

REAL-TIME SCENE SIMULATOR FOR THERMAL INFRARED LOCALIZATION

Daniel Hauschildt
Jürgen Kemper
Nicolaj Kirchhof
Robotics Research Institute
Technische Universität Dortmund
Otto-Hahn-Str. 8
44227 Dortmund, Germany

Benedict Juretko
Holger Linde
Ambiplex GmbH & Co. KG
Idastr. 11
44388 Dortmund, Germany

ABSTRACT

Exploiting the natural thermal infrared radiation of humans is a promising approach for an accurate, comfortable and inexpensive indoor localization system. However, different sources of disturbance make the development challenging. In order to provide valid sensor data for various scenarios an adequate simulation environment is needed. In this paper we present a real-time scene simulator that allows the simulation of dynamic indoor environments and the resulting output signals of infrared sensors. The composition of such environments is simplified by using an object and sensor database. In order to enable real-time processing, OpenGL and hardware acceleration is applied. Evaluations show that the accuracy of the chosen approach is sufficient to develop algorithms for a Thermal Infrared Localization System (ThILO). Furthermore, it can be shown that real-time processing is possible for a complete location system in typical indoor environments.

1 INTRODUCTION

The ability to locate people is an essential prerequisite to enable *location-based services*. In recent years, several indoor location systems for the field of ubiquitous computing have been developed. Mostly, these systems use technologies like active infrared, radio, ultrasound or cameras to gather the required measurements for localization ([Hightower and Borriello 2001](#), [Tauber 2002](#), [Kemper and Linde 2008](#)).

In fact, none of the common approaches completely meets the special requirements arising in the field of home automation and *Ambient Assisted Living (AAL)*. On the one hand, systems applying active infrared, radio and ultrasound require carrying an active hardware device, called tag or badge that is able to send or receive some kind of measurement signal. On the other hand, camera based approaches that allow locating people without an additional tag depend on lighting conditions and result in a violation of privacy. Hence, they suffer from a lack of consumer acceptance. An alternative approach for the field of home automation and AAL was presented by [Kemper and Linde \(2008\)](#). It exploits the thermal radiation of humans for localization. This makes the tag redundant, since the user is the active source himself. Furthermore, by using low resolution array sensors for radiation detection privacy is also guaranteed.

However, ThILO is challenging since not only humans but every object in the environment with a temperature above absolute zero (see Sect. 3) emits thermal radiation which is called *background radiation* from now on. Since this background radiation cannot be distinguished from that of a human, it has a share in the sensor signal as well and is a potential source of disturbance. Additionally, a discrimination of different users is also not possible.

Thus, in order to realize a robust and accurate localization based on the detection of thermal radiation, the challenges of handling *background radiation* and discriminating different users, have to be tackled by appropriate counteractive measures. Hence, adequate localization algorithms and appropriate filters have to be developed ([Kemper and Linde 2008](#)).

A development based on real measurements only is costly and time-consuming. An adequate simulation environment however, allows a fast and simple set-up of different test environments, the usage and comparison of different sensors and an easier way to test the behaviour of the resulting algorithms under changing conditions. Thus, the development should be eased significantly by a simulation environment.

The most crucial challenges of ThILo arise from environmental dynamics. Devices that heat up or cool down as well as users that move, leave or enter a room are examples for difficult situations that must be handled by the localization algorithms. In order to create and simulate such scenes, an efficient simulation approach is required that is able to process the resulting sensor outputs, at best in real-time.

In this paper we present such a scene simulator that allows a simple composition of test environments based on object and sensor databases. Moreover, real-time processing is enabled by hardware acceleration based on OpenGL.

2 RELATED WORK

Generally, different simulation approaches for the propagation of electromagnetic radiation exist. The most common ones are ray tracing and radiosity.

The idea of ray tracing is to send out rays into the scene and trace their ways in order to determine the visibility or occlusion of objects. Since the propagation of light is reversible, mostly, an inverse approach is applied that starts the rays at camera location. Thus, only rays seen by the camera have to be calculated.

When applying radiosity, all surfaces of the scene are divided into small patches and the pairwise exchange of radiation is calculated between these patches. Thus, the computed result is not position dependent. The advantage of radiosity is that it supports diffuse emission and reflection implicitly. However, the required computing power and the memory usage are high. On the contrary, for raytracers diffuse reflections are quite expensive, but specular reflections and transparency are supported directly.

In general, both approaches cannot be processed in real-time, though real-time ray tracing by massive parallelization is a topic of current research (Schmittler et al. 2002).

