the perpendicular distance between any pixel in the root groups and the effective diameter line. Also, the actual depth (12) can be calculated as the perpendicular distance between any pixel on the upper profile crest groups and the root line of the upper profile.

4. WORK PROCEDURES

To perform the measurement or inspection processes, the screw thread to be measured is set on the backlighting table then an image is captured and saved to a BMP file using the capturing software provided with the CCD camear. The captured image is then opened by the *MISTVision* software to perform the measurement and inspection processes as shown in the following sub-sections.

4.1 Measurement of screw threads

By referring to the main interface shown in Fig. 2, if the Automatic calculation check box is checked, the measurement process is performed automatically after opening the captured image of the screw thread to be measured. Otherwise, the user should click the Calculate button. The calculated features of the screw thread appear in the Thread Measurement section. If the system is calibrated, as it will be discussed in section 5, the thread features will be calculated in millimeters or inches. Otherwise, it will be calculated in pixels. The calculated features can be saved to a text file and it can be used as a reference data for the inspection process as it will be discussed in the next section.



(1)	Major diameter	(7)	Root Truncation	(13)	Theoretical depth
(2)	Minor diameter	(8)	Crest width/Radius	(14)	Flank angle 1
(3)	Effective diameter	(9)	Root width/Radius	(15)	Flank angle 2
(4)	Pitch	(10)	Addendum	(16)	Thread angle
(5)	Axial thickness	(11)	Dedendum	(17)	No of threads per inch
(6)	Crest Truncation	(12)	Actual depth	(18)	No of threads per cm.

Fig. 13: Thread features that can be measured by the MISTVision software

4.2 Multi Measurement Process

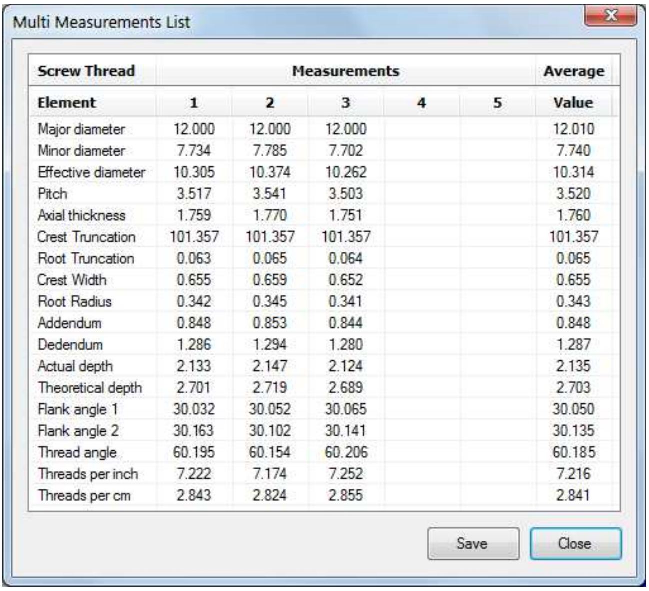
In metrology, usually one measurement is not enough to give accurate results; therefore, the *MISTVision* software can store up to five measurements for the same screw thread in order to calculate the average values of the screw thread features. These measurements are stored to a list called Multi Measurement List (*MML*) and each measurement can be obtained from a different image for the same screw thread. Fig. 14 shows the *MML* for three measurements obtained from three different images for the same screw thread and the average value for each screw thread parameter. The *MML* could be saved to a file for further usage.

4.3 Inspection of the screw threads

The inspection process is performed by comparing the measured features of the screw thread to be inspected with the features of a standard screw thread within specified tolerances. In the main interface of the MISTVision software (Fig. 2), the Inspection section deals with this process. The inspection process is performed using the following four steps:

1. Entering the features for the reference screw thread:

The standard features for the reference screw thread can be entered to the *MISTVision* software by two methods. In the first method, the standard screw thread is measured by the vision system and the calculated features are stored as reference values by clicking the *Set as Reference* button in the Thread measurement section. One reference screw thread could be used to check many screw threads. In the second method, the reference values are set manually by clicking the *Manual Ref*. button.


