Journal of
Energy
Technology

VISUALISATION OF 4D THERMAL MAPS

4D VIZUALIZACIJA TEMPERATURNEGA POLJA

Gorazd Hren ℜ

Keywords: real-time visualisation, volume rendering, spatial interpolation

Abstract

Today, there are many building energy simulation systems. Most rely on theoretical thermal models and numerical simulations to improve the building design and achieve good energy-efficient living conditions. The 3D visualisation technique proposed in this paper analyses temperature distribution in a room. A room is divided into uniform grids, and the temperature field is established by calculating temperature values for each cell using an inverse distance weighted interpolation algorithm with sensor data. These cells are visualised by voxel rendering. An X3D simulation system coupled with a cost-effective set of temperature sensors can provide valuable insights into room design. A measured thermal status can lead to energy savings and improved thermal comfort with improved efficiency. The paper discusses the cost-effective and simplified process for a hardware-software prototype system that can provide real data-driven 4D thermal maps for buildings. The system can be extended to provide 4D thermal maps for more rooms and multi-storey buildings.

Povzetek

Dandanes obstaja veliko aplikacij za energijsko simulacijo stavb in večina uporablja teoretične toplotne modele in numerične simulacije za izboljšanje zasnove stavbe in doseganje energetsko učinkovitih pogojev človeškega udobja. V tem prispevku je za analizo porazdelitve temperature v prostoru predlagana tehnika 3D vizualizacije v predpisanih časovnih okvirjih.

Prostor je razdeljen na mrežo celic, temperaturno polje je izračunano z uporabo algoritma interpolacije inverzne razdalje. Iz podatkov o poziciji in vrednosti temperature v posameznem

---

ℜ   Corresponding author: Assoc. Prof. Gorazd Hren Ph. D., University of Maribor, Faculty of Energy Technology, Hočevarjev trg 1, 8270 Krško, Gorazd.Hren@um.si

senzorju v prostoru interpoliramo vrednosti temperaturnega polja v celotnem prostoru. Za vizualizacijo uporabimo metodo upodabljanja volumetričnih slikovnih pik (voxel). Simulacijski sistem X3D skupaj s stroškovno učinkovitim naborom temperaturnih senzorjev lahko zagotovi vpogled v temperaturno polje prostora na osnovi izmerjenih temperatur. Temperaturno polje prikaže stanje temperature v celotnem prostoru v časovnih korakih, kar lahko privede do prihrankov energije, izboljšanega toplotnega udobja z izboljšano učinkovitost upravljanja klimatizacije prostora. Prispevek predstavlja učinkovit postopek za izdelavo 4D temperaturnega polja na podlagi izmerjenih podatkov. Rešitev je mogoče razširiti za časovno vizualizacijo temperaturnega polja za večnadstropne zgradbe.

1   INTRODUCTION

Thermal management is an effective method to reduce energy consumption. [1] Thermal management protocols can optimise temperature distribution, decrease heating and cooling costs and increase human comfort. The crucial task of thermal management is understanding the airflow and finding the hot or cold spots. Visualisation of temperature distribution as a 3D temperature field is an efficient method to achieve this. Many attempts only visualise the temperature field in 2D, which then needs to be revised to present the entire temperature in space. [2,3] For the most part, computational fluid dynamics (CFD) methods, [4] which are time consuming and neglect the actual temperature state in spaces, lake home rooms, warehouses, or production halls, are adopted. However, its effectiveness is limited by the computational costs associated with predictions, which limit online operation and maintenance applications.

Consequently, effective temperature assessments still depend on the limited measurement data obtained by discrete temperature sensors. [5,6] Nowadays, many closed spaces are grided with temperature sensors. A method to produce 3D visualisation for thermal management with the temperature distribution by 3D particles from temperature data in time extracted from sensors, [7,8] but the results still need to be understood. This approach first establishes the temperature field with data from sensors and, after visualising the temperature distribution field in time, is performed.

Three aspects are taken into consideration to gain insight into temperature distribution in closed spaces:

- Practical 3D visualisation program by modelling temperature as a data volume set. From this, temperature distribution could be visualised as volume rendering.

