Electronic Supplementary Information

Time-gated multi-dimensional luminescence thermometry *via* carbon dots for precise temperature mobile sensing

Sílvia F. V. Silva,[1,2] Gonçalo Figueiredo,[2,3] Rui F. P. Pereira,[4] Verónica de Zea Bermudez,[5] Lianshe Fu,[1,2] Paulo S. André,[3] Albano N. Carneiro Neto,[1,2,] Rute A. S. Ferreira[1,2,*]*

[1] Department of Physics, University of Aveiro, 3810-193 Aveiro, Portugal

[2] CICECO, Aveiro Institute of Materials, University of Aveiro, 3810-193 Aveiro, Portugal

[3] Department of Electrical and Computer Engineering and Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal

[4] Chemistry Center and Chemistry Department, University of Minho, 4710-057 Braga, Portugal

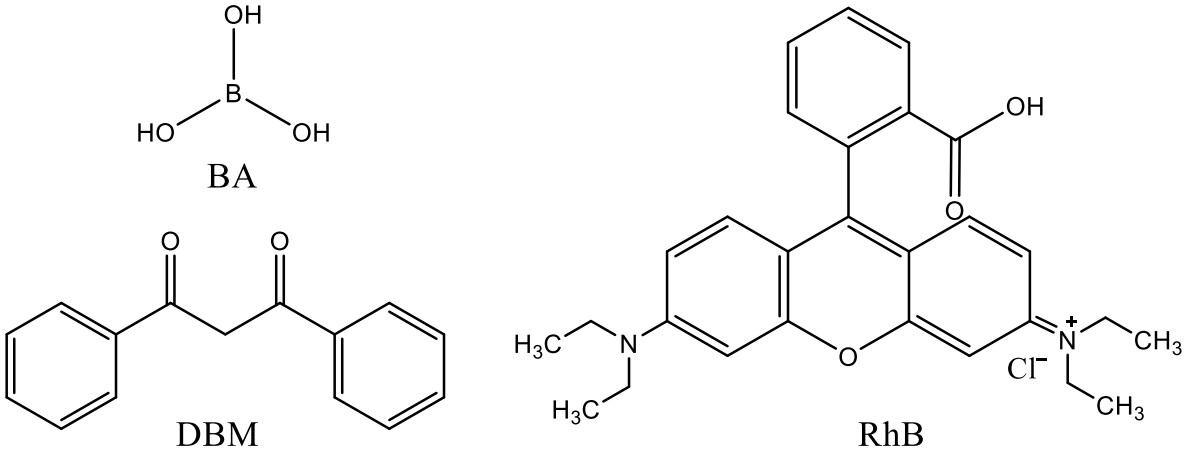
[5] Chemistry Department and CQ-VR, University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal

Contents

1. Materials and methods

1.1. Materials

Dibenzoylmethane (DBM, ABCR), rhodamine B (RhB, Sigma-Aldrich), boric acid (BA, ABCR), ethanol (EtOH, VWR), and sodium hydroxide (NaOH, VWR) were used as received. The molecular structures of DBM, RhB and BA are shown in **Scheme S1**.



Scheme S1. Molecular structures of the precursors.

1.2. Synthesis

To obtain carbon dots (CDs) with long persistent phosphorescence emission, we used a solid state reaction by mixing DBM or RhBCDs with BA, calcined at high temperatures. The introduction of an electron-withdrawing B atom with vacant p orbitals facilitates the intersystem crossing and increasing triplet excitons. In addition, covalent B−C bonds between CDs, structural rigidity, and spatial confinement within the B_2O_3 polycrystalline network also restrict nonradiative transitions, isolating quenchers from the environment and preserving the CDs' phosphorescence.[1] To ensure the homogeneity of final CDs and reproducibility of the experiments, DBM or RhBCDs was added to H_3BO_3 in solution, guaranteeing that a small amount of DBM or RhBCDs was added precisely. In addition, the experimental conditions were carefully controlled, which endows homogeneity to the final CD composites, CDs@B_2O_3, i.e., the appropriate embedding of the CDs in the B_2O_3 host. From the above measures, the CD composites prepared from distinct batches showed the same photoluminescece profiles.

1.2.1. Synthesis of CDs e-tag CD_A

CD_A was synthesized according to literature.[1] Typically, 10.0 mg of DBM (0.40 mL of DBM in EtOH, 25.0 mg/mL) was mixed with 1.0 g of BA. The mixture was mixed well and calcinated at 225 °C for 15 min, taken out from the oven, and cooled down to room temperature naturally.

1.2.2. Synthesis of CDs RhBCDs

RhBCDs were synthesized according to literature.[2] Typically, 96 mg of RhB was dissolved in 15 mL of a 0.67 M NaOH solution with ultrasonic treatment. The clear solution was transferred into a poly(tetrafluoroethylene)-lined autoclave (20 mL) and heated at 180 °C for 8 h and cooled down to room temperature naturally. The resulting aqueous solution was diluted to 20 mL with water and named as RhBCDs.

1.2.3. Synthesis of CDs e-tag CD_B

CD_B was synthesized according to literature.[3,4] Typically, 0.25 mL of RhBCDs aqueous solution was mixed with 2.0 g BA. The solid was ground well and dried at 90 °C to get a dried powder. The powder was calcinated at 235 °C for 15 min, moved out of the oven, and cooled down to room temperature naturally.

1.2.4. Preparation of CDs e-tags CD_{AB}

E-tags CD_{AB} were obtained by mixing of CD_A and CD_B in a weight ratio (A:B) of 1:1, 1:4, and 4:1, respectively. These mixtures were ground thoroughly, and pellets were made for further characterization.

1.2.5. Synthesis of undoped sample

For control experiment, 2 g of BA was also calcined at 230 °C for 15 min to obtain the heated BA.

1.3. Characterization

Transmission electron microscopy (TEM) images were obtained at the Iberian Nanotechnology Laboratory using a JEOL JEM 2100 (200 kV) microscope. Samples CD_A and CD_B were dispersed in water and placed into the analyzing grids (UC-A on holey 400 mesh Cu grids, Ted Pella ref. 01824) by drop-casting, and dried at room temperature. Fourier transform infrared (FT-IR) spectra from 4000 to 400 cm^{-1} with 64 scans and 2 cm^{-1} resolution were obtained by using the potassium bromide pellet technique on JASCO FT/IR-4X FTIR Spectrometer. The morphologies and the sizes that investigated by High-resolution TEM, the

surface compositions that examined by XPS and the components of the CDs that studied by Raman spectra were carried out previously [1].

UV–visible absorption spectra were measured on Perkin Elmer Lambada 950 UV/VIS/NIR spectrometer in the range of 200 to 700 nm with scan speed of 110 nm/min at room temperature. The photoluminescence spectra were recorded with a modular double-grating excitation spectrofluorimeter with a TRIAX 320 emission monochromator (Fluorolog-3, Horiba Scientific) coupled to an R928 Hamamatsu photomultiplier. The spectra were corrected for the detection and optical spectral response of the spectrofluorometer. The emission quantum yield values were measured at room temperature using a system (C9920-02, Hamamatsu) with a 150 W xenon lamp coupled to a monochromator for wavelength discrimination, an integrating sphere as the sample chamber, and a multichannel analyzer for signal detection. The method is accurate to within 10%.

An exploration of the Scopus platform using search terms such as "*Carbon Dots*", "*thermometry, thermometer, thermometric*", and "*luminescent, luminescence, emission*" in the title, abstract, and keywords reveals only 36 papers since 2015, as shows **Figure S1a**. Notably, this search neglects the pioneering works by Yu *et al.*,[5] as it deviated from the specific terminology by utilizing "*temperature-dependent*" instead of "*thermometry*" and its derivatives.

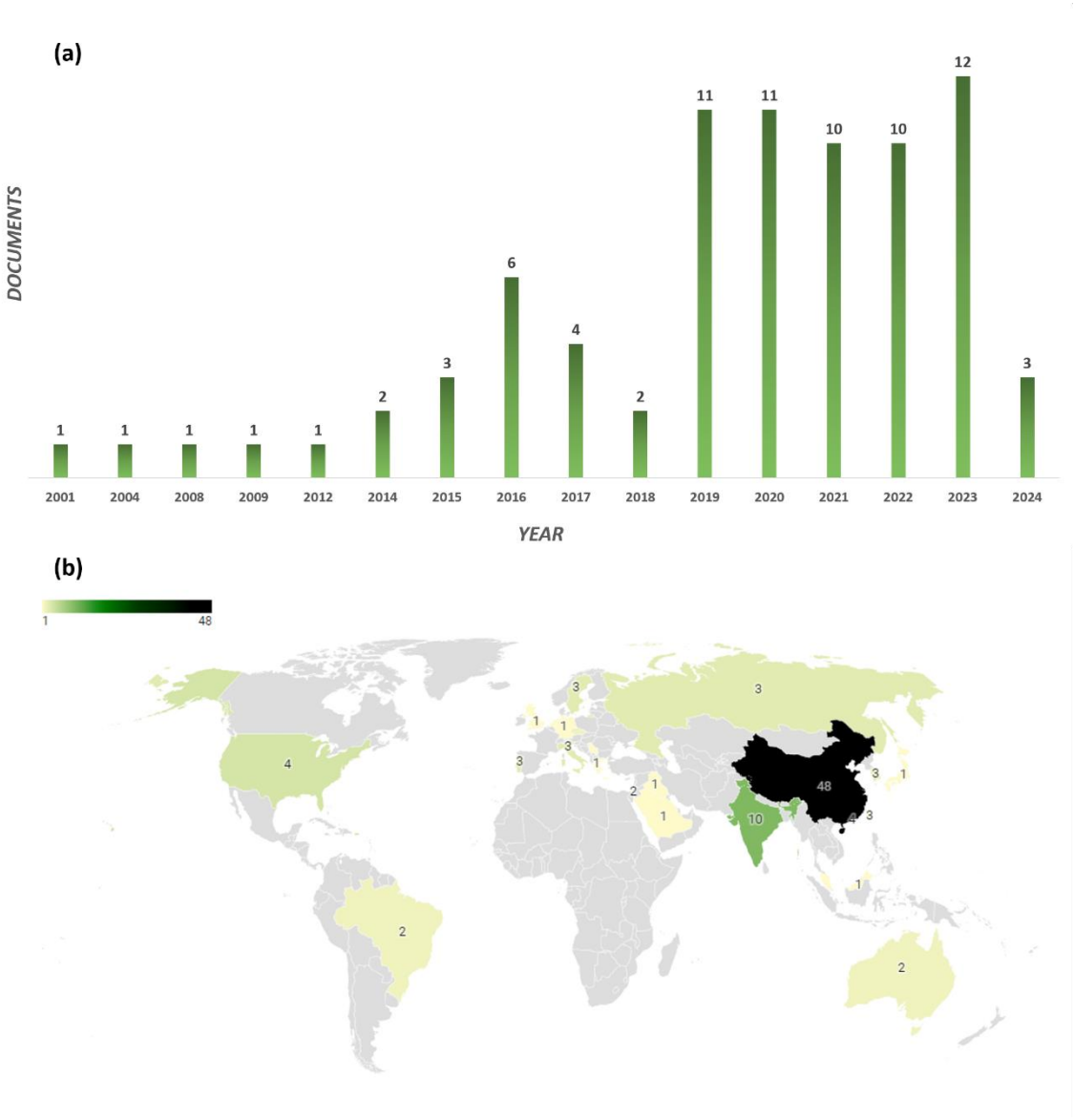


Figure S1. Literature search in the Scopus database for the terms "*Carbon Dots*", "*thermometry*, *thermometer, thermometric*", and "*luminescent, luminescence, emission*" in the title, abstract, and keywords. a) Histogram of the number of papers per year. b) World map representing the distribution of the number pf papers per country.

2. Transmission electronic microscopy (TEM)

The TEM images of CD_A and CD_B are shown in **Figure S2**. The presence and structural characterization of CDs in CD_A and CD_B was previously reported in a recent article of some of us.[1] High-resolution TEM images reveal the presence of CDs with a nearly spherical shape and average diameter sizes of 4±1 and 5±1 nm for CD_A and CD_B, respectively. Fast Fourier Transform analysis of TEM images expose the presence of highly crystalline structures with a lattice fringe spacing of 0.21 nm corresponding to the (100) interplanar spacing of graphene.

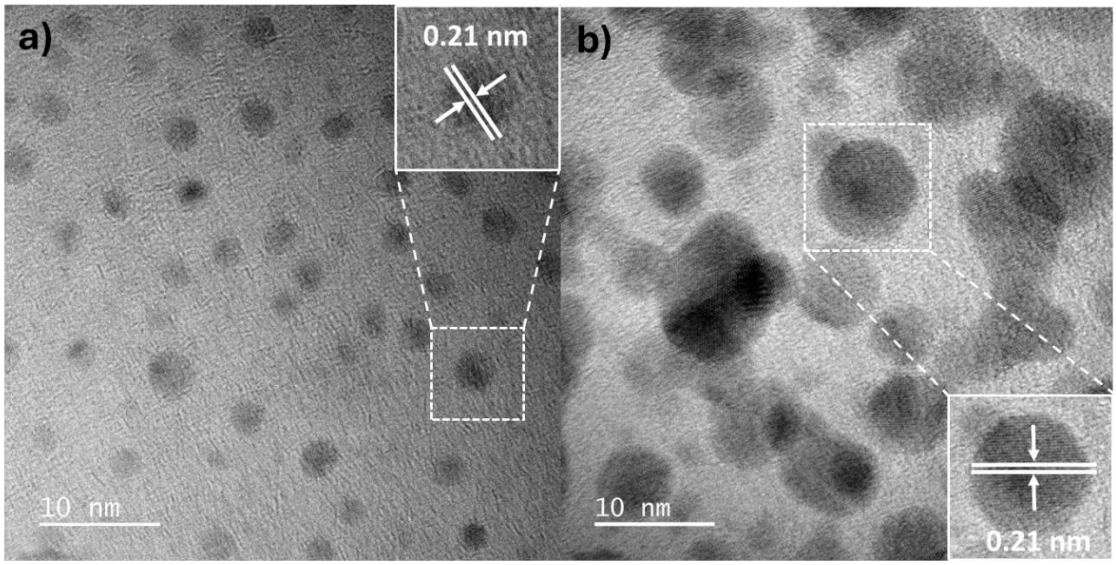


Figure S2. High-resolution TEM images of a) CD_A and b) CD_B. Inset images identify individual CDs showing lattice fringes with *d*-spacing determined by Fast Fourier Transform analysis. [1]

3. FTIR spectra

The FTIR spectra of BA and heated BA are shown in **Figure S3**. According to the literature [6] the peak at 1460 cm^{-1} is attributed to the asymmetric stretching of B–O in the BO_3 structure. The peak at 1196 cm^{-1} is due to B–OH plane bending vibrations. The weak band at 886 cm^{-1} is ascribed to the symmetric stretching vibration of B–O. The broad band at 775 cm^{-1} is assigned to the B–OH out-of-plane bending. The peak at 650 cm^{-1} belongs to the deformation vibrations of the B–O bond. The peak at 550 cm^{-1} is from the O–B–O ring bending. For the heated BA, it shows the similar vibrations to those of the BA. For CD_A and CD_B, they display similar absorption bands to heated BA. The broad band cantered at 3230 cm^{-1} can be assigned to –OH vibration, whereas the bands at 1475 cm^{-1} for sample A, 1460 cm^{-1} for CD_B and 1198 cm^{-1} are formed by the asymmetric stretching vibration of B–O and plane bending vibration of B–OH. The weak peak at around 945 cm^{-1} is the characteristic vibration from B–C bond, indicating that the CDs in situ synthesised are covalently bonded to the B_2O_3 host, which has been reported in other CDs based composites.[7]

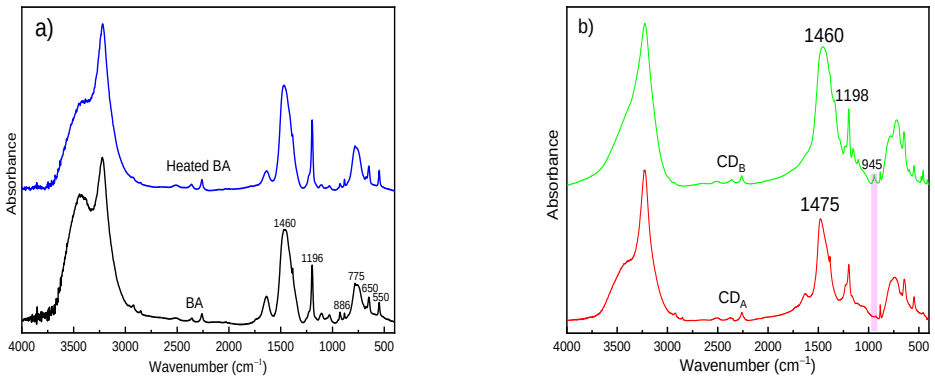


Figure S3. FTIR spectra of BA and heated BA (a) and CD_A and CD_B (b).