Multi Measurements List

Screw Thread	Measurements					Average
Element	1	2	3	4	5	Value
Major diameter	12.000	12.000	12.000			12.010
Minor diameter	7.734	7.785	7.702			7.740
Effective diameter	10.305	10.374	10.262			10.314
Pitch	3.517	3.541	3.503			3.520
Axial thickness	1.759	1.770	1.751			1.760
Crest Truncation	101.357	101.357	101.357			101.357
Root Truncation	0.063	0.065	0.064			0.065
Crest Width	0.655	0.659	0.652			0.655
Root Radius	0.342	0.345	0.341			0.343
Addendum	0.848	0.853	0.844			0.848
Dedendum	1.286	1.294	1.280			1.287
Actual depth	2.133	2.147	2.124			2.135
Theoretical depth	2.701	2.719	2.689			2.703
Flank angle 1	30.032	30.052	30.065			30.050
Flank angle 2	30.163	30.102	30.141			30.135
Thread angle	60.195	60.154	60.206			60.185
Threads per inch	7.222	7.174	7.252			7.216
Threads per cm	2.843	2.824	2.855			2.841

Fig. 14 the Multi Measurement List (MML)

2. Selecting the screw thread features to be inspected:

This can be done through the Inspection Tolerances dialog box shown in Fig. 15, which can be displayed by clicking the *Tolerances* button. The features to be inspected should be checked and their allowable tolerances are assigned. Each parameter can be assigned unique tolerance values (lower and upper). Alternatively, the same tolerance value can be applied to all thread items by entering the required value in the text box *Value for all thread items* then clicking the *Assign* button.

3. Measuring the screw thread to be inspected:

The screw thread to be inspected is measured by the vision system, then the calculated features are compared with the standard screw thread features according to the tolerances given through the inspection tolerance dialog box.

4. Taking an inspection decision:

The inspection decision is displayed automatically in the Inspection section (Fig. 2). The decision will be “Accepted” if all inspected features of the screw thread to be inspected meet the corresponding features of the standard screw thread within the specified tolerances. Otherwise, the decision will be “Rejected”. The details of the comparison process can be displayed through the Inspection Details dialog box (Fig. 16), which can be displayed by clicking the Details button in Fig. 2.

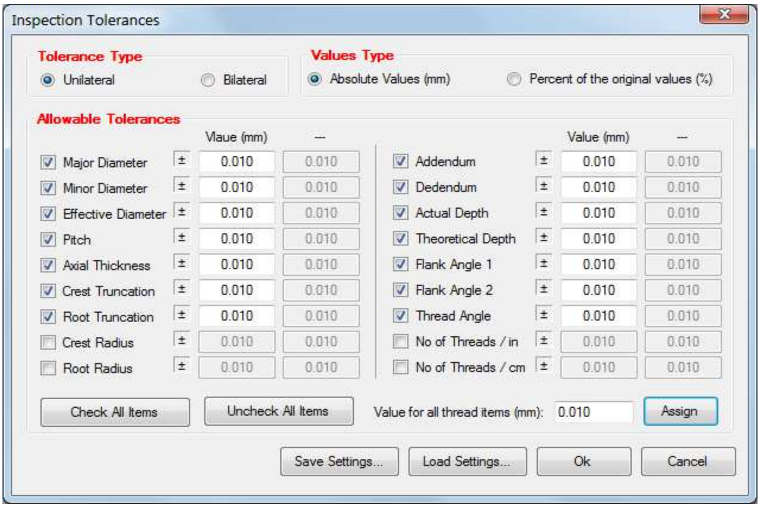
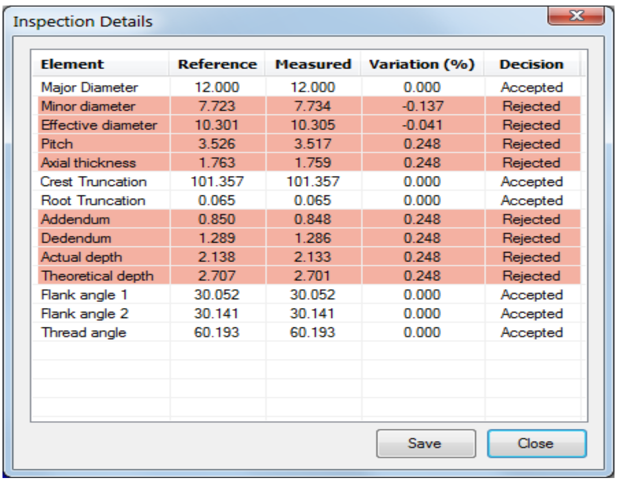


