www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

archives
of thermodynamics
Vol. **39**(2018), No. 1, 111–128
DOI: 10.1515/aoter-2018-0006

Time performance of RGB to HSI colour space transformation methods

ANDRZEJ ZIEMBA
ELŻBIETA FORNALIK-WAJS*

Department of Fundamental Research in Energy Engineering, Faculty of Energy and Fuels, AGH University of Science and Technology,
30 Mickiewicza Ave., 30-059 Krakow, Poland

Abstract Present paper is a continuation of works on evaluation of red, green, blue (RGB) to hue, saturation, intensity (HSI) colour space transformation in regard to digital image processing application in optical measurements methods. HSI colour space seems to be the most suitable domain for engineering applications due to its immunity to non-uniform lightning. Previous stages referred to the analysis of various RGB to HSI colour space transformations equivalence and programming platform configuration influence on the algorithms execution. The main purpose of this step is to understand the influence of computer processor architecture on the computing time, since analysis of images requires considerable computer resources. The technical development of computer components is very fast and selection of particular processor architecture can be an advantage for fastening the image analysis and then the measurements results. In this paper the colour space transformation algorithms, their complexity and execution time are discussed. The most common algorithms were compared with the authors own one. Computing time was considered as the main criterion taking into account a technical advancement of two computer processor architectures. It was shown that proposed algorithm was characterized by shorter execution time than in reported previously results.

Keywords: Digital image processing; Optical measurement method; RGB to HSI colour space transformation; Linear transformation; Time performance; Relative computational time

*Corresponding Author. Email: elzbieta.fornalik@agh.edu.pl

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

Nomenclature

b	–	relative blue coordinate in Smith's algorithm
$blue$	–	relative blue coordinate in *java.awt.Color* algorithm
B	–	blue coordinate in RGB colour space
cen	–	central component
C	–	cyan coordinate in CMY colour space
$delta$	–	variable in Foley's algorithm
g	–	relative green coordinate in Smith's algorithm
$green$	–	relative green coordinate in *java.awt.Color* algorithm
G	–	green coordinate in RGB colour space
H	–	hue coordinate in HSI, HSV, HSL colour spaces
H_B	–	hue value for $B = max$ in Foley's algorithm
H_G	–	hue value for $G = max$ in Foley's algorithm
H_R	–	hue value for $R = max$ in Foley's algorithm
H_u	–	hue value from out of the range, $[0°,\ 360°]$ in Foley's algorithm
I	–	intensity coordinate in HSI colour space
K	–	black colour defined by (0,0,0) coordinates
KR	–	interval between K and R vertices
KW	–	interval between K and W vertices
L	–	lightness coordinate in HSL colour space
max	–	maximal component, variable in Foley's and *java.awt.Color* algorithm
min	–	minimal component, variable in Foley's and *java.awt.Color* algorithm
M	–	magenta coordinate in CMY colour space
P	–	pink colour defined by (1,1/2,1/2) coordinates
r	–	relative red coordinate in Smith's algorithm
red	–	relative red coordinate in *java.awt.Color* algorithm
R	–	red coordinate in RGB colour space
RW	–	interval between R and W vertices
S	–	saturation coordinate in HSI, HSV, HSL colour spaces
T	–	teal colour defined by (0,1/2,1/2) coordinates
V	–	value coordinate in HSV colour space
W	–	white colour defined by (1,1,1) coordinates
xyI	–	transient coordinate system
X	–	variable in Smith's algorithm
Y	–	yellow coordinate in CMY colour space

1 Introduction

Image analysis can be called as an achievement of the technical development, because its potential to investigate numerous process. From one point of view there are the computers, which fasten the analysis but from another there is an intensive progress in devices recording the images with high resolution.

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

Information coded in the form of images has to be acquired, browsed, analyzed and stored. It can be stored as a raster or vector graphics. The raster graphics consists of many points (pixels) of various colours placed close to each other, which finally give seemingly continuous image (dot matrix data structure called bitmap). The vector graphics basis on the mathematical formulas – each displayed the graphical element is described by some features (location, colour, etc.). The first type of graphics is used in image generating devices (e.g., cameras), while the second one in the graphical software of special applications. The raster images are characterized by a resolution and number of bits per one pixel. The colour is represented by RGB (red, green,blue) colour space and the number of colours depends on the number of bits.