Another approach which is used in today's computer graphics is rendering. It is less exact, since it does not support diffuse reflections at all but allows real-time processing due to hardware acceleration. In recent years, the objective of the predominant part of research in the area of simulation of electromagnetic radiation was the analysis of communication channels. Different realization, applying radiosity and calculating the time response at the receiver, were for example presented by Gfeller and Bapst (1979) and ?. The runtime of these approaches depends on the number of reflections k , the number of patches N and equals $O(N^k)$.

A simulator that applies inverse ray tracing was for instance presented by Iske et al. (2004). It allows a scene composition based on simple primitives like spheres, cuboids and also determines the time response of the radiation at the receiver. Furthermore, it offers the evaluation of different receiver characteristics. The runtime of this approach is given by the number of objects o in the scene, the number of point sources q , the number of simulated rays a , the number of rays per reflections n and the number of reflections t itself and equals $O(aoqtn^t)$.

A virtual-reality realization that does not only simulate thermal radiation but also thermal conduction was presented by ?. Again, radiation is simulated by radiosity. The convection, however, is modelled by a simple differential equation. Additionally, the simulator allows the processing of dynamics. The user is enabled to walk through the scene and to switch objects on and off. The approach offers real-time processing by hardware acceleration. For example, on a multi-core test system a scene with 20.000 triangles was calculated with a frame rate of 25 Hz.

An implementation, similar to the one presented in this paper, was described by ?. The authors describe a system simulating infrared array sensors in the automotive field in order to detect pedestrians. They use OpenGL and estimate detector parameters by *Simulated Annealing*. In the contrary, in our realization, measured or idealized sensor characteristics are employed. Furthermore, the authors do not address the aspect of real-time simulation.

3 FUNDAMENTALS OF INFRARED RADIATION & SENSING

In order to be able to specify the requirements on simulation, a short introduction into the fundamentals of thermal radiation and ways of detection are given in this section. In addition, a short overview of challenges which have to be handled in ThILo, is given.

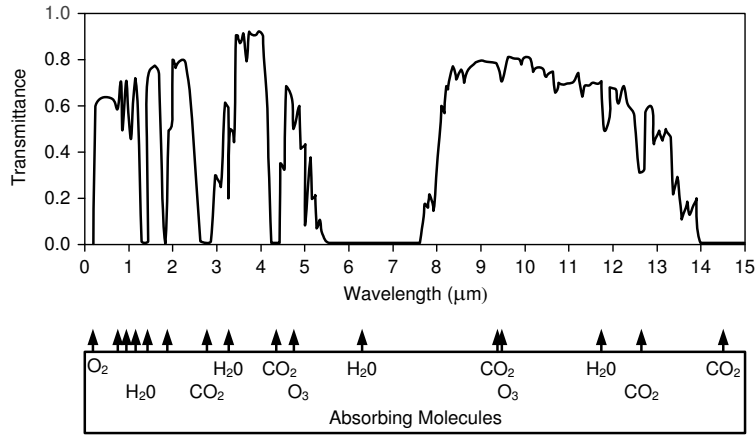


Figure 1: Absorption of thermal radiation in air per seamile

3.1 Radiation Basics

As already stated, every object with a temperature above *absolute zero* emits energy via electromagnetic radiation. The overall flux and the wavelength of this radiation depends on the surface temperature and emissivity ϵ . The latter lies in an interval of $[0..1]$ and describes the ability of a surface to emit radiation with respect to a perfect emitter at the same temperature, a so called blackbody (Gustav Robert Kirchhoff 1859). The emissivity is wavelength dependent, however, in most practical cases, where the relevant wavelength range is strongly limited, it can be assumed as constant (Infrared 2005).

The relation between wavelength λ , temperature T , emissivity ϵ and the radiant exitance M_λ is given by Planck's law:

$$M_\lambda = \frac{\epsilon(\lambda) \cdot C_1}{\lambda^5 (e^{C_2/\lambda T} - 1)} \quad (1)$$

with $C_1 = 3.74 \cdot 10^{-12} \text{W/cm}^2$ and $C_2 = 1.44 \text{cmK}$. The total radiated power within a certain bandwidth B can be calculated by integration, whereas in general the integral of Formula 1 over λ cannot be solved analytically. An approximation for a broad bandwidth is given by the Stefan-Boltzmann law:

$$\Phi_B = A \cdot \epsilon \cdot \sigma \cdot T^4 \quad (2)$$

where $\sigma = 5.67 \cdot 10^{-8} \text{W}/(\text{m}^2\text{K}^4)$, A is the observed surface area and ϵ is assumed to be wavelength independent (Fraden 2004).

