RoomCanvas: A Visualization System for Spatiotemporal Temperature Data in Smart Homes

B. König, D. Limberger, J. Klimke, B. Hagedorn, and J. Döllner

Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam, Germany

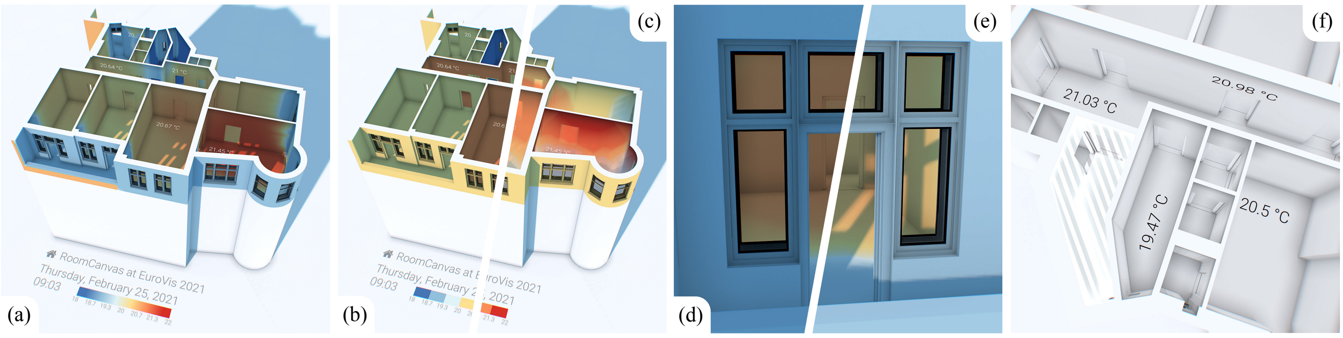


Figure 1: *RoomCanvas depicting data from multiple temperature sensors of an apartment using a diverging color scale: (a) interpolated and projected to walls, floors, and ceilings, (b) aggregated per-room, or (c) displayed using volumetric rendering. Our interpolation has spatial awareness of room-boundaries, doors, and windows. (d) and (e) showcase the geometric detail and different times of day, enabling indoor views and recognizable structural elements. (f) shows our adaptive label positioning and interactive highlighting of rooms. Sun position and lighting are simulated for the time and day at the geographic position in (a) to (e) and the outside temperature is taken into account as well.*

Abstract

Spatiotemporal measurements such as power consumption, temperature, humidity, movement, noise, brightness, etc., will become ubiquitously available in both old and modern homes to capture and analyze behavioral patterns. The data is fed into analytics platforms and tapped by services but is generally not readily available to consumers for exploration due in part to its inherent complexity and volume. We present an interactive visualization system that uses a simplified 3D representation of building interiors as a canvas for a unified sensor data display. The system's underlying visualization supports spatial as well as temporal accumulation of data, e.g., temperature and humidity values. It introduces a volumetric data interpolation approach which takes 3D room boundaries such as walls, doors, and windows into account. We showcase an interactive, web-based prototype that allows for the exploration of historical as well as real-time data of multiple temperature and humidity sensors. Finally, we sketch an integrated pipeline from sensor data acquisition to visualization, discuss the creation of semantic geometry and subsequent preprocessing, and provide insights into our real-time rendering implementation.

CCS Concepts

• ***Human-centered computing*** → *Visualization toolkits;* ***Visualization systems and tools; Information visualization;***

1. Introduction

The living comfort in apartments and houses depends, among other factors, strongly on the thermal comfort of the occupants and thus on the thermal environment indoors [YWY*12, YYL*20]. The availability of low-cost, low-energy, and easy-to-install IoT-enabled sensors has simplified data monitoring and data gathering for consumers. Thus, we need effective tools to explore, monitor, and understand such measurements in their spatial and temporal context. The temporal aspect of collected temperature data requires applying temporal visualization techniques, such as line graphs with a calendar/week/month selection—but those techniques are usually limited to this temporal dimension alone and do not take spatial relations into account. The spatial propagation of heat in buildings creates the need to apply visualization techniques for spatial data, such as heat