Undoubtedly technical progress influences the utilization of quantitative image (raster graphics) analysis, which can be found in many fields, e.g., medicine, cartography, geology, geodesy, engineering, etc. Authors took part in the analysis of images oriented on getting a quantitative information about temperature and velocity fields by particle image velocimetry (PIV) and particle image thermometry (PIT) with utilization of thermochromic liquid crystals in the convection problems [1–3], similar topics were presented in [4]. The utilization of image analysis in technical application is not limited to them. Medical and engineering application of liquid crystal images was reported in [5,6], determination of phase change phenomena such as freezing of water was shown in [7], whereas boiling was described in [8] and detection of dryout type critical heat flux was presented in [9]. In mentioned applications image analysis directly described searched parameters and the colour-quantity relation. However discussed colour space transformations can be utilized as indirect method of gaining the information. In example, the images obtained by infrared thermography or other optical techniques can be analysed with digital image processing (DIP) to verify the method and to receive unequivocal data. The great achievement of colour RGB to HSI (hue, saturation, intensity) space transformations and proposed own linear algorithm would be verification of theoretical models (e.g., [10,11]) with the optical measurements methods. The idea of this paper was born during the image analysis with application of digital particle image thermometry (DPIT). In this method, for the quantitative data acquisition, the colour information is crucial, in opposite to the luminance or intensity, which are not important. Therefore direct application of RGB colour space is difficult and the colour spaces based on hue, H, (synony-

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

mous: colour, tone, shade, tint) are more suitable. Hue very often is used in the experimental analysis since it is giving a proper information about colour regardless of the shadows or non-uniform lightning. However, there are some disadvantages, which at first come from a number of the available colour spaces and the number of transformations necessary for getting the quantitative information. These disadvantages are mainly connected with proper choice of colour space and then with the time and computer resources needed for the processing. Additional difficulty comes from the fact that in the literature various transformations for the same colour spaces are reported. This paper contains presentation of the most common colour space transformation algorithms and comparison of their computing time. It is a continuation of works [12,13] regarding proposal of simpler RGB to HSI colour space transformation, suitable also for an advanced computer processor architecture. Previous stages considered equivalence of own algorithm with other algorithms and also computational time related to various configurations of programming platform. Present phase of analysis extends towards significance of hardware architecture and its influence on computing time. Due to this it was possible to get the results, which confirmed that authors' idea of colour space transformation does not depreciate with technical development (like one presented algorithm), but in contrary it seems to be more valuable. It should be emphasized that some of RGB to HSI space transformations shown here have never appeared in the open form and they were transcripted from the source program. The way of computing time comparison is also the original idea of authors.

2 Colour spaces

The colour spaces RGB and CMY (cyan, magenta, yellow) are devices' oriented. The first space is displays oriented, while the second one is printer oriented. RGB colour space is coming from the camera sensors, screens and eyes ways of operation [14]. The space defined by RGB has a form of colour cube, in which independent signals – red, green, blue can represent any colour [15]. The 20th century state of art about digital colour imaging can be found in [16].

In the range of authors' interest is the Hue value, due to the reasons mentioned above – reduction of variables number, effectiveness and also better suitability of the colour spaces referring to the human colour sensation. That is why, the available colour spaces are as follows: HSV (hue,

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

saturation, value), HSI (hue, saturation, intensity), and HLS (hue, lightness, saturation). These colour spaces refer to the human eye perception: tint (hue+white), shade (hue+black) and tone (hue+grey). Saturation represents 'amount' of colour, therefore the difference between the red and pink colours can be distinguished. The third variable – value, lightness or intensity, refers to the amount of light, which enables determination of light and dark colours or the levels of grey.

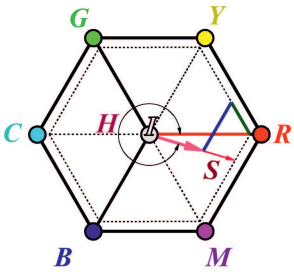


Figure 1: Colour wheel.

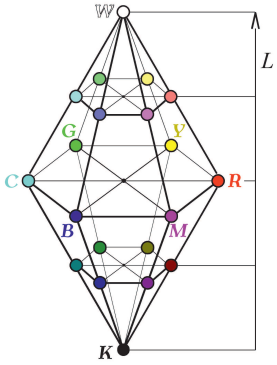


Figure 2: Double cone of RGB gamut.

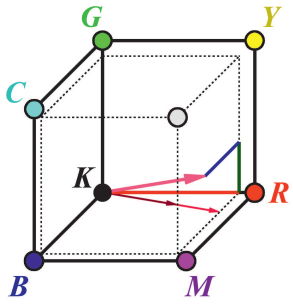


Figure 3: RGB cube.