The wavelength yielding the highest density of radiated power λ_m for a given T in Kelvin can be calculated using Wien's law:

$$\lambda_m = \frac{2898 \mu\text{m}^\circ\text{K}}{T} \quad (3)$$

with T given in Kelvin. It states, that the higher the surface temperature of the radiating object, the shorter the peak wavelength. Consequently, a human with a skin temperature of 34°C emits most of his radiation near a wavelength of $9.44 \mu\text{m}$. When assuming a relevant temperature range for indoor localization between 0°C and 70°C , most of the energy is radiated in an interval of approximately 8 to $14 \mu\text{m}$. That implies a limitation of the sensitivity sensor to this range. A positive side effect in this context is that air has a so called *atmospheric window* in the interval of 8 to $14 \mu\text{m}$, as illustrated in Fig. 1 (?). Consequently, the attenuation of the relevant thermal radiation by air is very low over short distances ($\leq 30\text{m}$) and thus negligible.

Another interesting aspect with respect to thermal radiation is that emissivity and absorption are equivalent. For an opaque object the following relationship for the reflectivity ρ is valid:

$$\rho = 1 - \epsilon.$$

A blackbody not only emits at maximum but also absorbs all incident radiation. Therefore, the real temperature of a highly reflective surfaces like polished aluminum ($\epsilon = 0.04 - 0.06$) cannot be detected, but only that of the reflected environment. Fortunately, human skin has an emissivity of 0.93 to 0.96 and is thus suitable for thermal detection.

3.2 Lambert's Cosine Law & Flux Transfer Calculation

All non-metallic materials are good diffuse emitters (Fraden 2004). Their radiance does not depend on the point of view. However, the the active area of source and receiver depends on the angle between the normal of the surface and the direction between source and detector and so does the received radiation. This relation is stated by *Lambert's Cosine Law*:

$$I(\alpha) = L \cdot A \cos(\alpha) \quad (4)$$

where I is the radiant intensity in W/sr, L is the radiance in W/(m²sr) and A is the size of the radiating surface.

Based on Lambert's Cosine Law the flux transfer between two objects can be calculated as integration over both, the surface of the receiver and the surface of the source

$$\Phi = \int_{A_S} \int_{A_R} L \cdot \frac{\cos(\gamma_1) \cdot \cos(\gamma_2)}{r^2} dA_R \cdot dA_S, \quad (5)$$

where A_S and A_R specify the size of source and receiver surface, respectively. L equals the radiance and r describes the distance between two infinitesimal patches. Furthermore, γ_1 and γ_2 equal the angles between the connection line of the patches and the respective surface normals.

3.3 Detection of Infrared Radiation

For the detection of thermal radiation different kinds of sensors exist. They are classified into quantum and thermal detectors. Quantum detectors exploit the photoelectric effect. They are highly sensitive and have a short response time. In order to perform well, they have to be cooled to a temperature of 70 K. Additionally, their detectivity strongly depends on the wavelength. Hence, quantum detectors are not appropriate for the intended application. Thermal detectors convert the received infrared radiation into heat and their detectivity is nearly wavelength independent. They do not necessarily need cooling, but their response time is higher and their detectivity is lower. Today, the most common thermal detectors are pyroelectric sensors, applied in motion detectors, microbolometer arrays, which are used in infrared cameras and thermopiles. Since pyroelectric sensors can only detect changes in heat flow, they are not adequate for ThILo, as well as microbolometer arrays, which need temperature stabilization for proper function. Furthermore, they are quite expensive, since they are mainly produced with high resolutions.

In comparison, thermopiles offer the best detection capabilities (Photonics 2004), are inexpensive and available as low resolution line and array sensors. Therefore, they are well suited for appliance in ThILo. Their working principle is based on a serial connection, in order to increase the output voltage, of thermocouples. The resulting junctions are divided into active and passive ones. The active junctions are attached to a thermally isolated membrane that is exposed to radiation (absorber) as shown in Fig. 2. The passive ones are only influenced by the ambient temperature. Hence, the output signal of a thermopile depends on the difference between object and ambient radiation (temperature). With response times between 20 - 50 ms, they are fast enough to detect human motion adequately.

Localization based on thermopile arrays can be realized by determining the angle of arrival (AoA) under which an object is seen. If several sensors are placed at different positions in a room, the position of a human can be computed via triangulation. When using a low resolution line sensor whose pixels exhibit slightly different field of views, the assumed AoA equals (simplified) the orientation of that pixel with the highest outcome, as illustrated in Fig. 3.