- For calculating the temperature field, the inverse distance weighted interpolation (IDW) is used. Heat transfer rules and transfer distance are used for the method, which makes the visualisation results more reliable. The execution of complex algorithms is a question of the available CPU RAM. [9]

- The design rendering method (and variations) for different analysis purposes. Rendering temperature field grid of voxels where each is represented as a box using transparent colour as a visual element.

- The approach can be roughly divided into two steps: establishing the temperature field from sensor data and visualising the temperature field.

2   SPATIAL TEMPERATURE FIELD CALCULATION

The first step is to divide the room into a uniform grid, with each cell becoming the element for the visual component. The temperature of each cell is calculated by the IDW method, which simulates the sensors as heat sources and bounds the transfer distance of sensors. IDW reduces the complexity of computation and maintains the locality.

The temperature field is divided into space voxels, transforming the problem into volume data. [10-13] The abstracted model is a single room r=(w,h,d), w-width, h-height, and d-depth. The room is divided into a 3D grid that is regularly spaced. The cell size determines the density of the grid and space division. Each cell is considered a voxel, and the whole room represents a collection of voxels.

The temperature field describes the temperature distribution in the entire room. The voxels define it, and each cell's temperature value is defined. The collection of all voxels with temperature data is volume data. The temperature field is equivalent to a set of volume data, and visualising the temperature distribution can be transformed into a volume rendering problem, specifically to calculate colours representing each voxel in space.

For visualising the volume data set, the temperature value of each voxel must first be calculated by a spatial interpolation method with data from sensors. Temperature distribution is a result of heat transfer. Voxels close to each other have similar temperature values, while voxels at larger distances are likely different. Thermal distribution is thus distance-related. Inverse distance interpolation IDW is a widely used method with known values in scattered points corresponding to the thermal distribution characteristic. The basic IDW function is shown below:

$$T(x) = \sum_{i=1}^{n} \frac{\frac{T_i}{d(x,x_i)^p}}{\sum_{j=0}^{s}\frac{1}{d(x,x_j)^p}} \quad T(x) = \sum_{i=1}^{n} \frac{\frac{T_i}{d(x,x_i)^p}}{\sum_{j=0}^{s}\frac{1}{d(x,x_j)^p}} n \qquad (2.1)$$

where T is the interpolation value at point x from data Ti=T(xi) and i∈ {0,1,...n}; x_i is the known point, d is the Euclidean distance between the estimated point and the observation point, x, and xi, and s is the number of points with known values, and n is the total number of voxels. p is the positive real number, a power parameter; greater values of p assign greater influence on values closest to the interpolation point. The implicit assumption of the classical IDW interpolation method is that the objects to be estimated are uniformly distributed in space with the power parameter p=2.

2.1   Modelling and visualisation in application

The abstracted model, a single room 3x3 m (SR33), was produced as a testbed to develop an approach, check different options, and verify the results. MatLab was used as the programming language for calculations and preliminary visualisation. MatLab applies *interp1*, *interp2*, and *interp3* functions in [14] for uniform data. These functions use polynomial techniques, fitting the supplied data with polynomial functions between data points and evaluating the appropriate function at the desired interpolation points. When the sample data is scattered, the interpolation techniques use a triangulation-based approach to compute interpolated values. The interpolating function passes through the data points. This is an important distinction between interpolation

and curve/surface fitting. The function does not necessarily pass through the sample data points in the fitting. We do have measurements in space: the exact position and temperature value at that point.

The SR33 is grided with evenly spaced points. The sensors' location and temperature data are prescribed. In the SR33 corner, a temperature of -10 °C is prescribed. In the opposite corner, a temperature of 10 °C is prescribed, while in the middle of the SR33, 25 °C is prescribed. The temperature field of SR33 was organised as a set of volume data, thus allowing the temperature field to be visualised by rendering methods.

Each voxel is assumed to be an element (sphere or box) and is rendered as an element. The colour and scale of transparent elements are changed to determine heat distribution in the room. Visualisation of spatial data is challenging to achieve.