4. UV-visible absorption spectra

Figure S4 shows the UV–visible absorption spectra of Heated BA, CD_A and CD_B. From 200 to 700 nm, there is nearly no absorption for Heated BA. The peak at 250 nm for sample **A** and at 230/275 nm for sample B are attributed to the π–π* transitions of C=C bonds, while the peak at 350 nm for CD_A and the weak peak at 315 nm for CD_B are attributed to the n–π* transitions of C=O bonds, of the CDs derived from different organic molecules. The broad band in the visible region at 484 nm with a shoulder at 455 nm for CD_B are associated with the absorptions from RhB. These visible absorption bands lead to the fact that these CDs can be excited with a visible light and give persistent luminescence at room temperature.

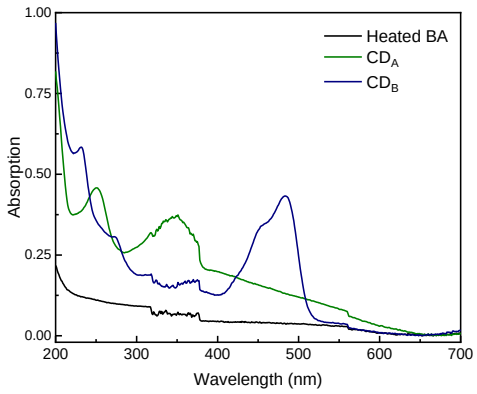


Figure S4. UV–visible absorption spectra of Heated BA, CD_A and CD_B. Concentrations: 10 mg/mL aqueous solutions for Heated BA and CD_B and 2.5 mg/mL aqueous solution for CD_A. 1 cm quartz cell was used for all the measurements.

5. Photoluminescence spectra

CD_A displays a broad excitation band from 240 to 420 nm (**Figure S5**). Upon excitation with different wavelength UV light, CD_A shows excitation-independent emission with a maximum emission wavelength of 443 nm, which indicates that CD_A has homogeneous surface states. The relative intensity depends only on the selected excitation wavelength. The emission band extends from around 400 to higher than 600 nm, giving a bluish color with CIE of (0.21,0.25) (**Figure S7**).

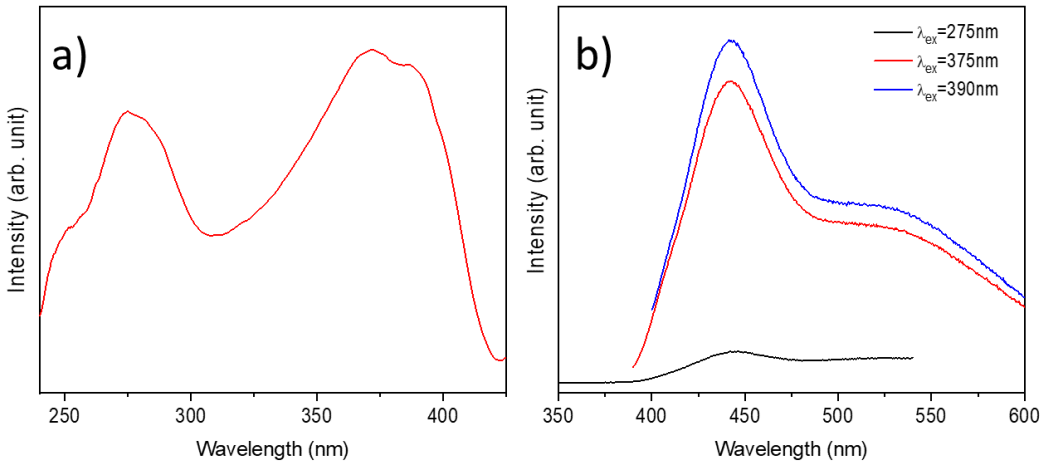


Figure S5. Excitation (a, λ_{em} = 443 nm) and emission (b, different excitation wavelengths) spectra for CD_A.

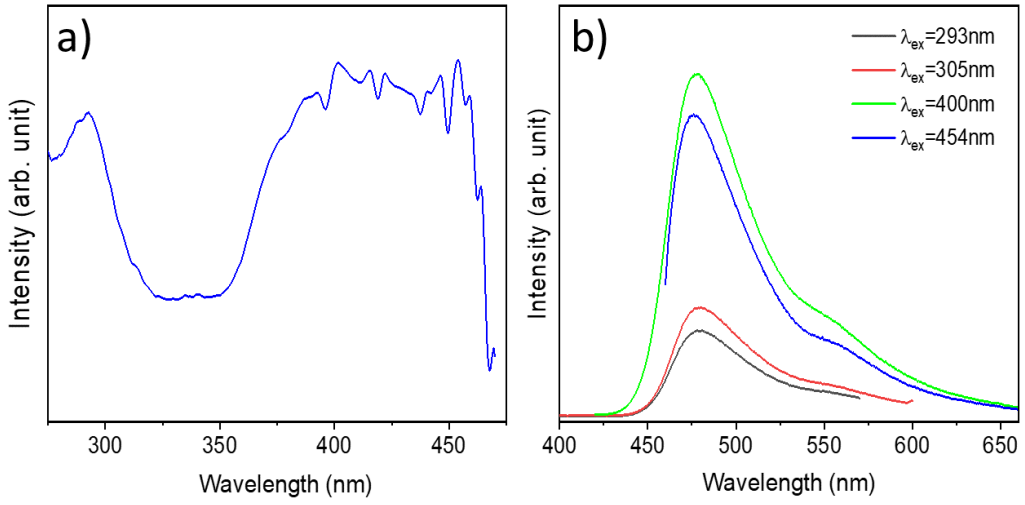


Figure S6. Excitation (a, λ_{em} = 479 nm) and emission (b, different excitation wavelengths) spectra for CD_B.

Compared to CD_A, CD_B also shows broad excitation band, extending to visible spectral region. Upon excitation with either UV light (293 and 305 nm), near-UV light (400 nm) or visible light (454 nm), CD_B also exhibits excitation-independent emission with the maximum

emission wavelength of 479 nm (**Figure S6**), a similar luminescence feature to CD_A (**Figure S5**). The emission band ranges from 425 to 650 nm, showing a cyan color with CIE of (0.20,0.35) (**Figure S7**).

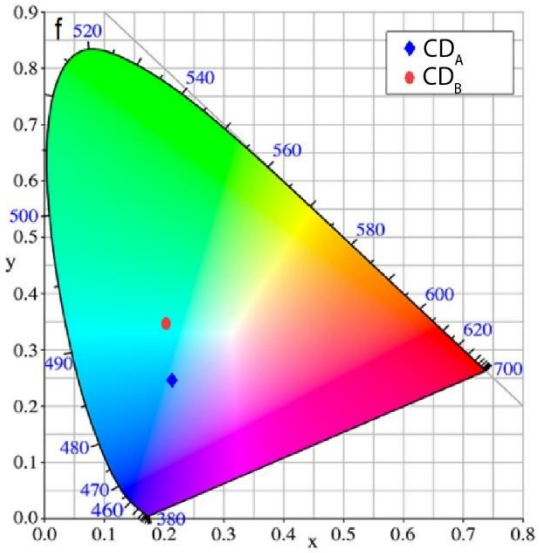


Figure S7. CIE chromaticity diagram of CD_A and CD_B.

The quantum yield (QY) values were measured for CD_A and CD_B in powder under different excitation wavelengths and summarised in **Table S1**. Contrary to the aggregation-induced quenching for most CDs in the solid state, CD_A and CD_B display strong emission with a high QY value of 0.35±0.03. It should be noted that the mixed e-tags CD_{AB} show even bigger QY values.[1]

Table S1. Absolute emission quantum yield (QY) for samples CD_A, CD_B and CD_{41} excited at different wavelengths (λ_{ex}).

CDs	**λ_{ex} (nm)**	**QY**	**CDs**	**λ_{ex} (nm)**	**QY**	**CDs**	**λ_{ex} (nm)**	**QY**
CD_A	275	0.19±0.02	CD_B	293	0.23±0.02	CD_{41}	375	0.52±0.05
	375	0.33±0.03		305	0.26±0.03		405	0.46±0.05
	390	0.35±0.03		400	0.35±0.03		430	0.39±0.04
				454	0.35±0.03		470	0.28±0.03
							490	0.20±0.02

6. Thermometric analysis

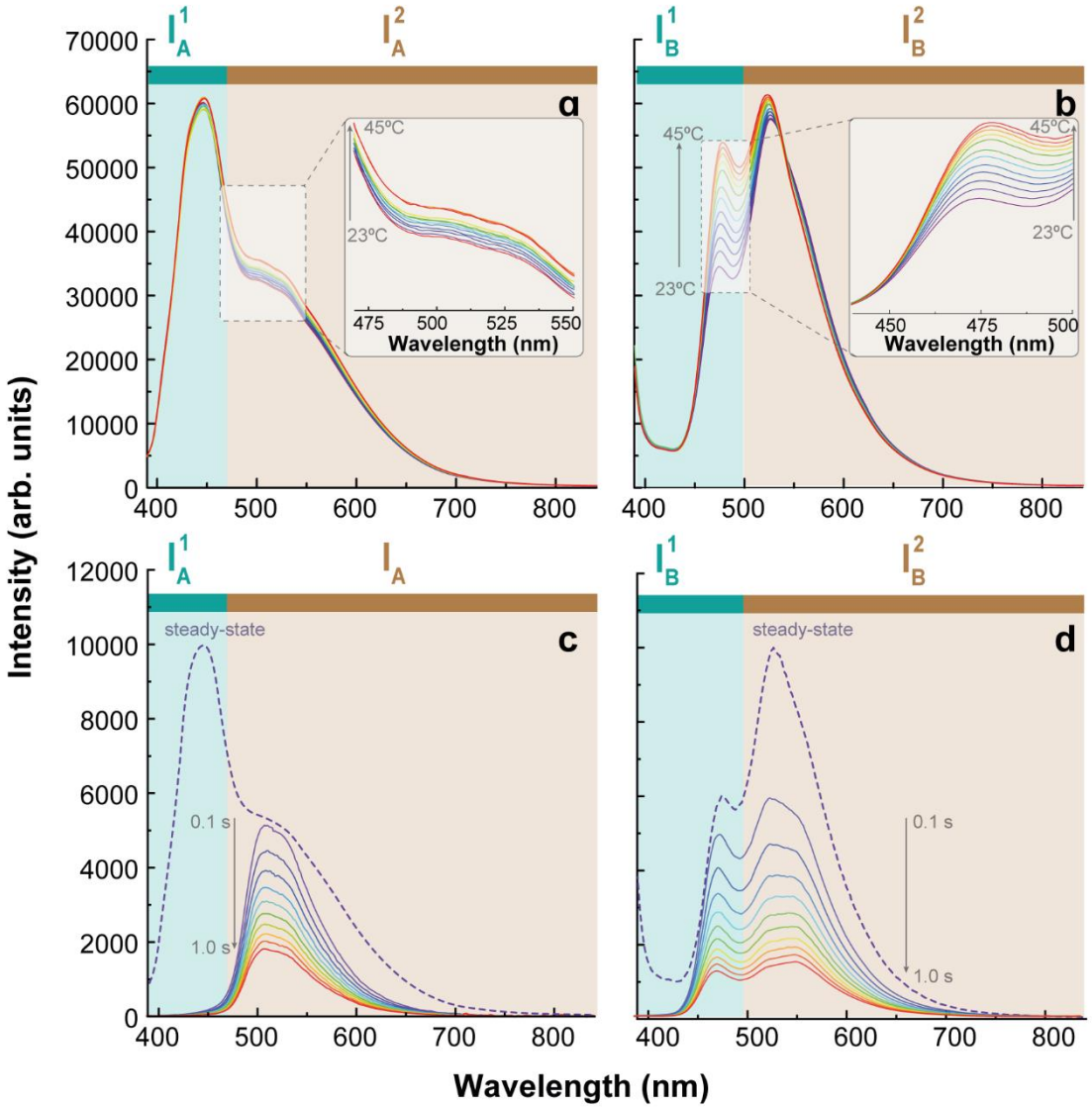


Figure S8. **Thermal and temporal behaviors of pure CDs emission.** Emission spectra excited at 365 nm as function of the temperature (23-45 °C) in the steady-state regime of (a) CD_A and (b) CD_B. Panels (c) and (d) display the time-resolved emission with different starting delay values (0.1-1.0 s) at 23 °C. The integrated intensity values within the shadowed areas I_A^1 and I_A^2 (I_B^1 and I_B^2) are assigned.

Tables S2 and S3 present all defined ratiometric thermometric parameters based on different integrated areas of the emission spectra of CDs in **Figure S8** (CD_A and CD_B) and **Figure 4** (CD_{AB}), respectively. Specifically, **Table S2** includes Δ_n with n ranging from 1 to 6 for CD_A and CD_B, while Table S3 includes Δ_n with n ranging from 1 to 30 for mixed CD_{AB}.

Table S2. Definitions of the thermometric parameter for CD_A and CD_B. The integrated intensities I_A^1, I_A^2, I_B^1, and I_A^2 are depicted in Figure 2.

	Thermometric parameter	Definition
CD_A	Δ_1	I_A^1/I_A^2
	Δ_2	I_A^2/I_A^1
	Δ_3	$I_A^1/(I_A^1+I_A^2)$
	Δ_4	$I_A^2/(I_A^1+I_A^2)$
	Δ_5	$(I_A^2+I_A^1)/ I_A^1$
	Δ_6	$(I_A^2+I_A^1)/ I_A^2$
CD_B	Δ_1	I_B^1/I_B^2
	Δ_2	I_B^2/I_B^1
	Δ_3	$I_B^1/(I_B^1+I_B^2)$
	Δ_4	$I_B^2/(I_B^1+I_B^2)$
	Δ_5	$(I_B^2+I_B^1)/ I_B^1$
	Δ_6	$(I_B^2+I_B^1)/ I_B^2$

Table S3. Definitions of the thermometric parameters for CD_{11}, CD_{14}, and CD_{41}, as well as definitions for the RGB thermometric parameters acquired through both the phone flash and UV lamp illumination. The integrated intensities I_{AB}^1 (389 – 450 nm), I_{AB}^2 (450 – 575 nm), and I_{AB}^3 (575 – 841 nm) are represented in **Figure 4**.

Mixed CDs		Processed e-tags	
Spectroscopic	Integrated intensity	UV (365 nm) or flash	RGB coordinates
Δ_1	I_{AB}^1/I_{AB}^2	Δ_1^{IoT}	B/G
Δ_2	I_{AB}^1/I_{AB}^3	Δ_2^{IoT}	B/R_{mean}
Δ_3	I_{AB}^2/I_{AB}^3	Δ_3^{IoT}	G/R_{mean}
Δ_4	I_{AB}^2/I_{AB}^1	Δ_4^{IoT}	G/B
Δ_5	I_{AB}^3/I_{AB}^1	Δ_5^{IoT}	R_{mean}/B
Δ_6	I_{AB}^3/I_{AB}^2	Δ_6^{IoT}	R_{mean}/G
Δ_7	$I_{AB}^1/(I_{AB}^1+I_{AB}^2)$	Δ_7^{IoT}	B/(B+G)
Δ_8	$I_{AB}^1/(I_{AB}^1+I_{AB}^3)$	Δ_8^{IoT}	$B/(B+R_{mean})$
Δ_9	$I_{AB}^1/(I_{AB}^2+I_{AB}^3)$	Δ_9^{IoT}	$B/(G+R_{mean})$

Δ_{10}	$I^1_{AB}/(I^1_{AB}+I^2_{AB}+I^3_{AB})$	Δ^{IoT}_{10}	$B/(B+G+R_{mean})$
Δ_{11}	$I^2_{AB}/(I^1_{AB}+I^2_{AB})$	Δ^{IoT}_{11}	$G/(B+G)$
Δ_{12}	$I^2_{AB}/(I^1_{AB}+I^3_{AB})$	Δ^{IoT}_{12}	$G/(B+R_{mean})$
Δ_{13}	$I^2_{AB}/(I^2_{AB}+I^3_{AB})$	Δ^{IoT}_{13}	$G/(G+R_{mean})$
Δ_{14}	$I^2_{AB}/(I^1_{AB}+I^2_{AB}+I^3_{AB})$	Δ^{IoT}_{14}	$G/(B+G+R_{mean})$
Δ_{15}	$I^3_{AB}/(I^1_{AB}+I^2_{AB})$	Δ^{IoT}_{15}	$R_{mean}/(B+G)$
Δ_{16}	$I^3_{AB}/(I^1_{AB}+I^3_{AB})$	Δ^{IoT}_{16}	$R_{mean}/(B+R_{mean})$
Δ_{17}	$I^3_{AB}/(I^2_{AB}+I^3_{AB})$	Δ^{IoT}_{17}	$R_{mean}/(G+R_{mean})$
Δ_{18}	$I^3_{AB}/(I^1_{AB}+I^3_{AB}+I^2_{AB})$	Δ^{IoT}_{18}	$R_{mean}/(B+G+R_{mean})$
Δ_{19}	$(I^1_{AB}+I^2_{AB})/ I^1_{AB}$	Δ^{IoT}_{19}	$(B+G)/B$
Δ_{20}	$(I^1_{AB}+I^3_{AB})/ I^1_{AB}$	Δ^{IoT}_{20}	$(B+R_{mean})/B$
Δ_{21}	$(I^2_{AB}+I^3_{AB})/ I^1_{AB}$	Δ^{IoT}_{21}	$(G+R_{mean})/B$
Δ_{22}	$(I^1_{AB}+I^2_{AB}+I^3_{AB})/ I^1_{AB}$	Δ^{IoT}_{22}	$(B+G+R_{mean})/B$
Δ_{23}	$(I^1_{AB}+I^2_{AB})/ I^2_{AB}$	Δ^{IoT}_{23}	$(B+G)/G$
Δ_{24}	$(I^1_{AB}+I^3_{AB})/ I^2_{AB}$	Δ^{IoT}_{24}	$(B+R_{mean})/G$
Δ_{25}	$(I^2_{AB}+I^3_{AB})/ I^2_{AB}$	Δ^{IoT}_{25}	$(G+R_{mean})/G$
Δ_{26}	$(I^1_{AB}+I^3_{AB}+I^2_{AB})/ I^2_{AB}$	Δ^{IoT}_{26}	$(B+G+R_{mean})/G$
Δ_{27}	$(I^1_{AB}+I^2_{AB})/ I^3_{AB}$	Δ^{IoT}_{27}	$(B+G)/ R_{mean}$
Δ_{28}	$(I^1_{AB}+I^3_{AB})/ I^3_{AB}$	Δ^{IoT}_{28}	$(B+R_{mean})/ R_{mean}$
Δ_{29}	$(I^2_{AB}+I^3_{AB})/ I^3_{AB}$	Δ^{IoT}_{29}	$(G+R_{mean})/ R_{mean}$
Δ_{30}	$(I^1_{AB}+I^3_{AB}+I^2_{AB})/ I^3_{AB}$	Δ^{IoT}_{30}	$(B+G+R_{mean})/ R_{mean}$

6.1. Python code

The following code includes generating a *meshgrid* from the T and t arrays, which facilitates a 2D interpolation process. By utilizing the *griddata* function from the *scipy.interpolate* module, $\Delta(T,t)$ values are interpolated on this grid. The ensuing step encompasses the computation of partial derivatives concerning both the T and t axes, achieved through the application of the *np.gradient* function.