3.4 Challenges of Passive Infrared Localization

After describing the radiation and sensing basics, a short overview of the arising challenges of PIL is given in order to derive the required capabilities of the simulator. A more detailed description can be found in (Kemper and Linde 2008).

As already stated, thermal radiation of different objects cannot be discriminated. Hence, thermal radiation, not emanated by humans (**background radiation**), can tamper the measured AoA. Especially, when its radiant intensity is higher. Generally, a sub-division into static (e. g. electronic devices that are already switched on for a while)

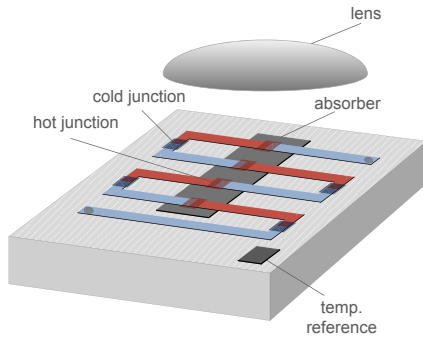


Figure 2: Construction of a thermopile

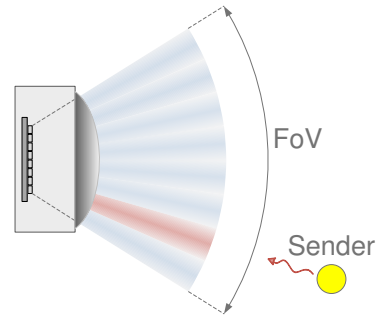


Figure 3: Principle of AoA estimation

and dynamic (e. g. switched or moving devices) background radiation is convenient. The static part only changes slowly or not at all. Thus, filtering can simply be done by its determination and subtraction from the sensor signal. On the contrary, the dynamic part, changes fast and its effects can therefore be misinterpreted as a moving person. That especially applies to **specular reflections** on metallic surfaces, since they "behave" exactly like the reflected source itself. For an accurate localization, it is necessary to determine such dynamic distortions and to develop appropriate recognition and filter algorithms. Consequently, a simulation of dynamic environments is necessary.

Additionally, it should be possible to locate several people per room. This requires the development of elaborate localization algorithms adapted to ThILo. A simulation avoids the need of several test persons and thus makes the development more flexible.

4 BASICS OF OPENGL & REQUIRED SIMPLIFICATIONS

In the following, the basics of OpenGL are described with respect to the developed simulator. Furthermore, it is explained, what kinds of simplifications OpenGL requires.

4.1 OpenGL

The *Open Graphics Library (OpenGL)* is an open and platform independent interface specification that enables direct access to the graphic card and allows the generation of two and three dimensional graphical content in real-time (Rendering). The composition of the rendered objects is based on simple geometrical primitives like triangles or polygons.

The image generation is processed in several stages that transform the geometrical and visual data of objects into a two dimensional image. OpenGL 2.0 introduced vertex and fragment processors which replace the former vertex and fragment operation and enable programmability by users. The code, written for those processors, is called shader. The simulator uses such manipulations to fasten the calculation of sensor values.

Although OpenGL is not limited to a special kind of transformation, for visual presentation on screen, typically a perspective projection is used that equals the image of a pinhole camera which is of relevance for the simulation.

4.2 Simplifications

Due to the limitations of OpenGL and in order to enable high simulation performance, several simplifications are necessary:

- *Wavelength independence of emissivity* - As already stated the emissivity can be assumed as wavelength independent, if the relevant bandwidth is small. Due to air-transmittance and the typically used silicon lenses, in case of ThILo the wavelength range is limited to 7.5 to 13.5 μm . Hence, this assumption is valid.
- *Limitation to diffuse radiation* - According to Fraden (2004), all materials except metals are good diffuse radiators in the relevant wavelength range. Consequently, modelling of non-diffuse radiation can be neglected, except specular reflections on metallic surfaces. However, this issue is not addressed in this paper.
- *Diffuse reflection are negligible* - Due to the diffuse character of reflections and the high attenuation of all materials except metals diffuse reflections can be neglected. This will be proven in the evaluation.

- *Transmission of infrared radiation in objects is negligible* - Solid objects, except special optical materials like gallium arsenide, are opaque for infrared radiation of the relevant wavelength range.
- *Negligence of fuzzy imaging* - Since infrared sensors have a fixed focus, they only produce a sharp image, if the distance to the object is large with respect to the diameter of the lens. As the distance between objects and sensors is typically high with respect to lens size, also this assumption can be made.