The hue is a property of pure colour, very often identified with the colour itself, which helps to distinguish particular colours among the others. It is closely connected with the light wavelength. Hexagonal structure (Fig. 1), inscribed in the circle creates the colour wheel, in which angle of pure colour represents hue H. Each pure colour can be represented by a mixture of two neighbouring colours. To illustrate lightness, the additional parallel circles of decreasing diameter should be added above and below the basic colour circle. The double cone of colour is created (Fig. 2). The circles placed above the basic circle, tend towards the white colour, therefore they become lighter. The circles placed below the basic one, tend towards the black colour becoming darker. The axis of such body corresponds to the grey scale from black to white colours and it describes the lightness axis. Distance between the particular colour and this axis is a measure of saturation. The angle between two planes: the one containing this axis and red colour and the second containing this axis and any other colour, is called this colours hue [17]. Value can be illustrated by height of inverted cone, while intensity

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

by diagonal of RGB cube (Fig. 3).

The RGB colour space is applied in the process of image generation (image capturing or displaying) but its suitability for colour describing in a way understandable by human eyes is limited. In HSI colour space, intensity – the achromatic component is separated from the chromatic ones giving the information about colour – hue and saturation. This feature specifies the HSI system as an ideal tool for development of the image processing algorithms based on the natural colour description intuitive for human being.

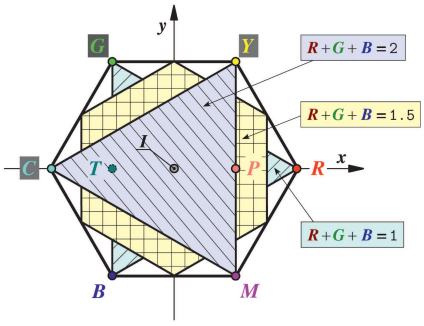


Figure 4: Intersections of colour cube by the planes perpendicular to intensity axis I.

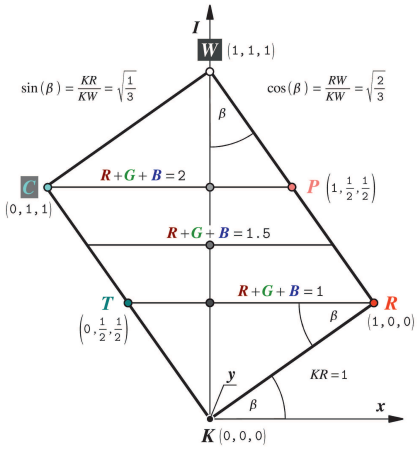


Figure 5: Intersection of RGB colour cube by xI plane.

Colour image recorded as three monochromatic, various intensity components (red, green, blue) is a source of information about the total intensity. If the colour cube is oriented in such a way, that black colour $K = (0, 0, 0)$ is at the bottom and forms the first point of cube main diagonal. This diagonal is a vertical axis of intensity I, connecting white and black colours, perpendicular to the base xy-plane (Fig. 4). From the transient xyI coordinate system, the HSI system can be derived. To find the saturation component, S, of any colour, the axis perpendicular to intensity should be drawn. The intersection point has the intensity value belonging to the range (0,1). The saturation (purity) of colour increases with increasing distance from the intensity axis. On the intensity axis the

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

saturation is equal to zero, because there are placed only grey colours [18]. In Fig. 4 three planes intersecting perpendicularly the intensity axis and projected on the base xy-plane are presented. They are also shown in Fig. 5 but from another perspective, which help to understand the orientation in space. The large number of colour spaces applied in the various fields is coming from a different needs of researchers [18].

3 Common transformations of RGB to HSI colour space

In digital image processing transformations coming from geometrical relations between cartesian RGB colour space and xyI space are applied very often. As it was mentioned in previous sections, the xyI space is constructed by the projection of colour cube in the direction of its main diagonal (interpreted as an intensity I axis) on the xy-plane perpendicular to it. This transformation presented in [17–21] is non-linear (requires mathematical operations with application of trigonometrical functions, raising to a power or extraction of roots). During the image processing there is a need of multiple determinations of HSI values, what can significantly increase time of necessary calculations. Moreover, the non-linear transformation has a plane of singularity $2R - G - B = 0$, what is an additional difficulty. In the scientific literature regarding digital image processing the examples of linear transformation of RGB to HSI can be found, but they are presented without derivation of formulas.

The linear transformations are obtained on the basis of assumption that in the range, where one of RGB components is maximal and another one is minimal, the value of hue can be approximated by linear function of third component equal to the central value.

The hue and saturation components of HSI colour space coming from non-linear transformation reported by [14] have their origin in the linear transformation. Many researchers are using the linear transformation formula to the saturation calculations [21].