The final facet of the code involves the blend of the interpolated data, partial derivatives, original dataset, and sensitivity metric into a tabular structure, instantiated as a Pandas DataFrame. This tabular representation is subsequently transcribed into an output CSV file, featuring columns such as T, $\partial\Delta/\partial$T, t, $\partial\Delta/\partial$t, Δ, and S_r (in %·C^{-1}). The nomenclature of this output file is derived from the input file name, with the CSV format and semicolon delimiter employed for storage.

In summary, the code below treats and analyzes the input data concerning the measured thermometric parameter into three steps: 1) making 2D interpolation of $\Delta(T,t)$ consisting of 500 points between each consecutive measured one; 2) derivative calculations; and 3) creating output data.

```
import numpy as np
from scipy.interpolate import griddata
import pandas as pd
import argparse
import matplotlib.pyplot as plt
from lmfit import Model

def main(input_file):
    data = np.loadtxt(input_file)
    T = data[:, 0]
    t = data[:, 1]
    Delta = data[:, 2]

    T_grid, t_grid = np.meshgrid(np.linspace(min(T), max(T), 500), np.linspace(min(t), max(t), 500))

    points = np.column_stack((T, t))
    Delta_interp = griddata(points, Delta, (T_grid, t_grid), method='cubic')

    # derivatives with respect to T and t
    dDelta_dT = np.gradient(Delta_interp, axis=1)
    dDelta_dt = np.gradient(Delta_interp, axis=0)

    # relative sensitivity
    S_r = (100 / np.abs(Delta_interp.flatten())) * np.abs(dDelta_dT.flatten())

    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    plt.imshow(dDelta_dT, extent=(min(T), max(T), min(t), max(t)), origin='lower', aspect='auto',
cmap='viridis')
    plt.colorbar()
    plt.xlabel('Temperature (°C)')
    plt.ylabel('Time (s)')
    plt.title('Partial derivative ∂Δ(T,t)/∂T')

    plt.subplot(1, 2, 2)
    plt.imshow(dDelta_dt, extent=(min(T), max(T), min(t), max(t)), origin='lower', aspect='auto',
cmap='viridis')
    plt.colorbar()
    plt.xlabel('Temperature (°C)')
    plt.ylabel('Time (s)')
```

```
    plt.title('Partial derivative ∂Δ(T,t)/∂t')

    plt.tight_layout()
    plt.savefig(f'{input_file}_plot.jpg', dpi=600)
    plt.close()

    output_data = {
        'T': T_grid.flatten(),
        '∂Δ/∂T': dDelta_dT.flatten(),
        't': t_grid.flatten(),
        '∂Δ/∂t': dDelta_dt.flatten(),
        'Δ': Delta_interp.flatten(),
        'Sr(T,t) (%)': S_r
    }
    dDelta_output = pd.DataFrame(output_data)
    output_file = input_file.replace('.txt', '_output.csv')
    dDelta_output.to_csv(output_file, index=False, sep=';')

    print("Output file created successfully:", output_file)

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Process data file and calculate derivatives.")
    parser.add_argument("input_file", help="Path to the input .txt file with T, t, and Delta
coordinates.")
    args = parser.parse_args()
    main(args.input_file)
```

6.2. MATLAB routine for RGB analysis

The following MATLAB® code reads a video file, extracts individual frames, and applies various image processing functions to each frame. It computes the mean gray, red, green, and blue levels for each frame and plots the intensity profiles over time. It also allows the user to save individual frames as image files and then recall them back into a movie. The code also employs a background differencing technique to detect differences between successive frames. Finally, it displays the processed frames and allows the user to view the movie made from the recalled frames.

```
try
    videoObject = VideoReader(outputVideoPath)
    numberOfFrames = videoObject.NumFrames;
    vidHeight = videoObject.Height;
    vidWidth = videoObject.Width;
    numberOfFramesWritten = 0;
    figure;
    set(gcf, 'units','normalized','outerposition',[0 0 1 1]);

    promptMessage = sprintf('Do you want to save the individual frames out to individual disk files?');
    button = questdlg(promptMessage, 'Save individual frames?', 'Yes', 'No', 'Yes');

    if strcmp(button, 'Yes')
        writeToDisk = true;
        [folder, baseFileName, extentions] = fileparts('x1.mp4');
        folder = pwd;
        outputFolder = sprintf('%s/Movie Frames from %s', folder, baseFileName);

        if ~exist(outputFolder, 'dir')
            mkdir(outputFolder);
        end
    else
        writeToDisk = false;
    end

    meanGrayLevels = zeros(numberOfFrames, 1);
    meanRedLevels = zeros(numberOfFrames, 1);
    meanGreenLevels = zeros(numberOfFrames, 1);
    meanBlueLevels = zeros(numberOfFrames, 1);
    clear image
    for frame = 1 : numberOfFrames

        thisFrame = read(videoObject, frame);
        hImage = subplot(3, 2, [1 3 5]);
        image(thisFrame);
        caption = sprintf('Video - Frame %4d of %d', frame, numberOfFrames);
        title(caption, 'FontSize', fontSize);
        drawnow; % to refresh the window.

        if writeToDisk
            outputBaseFileName = sprintf('Video - Frame %4.4d.png', frame);
            outputFullFileName = fullfile(outputFolder, outputBaseFileName);
            text(5, 15, outputBaseFileName, 'FontSize', 20);

            frameWithText = getframe(gca);
            imwrite(frameWithText.cdata, outputFullFileName, 'png');
        end

        % mean gray level.
        grayImage = rgb2gray(thisFrame);
        meanGrayLevels(frame) = mean(grayImage(:));
        % R, G, and B levels.
        meanRedLevels(frame) = mean(mean(thisFrame(:, :, 1)));
        meanGreenLevels(frame) = mean(mean(thisFrame(:, :, 2)));
        meanBlueLevels(frame) = mean(mean(thisFrame(:, :, 3)));

        subplot(3, 2, 6)
```

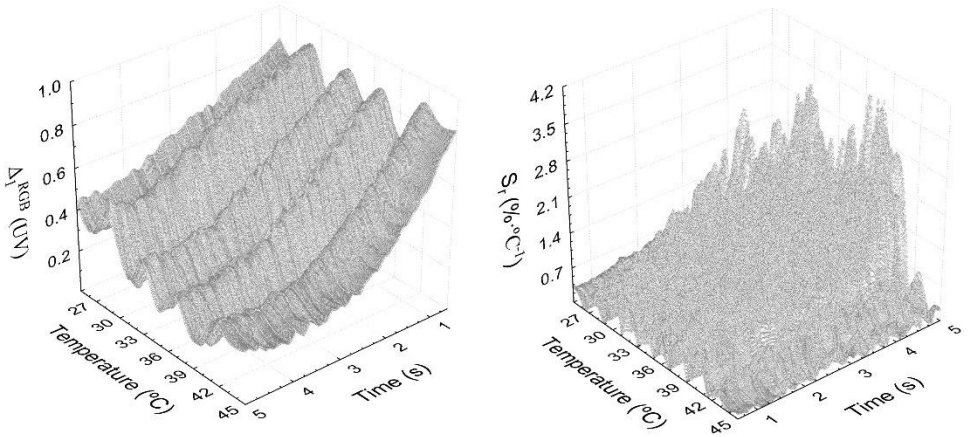
```
        plot(meanBlueLevels, 'b-','LineWidth',1);
        legend('Blue Component')
        ylabel('Intensity (a.u.)','FontWeight', 'bold')
        xlabel('Frame number','FontWeight', 'bold')
        xlim([0 numberOfFrames])
        grid on

        subplot(3, 2, 4)
        plot(meanGreenLevels, 'g-','LineWidth',1);
        legend('Green Component')
        ylabel('Intensity (a.u.)','FontWeight', 'bold')
        xlabel('Frame number','FontWeight', 'bold')
        xlim([0 numberOfFrames])
        grid on

        subplot(3, 2, 2)
        plot(meanRedLevels, 'r-','LineWidth',1);
        title('RGB Video Analysis', 'FontSize', fontSize);
        legend('Red Component')
        ylabel('Intensity (a.u.)','FontWeight', 'bold')
        xlabel('Frame number','FontWeight', 'bold')
        xlim([0 numberOfFrames])
        grid on

        if frame == 1
            xlabel('Frame Number');
            ylabel('Gray Level');
            [rows, columns, numberOfColorChannels] = size(thisFrame);
        end

        if writeToDisk
            progressIndication = sprintf('Wrote frame %4d of %d.', frame, numberOfFrames);
        else
            progressIndication = sprintf('Processed frame %4d of %d.', frame, numberOfFrames);
        end
        disp(progressIndication);

        numberOfFramesWritten = numberOfFramesWritten + 1;

        alpha = 0.5;
        if frame == 1
            Background = thisFrame;
        else
            % Change background slightly at each frame
            %           Background(t+1)=(1-alpha)*I+alpha*Background
            Background = (1-alpha)* thisFrame + alpha * Background;
        end

    end

    if writeToDisk
        finishedMessage = sprintf('Done!  It wrote %d frames to folder\n"%s"', numberOfFramesWritten,
outputFolder);
    else
        finishedMessage = sprintf('Done!  It processed %d frames of\n"%s"', numberOfFramesWritten,
movieFullFileName);
    end
    disp(finishedMessage);
    uiwait(msgbox(finishedMessage));

    if ~writeToDisk
        return;
    end
    promptMessage = sprintf('Do you want to recall the individual frames\nback from disk into a
movie?\n(This will take several seconds.)');
    button = questdlg(promptMessage, 'Recall Movie?', 'Yes', 'No', 'Yes');
    if strcmp(button, 'No')
        return;
    end
    writerObj = VideoWriter('vlc.mp4');
    open(writerObj);

    allTheFrames = cell(numberOfFrames,1);
    allTheFrames(:) = {zeros(vidHeight, vidWidth, 3, 'uint8')};
    allTheColorMaps = cell(numberOfFrames,1);
    allTheColorMaps(:) = {zeros(256, 3)};
```

```
    recalledMovie = struct('cdata', allTheFrames, 'colormap', allTheColorMaps);
    for frame = 1 : numberOfFrames
        % Construct an output image file name.
        outputBaseFileName = sprintf('Frame %4.4d.png', frame);
        outputFullFileName = fullfile(outputFolder, outputBaseFileName);
        thisFrame = imread(outputFullFileName);
        recalledMovie(frame) = im2frame(thisFrame);
        writeVideo(writerObj, thisFrame);
    end
    close(writerObj);
    delete(hImage);
    delete(hPlot);
    subplot(1, 3, 2);
    axis off;  % To turn off axes numbers.
    title('Movie recalled from disk', 'FontSize', fontSize);
    movie(recalledMovie);
    msgbox('Done with this demo!');
catch ME
end
```

6.3. Thermal behavior under UV excitation for CD_{11}

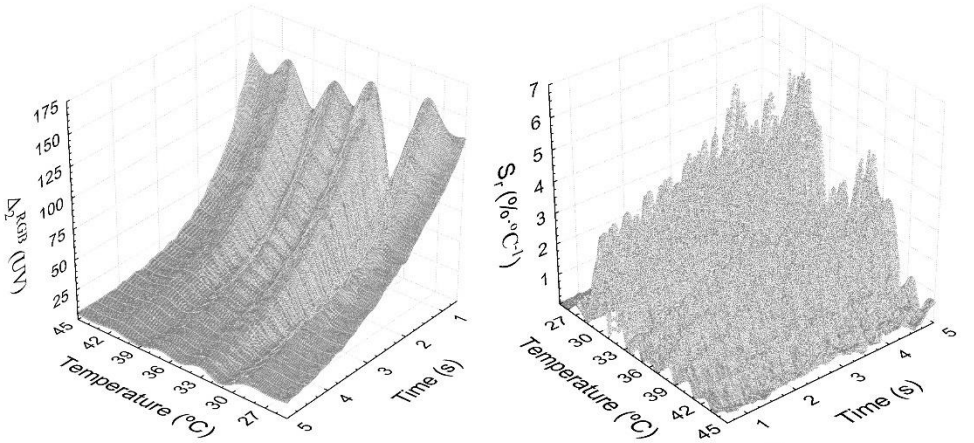
(I)

$$\Delta_1^{IoT} = \frac{B}{G}$$



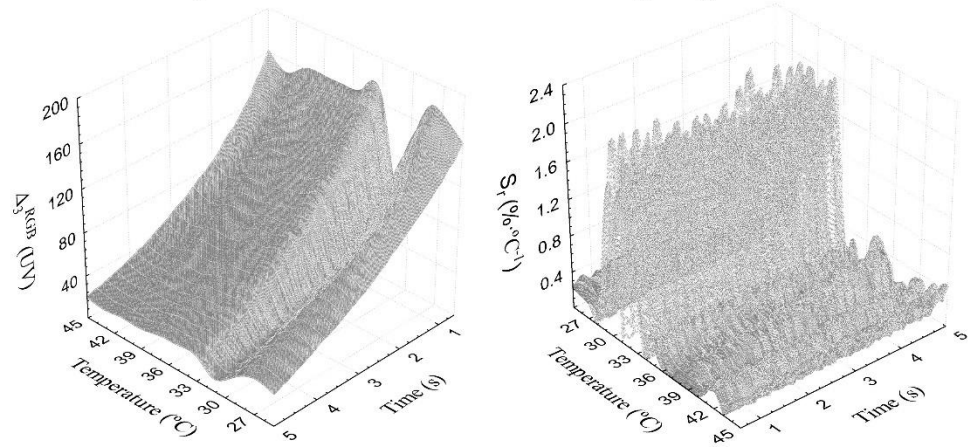
(II)

$$\Delta_2^{IoT} = \frac{B}{R_{mean}}$$



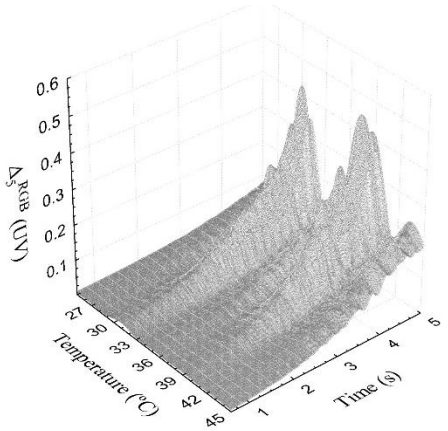
(III)