In order to visualise the volumetric data set, the IWD must be interpolated from the sensor data to calculate the temperature value of each voxel. The space is modelled with a rough grid for easier and faster calculations. The sensors' position, their temperature data, and the calculated temperatures in all interpolation points are determined. For the refined mesh, the mapping is performed using the MatLab *interp3* spatial interpolation function.

The representation and visualisation of room temperature in MatLab is limited, although MatLab contains many functions in the libraries for drawing graphs and pictures. Various functions can be applied to 3D volume data; Figures 1-3 show the various functions and some of their variations and options. Only three functions were relevant: *scatter*, *slice,* [14] and *VOXview.* [15] Figure 1 shows the results using the *scatter* and *VOXview* functions, while Figure 2 uses the slice function, which produces multiple slices. Figure 3 presents different IWD interpolations with power parameters p=1 and p=2. In all images, the temperature mapping is scaled between the highest and lowest temperature from all temperature data values.
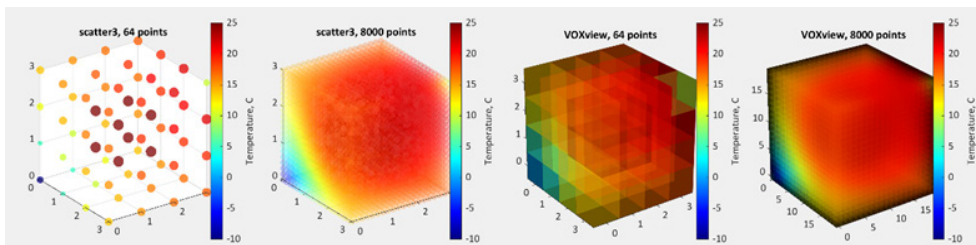

Figure 1: *Results of IDW interpolation presented by scatter and VOXview function; with basic grid and with inerp3 refined grid, respectively*

Visualisation of the temperature field helps to analyse and manage heat flows. Relevant questions include whether there is a hot spot and the overall thermal condition. An overall view and details are required to determine the general thermal state of the room. To produce a visualisation of changes in the temperature field over time, looping with a pause is necessary.

There needs to be more quality visualisation since the figures present the temperature field as a sequence over time that cannot be controlled. A detailed view could be achieved with cross-section views, but phenomena such as hot spots or sudden temperature changes could not be predicted in advance.
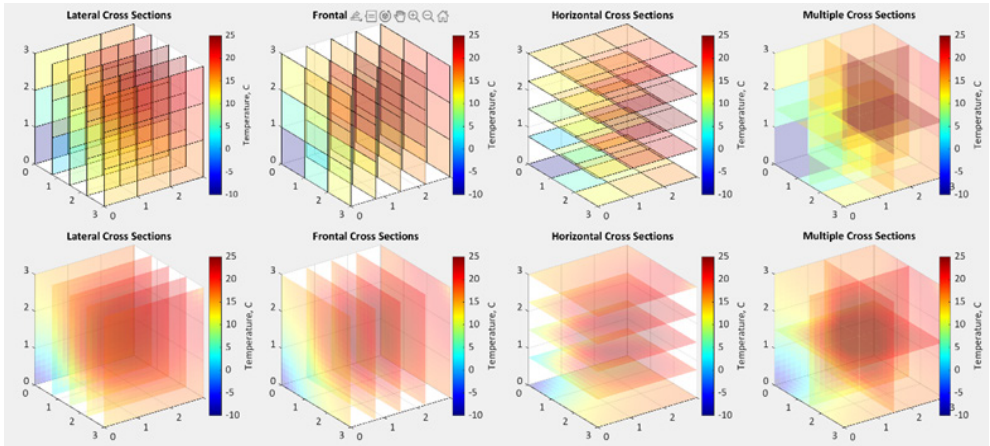
Figure 2: Results of IDW interpolation presented by cross sections and slice function; with basic grid and with inerp3 refined grid, respectively
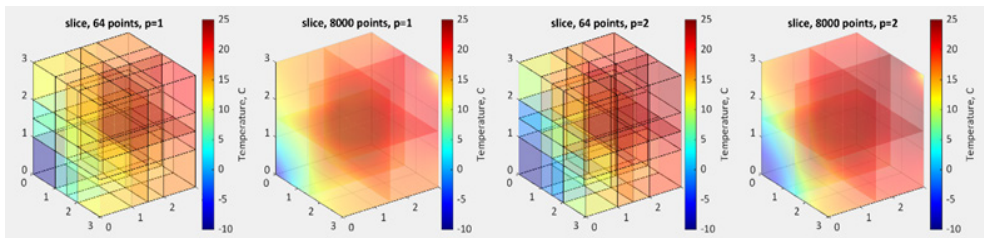