Smith in [22] presented an example of linear transformation with entire derivation of formulas and their geometrical interpretation. Instead of intensity I he was using the maximal brightness called value V. The RGB values are the input parameters, while HSV values – output ones. All numbers are normalized to the range (0,1). The concept of Smith's transformation was presented in the form of following algorithm:

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

1. $V = \mathrm{Max}\,(R, G, B)$; $X = \mathrm{Min}\,(R, G, B)$, (authors' comment – supplement for black colour $K = (0,\ 0,\ 0)$ if $V = 0$; exit from procedure)
2. $S = \frac{(V-X)}{V}$; if $S = 0$, if there is achromatic colour; exit from procedure,
3. $r = \frac{V-R}{V-X}$; $g = \frac{V-G}{V-X}$; $b = \frac{V-B}{V-X}$,
4. if: $R = V$, then:
   if: $G = X$, then
   $H = 5 + b$,
   else:
   $H = 1 - g$; go to 5,
   if: $G = V$, then:
   if: $B = X$, then:
   $H = 1 + r$,
   else:
   $H = 3 - b$; go to 5,
   if: $B = V$, then:
   if: $R = X$, then:
   $H = 3 + g$,
   else:
   $H = 5 - r$,
5. $H = \frac{H}{6}$.

Smith [22] introduced the formula defining saturation S as $S = [V - \mathrm{Min}\,(R, G, B)]\,/V$ and relations describing colour angle hue H in individual triangles of hexagon, consistent with the algorithm presented above. The hue H value indicates the pure colours located on the sides of the largest regular hexagon of $V = 1$ size [22].

The other example of linear transformation of RGB colour space to HSV one was described by Foley [23], who presented it in the form of a program written in C language, however the derivation was omitted. In his case the variable V is slightly different than the intensity I. The following algorithm was transcripted by authors from Foley's program [23] to prepare the form suitable for comparison with the other algorithms:

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

1. R, G, B values are given as input parameters normalized to $[0,\ 1]$,
2. $V = max = \mathrm{Max}\,(R, G, B)$; $min = \mathrm{Min}\,(R, G, B)$,
3. if max$= 0.0$, then $S = 0.0$;

   else $delta = max - min$, $S = delta/max$,
4. if $S = 0.0$, then $H = H_u$ the hue value is out of the range $[0°, 360°]$, and the algorithm gives negative value

   $(H, S, V) = (\,(H_u < 0°) \vee (H_u > 360°)\,,\ \ 0,\ \ 0)$;

   else the colour is chromatic and procedure is continued,
5. if $R = max$, then colour is within the red R colour zone between the yellow Y and magenta M colours,

   $H_R = 0$; $H = H_R + \frac{(G-B)}{delta}$;

   if $G = max$, then colour is within the green G colour zone between the cyan C and yellow Y colours,

   $H_G = 2.0$; $H = H_G\ +\ \frac{(B-R)}{delta}$;

   if $B = max$, then colour is within the blue B colour zone between the magenta M and cyan C colours,

   $H_B = 4.0$; $H = H_B\ +\ \frac{(R-G)}{delta}$,
6. conversion to the range $[0°, 360°]$ and assurances of positive value,

   $H\ \ \leftarrow\ \ 60° \cdot H$

   if $H < 0°$, then $H \leftarrow 360° + H$.

As the output parameters the values of $H \in [0°, 360°]$ and $S,\ V \in [0, 1]$ are obtained.

The third example of discussed algorithm is connected with Java platform. In accordance with documentation of programming Java platform [24] the class *java.awt.Color* includes a procedure, which allows transformation of RGB colour space to *HSB* one, in which the brightness B values, similar to the definition of value V is determined in the same way (Java classes are template for object you are going to create). Based on the class source code, authors wrote down the algorithm of colour space transformation. As input parameters the R, G, B values are given in a form of integer number from the range [0, 255], while as an output parameters the values of H, S, B in a form of floating point numbers, normalized to the range [0,1] are obtained. The algorithm is as follows:

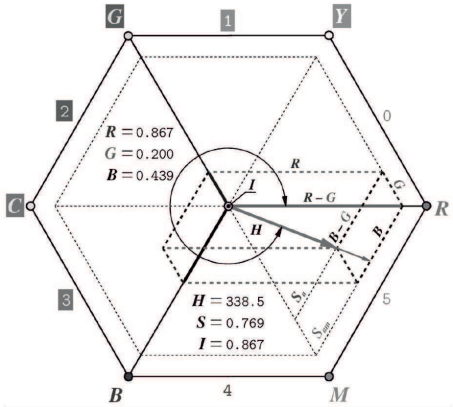
www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

1. $max = \mathrm{Max}\,(R, G, B)$; $min = \mathrm{Min}\,(R, G, B)$,
2. $B = \frac{max}{255}$,
   if $B \neq 0$, then $S = \frac{max - min}{max}$;
   if $B = 0$, then $S = 0$;
3. if $S = 0$, then $H = 0$; return,
4. $red = \frac{max - R}{max - min}$; $green = \frac{max - G}{max - min}$; $blue = \frac{max - B}{max - min}$,
5. if: $R = max$, then
   $H = 0.0 + blue - green$;
   if: $G = max$, then
   $H = 2.0 + red - blue$;
   if: $B = max$, then
   $H = 4.0 + green - red$;
6. $H = \frac{H}{6.0}$
7. if: $H < 0$, then
   $H = H + 1.0$;
8. return.