Fig. 15 Inspection tolerance dialog box



Element	Reference	Measured	Variation (%)	Decision
Major Diameter	12.000	12.000	0.000	Accepted
Minor diameter	7.723	7.734	-0.137	Rejected
Effective diameter	10.301	10.305	-0.041	Rejected
Pitch	3.526	3.517	0.248	Rejected
Axial thickness	1.763	1.759	0.248	Rejected
Crest Truncation	101.357	101.357	0.000	Accepted
Root Truncation	0.065	0.065	0.000	Accepted
Addendum	0.850	0.848	0.248	Rejected
Dedendum	1.289	1.286	0.248	Rejected
Actual depth	2.138	2.133	0.248	Rejected
Theoretical depth	2.707	2.701	0.248	Rejected
Flank angle 1	30.052	30.052	0.000	Accepted
Flank angle 2	30.141	30.141	0.000	Accepted
Thread angle	60.193	60.193	0.000	Accepted

Fig. 16 The Inspection details dialog box

5. SYSTEM VERIFICATION

This section explains the calibration and the accuracy of the employed vision system. In addition, the verification of the system accuracy is explained.

5.1 System Calibration

The units of the captured images are pixels. To calibrate the proposed vision system for actual units (i.e. millimeter or inches), the pixel size in y direction is calculated according to the major diameter of the screw thread to be measured as follows:

1- The major diameter (D_{major}) of the screw thread to be measured or inspected should be entered by the user to the MISCVision software through the Input Data section in the main interface (Fig. 2). Also, the units of measurements should be selected from the Units combo box. If the major diameter is specified, the system will be calibrated automatically and the measurements will be calculated in actual units, otherwise, it will be calculated in image pixels.

2- The MISCVision software searches the edge pixels of the upper and lower profiles for the opened image to find the two pixels having the minimum and maximum y coordinates (Y_{min}, Y_{max}) then it calculates the maximum diameter of the screw thread in y direction (D_{max}) as follows:

$$D_{max} = Y_{max} - Y_{min} \quad (15)$$

3- A calibration factor (CF) is calculated as follows:

$$CF = D_{major} / D_{max} \quad (16)$$

4- All thread features are multiplied by the calibration factor (CF) except the flank and thread angles and the number of threads per inch or centimeter.

5.2 System Accuracy

The accuracy of the employed vision system can be affected by two factors (1) the size of the measured screw thread and (2) the size of the captured image. For the first factor, increasing the size of the measured screw thread increases the pixel size, i.e. decreases the accuracy of measurement. For the second factor, increasing the size of the captured image decreases the pixel size, i.e. increases the accuracy of measurement and increases the processing time. Fig 17 shows the relationship between the size of the captured image and both of the pixel size and the processing time for thread type M10. From the figure, it can be seen that images with size of 4.2 Mp (2048*2048 pixels) can produce a resolution of 5 μm/pixel for screw threads of 10 mm major diameter in a reasonable processing time (5.5 sec).

5.3 System Verification

To verify the proposed vision system, an ISO metric thread plug gage (M16 coarse) with nominal major diameter equals to 16 mm and a pitch equals to 2 mm was measured by the proposed vision system. The measurement process was performed three times using the MML feature, offered by the proposed software, and the averages of the measurements were recorded. The features of the ISO metric thread plug gage were calculated according to its basic geometry. Table 3 shows a comparison between the values of the thread features for both the standard thread plug gauge and the measured features by the developed system. The differences between the standard and measured features are listed in μm. It can be noticed that the maximum difference between the standard and measured values was ±5.4 μm.

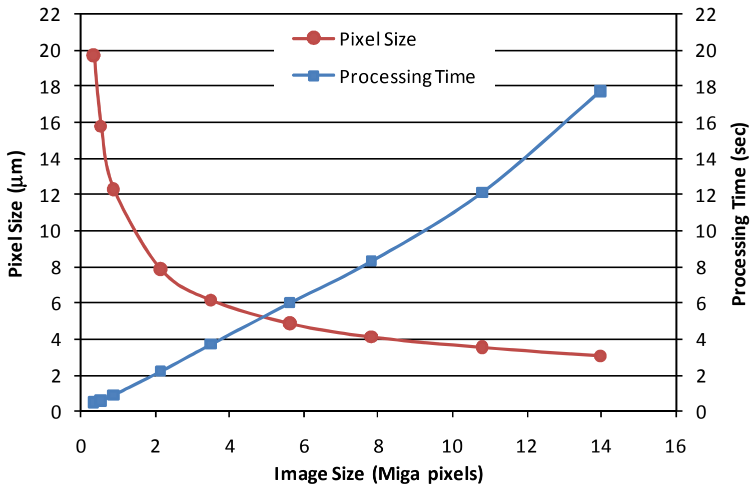


Fig. 17 Relationship between the captured image size and both processing time and pixel size for thread M10 (Major diameter = 10 mm)