Figure 3: Results of IDW interpolation presented by cross sections and slice function with power parameters of p=1 and p=2; with basic grid and with inerp3 refined grid, respectively

2.2   X3D visualisation

X3D is an open standard with worldwide validity certified by the International Organisation for Standardisation. The openness of X3D ensures the independence of a data format. So, like an XML document, an X3D document consists of nodes organised into parent and child elements. Each element has a unique name, defined in the XML Schema Definition. These names are self-explanatory, and are described in a computer language that is also readable and writeable by users. Four predefined positions are accessible from the view bar at the bottom of the browser: top, front, back, and front-top view. Navigation is defined as both flying and walking through the scene.

X3D offers a wide range of functionalities and node types to include data from many sources. There are various ways to create 3D objects in X3D. X3D geometric primitives (e.g., *Box*, *Sphere*) are good possibilities for 3D thermal map representation as their RGB and $\gamma-$values can be controlled straightforwardly, the author chooses to model the volume in the room as a set (mesh) of semi-transparent colour cubes.

The cube's colour represents the temperature at a certain location. Values are interpolated to determine intermediary temperature values. IDW interpolation is used currently, as illustrated in the previous chapter. However, other more complex models may enhance the visual representation of the sensor data.

X3D scene graph structure consists of nodes with properties called fields. A field includes an attribute storing the basic data type (e.g. a floating-point value or a vector of three values). The routing of an output field of a geometrical child node of an X3D to the input field of another makes animation within the scene graph possible. X3D event animation at author-time is an event cascade across sensor, interpolator, and transform nodes. The *ROUTE* mechanism connects the node sending the event and the node receiving that event. A *TouchSensor* is placed on the keyword geometry used as a trigger. The event is first routed to a *TimeSensor* node, where the cycle interval is controlled and can be easily modified. To complete the colouring effect, the *startTime* field of this *TimeSensor* is routed to a *ColorInterpolator*, and then to the *Material* node. 3D worlds can be built without programming, just by describing the scene graph using X3D syntax, particularly in geoscience, BIM. [19]

In our case, geometry is not an issue, but it is necessary to transform a vast amount of data, which makes it possible to perform visualisation with computer programming. There has been much experience with visualisations using X3D in the scientific community. A voxel representation algorithm is used to describe the 3D data. This algorithm describes 3D objects with flexible volumes for the whole internal space with predefined shapes (box and sphere) as a 3D mesh. The result of the study should give a good indication of the possibility of producing sufficient 4D maps that could be published via Internet Standards. While X3D is an open standard, powerful, extensible, and has a consortium that maintains and supports it, it needs to provide a more complex environment of quick development. [16]

3   RESULTS AND DISCUSSION

The proposed method has been deployed in one room of the Faculty of Energy Technology (LABB) laboratory building. The LABB is a standalone building consisting only of one room with computer hardware equipment. Data collected from 24 sensors are spread in all corners of the LABB (inside, in-wall, and outside) and the middle of the roof. The distribution and temperature values of these sensors are demonstrated in Figure 4.

All sensors are linked to a SCADA system that continuously stores data every 5 minutes. The positions of the sensors can be easily seen in Figure 4. The data were obtained in an MS Excel file and reduced to include the temperature values for each hour of the day for each month. Only in-room temperature measurements were considered for visualisation. Since the MatLab application was developed to read and calculate the temperature distribution field in the room, it was possible to change the time step and duration of the measurements by editing the input file.