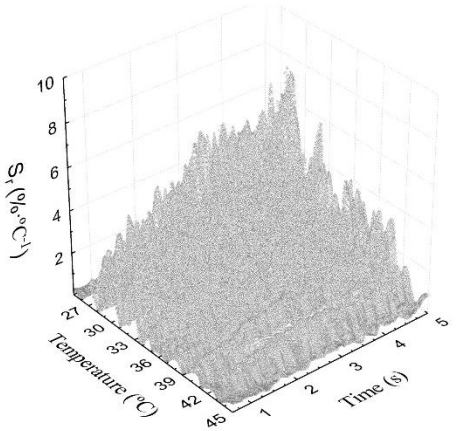
$$\Delta_3^{IoT} = \frac{G}{R_{mean}}$$



(IV)

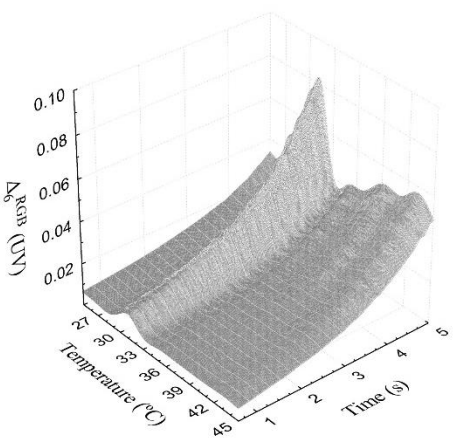
$$\Delta_5^{RGB} = \frac{R_{mean}}{B}$$

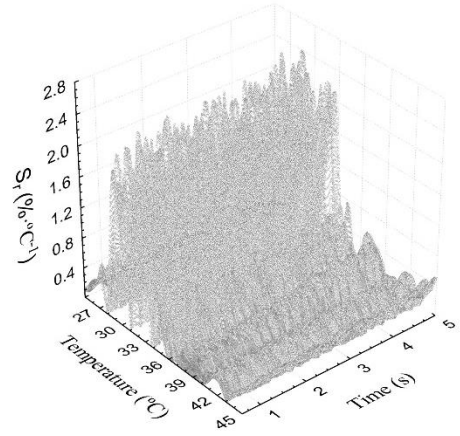




(V)

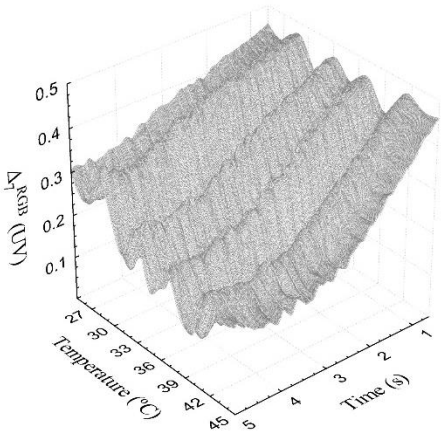
$$\Delta_6^{RGB} = \frac{R_{mean}}{G}$$

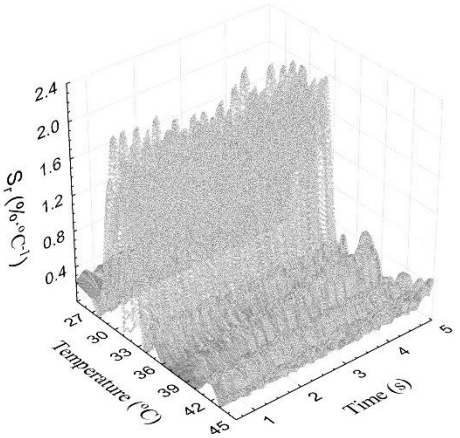




(VI)

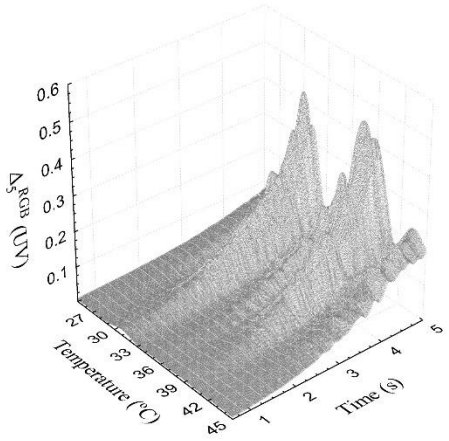
$$\Delta_7^{RGB} = \frac{B}{(B+G)}$$

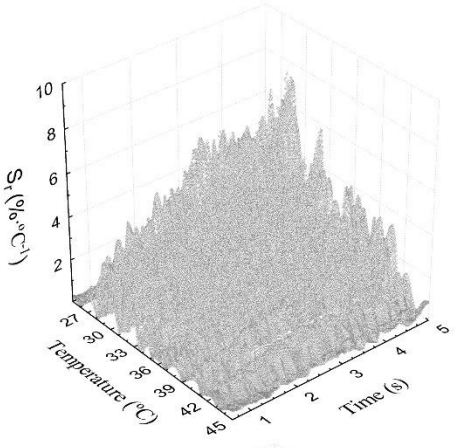




(IV)

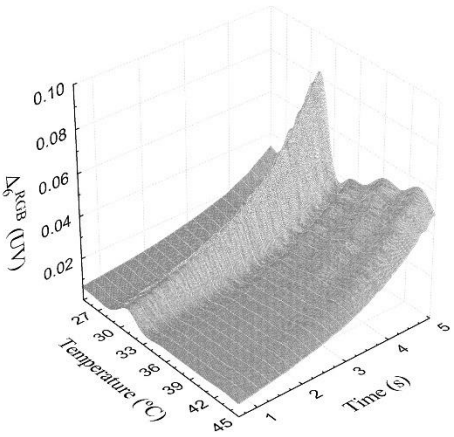
$$\Delta_5^{IoT} = \frac{R_{mean}}{B}$$

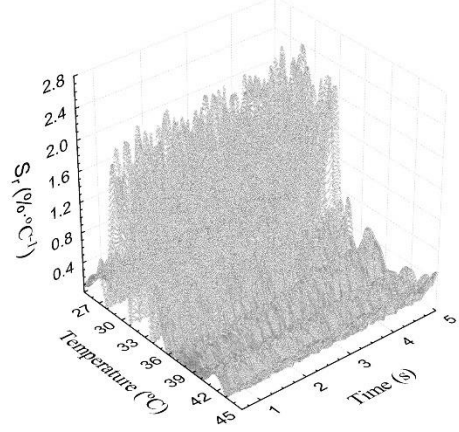




(V)

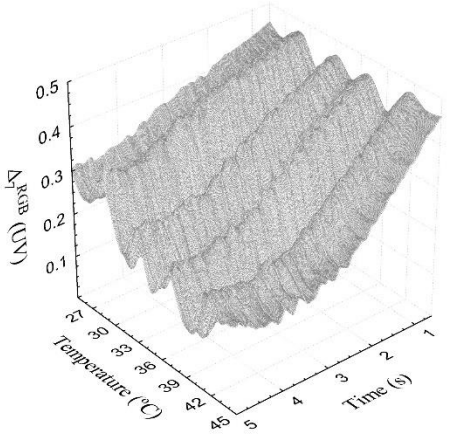
$$\Delta_6^{IoT} = \frac{R_{mean}}{G}$$

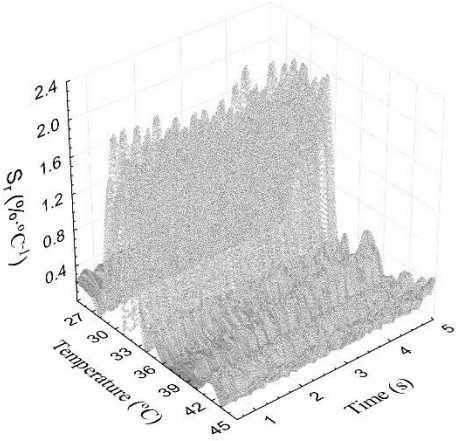




(VI)

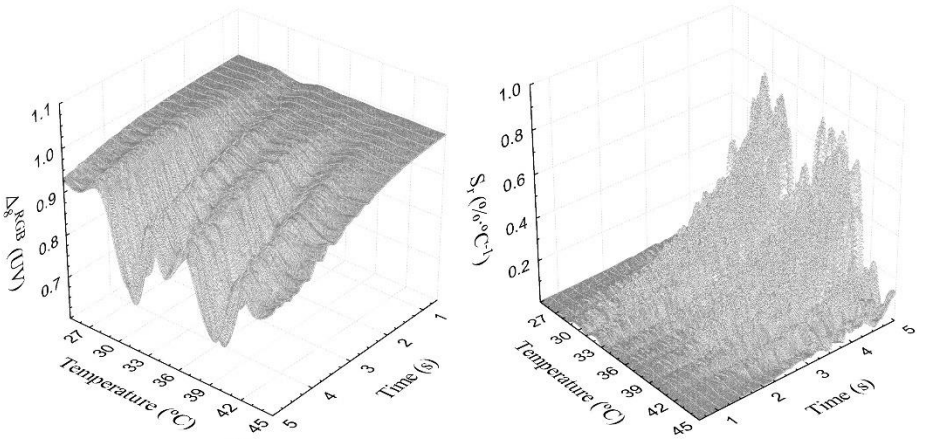
$$\Delta_7^{IoT} = \frac{B}{(B+G)}$$





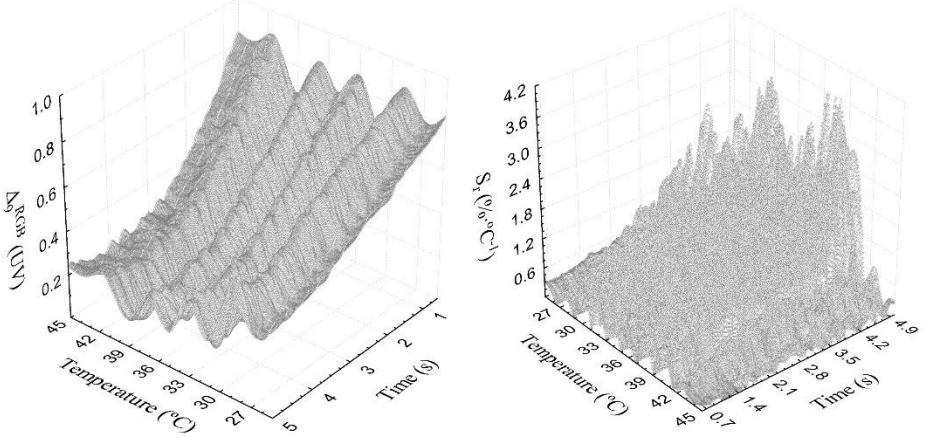
(VII)

$$\Delta_8^{IoT} = \frac{B}{(B+R_{mean})}$$



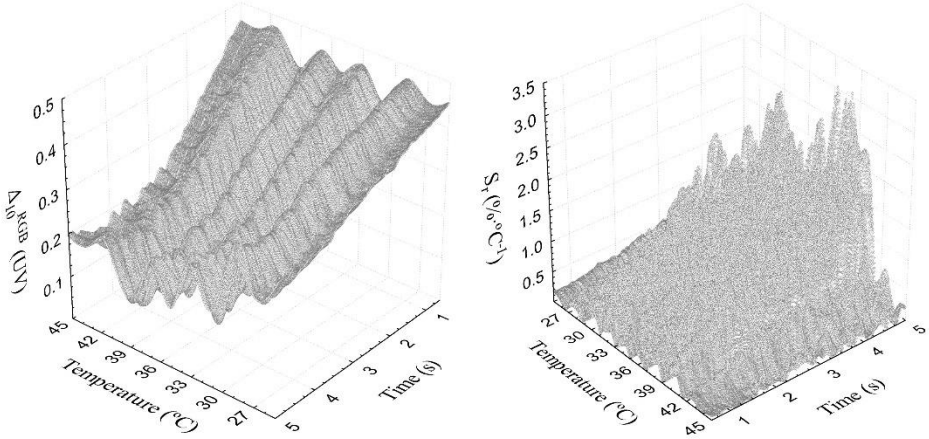
(VIII)

$$\Delta_9^{IoT} = \frac{B}{(G+R_{mean})}$$



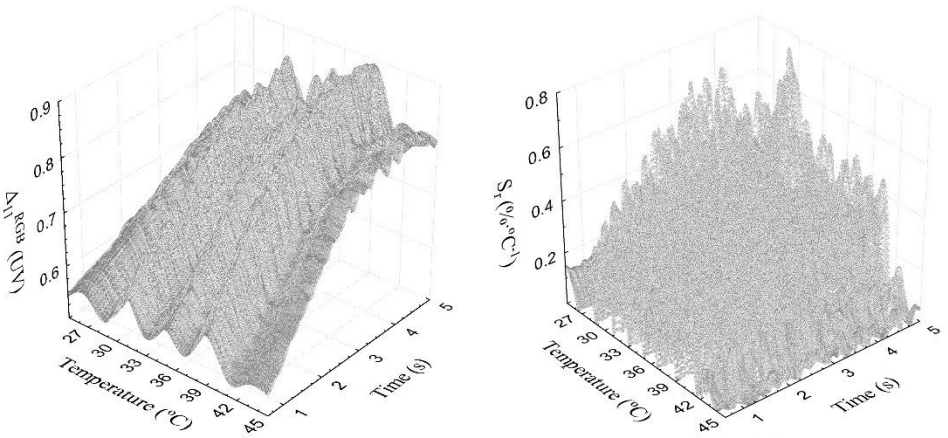
(IX)

$$\Delta_{10}^{IoT} = \frac{B}{(B+G+R_{mean})}$$



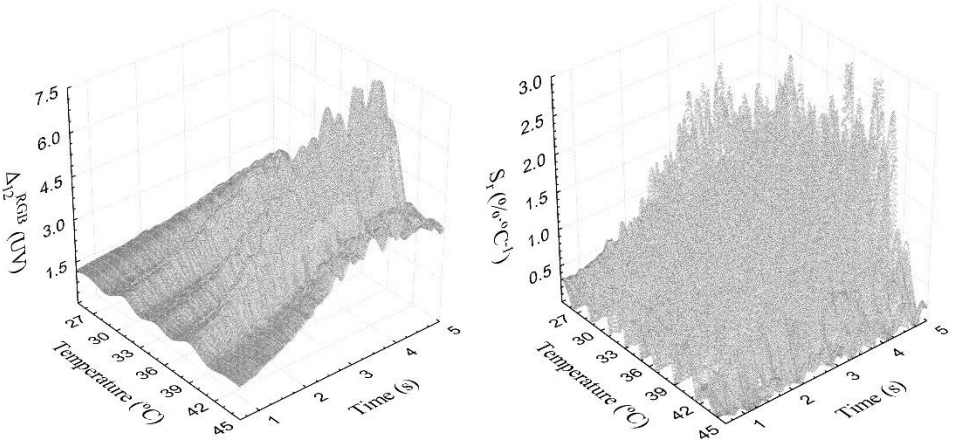
(X)

$$\Delta_{11}^{IoT} = \frac{G}{(B+G)}$$



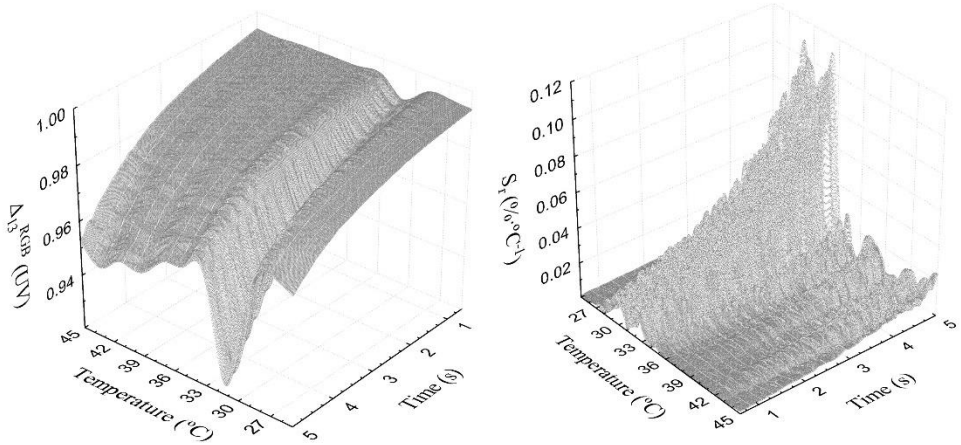
(XI)

$$\Delta_{12}^{IoT} = \frac{G}{(B+R_{mean})}$$



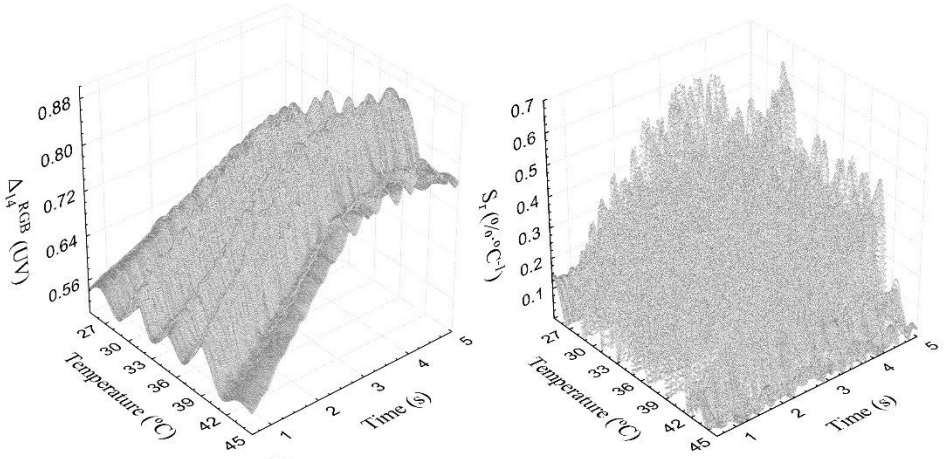
(XII)