In the algorithm based on Foley's program [23], the colour angle hue H belongs to the range $[0°, 360°]$, while the java procedure RGBtoHSB [24] and algorithm coming from Smith's program [22] are giving values from the range [0,1]. When saturation is equal to zero $S = 0$, in Smith's program hue H value is undefined, in RGBtoHSB procedure the hue value equals to zero and in Foley's program the Hue reaches the value outside the range of $[0°, 360°]$.

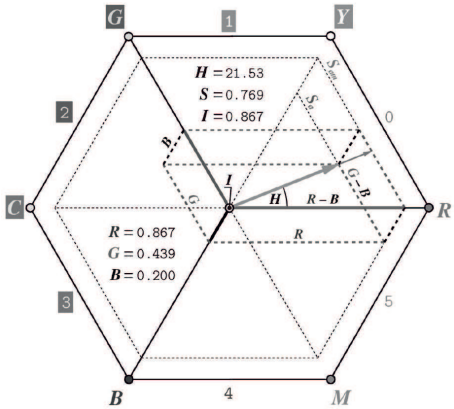
4 Own linear transformation of RGB to HSI colour space

The number of available RGB to HSI colour space transformations raises a question regarding the reasons of such variety and an equality of the results. Therefore authors decided to go through their derivations and to compare the results. During the process new transformation was found and it was presented in [13]. This own linear transformation is just recalled



R	Y	G	C	B	M
0°	60°	120°	180°	240°	300°

Figure 6: The colour example in triangle No. 5.



R	Y	G	C	B	M
0°	60°	120°	180°	240°	300°

Figure 7: The colour example in triangle No. 0.

here and presented together with graphical representation; for the detail information, please refer to mentioned paper.

Figures 6 and 7 present colour hexagon (similar like in Fig. 1) divided into six equilateral triangles, marked from 0 to 5. Two neighbouring triangles (being next to each other) having the same side and laying on the projection of one of RGB system axis (segment of solid lines shown in Figs. 6 and 7) form rhombus. Inside of them there are the projections of colours characterized by maximal value of one of primaries. The values of RGB primary components in particular triangles of hexagon from maximal (*max*), trough central (*cen*) to minimal (*min*) are listed in Tab. 1.

Table 1: Colour basic component values in dependence on triangle number [13].

Colour of rhombus	No. of triangle	Component		
		R	***G***	***B***
red	5	*max*	*min*	*cen*
	0	*max*	*cen*	*min*
green	1	*cen*	*max*	*min*
	2	*min*	*max*	*cen*
blue	3	*min*	*cen*	*max*
	4	*cen*	*min*	*max*

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

Definition of colour intensity I generally applied has the following form [17,18,19]

$$I = max = max\ (R, G, B)\ . \tag{1}$$

Saturation S is a measure of colour purity and measure of a distance from intensity I-axis [22]. It can be calculated from the equation

$$S = \ 1 - min/max\ . \tag{2}$$

Hue H is the most difficult to define since there are six equations describing it in particular triangles forming colour hexagon. They are as follows for triangle 5, triangle 0, triangle 1, triangle 2, triangles 3 and 4, respectively:

$$H = 60^\circ \cdot (-\ (cen - min)\ /\ (max - min)\ +\ 6)\ , \tag{3a}$$

$$H = 60^\circ \cdot (+\ (cen - min)\ /\ (max - min)\ +\ 0)\ , \tag{3b}$$

$$H = 60^\circ \cdot (-\ (cen - min)\ /\ (max - min)\ +\ 2)\ , \tag{3c}$$

$$H = 60^\circ \cdot (+(cen - min)\ /\ (max - min)\ +\ 2)\ , \tag{3d}$$

$$H = 60^\circ \cdot (-\ (cen - min)\ /\ (max - min)\ +\ 4)\ , \tag{3e}$$

$$H = 60^\circ \cdot (+(cen - min)\ /\ (max - min)\ +\ 4)\ . \tag{3f}$$

All Eqs. (3a)–(3f) form set of equations, from which hue H can be calculated in whole range. This set of equations is called *set_3*.

In this paper four linear transformations were presented. Each of them represents different way of going to the target but their equivalence was proved and presented in [13].

5 Calculations

In paper [25] the analysis of time needed for four presented transformations of RGB to HSI colour system was evaluated. Only one example was consistent with described in this paper algorithms and corresponded to the Foley algorithm. It was concluded that for the object recognition purposes the transformation of RGB to HLS is better than to HSI due to application of the luminance L definition instead of intensity I. Luminance, in their opinion, is more suitable for calculations and is giving satisfactory time of computing.