DOI: 10.2312/evs.20211048

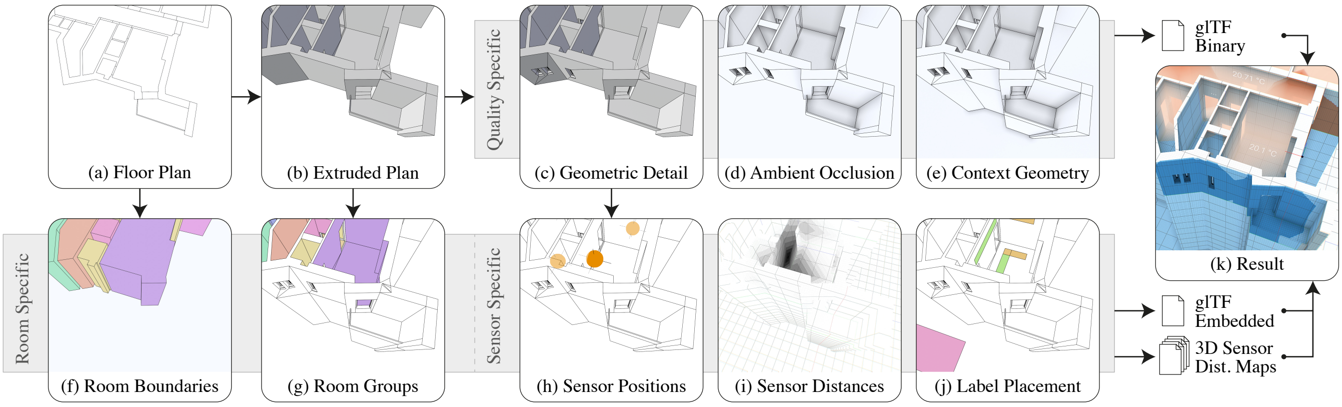


Figure 2: *From a given (a) floor plan, a (b) simplified 3D apartment representation is created via extrusion, which is then used for further preprocessing to increase visual quality and to compute and store data attributes: (c) detailed geometry is added, (d) ambient occlusion is prerendered, and (e) a block for lower floors—indicating the apartment's floor level—and a ground plane are added. At the same time, (f) floor-plan-based extrusion of simplified meshes representing the individual rooms' boundaries, (g) assignment of faces to the corresponding room surfaces, and (h) configuration of the sensors' positions allow for corresponding aggregations and interaction events. For each sensor, (i) distances to every point in a discretized volume spanning the apartment's bounding box are precomputed using a three-dimensional flood fill variant and encoded in 3D textures, and (j) candidates for dynamic label placement are precalculated. (k) based on the outputs of those preprocessing steps, our visualization can interpolate, display, and label sensor data at interactive frame rates (even on smartphones).*

maps with a per-building or per-floor aggregation—often lacking a graphical connection to the building's structure. The spatial structure, however, is also often already readily available for newly built buildings due to the increased integration of *building information modeling* (BIM) techniques and standards, and allows for interesting and insightful visual display of temperature and humidity.

We propose an interactive, web-based, real-time 3D visualization system, called *RoomCanvas*, that is built around the spatial nature of buildings, apartments, and rooms. It allows for the visualization of spatiotemporal data directly at its respective location, employing techniques such as spatial and temporal interpolation, color mapping, and automated label placement. As opposed to existing, expert-targeted BIM simulation and visualization software systems [CDW18], which focus on being mechanically precise and use sufficiently aesthetic models of building structures and contents, we chose to build a system for a broader target audience. Available systems typically do not interpolate sensor values spatially [DBB*18, MCKM15], are not or only partially web-based [MCKM15], and in some cases tightly integrated with existing expert tools such as Revit. To this end, we have slightly different design goals, namely, the suitability for web-based, real-time rendering, a focus on increased aesthetic quality and recognizability using a sufficiently precise model representation of the built environment, and interaction tools targeted towards non-experts.

2. Data Provisioning and 3D Floor Plan Modeling

Our visualization relies on (1) access to real-time or historic temperature data as much as on (2) an accurate and up-to-date 3D floor plan with (3) room semantics, (4) preferred locations for view-dependent labeling, and, finally, (5) precomputed spatial distance data for every data source, i.e., every single sensor.