$$\Delta_{13}^{IoT} = \frac{G}{(G+R_{mean})}$$



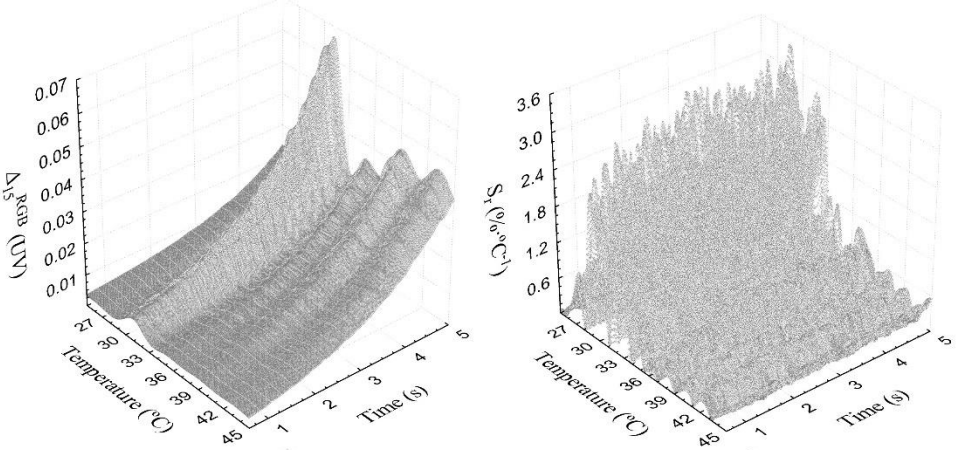
(XIII)

$$\Delta_{14}^{IoT} = \frac{G}{(B+G+R_{mean})}$$



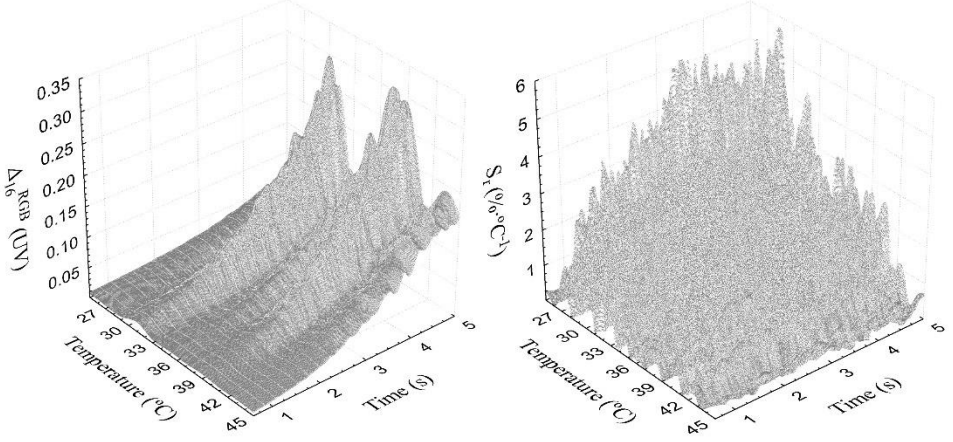
(XIV)

$$\Delta_{15}^{IoT} = \frac{R_{mean}}{(B+G)}$$



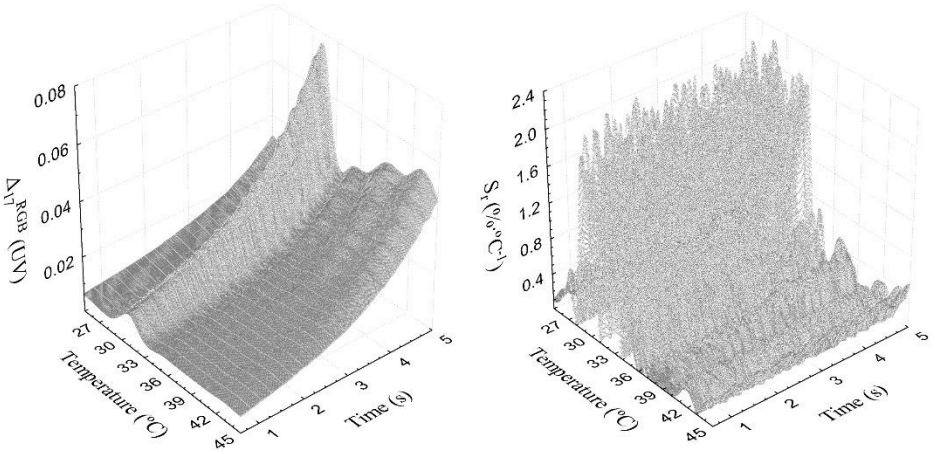
(XV)

$$\Delta_{16}^{IoT} = \frac{R_{mean}}{(B+R_{mean})}$$



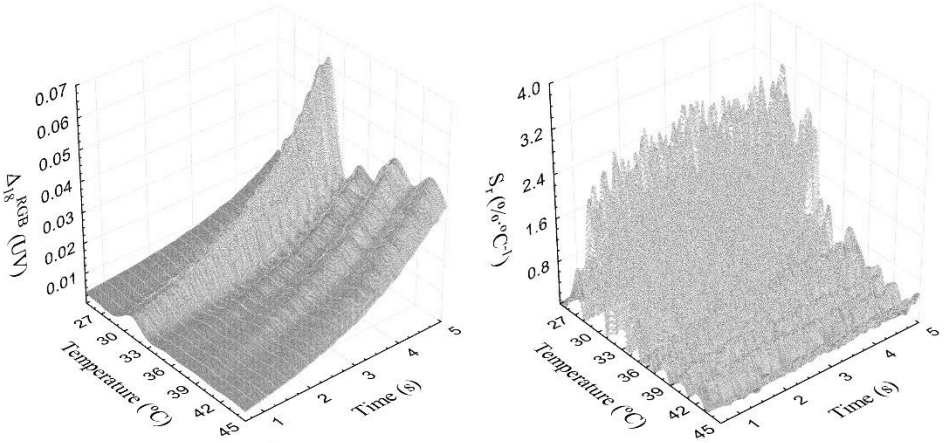
(XVI)

$$\Delta_{17}^{IoT} = \frac{R_{mean}}{(G+R_{mean})}$$



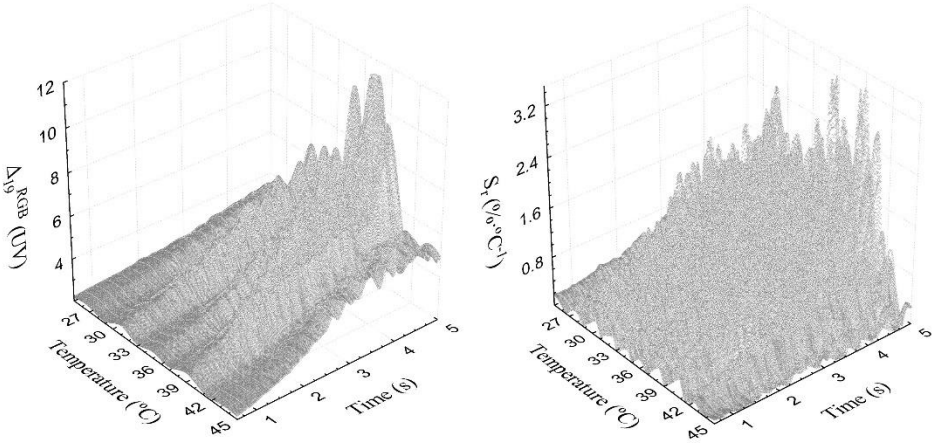
(XVII)

$$\Delta_{18}^{IoT} = \frac{R_{mean}}{(B+G+R_{mean})}$$



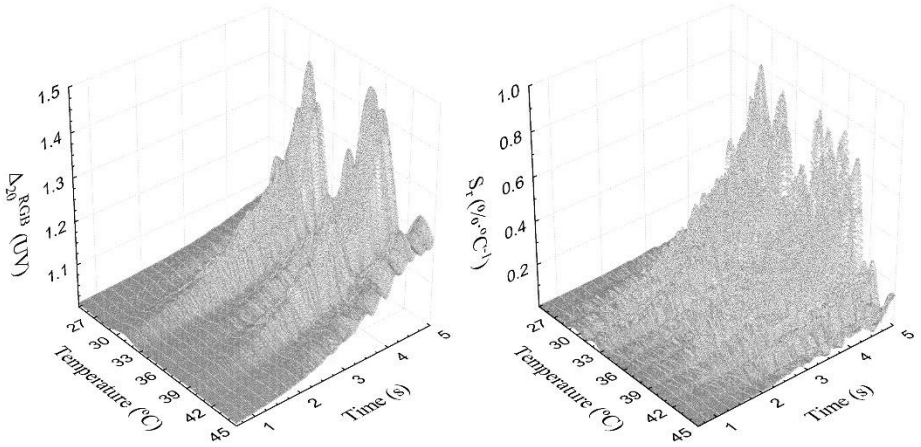
(XVIII)

$$\Delta_{19}^{IoT} = \frac{(B+G)}{B}$$



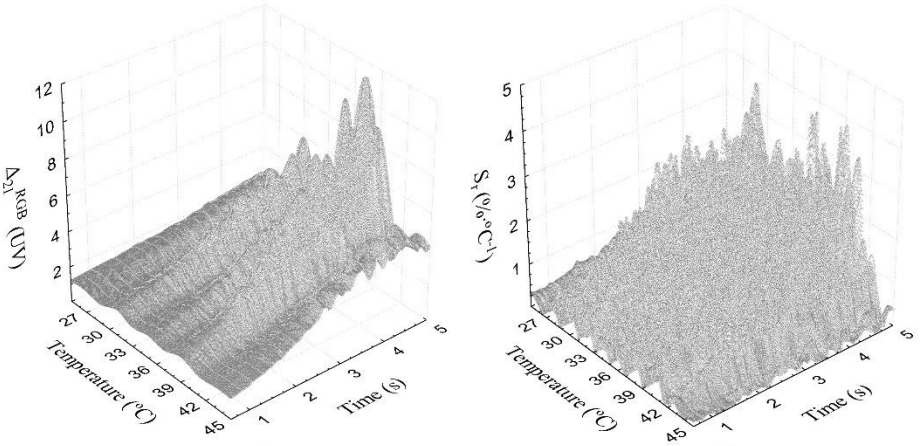
(XIX)

$$\Delta_{20}^{IoT} = \frac{(B+R_{mean})}{B}$$



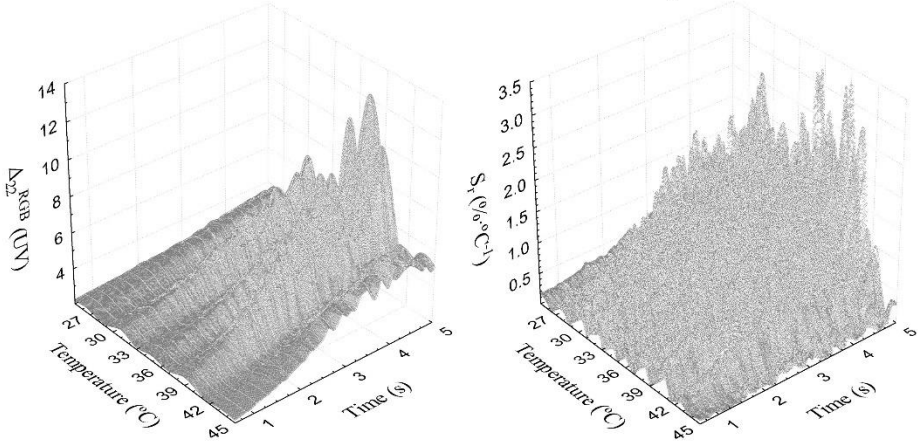
(XX)

$$\Delta_{21}^{IoT} = \frac{(G+R_{mean})}{B}$$



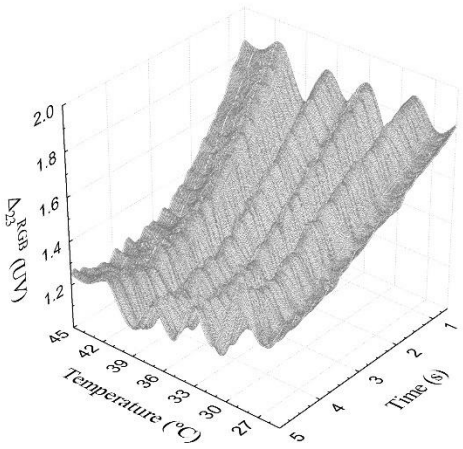
(XXI)

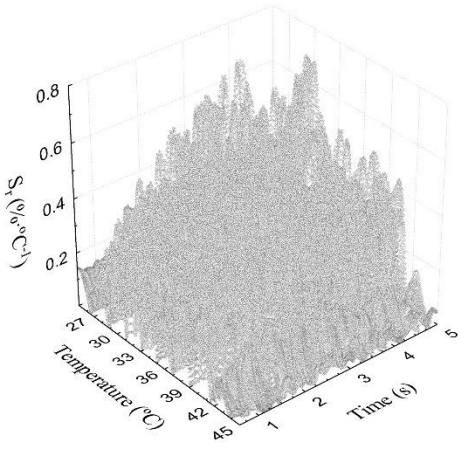
$$\Delta_{22}^{IoT} = \frac{(B+G+R_{mean})}{B}$$



(XXII)

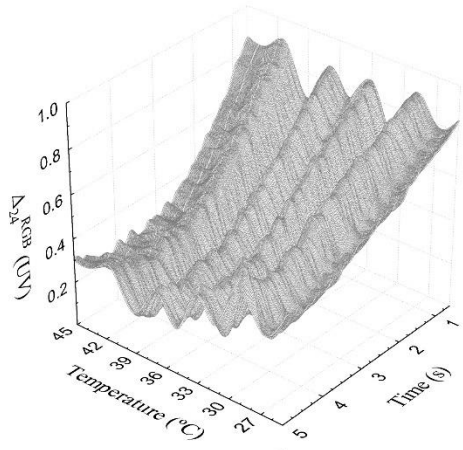
$$\Delta_{23}^{IoT} = \frac{(B+G)}{G}$$

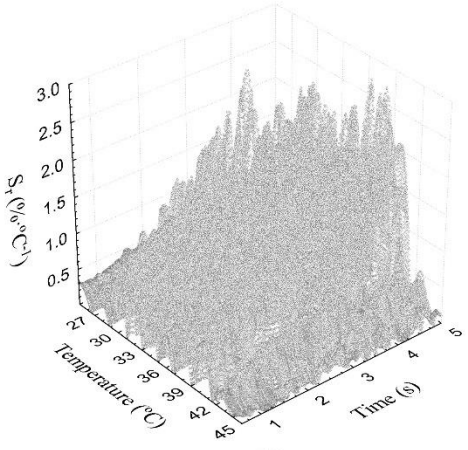




(XXIII)

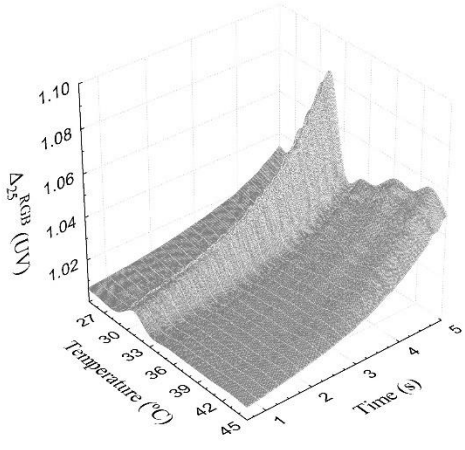
$$\Delta_{24}^{IoT} = \frac{(B+R_{mean})}{G}$$

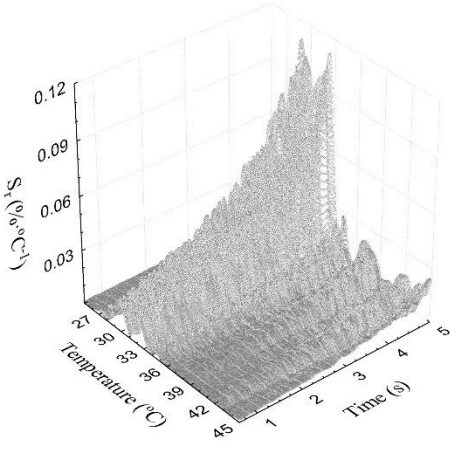




(XXIV)