Given the number of sensors deployed in the room is s, and the sensor set is represented by S, then a sensor is denoted by S_i, i$\hat{\mathbf{I}}$ {1,2, ...s} and S_i=(x,y,z,T), where values represent the location coordinate in width, height, depth, and the value of the temperature respectively.
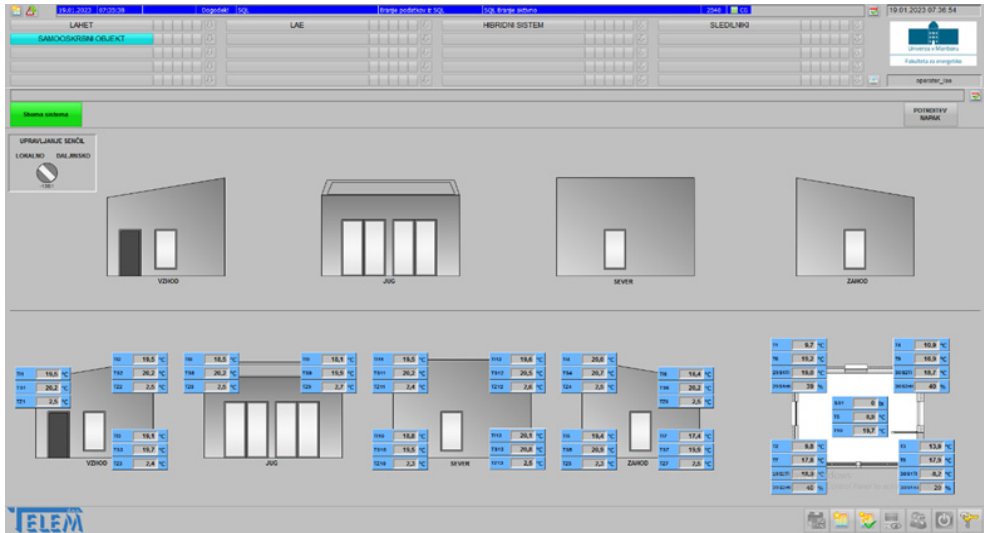
Figure 4: *SCADA layout showing temperature measuring positions sensors*

The starting point is the LABB geometric model, modelled in the SolidWorks CAD system, extracted to VRML97, and then translated to the currently active X3D format. The walls and glass in the windows are set to a semi-transparent colour, and the computer and projector parts are pink, as can be seen in Figure 5, for easy monitoring.

The geometry model and grid of points from the application are linked (the coordinate systems were very helpful). The application reads the temperature values from the Excel file for nine sensors. All considered sensors have a temperature value for each hour for the whole month (720 values for each sensor). Adding more data could significantly slow down the calculation speed. The application transfers data to nodes and X3D files, simultaneously produces all geometric data and all *ColorInterpolator* data for time frames and calculates a representative colour for the current temperature in a fixed colour bar that ranges from the lowest to the highest temperature value.