Temperature Data Integration For our measurements, we combine multiple sensors for temperature, humidity, and atmospheric pressure (inside and outside, usually one per room), with sensors for opening states of windows and doors based on the wireless, low-energy ZigBee protocol, as well as public weather services for outside temperature tracking. The sensors are connected to a ZigBee gateway (Raspbee) on a Raspberry Pi running deCONZ home automation software. Node-RED is then used to filter, transform, and transmit raw sensor data to a real-time data analytics platform via the MQTT protocol. There, the sensor data is semantically structured, analyzed as needed, and stored for efficient provision via authorized API requests. Additional data resources can and—in the case of outside weather information—are added and integrated here. In addition to raw data access, the analytics platform API supports requests for custom data resolutions, time ranges, accumulations, filtering, and more. This, for example, allows to query for time-series as complex as the daily average in temperature around noon for every march of the last five years (assuming data was collected for that time span, obviously).

Drawing of 3D Floor Plans One of our main concerns when designing our 3D visualization was to prevent visual clutter from the beginning on. Instead of using 3D LiDAR scans, photogrammetry, or complex CAD drawings, we model, enhance and optimize the 3D geometry ourselves. To this end, we create a minimal but accurate floor plan (Figure 2 (a)), extrude walls, add ceilings and floors, account for special areas such as balconies, steps, etc., and finally, add doors and windows. During the whole process, we keep the number of faces and vertices as low as reasonably possible.

To increase the authenticity of our floor plans, to provide a reference to the location of the apartment in its surrounding building (context), and to reduce the rendering complexity at run-time, we precompute a light map, e.g., accounting for local ambient occlu-

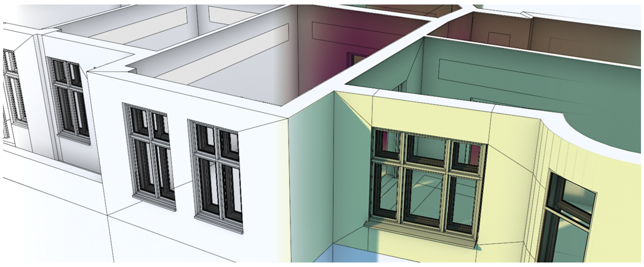


Figure 3: *An optimized, low-polygon 3D floor plan, enhanced with precomputed ambient occlusion maps and real-time shadows to serve as a canvas for the visual display of interpolated sensor values.*

sion [AMHH*18] (Figure 2 (d)) and add a low-polygon representation of lower floors and (optionally) the immediate environment (Figure 2 (e)). Furthermore, we compute the sun position [KS11] and approximate physically plausible soft-shadows [ML19] in real-time using multi-frame sampling [LPKD17].

Semantic Enrichment of 3D Floor Plans We group every polygon of the geometry into semantic units and layers such as lateral walls, ceilings, floors (Figure 2 (f–g)). To allow for user-interaction with individual rooms, we apply a naming scheme and finally store everything as a glTF model. Applying this process enabled us to understand the optimal characteristics of the 3D model and to identify techniques for its semi-automated computation in the future.

Per-Sensor Distance Map Computation To enable spatially unrestricted sampling of temperature values in 3D, we use precomputed 3D distance maps (distance volume) for every single sensor (Figure 2 (i)). A distance volume encodes, at every cell (voxel), the shortest path to its sensor's location in three dimensions, taking the complete geometry into account in 3D, i.e., walls, doors, windows, small openings, etc. First, we create a cuboid with sufficient extent and reasonable resolution, i.e., the oriented bounding box of the 3D floor plan, and a voxel extent of, e.g., 10 to 20 cm. Second, we compute a binary 3D floor plan, storing whether or not a cell intersects with at least one polygon (ceiling, floor, walls, etc.). Finally, we use a modified flooding algorithm to create a 16 bit distance map per sensor. This needs to be done only once per sensor. Adding new sensors does not invalidate earlier distance maps, and only requires to compute a single additional distance map. Since WebAPIs do not have native support for 3D images, we use multiple distance maps—sliced into rows and columns of regular 2D images—encoded in one or more texture atlases. Since there is no 16 bit support either, the low and high bytes are spread over subsequent color channels.