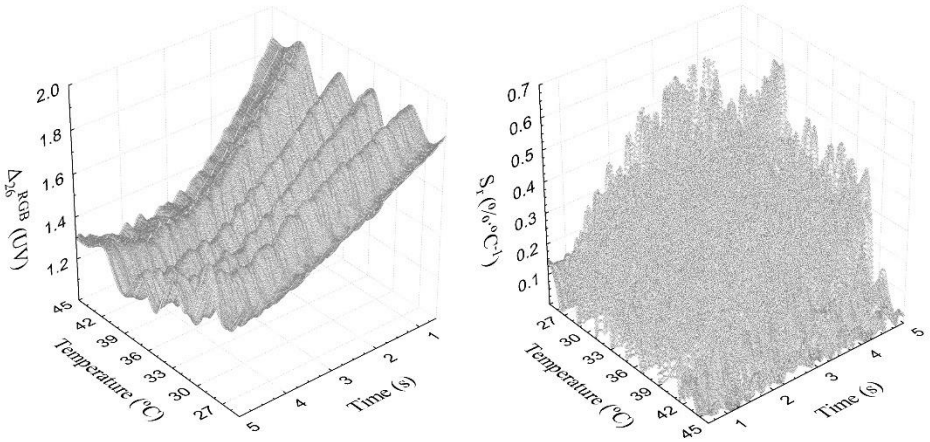
$$\Delta_{25}^{IoT} = \frac{(G+R_{mean})}{G}$$





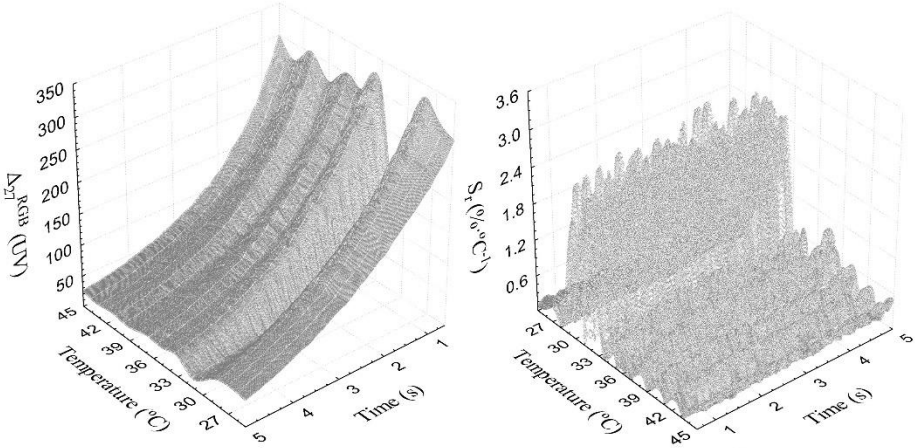
(XXV)

$$\Delta_{26}^{IoT} = \frac{(B+G+R_{mean})}{G}$$



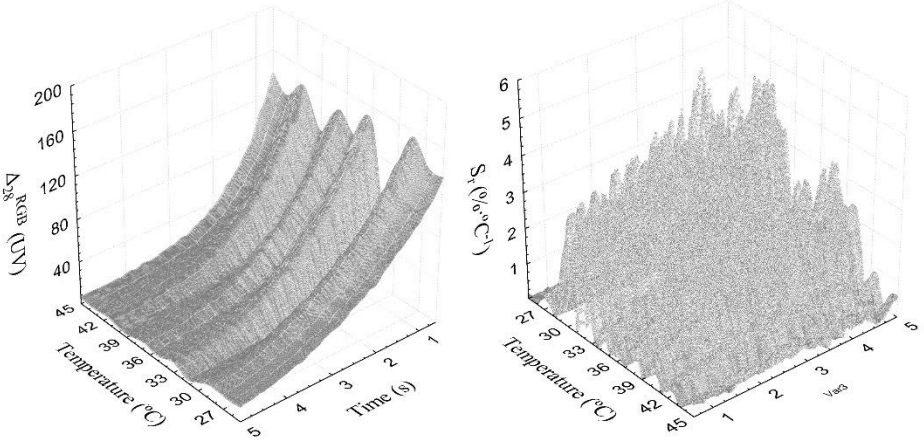
(XXVI)

$$\Delta_{27}^{IoT} = \frac{(B+G)}{R_{mean}}$$



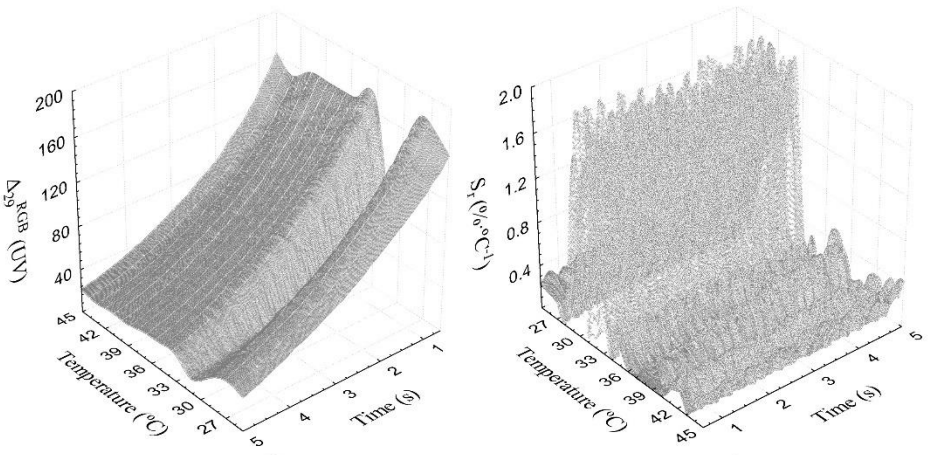
(XXVII)

$$\Delta_{28}^{IoT} = \frac{(B+R_{mean})}{R_{mean}}$$



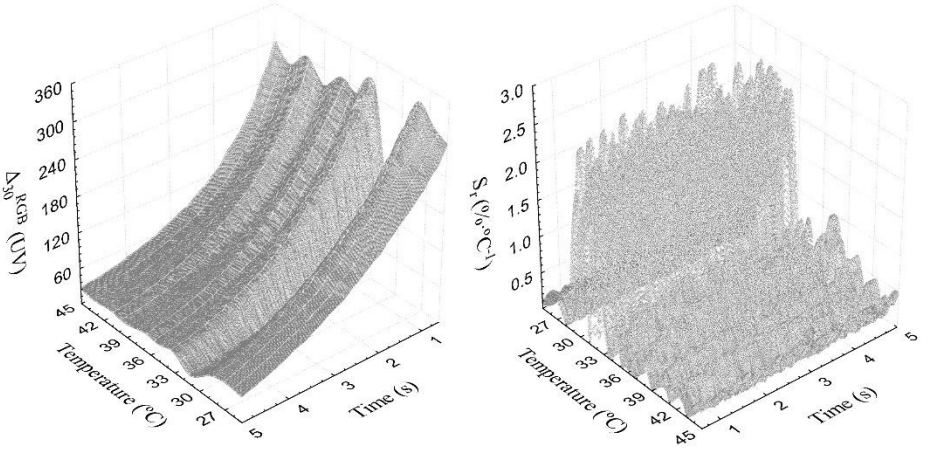
(XXVIII)

$$\Delta_{29}^{IoT} = \frac{R_{mean}}{(G+R_{mean})}$$



(XXIX)

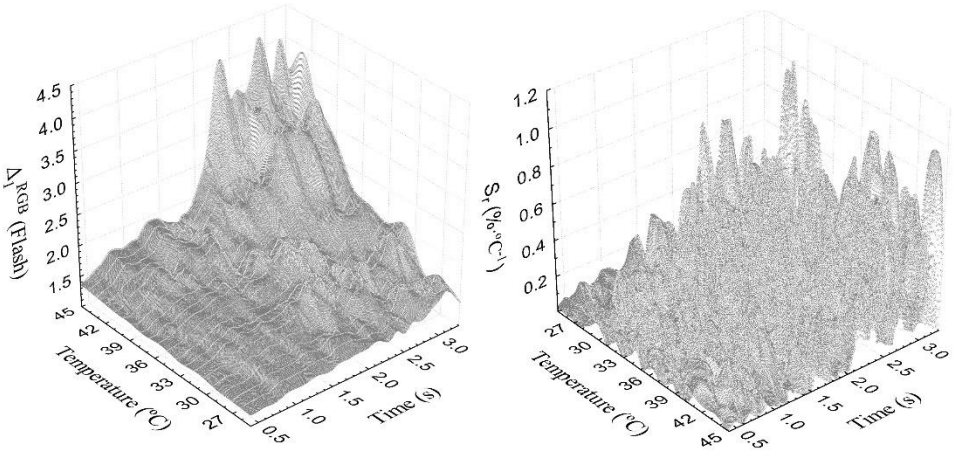
$$\Delta_{30}^{IoT} = \frac{R_{mean}}{(B+G+R_{mean})}$$



6.4. Thermal behavior using phone flash for CD_{11}

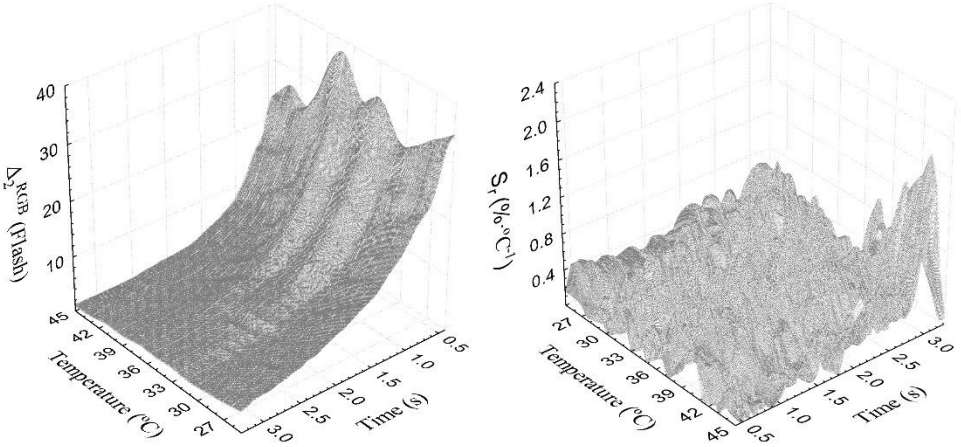
(I)

$$\Delta_1^{IoT} = \frac{B}{G}$$



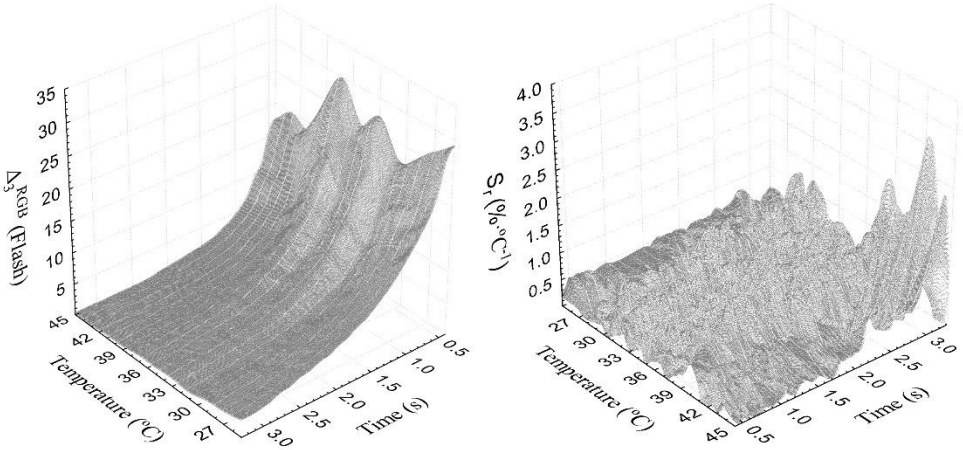
(II)

$$\Delta_2^{IoT} = \frac{B}{R_{mean}}$$



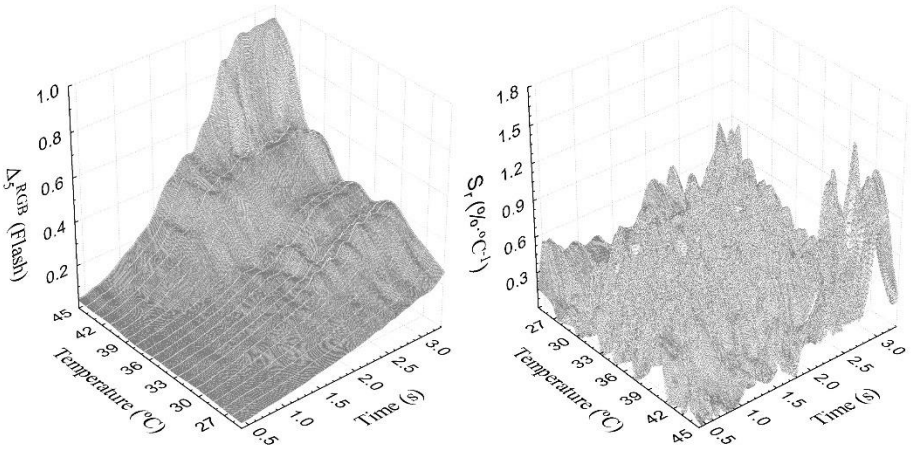
(III)

$$\Delta_3^{IoT} = \frac{G}{R_{mean}}$$



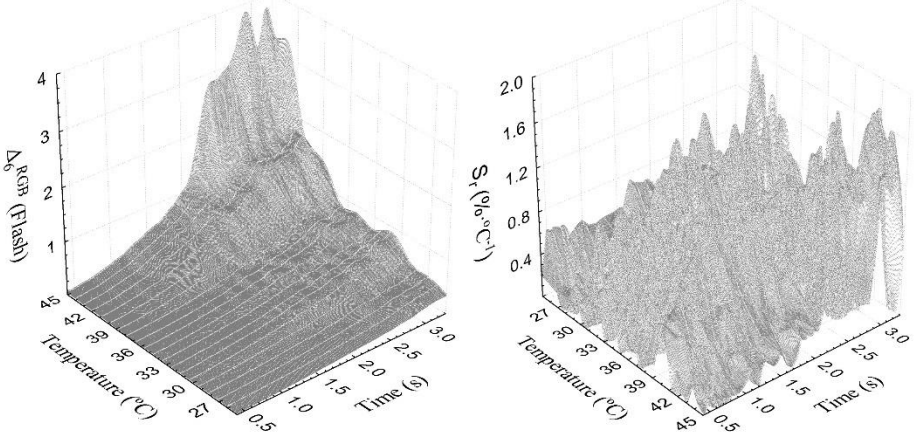
(IV)

$$\Delta_5^{IoT} = \frac{R_{mean}}{B}$$



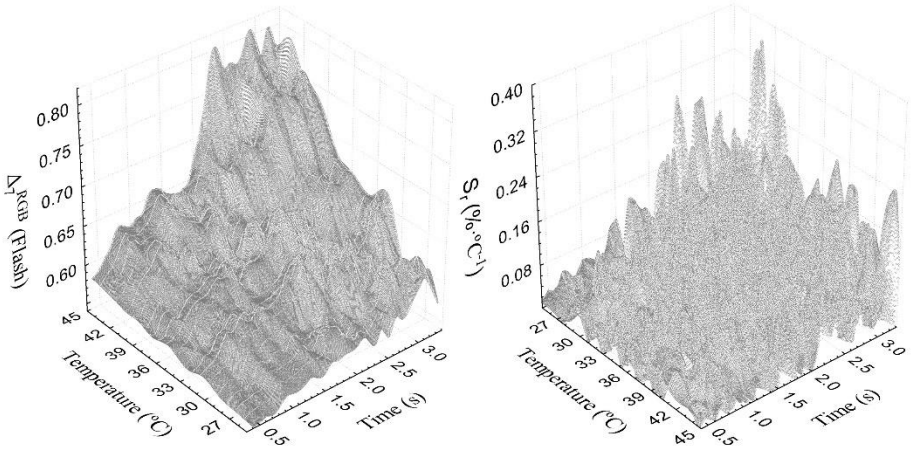
(V)

$$\Delta_6^{IoT} = \frac{R_{mean}}{G}$$



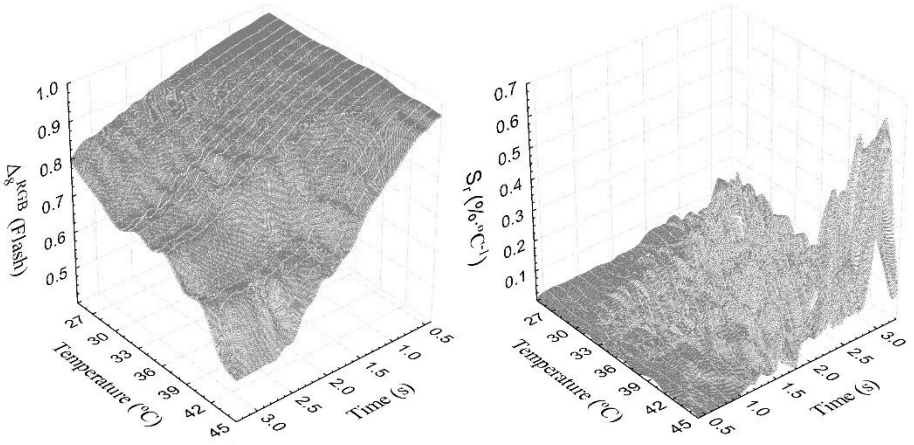
(VI)