To compare various RGB to HSI colour spaces transformations described in previous sections the special tool was created. This tool was

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

prepared in *java* language and enabled speed testing of conducted transformations for following methods: *empty* – it didn't perform any calculations, that is why it represented theoretically the fastest possible reference transformation; *modJava* – it was a source code of Color.RGBtoHSB method [26], in which the input and output parameters were changed to adopt them to other methods; *set_3* – represented proposed equations system (3); *Smith* – regarded to the Smith's algorithm; *stdFoley* – indicated the standard Foley's algorithm; *optFoley* – referred to the optimized Foley's algorithm of 25%–40% better performance [27]. Applied tool six time calculated HSI values for each transformation in 256^3 points of RGB standard colour cube, which corresponded to 49 images of 1920×1080 pixels resolution.

The calculations were conducted on two different computers to valorise an influence of computer hardware configuration on the computational time. Both computers consisted of 8 GB RAM, 64-bit Windows 10 Pro v. 1703 system, in the environment of Java Development Kit(JDK) *v.8 update 141*, disconnected from the Internet during calculations. The first computer included two-cores 3.2 GHz processor of Haswell type, while the second one four-cores 2.4 GHz processor of Kentsville type. Time estimation was done in the basis of *System.nanoTime()* method [26]. To averaged the influence of *java VM* (java virtual machine) operational way on the calculation time, at first the same speed test was performed but without time estimation. In this step the standard option of *VM –server* was used, because in 64-bit system it appeared only in this form [28]. The analysis regarding *–server* and *–client* options in 32-bit system were reported in [13].

The results of six successive calculations were arithmetically averaged for each of the methods and computers separately. Comparison test between all considered methods (except the empty method) was also done and it confirmed that they gave similar HSI results. The differences were at the level of 10^{-12}. Double precision computing was conducted in each case; the results are presented as the percentage relative values. In Fig. 8 the program relative time is presented in accordance with following equation:

$$\text{Program Relative Time} = \frac{\text{methods time}}{\text{empty method time}} . \tag{4}$$

In Fig. 9 the methods relative time is shown representing the relation

$$\text{Method Relative Time} = \frac{\text{methods time} - \text{empty method time}}{\text{modJava time} - \text{empty method time}} . \tag{5}$$

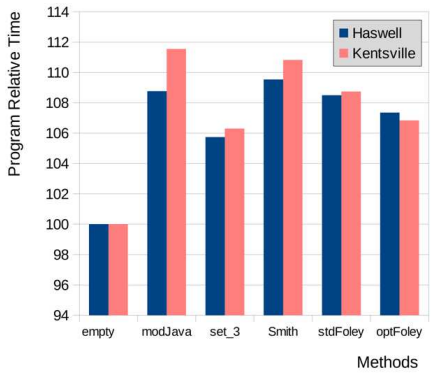


Figure 8: Time in relation to *empty* method.

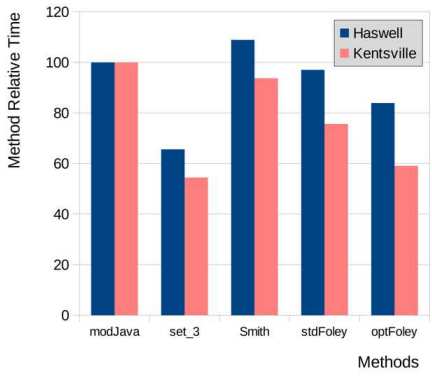


Figure 9: Time in relation to *modJava* method.

It was the attempt to estimation of the relative time of pure transformation methods, which is not precise due to the way of *VM* operation. Figures 10 and 11 base on the same results as presented in Figs. 8 and 9, but in Eqs. (4) and (5) the denominators were changed to the time calculations on the Haswell computer. The equations took the form

$$\text{Program Haswell Relative Time} \;=\; \frac{\text{methods time}}{(\text{empty method time})_{\text{Haswell}}}\,, \tag{6}$$

$$\text{Method Haswell Relative Time} \;= \\ = \frac{\text{methods time} \;-\; \text{empty method time}}{(\text{modJava time} \;-\; \text{empty method time})_{\text{Haswell}}}\,. \tag{7}$$

It was done to highlight the fact that the computations regarding particular methods of colour space transformation on Haswell computer were about 1.85 times faster than on the Kentsville one. The differences between the methods were small, but still clearly visible. It was again confirmed that method *set_3* gave the shortest calculation time.