The results are difficult to show on screen or paper. The temperature in the room is very consistent, with only very small differences between the data of the individual sensors being recorded. The temperature management of the LABB is very efficient. Therefore, two sequences are chosen to show the results: in Figure 6, when the central computer is active, which also has a sensor, and in Figure 7, which shows the temperature field when the projector is active in global view and local detail.
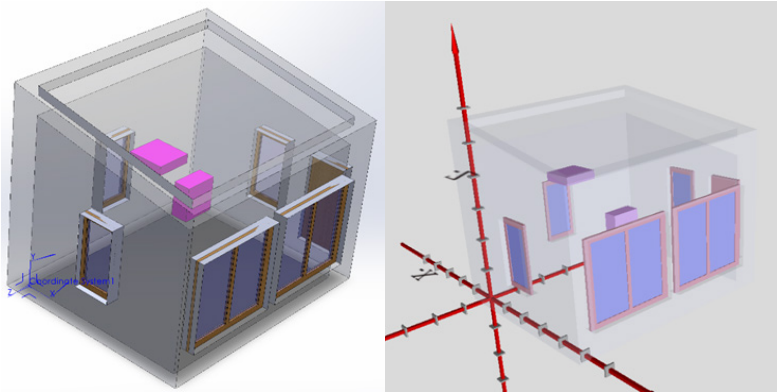
Figure 5: *CAD model of the LABB and its presentation in X3D viewer*
(the coordinate system is added for checking data)
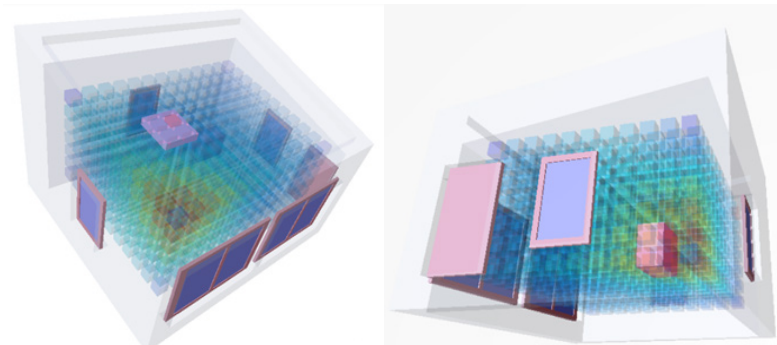

Figure 6: *Temperature distribution field in LABB, on X3D viewer while the computer runs*
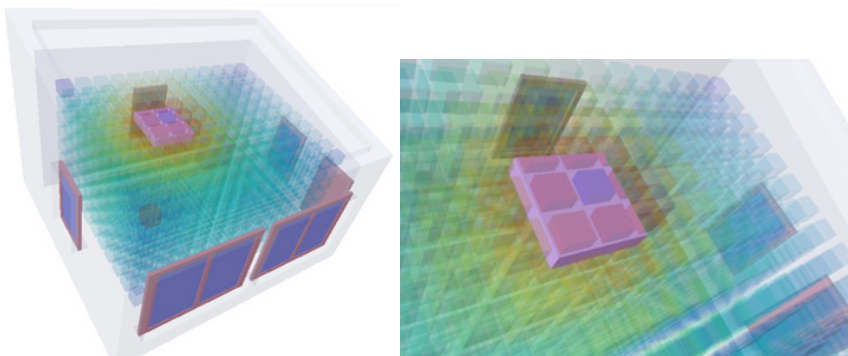

Figure 7: *Temperature distribution field in LABB, on X3D viewer, while the projector is on*

4   CONCLUSIONS AND FUTURE WORK

This paper has presented a basic and cost-effective system for temperature data acquisition and an X3D module for 3D Thermal Map representation. It models the temperature field as a volume data set. The voxel value is obtained by interpolating the sensor data using the IDW method. Voxel rendering is used for visualisation so that the results give a more comprehensive understanding of heat distribution. The proposed method was applied in an actual work environment. The results proved that our method could visualise heat fluxes in 3D and in real time, and help in efficient heat distribution analysis. Since voxel computation and rendering could be done in parallel, the method can be easily implemented on GPU to speed it up. This method can also be used in other applications, such as in the visualisation of humidity or in other environmental factors in closed spaces. In addition to this, optimised sensor distribution will provide better interpolation results. Even though the intelligence is lost due to the transfer of these files (mainly parametric objects) into the X3D environment, the proposed system replaces the virtual model of a building with a three-dimensional virtual prototype containing a swarm of sensors that generates important thermal information ready to be used. As a result, embedded information can be conveyed and input characteristics exported to other analysis tools, such as CFD or ultrasonic method. [17]

The most exciting part of this research is the low cost of the system to implement and the visualisation and production of heat maps without specialised software. The X3D concept of paths, interpolators, and sensors for animating elements still needs some improvement. The problem is that X3D needs a built-in event-based mechanism to trigger complex, reusable animations. The widespread use of the event approach in web technology shows that it is an effective concept that adapts to different needs. The lack of such a mechanism in X3D makes creating animated and interactive scenes complex in some scenarios.