$$\Delta_7^{IoT} = \frac{B}{(B+G)}$$



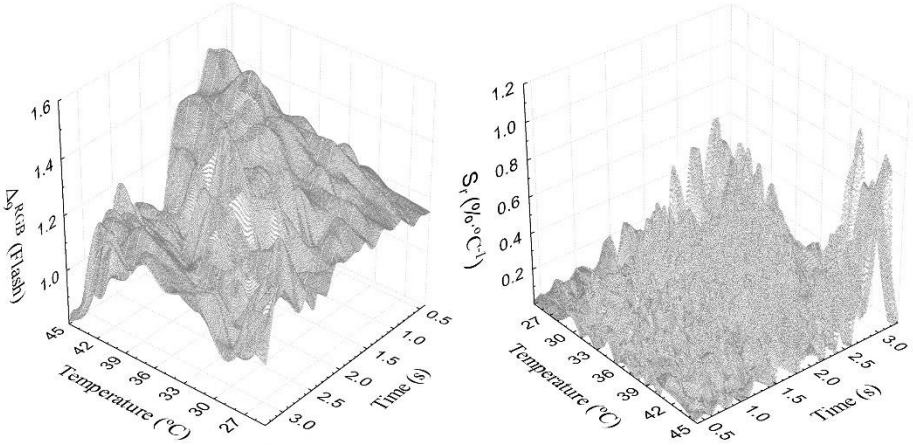
(VII)

$$\Delta_8^{IoT} = \frac{B}{(B+R_{mean})}$$



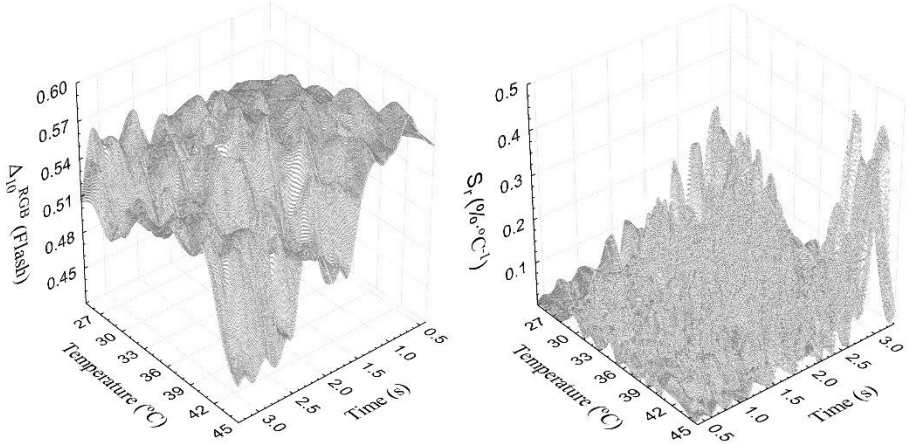
(VIII)

$$\Delta_9^{IoT} = \frac{B}{(G+R_{mean})}$$



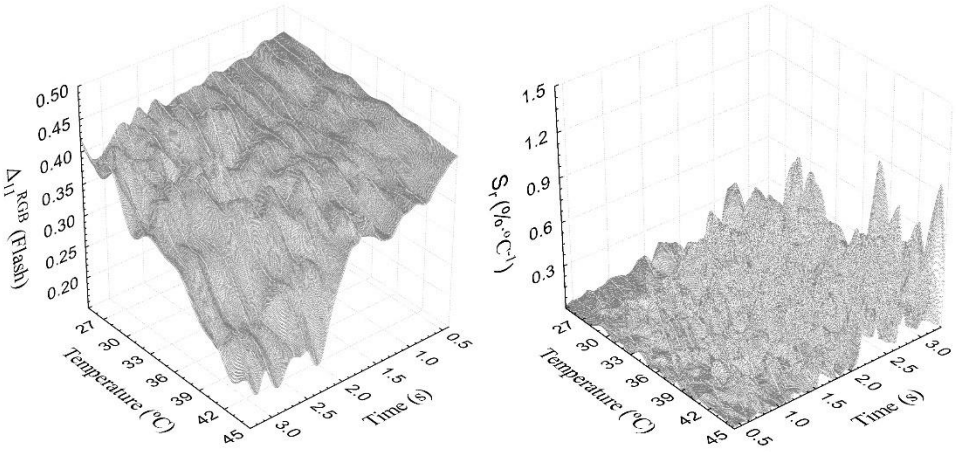
(IX)

$$\Delta_{10}^{IoT} = \frac{B}{(B+G+R_{mean})}$$



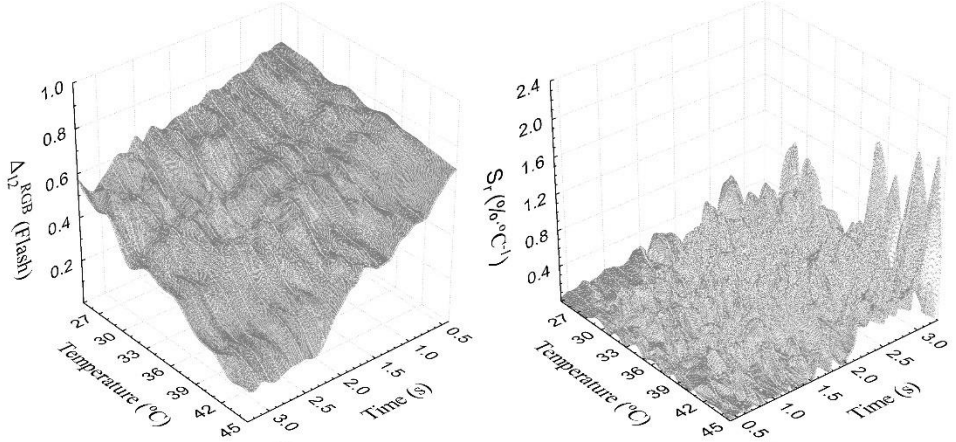
(X)

$$\Delta_{11}^{IoT} = \frac{G}{(B+G)}$$



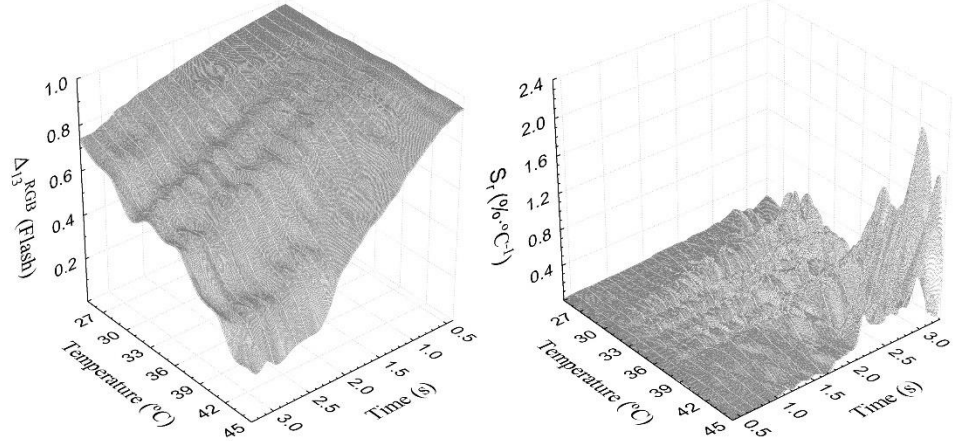
(XI)

$$\Delta_{12}^{IoT} = \frac{G}{(B+R_{mean})}$$



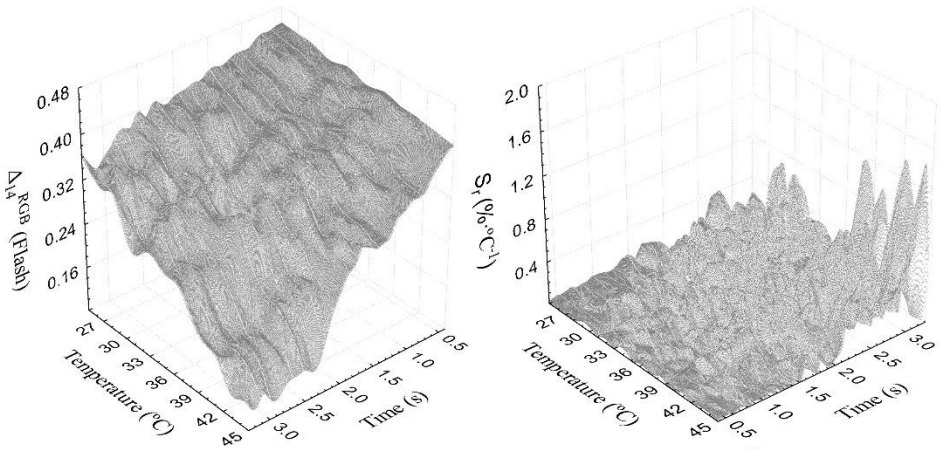
(XII)

$$\Delta_{13}^{IoT} = \frac{G}{(G+R_{mean})}$$



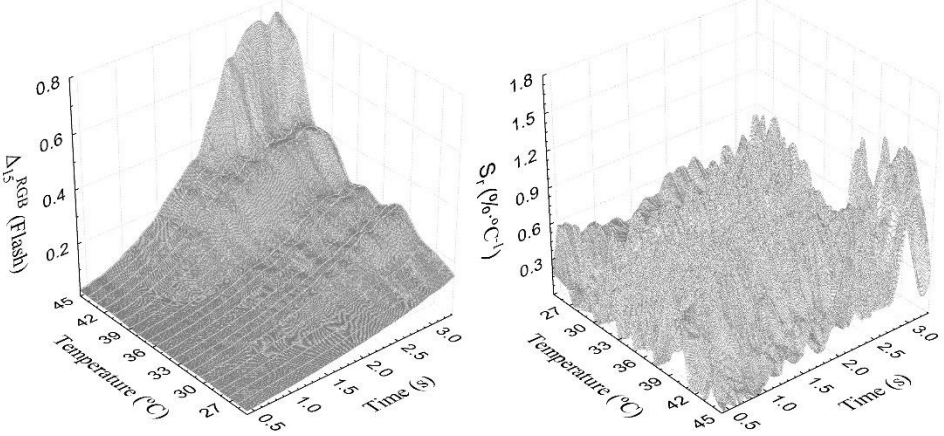
(XIII)

$$\Delta_{14}^{IoT} = \frac{G}{(B+G+R_{mean})}$$



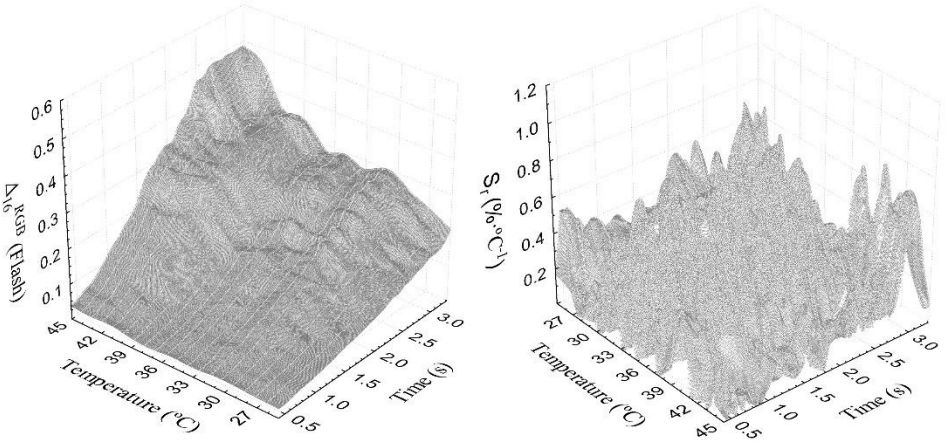
(XIV)

$$\Delta_{15}^{IoT} = \frac{R_{mean}}{(B+G)}$$



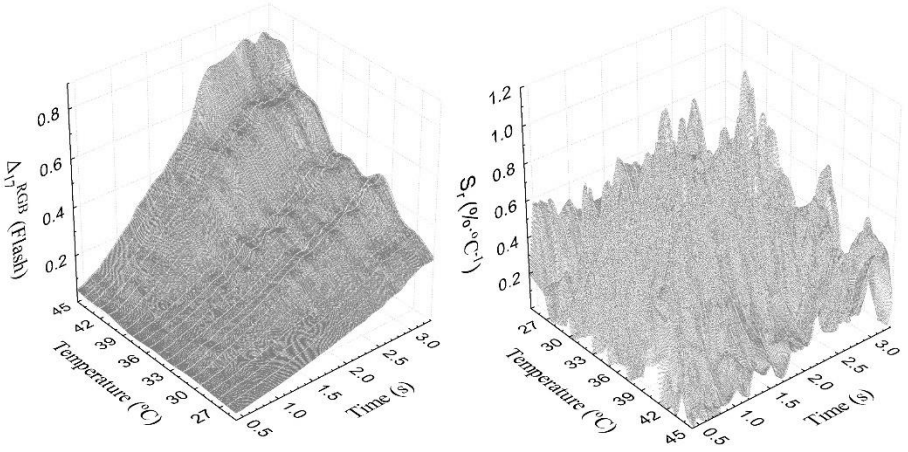
(XV)

$$\Delta_{16}^{IoT} = \frac{R_{mean}}{(B+R_{mean})}$$



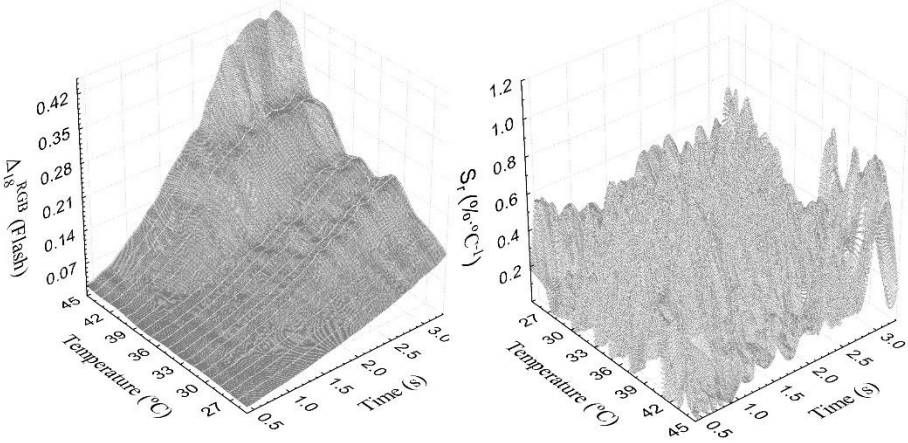
(XVI)

$$\Delta_{17}^{IoT} = \frac{R_{mean}}{(G+R_{mean})}$$



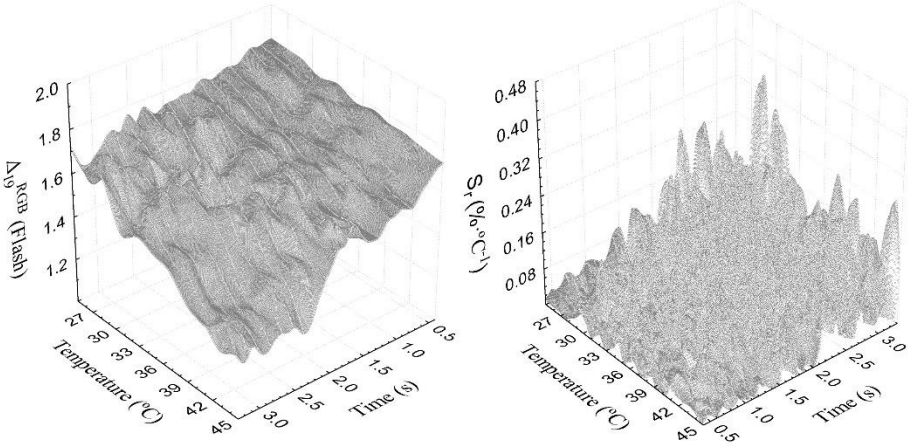
(XVII)

$$\Delta_{18}^{IoT} = \frac{R_{mean}}{(B+G+R_{mean})}$$



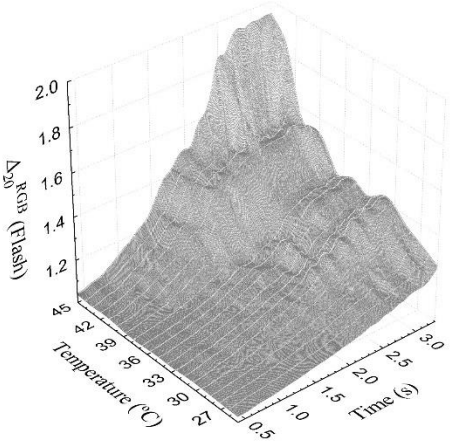
(XVIII)