Computation of Preferred Label Locations We manually specify suitable locations for automated text placement (Figure 2 (j)). More specifically, we define rectangles within the 3D floor plan, primarily aligned to floors and upper parts of lateral surfaces. We apply the naming scheme we used for the rooms to the rectangles and store them in an extra layer within the glTF assets. While loading the rectangles, label locations are derived, based on a starting point, an up vector, and a forward-facing vector, and a text running direction with its length encoding the maximum possible text length.

Outside Temperature Additional temperature sensors that are placed outside can provide spatial temperature information to points of interest (terrace, balcony, etc.) and, furthermore, can help to differentiate between opposite exterior facades (e.g., drafts). To establish a baseline for outside temperature, we use data available through open weather APIs, i.e., OpenWeatherMap. If no geographic location is available, heat transfer depending on type and structure of the windows could be calculated and integrated as a data source [AKK15]. However, we assume a uniformly distributed temperature for the entire outdoor area, since the correctness of our interpolation is not sufficiently verified.

3. Visualization

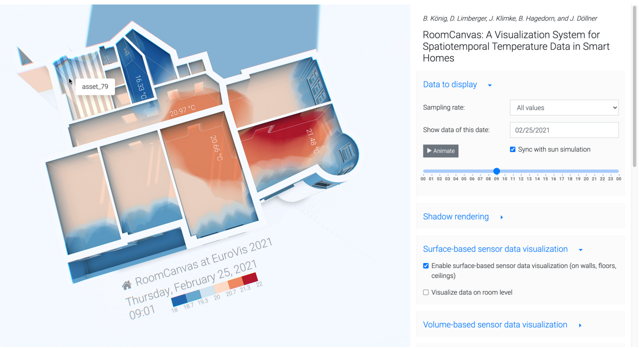
Our proposed visualization system combines the idea of visualizing sensor data in building models [CDW18] with spatial aggregation and interpolation methods, such as inverse distance weighting (IDW) [She68]. Therein, we use the base 3D building model as a canvas for direct in-place visualization of values (Figure 3).

Per-Room Display To provide insights on a room level basis, i.e., to allow for a fast judgment of the temperature distribution at a given time across a building's rooms, our system combines all sensor measurements from inside a room into one mean value per room, which is then mapped via a color mapping and rendered on all walls, the floor, and the ceiling of the corresponding room (Figure 1 (b)).

Fine-grained Spatial Interpolation To allow for a more fine-grained analysis of the approximated temperature distributions in a building at a given time, we use a spatial interpolation method. Figure 1 (a) shows an example for this interpolation: For each position on the projected-to surface, our system uses the distances to each sensor in the building for weighting the corresponding sensor value's influence on the resulting encoded value using IDW. To increase the quality of the spatial interpolation, advanced spatial interpolation methods such as Kriging [OW90]—which has already been successfully applied in interpolating sensor data inside buildings [LLM*18]—could be integrated into our visualization system.

Volumetric Display For the volumetric visualization of interpolated sensor data (Figure 1 (c)), we use a ray-marching-based approach of tracing through the rooms' volumes up to their respective back-faces (Figure 2 (g)) and aggregating the sampled, interpolated sensor values along the way. At each sampled volume location, we again color-map the respective interpolated sensor value to an RGB color value using the selected color scale, while at the same time using a configurable opacity transfer function to determine the sample's opacity. We then accumulate the samples using a standard front-to-back compositing equation and progressively enhance the visual quality of the volume rendering by shifting a pseudo-random generated hash used for jittering, and by adaptively increasing the sampling resolution.

Dynamic Label Placement For dynamic rendering and placement of three-dimensional text labels we use OpenLL [LGB*18] in combination with our precomputed label locations. This allows for algorithmic optimization of the most suitable position, orientation, and length for the to-be-displayed labels, taking into account the



B. König, D. Limberger, J. Klimke, B. Hagedorn, and J. Döllner

RoomCanvas: A Visualization System for Spatiotemporal Temperature Data in Smart Homes

Data to display ▾

Sampling rate: All values

Show data of this date: 02/25/2021

☑ Sync with sun simulation

Shadow rendering ▸

Surface-based sensor data visualization ▾

☑ Enable surface-based sensor data visualization (on walls, floors, ceilings)