Table 3: Comparison between the values of the thread features for both the standard thread plug gauge and the features measured by the vision system

No	Thread Feature	Measurements			Difference	
		Standard Specimen	Measured Specimen	Units	Value	units
1	Major diameter	16.0000	16.001	Mm	-1.00	µm
2	Minor diameter	13.8340	13.837	Mm	-3.0	µm
3	Effective diameter	14.7000	14.6979	Mm	2.1	µm
4	Pitch	2.0000	2.003	Mm	-3.0	µm
5	Axial thickness	1.0006	1.0003	Mm	0.3	µm
6	Crest truncation	0.2165	0.2171	Mm	-0.6	µm
7	Root truncation	0.4330	0.4342	Mm	-1.2	µm
8	Crest width	0.2500	0.2523	Mm	-2.3	µm
9	Root width	0.5000	0.5054	Mm	-5.4	µm
10	Addendum	0.6500	0.6516	Mm	-1.6	µm
11	Dedendum	0.4330	0.4344	Mm	-1.4	µm
12	Actual depth	1.0830	1.086	Mm	-3.0	µm
13	Theoretical depth	1.7325	1.7373	Mm	-4.8	µm
14	Flank angle 1	30.0000	29.9752	degree	0.025	degree
15	Flank angle 2	30.0000	29.9884	degree	0.012	degree
16	Thread angle	60.0000	59.9636	degree	0.036	degree
17	No of threads per inch	12.7000	12.6810	---	0.019	---
18	No of threads per centimeter	5.0000	4.99251	---	0.0075	---

6. CONCLUSIONS

A vision system has been utilized, as a new non-contact measurement system, for measurement and inspection of various types of screw threads. Six computer vision algorithms have been developed to analyze the captured images and perform the measurement and inspection processes from the captured images. The proposed vision system is capable of identifying most of the thread types automatically and calculating the most common screw thread features (18 features), which cannot be achieved by any other measuring system. The software is capable of inspecting screw threads based on reference values and specified tolerances for selected features. The system has been calibrated for both imperial and metric units and was verified by measuring a standard ISO metric thread plug gage and comparing the results with the standard values. The results showed that the maximum difference between the standard and measured values was ±5.4 μm, which provide a good accuracy.

REFERENCES

[1] Eldar Fetullazade, Hamza K. Akyildiz, S. Saritas, Effects of the machining conditions on the strain hardening and the residual stresses at the roots of screw threads, Materials and Design, vol. 31, pp 2025-2031, 2010.

[2] Degarmo, E.P., Black, J.T. and Kohser, R.A., Materials and processes in manufacturing, Prentice-Hall International Inc., 8th edition, 1997.

[3] Screw-thread gauging systems for determining conformance to thread standards, CRTD- American Society of Mechanical Engineers Center for Research and Technology Development, 354 East 47th Street, New York, NY 10017, 1996, 37.

[4] Aviation fastener audit, Federal Aviation Administration aircraft certification service, 800 Independence Avenue S.W. Washington, D.C. 20591, final report, June 13, 2001.

[5] Dong, Y. and Hess, D.P., The effect of thread dimensional conformance on vibration-induced loosening, ASME, Journal of Vibration and Acoustics, , vol. 121, 1999.

[6] Leon, F.M., Pai, N.G. and Hess, D.P., The effect of thread dimensional, conformance on yield and tensile strength, Department of Mechanical Engineering University of South Florida, Tampa, May 1999.

[7] Greenslade, J., Matievich, W., Kenneth, McCullough, Shoberg, R., Vass S., and Wilson, C., Thread geometry and fastener performance, Mechanical Engineering Magazine, Vol. 118, No. 12, Dec. 1996.

[8] Rashed, A.F. and Damir, M.N.H., Metrology of screw thread, Dar El-Manar, Egypt, 1977, 1.

[9] Jain, R.K., Engineering metrology, Khana Publisher, 17th edition, Khanna Pub., 2-B Nath Market, Nai Sarak, Delhi 110 006, 1999.