6 Conclusions

Presented paper summarizes successive stages of works on evaluation of RGB to HSI colour space transformation in regard to digital image processing application in optical measurements methods. Previously it was

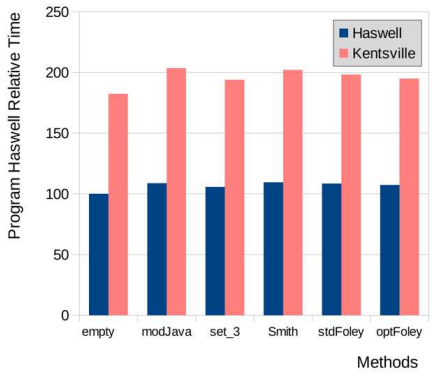


Figure 10: Time in relation to Haswell *empty* method.

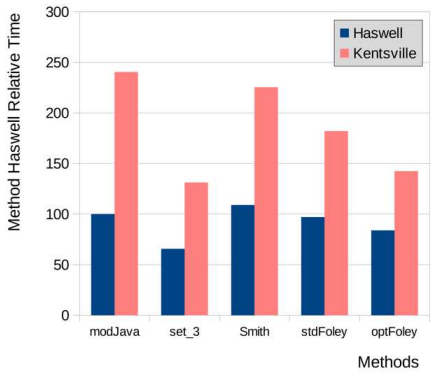


Figure 11: Time in relation to Haswell *modJava* method.

confirmed that all considered methods are equivalent analytically and numerically, but it was also shown that the algorithms execution was influenced by the programming platform configuration, what had its consequences in different computing time. This step focused on understanding of computer processor architecture influence on this parameter, therefore operation of two processors was compared. Computer Haswell was calculating the codes about 1.85 time faster, even if its processor timing was only 1.33 times faster. Additionally Kentsville processor had two times more cores (higher number of cores can speed up the computations, because *VM* executes the code as a multitask). It confirms higher effectiveness of Haswell type processor.

Analysis of the results confirms the conclusions presented in [13] in regard to the calculations with utilization of *VM* server, where the fastest executed codes have methods *set_3* and *optFoley*, while the slowest – the codes of *modeJava* and *Smith*'s methods. Execution of *stdFoley* method was on average level. Presentation of computing time as the relative time allows comparison between the results obtained on these two computers and together with processor timing and core type it can be used to an estimation of the program execution time for other computers.

Methods of *optFoley* is faster than *stdFoley* method of about 28% on Kentsville and 15.6% on Haswell (Fig. 9). It partially confirmed the results presented in [27] and at the same time shows decreasing significance of this optimization for increasing effectiveness of the processors and *VM*. In con-

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

trary the method *set_3* is faster than *optFoley* of about 8% on Kentsville and of about 28% on Haswell.

It can be concluded that image analysis performance is not only a question of applied algorithms but also the programming language and hardware configuration. The results exhibited influence of processor architecture on the algorithms computing time. Due to this and also taking into account the applied language, programming style and also the number of needed operations the significant increase of performance during image analysis can be gained. Additionally it can be pointed out that something like 'aging' of algorithms could be observed. As an example the *optFoley* can be given, which was optimised for particular platform configuration but it does not suit well to advanced processor architecture. Opposite tendency can be seen in the case of *set_3* algorithm, for which considerable improvement was observed. It seems to be an interesting proposal for demanding image analysis.

Acknowledgement The present work was supported by the Polish Ministry of Science (Grant AGH No. 11.11.210.312).

Received 20 October 2017

References

[1] Fornalik E., Leiner W., Szmyd J.S., Kowalewski T.: *Experimental simulation of mixed convection*. 4th Baltic Heat Transfer Conference, Kaunas, Lithuania, 2003, Begell House Inc., Advances in Heat Transfer Engineering, New York, ISBN 1-56700-198-1, ISBN 9986-492-78-5, 339–346.

[2] Fornalik E., Nakabe K., Yamamoto Y., Chen W., Suzuki K.: *Visualization of heat transfer enhancement regions modified by the interaction of inclined impinging jets into crossflow*. MG&V **8**(1999), 4, 597–609.

[3] Bednarz T., Fornalik E., Tagawa T., Ozoe H., Szmyd J.S.: *Convection of paramagnetic fluid in a cube heated and cooled from side walls and placed below a superconducting magnet: comparison between experiment and numerical computations*. Therm. Sci. Eng. **14**(2006), 4, 107–114.

[4] Kowalewski T.A.: *Particle Image Velocimetry and Thermometry using Liquid Crystals*. In: Proc. FLUVISU 99, 8me colloque nationale de visualisation et de traitment d'images en mecanique des fluids, Toulouse, June 1-4, 1999, ENSICA, 33–48, http://fluid.ippt.pan.pl/papers/fluvis99_tkowale.pdf

[5] Stasiek J., Jewartowski M., Kowalewski T.A.: *The use of liquid crystal thermography in selected technical and medical applications*. J. Crystall. Process . Technol. **4**(2014), 1, 46–59.