☐ Visualize data on room level

Volume-based sensor data visualization ▸

Figure 4: *The interactive, web-based, real-time visualization prototype allows for selecting time aggregation intervals and ranges, for animating historical data, for configuring the type, steps, and discretization of the used color scale, for navigating the 3D scene, and for interacting with elements of the building model.*

camera's position as well as visibility and size constraints. Our system first filters potential label placement positions by a set of criteria based on their orientation (front-facing normal points towards camera; text is readable from left-to-right in screen space co-ordinates) and position (label position is not obstructed by any geometry, determined by sampling the depth buffer; label's world position transformed to screen space co-ordinates is within the view frustum). Then, we sort the remaining potential placement positions using a set of comparison attributes, such as the label's size, its distance to the camera, and the angle between the label's front-facing normal and the vector from the camera's eye to the label's position.

4. Results & Discussion

The results of our work include exemplary static, dynamic, and interactive visualization artifacts for the sample apartment, as well as the design and implementation of an end-to-end system for data measurement, collection, aggregation, and visualization. With the implementation of a prototypical, web-based viewer, we aim to provide a basis for extensions of the proposed visualization system and for experimentation in related use case areas.

Static Visualizations Still images allow for analysis of spatial distributions of temperature values (Figure 1). Different spatial levels of detail, e.g., per-room or spatially interpolated, as well as different time-based aggregations are available for exploration. This already provides a promising stand-alone or complementary visualization for communicating temperature distributions in living scenarios.

Animated Visualizations As an extension of static visualizations, animations can be created based on the proposed visualization system, e.g., by letting the visualized timestamp slide through days or hours, or by moving the camera along a path for insights into different areas. Therein, the use of shadow mapping improves the perception of seasonal and daily time differences.

Sample Use Case One possible use case for RoomCanvas is to determine in which areas of an apartment (either abstracted to the room level, or freely on any given position) a certain temperature threshold is exceeded or fallen below, potentially averaged temporally over a certain time interval, such as in six-hours intervals. By combining the capability of RoomCanvas to precisely interpolate and display such data based on recorded temperature data with its ability of adding temporal and spatial context (through the simulated sun position, real-time rendered shadows, and recognizable visual details of the 3D floor plan), we aim to simplify such use cases—use cases which otherwise could possibly only be tackled by looking at 2D plots and charts and thus without spatial and temporal context.

Interactive Visualizations Our work includes a prototype for an interactive, WebGL-based 3D viewer (Figure 4, visit `http://roomcanvas.dev` for a live demo of the prototype). By loading preprocessed assets—a *glTF binary file* containing the presentation-focused, high-quality apartment model, a *glTF embedded file* containing the apartment's logical and hierarchical structure, and a set of 3D textures encoding precomputed, occlusion-aware distances to sensors—, and by fetching and interpolating data on-demand from the analytics platform API (section 2), the viewer implements the proposed visualization system and allows for data querying, selection, navigation, and spatial interactions.

5. Conclusion & Future Work

We have introduced a system for high-quality visualization of built environments, that combines spatial and temporal aspects of a variety of sensor data. The system can be used to generate aesthetic still images as well as to provide interactive access to real-time sensor data. The fusion of the 3D floor plan with the unified sensor data representation potentially facilitates exploration and correlation of temporal changes in data to past events and actions of occupants.

Following the development of the visualization system, further research needs to be conducted to extend this approach to be applied on larger buildings as well as industrial facilities, that are modeled using complex data models and standards in CAD, BIM (*IFC*), or city modeling (*CityGML*). Here, the challenge is in how to scale the solution to support those large-scale models in terms of data processing, transmission, and rendering. To make our system more accessible, one could simplify the currently manual process of drawing 3D floor plans, for example, by providing an export plug-in for consumer tools such as Sweet Home 3D.

As related work provided promising approaches to finding the optimal number of sensors for a given location in other use case domains [JL09], we believe that a minimum number of sensors can also be determined in an apartment or house, resulting in optimized resource usage while maintaining measurement quality. The work described could further benefit from an optimized location sampling and an improvement in the placement of the sensors [HSL*15], potentially in a semi-automated, assisted way.