[10] Laczik, B., Contributions to measuring of screw surfaces, Periodica Polytechnica Ser. Mech. Eng., Vol. 43, No. 2, pp. 89–96, 1999.

[11] Hongyan Zhang and Guoxiong Zhang, “An online measuring system for large screw threads”, Proc. SPIE, Vol. 7130, 71301X (2008); doi:10.1117/12.819608 ,

Fourth International Symposium on Precision Mechanical Measurements, December 2008.

[12] Q. B. Tong, Z. L. Ding, J. C. Chen, L. L. Ai and F. Yuan, “The Research of Screw Thread Parameter Measurement Based on Position Sensitive Detector and Laser”, Journal of Physics: Conference Series 48, pp. 561–565, 2006.

[13] Fujun H., Xuming C., Yonghong Z., and Zhaozi H., Non-contact measurement of oil tube thread and the application, Optical Technology and Image Processing for Fluids and Solids Diagnostics, SPIE-Beijing, Abstract 050, 2002.

[14] E.S. Gadelmawla “A novel system for automatic measurement and inspection of screw threads”, Journal of Engineering Manufacture, I MECH E Part B, Vol. 218, B5, pp. 545-556, 2004.

[15] Carfagni M., Furferi R. and Governi L., A real-time machine-vision system for monitoring the textile raising process, Computers in Industry, Vol. 56, No. 8-9, pp. 831-842, 2005.

[16] D. Kerr, J. Pengilley, R. Garwood, Assessment and visualization of machine tool wear using computer vision, International Journal of Advanced Manufacturing Technology, vol. 28, pp. 781-791, 2006.

[17] Franco Pedreschi, Jorge León, Domingo Mery, Pedro Moyano, Development of a computer vision system to measure the color of potato chips, Food Research International, vol. 39, pp. 1092-1098, 2006.

[18] P.J. Armstrong, J. Antonis, The development of an active computer vision system for reverse engineering, Journal of engineering manufacture (Part B), Vol. 214, pp. 615-618, 2000.

[19] T. Brosnan, D.W. Sun, Improving quality inspection of food products by computer vision, Journal of Food Engineering, Vol. 61, pp. 3–16, 2004.

[20] Gadelmawla E.S., “Development of Computer Vision Algorithms for Measurement and Inspection of Spur Gears”, Mansoura Engineering Journal (MEJ), Faculty of Engineering, Mansoura University, Egypt, Vol. 34, No. 1, pp M27-M38, March 2009.

ABBREVIATIONS

EPD	edge pixels detection
EPT	edge pixels thinning
EPL	edge pixel labeling
EPA	edge pixels arrangement
MISTVision	Screw Thread Measurement and Inspection by Vision
IPE	Inversion Pixels Extraction
CRPE	Crest and Root Pixels Extraction
CRPG	Crest and Root Pixels Grouping
CRGC	Crest and Root Groups Classification
CRDC	Crest and Root Data Calculation
STPC	Screw Thread Features Calculation
pd	perpendicular distance

تطوير خوارزمات رؤية بالحاسب لقياس وفحص القلاووظ الخارجي

تعتبر الطرق الحالية لقياس القلاووظ إما مضيعة للوقت أو مكلفة. بالإضافة إلي ذلك لا توجد طريقة واحدة متاحة يمكن استخدامها لقياس كل خصائص القلاووظ بدقة وفي وقت مناسب. ويقدم هذا البحث نظام رؤية بالحاسب لقياس وفحص معظم أنواع القلاووظ المعروفة مع التغلب علي المشكلتين السابقتين. وقد تم عمل وتطوير مجموعة من خوارزميات معالجة الصور وخوارزميات الرؤية بالحاسب اللازمة لتحليل الصور الملتقطة لأسنان الثلاووظ، ومن ثم إجراء عمليتي القياس والفحص. ومن خلال نظام الرؤية بالحاسب المقدم يمكن قياس وفحص معظم خصائص أسنان القلاووظ (18 خاصية). وقد تم معايرة النظام المقدم بحيث يمكن قياس القلاووظ باستخدام كل من النظامين الانجليزي والمتري. كذلك تم اختيار النظام المقدم بقياس قلاووظ قياسي لنظام الأيزو المتري، ثم مقارنة النتائج بالقيم القياسية للقلاووظ. وقد أثبتت النتائج أن أكبر فرق بين القيم المقاسة والقيم القياسية كان في حدود 5.4 ميكرون، مما يؤكد دقي القياس بالنظام المقدم.