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

[6] Stasiek J., Stasiek A., Jewartowski M., Collins M.W.: *Liquid crystal thermography and true-colour digital image processing*. Opt. Laser Technol. **38**(2006), 4–6, 234–256.

[7] Kowalewski T.A., Cybulski A., Rebow M.: *Particle Image Velocimetry and Thermometry in Freezing Water*. In: Proc 8th Int. Symp. Flow Visualization, Sorrento, Sept. 1-4, 1994.

[8] Piasecka M., Strąk K., Maciejewska B.: *Calculations of flow boiling heat transfer in a minichannel based on liquid crystal and infrared thermography data*. Heat Transfer Eng. **38**(2017), 3, 332–346.

[9] Mikielewicz D., Wajs J., Gliński M., Zrooga A.-B.R.S.: *Experimental investigation of dryout of SES 36, R134a, R123 and ethanol in vertical small diameter tubes*. Exp. Therm. Fluid Sci. **44**(2013), 556–564.

[10] Mikielewicz D., Mikielewicz J., Wajs J., Gliński M.: *Modelling of dryout process in an annular flow*. Heat Transfer Res. **39**(2008), 7, 587–596 (doi:10.1615/HeatTransRes.v39.i7.30).

[11] Wajs J., Mikielewicz D.: *Determination of dryout localization using a five-equation model of annular flow for boiling in minichannels*. Arch. Thermodyn. **38**(2017), 1, 123–139 (doi: 10.1515/aoter-2017-0007).

[12] Ziemba A., Fornalik-Wajs E.: *Transformation of colour space dedicated to an experimental analysis fulfilling the applicability criteria*. J. Phys. Conf. Ser. **530**(2014), 012047, (doi:10.1088/1742-6596/530/1/012047).

[13] Ziemba A., Fornalik-Wajs E.: *Evaluation of colour space transformation suitability to optical temperature measurements*. J. Physics Conf. Ser. **745**(2016), 032108, (doi: 10.1088/1742-6596/745/3/032108)

[14] Kowalewski T.A.: *Thermochromic Liquid Crystals, in Handbook of Experimental Fluid Mechanics*. (C Tropea, A Yarin and J F Foss Eds.), Berlin, Heidelberg: Springer–Verlag, Chapt. B7.1, 487–500, 2007.

[15] Russ J.C.: *The Image Processing Handbook,* 4th edn. CRC Press LLC Boca Raton, London, New York, Washington D.C. 2002.

[16] Sharma G.: *Digital Color Imaging Handbook*. CRC Press, Boca Raton 2002.

[17] Bunting F.: *The ColorShop Color Primer*. Light Source Computer Images, Inc. An X-Rite Company, 1998.

[18] Gonzalez R.C., Woods R.E.: *Digital Image Processing*, 2nd Edn. Prentice Hall Upper Saddle River, New Jersey 2002.

[19] Park H.G., Dabiri D., Gharib M.: *Digital Particle Image Velocimetry/Thermometry and Application to the Wake of Heated Circular Cylinder*. Exp. Fluids **30**(2001), 3, 327–338.

[20] Dabiri D., Gharib M.: *Digital particle image thermometry: The method and implementation*. Exp. Fluids **11**(1991), 2-3, 77–86.

[21] Hay J.L., Hollingsworth T.H.: *A comparison of trichromic systems for use in the calibration of polymer-dispersed thermochromic liquid crystals*. Exp. Therm. Fluid Sci. **12**(1996), 1, 1–12.

www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

[22] Smith A.R.: *Color Gamut Transform Pairs*. In: SIGGRAPH '78 Proc. 5th annual Conf. on Computer Graphics and Interactive Techniques (New York: ACM), Vol. 12, 12–19.

[23] Foley J.D. van Dam A., Feiner S.K., Hughes J.F.: *Computer Graphics: Principles and Practice in C, 2nd Edn.* Addison–Wesley, 1996.

[24] http://www.oracle.com/technetwork/java/javase/downloads/index.html (accessed 16.03.2018)

[25] Palus H., Bereska D.: *The comparison between transformations from RGB colour space to IHS colour space, used for object recognition*. In: Proc. 5th Int. Conf. Image Processing and its Applications, Edinburgh, July 4-6, 1995, doi:10.1049/cp:19950775.

[26] http://docs.oracle.com/javase/8/docs/api/index.html

[27] http://lolengine.net/blog/2013/01/13/fast-rgb-to-hsv (accessed 16.03.2018)

[28] http://docs.oracle.com/javase/8/docs/technotes/guides/vm/server-class.html (accessed 16.03.2018)