Acknowledgements

This work has been supported by the German Federal Ministry of Education and Research (BMBF) through grant 01IS19006 ("KI-Labor ITSE").

References

[AKK15] Arici M., Karabay H., Kan M.: Flow and heat transfer in double, triple and quadruple pane windows. *Energy and Buildings 86* (2015), 394–402. doi:10.1016/j.enbuild.2014.10.043. 3

[AMHH*18] Akenine-Möller T., Haines E., Hoffman N., Pesce A., Iwanicki M., Hillaire S.: *Real-Time Rendering 4th Edition*. A K Peters/CRC Press, 2018. 3

[CDW18] Chang K.-M., Dzeng R.-J., Wu Y.-J.: An automated IoT visualization BIM platform for decision support in facilities management. *Applied Sciences 8*, 7 (2018). doi:10.3390/app8071086. 2, 3

[DBB*18] Delgado J. M. D., Butler L. J., Brilakis I., Elshafie M. Z. E. B., Middleton C. R.: Structural performance monitoring using a dynamic data-driven BIM environment. *Journal of Computing in Civil Engineering 32* (2018). doi:10.1061/(ASCE)CP.1943-5487.0000749. 2

[HSL*15] Huang Y., Shen X., Li J., Li B., Duan R., Lin C.-H., Liu J., Chen Q.: A method to optimize sampling locations for measuring indoor air distributions. *Atmospheric Environment 102* (2015), 355–365. doi:10.1016/j.atmosenv.2014.12.017. 4

[JL09] Jedermann R., Lang W.: The minimum number of sensors – interpolation of spatial temperature profiles in chilled transports. In *Wireless Sensor Networks* (2009), Springer Berlin Heidelberg, pp. 232–246. doi:10.1007/978-3-642-00224-3_15. 4

[KS11] Kolivand H., Sunar M.: Real-time projection shadow with respect to sun's position in virtual environments. *IJCSI 8*, 6 (2011), 80–84. 3

[LGB*18] Limberger D., Gropler A., Buschmann S., Wasty B., Döllner J.: OpenLL: An API for dynamic 2D and 3D labeling. In *22nd International Conference Information Visualisation (IV)* (2018), pp. 175–181. doi:10.1109/iV.2018.00039. 3

[LLM*18] Li J., Li H., Ma Y., Wang Y., Abokifa A. A., Lu C., Biswas P.: Spatiotemporal distribution of indoor particulate matter concentration with a low-cost sensor network. *Building and Environment 127* (2018), 138–147. doi:10.1016/j.buildenv.2017.11.001. 3

[LPKD17] Limberger D., Pursche M., Klimke J., Döllner J.: Progressive high-quality rendering for interactive information cartography using WebGL. In *Proc. ACM Web3D* (2017), pp. 8:1–8:4. doi:10.1145/3055624.3075951. 3

[MCKM15] McCaffrey R., Coakley D., Keane M., Melvin H.: Development of a web-based BMS data visualization platform using building information models. In *Proc. CIBSE 2015 Technical Symposium, London, UK* (2015), pp. 16–17. 2

[ML19] Molzer G., Ledermann F.: Interactive 3D time-integrated solar shadow maps. *Abstracts of the ICA 1* (2019), 255. doi:10.5194/ica-abs-1-255-2019. 3

[OW90] Oliver M. A., Webster R.: Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information System 4*, 3 (1990), 313–332. doi:10.1080/02693799008941549. 3

[She68] Shepard D.: A two-dimensional interpolation function for irregularly-spaced data. In *Proc. 23rd ACM National Conference* (1968), p. 517–524. doi:10.1145/800186.810616. 3

[YWY*12] Ye X., Wolff R., Yu W., Vaneckova P., Pan X., Tong S.: Ambient temperature and morbidity: A review of epidemiological evidence. *Environmental Health Perspectives 120*, 1 (2012), 19–28. doi:10.1289/ehp.1003198. 1

[YYL*20] Yun H., Yang J., Lee B. H., Kim J., Sohn J.-R.: Indoor thermal environment long-term data analytics using IoT devices in Korean apartments: A case study. *International Journal of Environmental Research and Public Health 17*, 19 (2020). doi:10.3390/ijerph17197334. 1