References

[1]   **P.X. Gao, A.R. Curtis,B. Wong,S. Keshav:** *It's not easy being green.* ACM SIGCOMM Computer Communication Review, 42(4), 211-22, 2012

[2]   **M.C. Hao, R.K. Sharma, D.A. Keim, U. Dayal, C. Patel, R. Vennelakanti:** *Application of Visual Analytics for Thermal State Management in Large Data Centres,* Computer Graphics Forum, 29(6):1895-904, 2010

[3]   **S. Yan, X. Li:** *Analytical expression of indoor temperature distribution in generally ventilated room with arbitrary boundary conditions*, Energy and Buildings, Vol. 208, 2020

[4]   **H.Edtmayer, D. Brandl, T. Mach, E. Schlager, H. Gursch, M. Lugmair, C. Hochenauer:** *Modelling virtual sensors for real-time indoor comfort control*, Journal of Building Engineering, Vol. 67, 2023

[5]   **C. Lou, H.C. Zhou:** *Deduction of the two-dimensional distribution of temperature in a cross section of a boiler furnace from images of flame radiation*, Combust. Flame, Vol. 143 (1–2), p.p. 97-105, 2005

[6]   **G.Jiang, M.Kang, Z.Cai, H.Wang, Y.Liu, W.Wang:** *Online reconstruction of 3D temperature field fused with POD-based reduced order approach and sparse sensor data*, International Journal of Thermal Sciences, Vol. 175, 2022

[7]   **B. Lange, N. Rodriguez, W. Puech, H. Rey, X. Vasques:** *A 3D particle visualization system for temperature management.* Proceedings of SPIE - The International Society for Optical Engineering, 7868, 2011

[8]   **H.Edtmayer, D.Brandl, T.Mach, E.Schlager, H.Gursch, M.Lugmair, C.Hochenauer:** *Modelling virtual sensors for real-time indoor comfort control*, Journal of Building Engineering. 67, 2023

[9]   **R. Jedermann, J. Palafox-Albarrán, P. Barreiro, L. Ruiz-García, J. Ignacio Robla, W. Lang**, "*Interpolation of spatial temperature profiles by sensor networks*," *SENSORS, 2011 IEEE*, Ireland, 778-781, 2011

[10]   **I. Cohen, D. Gordon:** *VS: a surface-based system for topological analysis, quantization and visualization of voxel data.* Med Image Anal. 13(2), 245-56, 2009

[11]   **J. Kruger, P. Kipfer, P. Kondratieva,RD. Westermann:** *A particle system for interactive visualization of 3D flows*, IEEE Trans Vis Comput Graph, 11(6), 744-56, 2005

[12]   **S.M. Seitz, C.R. Dyer:** *Photorealistic scene reconstruction by voxel coloring*. Int J Comput Vision, 35(2):151-73, 1999

[13]   **N.A. Taranukha, Z.A. Izabekov:** *A method for voxel visualization of 3D objects.* Program Comput Soft+; 33(6):336-42, 2007

[14]   **MatLab help.**

[15]   **R. Hajovsky, B. Filipova, M. Pies and S. Ozana:** *Using Matlab for thermal processes modeling and prediction at mining dumps*, *12th International Conference on Control, Automation and Systems*, p.p. 584-587, 2012

[16]   **Tim VOXview (https://**www.mathworks.com/matlabcentral/fileexchange/78745-voxview), MATLAB Central File Exchange. Retrieved June 1, 2023

[17]   **F. Quintella, L.P. Soares, A.B. Raposo:** *DWeb3D: a toolkit for developing X3D applications in a simplified environment*, Web3D '10: Proceedings of the 15th International Conference on Web 3D Technology, pp. 45–54, 2010

[18]   **X. Shen,H. Chen, T.M. Shih, X.Qingyu, Z. Hualin:** *Construction of three-dimensional temperature distribution using a network of ultrasonic transducers*, Scientific Reports 9, 12726, 2019

[19]   **A. Schweigkofler, O. Braholli, S. Akro, D. Siegele, P. Penna, C. Marcher, L. Tagliabue, D. Matt:** *Digital Twin as energy management tool through IoT and BIM data integration*, CLIMA 2022 conference, 2022