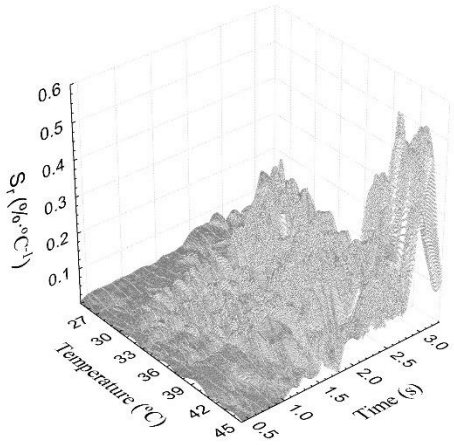
$$\Delta_{19}^{IoT} = \frac{(B+G)}{B}$$



(XIX)

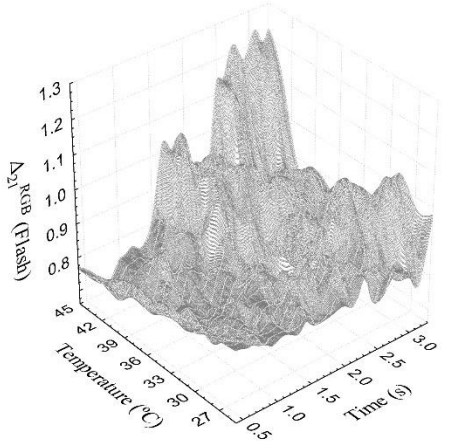
$$\Delta_{20}^{IoT} = \frac{(B+R_{mean})}{B}$$

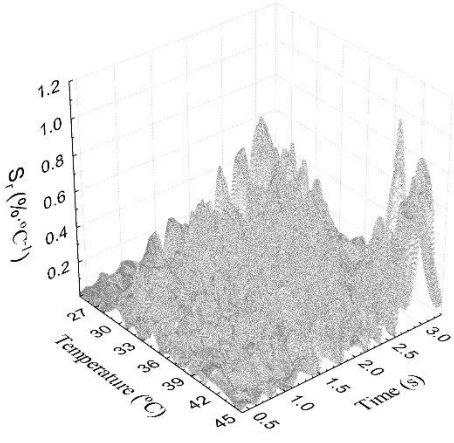




(XX)

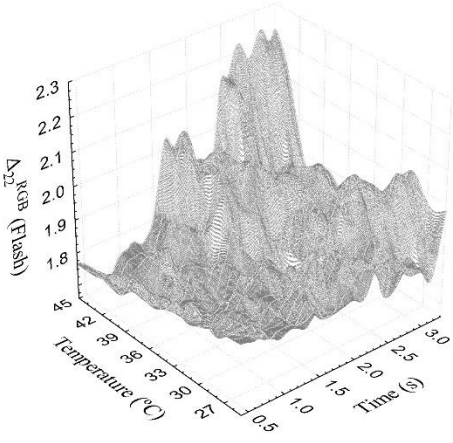
$$\Delta_{21}^{IoT} = \frac{(G+R_{mean})}{B}$$

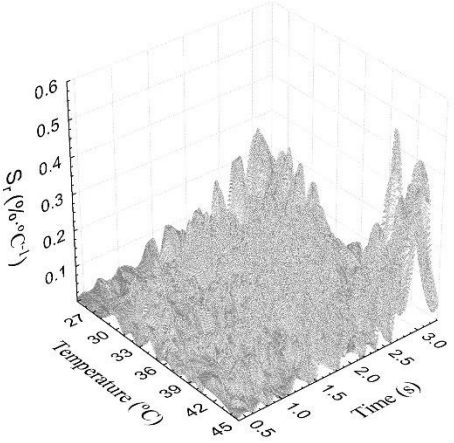




(XXI)

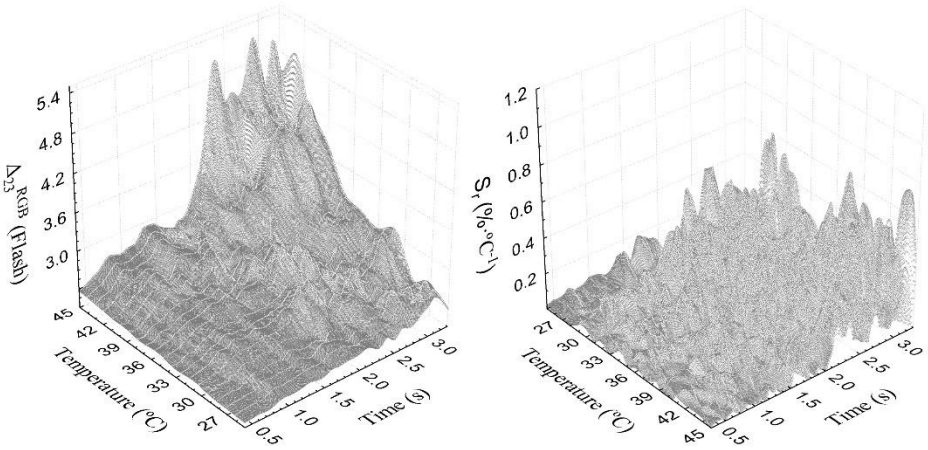
$$\Delta_{22}^{IoT} = \frac{(B+G+R_{mean})}{B}$$





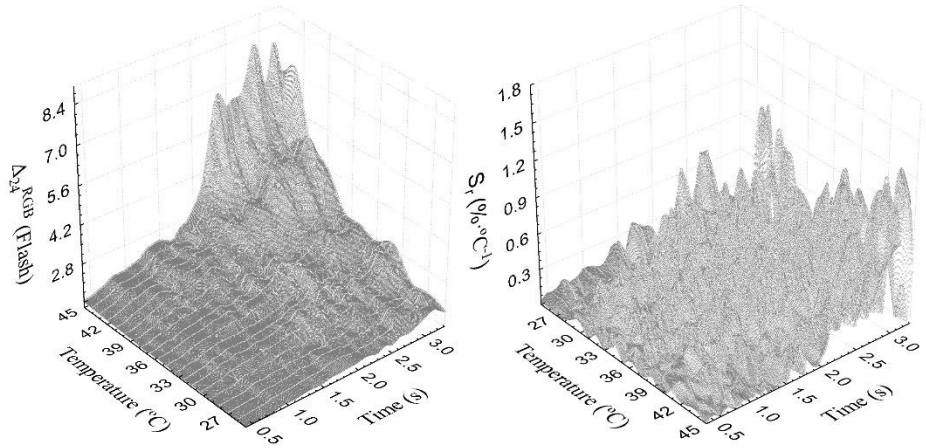
(XXII)

$$\Delta_{23}^{IoT} = \frac{(B+G)}{G}$$



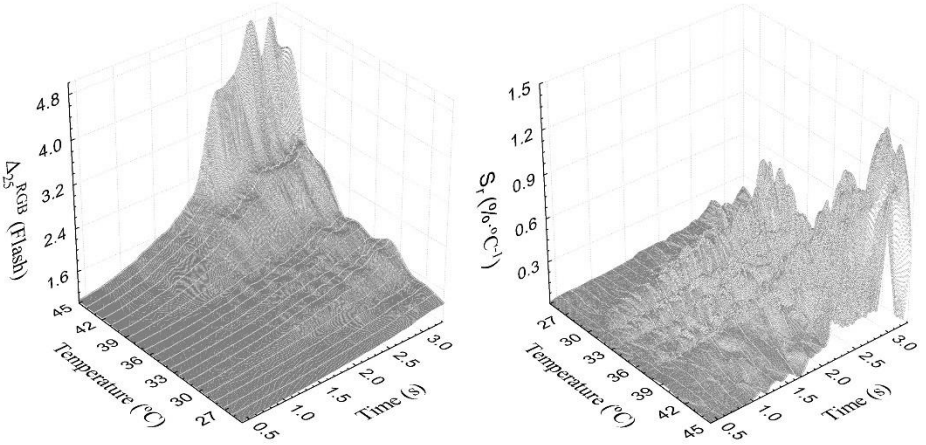
(XXIII)

$$\Delta_{24}^{IoT} = \frac{(B+R_{mean})}{G}$$



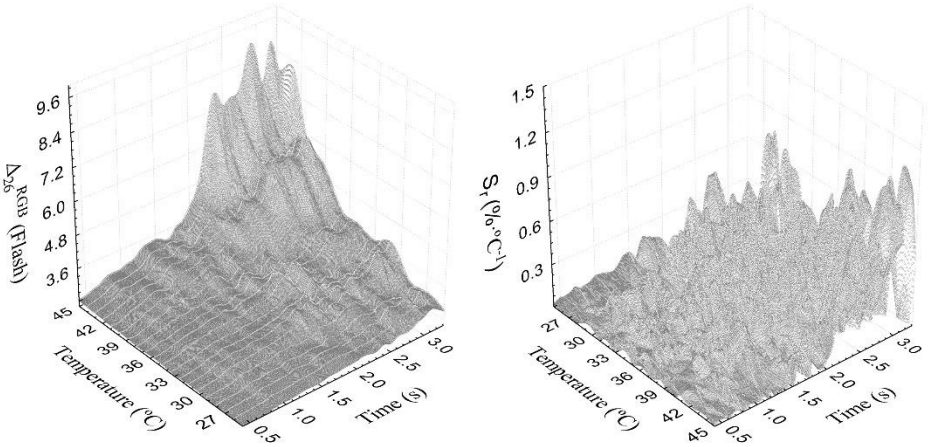
(XXIV)

$$\Delta_{25}^{IoT} = \frac{(G+R_{mean})}{G}$$



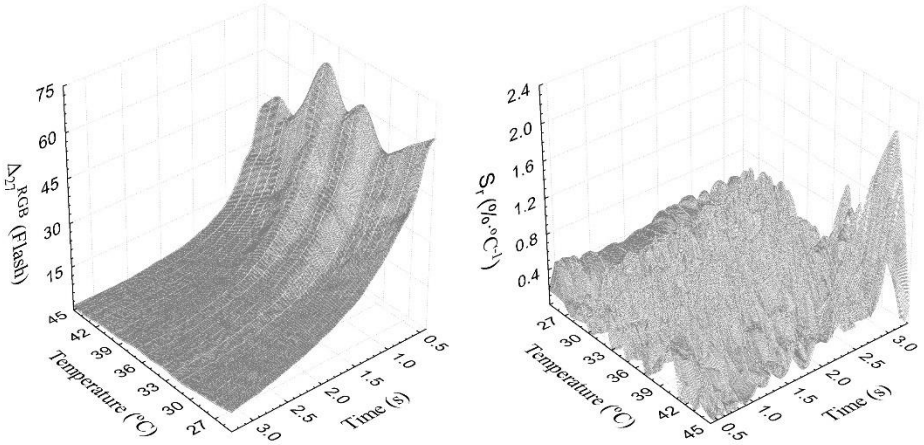
(XXV)

$$\Delta_{26}^{IoT} = \frac{(B+G+R_{mean})}{G}$$



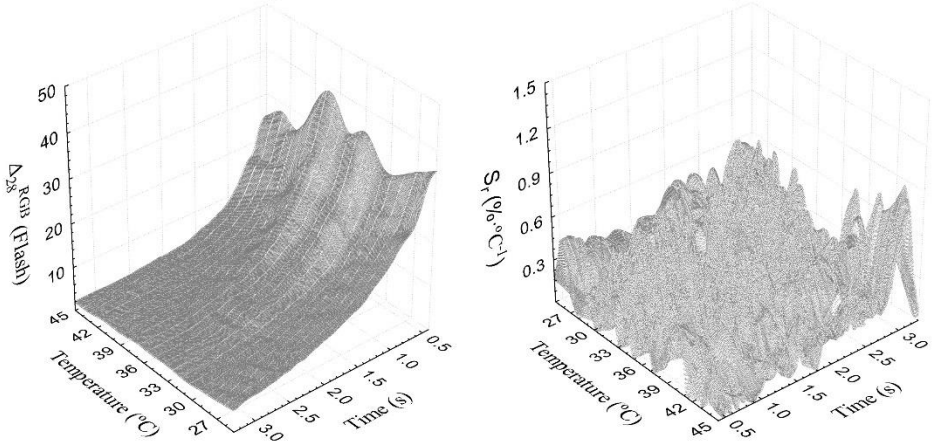
(XXVI)

$$\Delta_{27}^{IoT} = \frac{(B+G)}{R_{mean}}$$



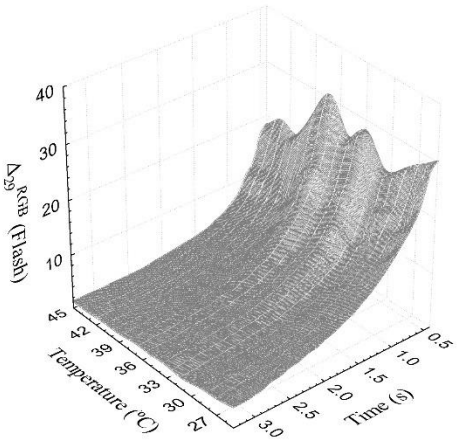
(XXVII)

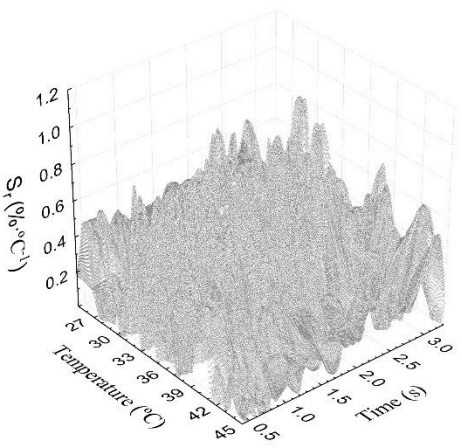
$$\Delta_{28}^{IoT} = \frac{(B+R_{mean})}{R_{mean}}$$



(XXVIII)

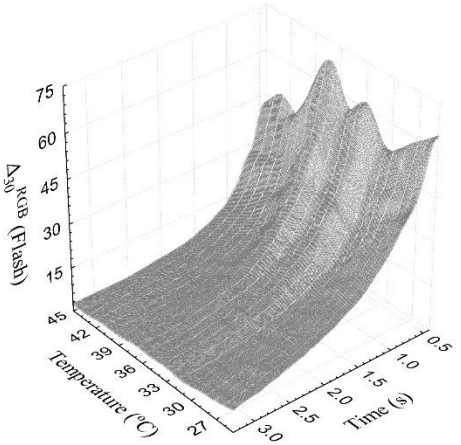
$$\Delta_{29}^{IoT} = \frac{R_{mean}}{(G+R_{mean})}$$

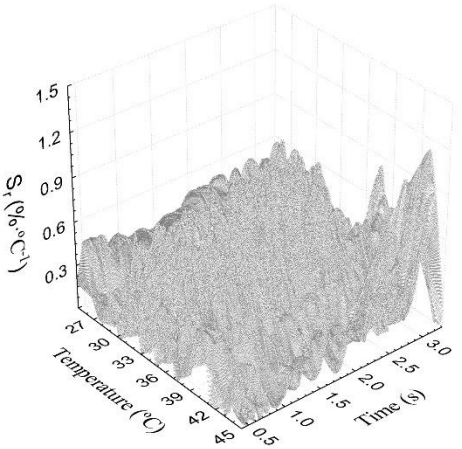




(XXIX)

$$\Delta_{30}^{IoT} = \frac{R_{mean}}{(B+G+R_{mean})}$$





References

1 L. M. S. Dias, L. Fu, R. F. P. Pereira, A. N. Carneiro Neto, V. de Z. Bermudez, P. S. André and R. A. S. Ferreira, *FlexMat*, 2024, **1**, 116–126.

2 Z. Xu, X. Sun, P. Ma, Y. Chen, W. Pan and J. Wang, *J. Mater. Chem. C*, 2020, **8**, 4557–4563.

3 W. He, X. Sun and X. Cao, *ACS Sustain. Chem. Eng.*, 2021, **9**, 4477–4486.

4 S. F. H. Correia, L. Fu, L. M. S. Dias, R. F. P. Pereira, V. de Zea Bermudez, P. S. André and R. A. S. Ferreira, *Nanoscale Adv.*, 2023, **5**, 3428–3438.

5 P. Yu, X. Wen, Y.-R. Toh and J. Tang, *J. Phys. Chem. C*, 2012, **116**, 25552–25557.

6 İ. Y. Elbeyli, *Hydrometallurgy*, 2015, **158**, 19–26.

7 Y. Jie, D. Wang, R. Chen, J. Zhang, W. Li, J. Huang, P. Dai, Y. Gao, F. Li and J. Fang, *Nanoscale*, 2023, **15**, 3337–3344.