5 SCENE SIMULATOR FOR PASSIVE INFRARED LOCALIZATION

A simplified illustration of the simulator with main focus on data exchange is given in Fig. 4. It clarifies that the scene composition is done based on an object and sensor library. Thus, it is possible to compose scenes by inserting objects and sensors from the library. Hence, only parameters like temperatures or certain animations have to be added. In order to allow a simple composition, the models as well as the simulator support a normal visualization and a mode, showing the infrared characteristics of objects which is needed for simulation. The object descriptions use **Collada**, which is a special XML format (Arnaud and Barnes 2006). It enables user specific enhancements that are essential in case of simulating the propagation of infrared radiation.

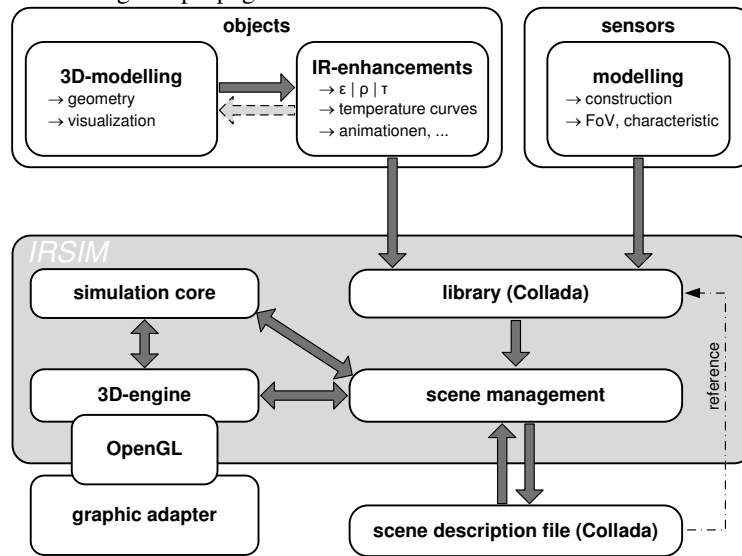


Figure 4: Simplified illustration of simulator concept

The main component of the simulator is the **simulation core**. It realizes the calculation of sensor values using a proprietary 3D-engine that encapsulates all necessary OpenGL functionality.

5.1 Simulation of Infrared Sensors

Following, the mathematics to calculate the sensor outputs based on an infrared image of the scene are presented. Moreover, a projection correction is described which is necessary as the optical characteristics of a sensor do not correspond to that of pinhole camera. Finally, it is shown, how these computations can be processed more efficiently by using shaders.

5.1.1 Mathematical Derivation

The output signal of thermopiles depends on the radiation reaching the sensitive element. Due to Lambert's Cosine Law but also because of the used optics the impact of this radiation depends on the direction of arrival. This direction dependence is described by the sensor characteristic C which exhibits an angle dependent attenuation. Consequently, the output signal presents a weighted average of the radiance L over the complete field of vision Ω :

$$U = K \cdot \left(\int_{\Omega} L(\phi, \theta) \cdot C(\phi, \theta) d\omega - E_a \right) \quad (6)$$

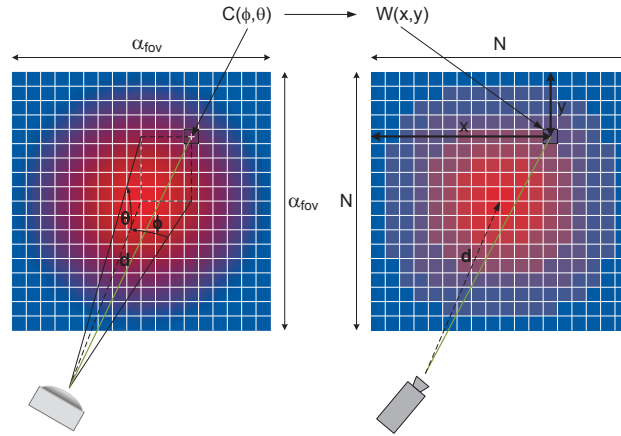


Figure 5: Relation between continuous and discrete sensor characteristic

where ϕ and θ equal azimuth and elevation with respect to the optical axis. K is a sensor specific factor in Vm^2/W giving the dependence between the weighted average of irradiance and the output voltage. Furthermore, E_a equates the radiance that corresponds to the ambient temperature.

However, as the simulator is working on a pixel based picture, an adequate mapping of the characteristic is required, as shown in Fig. 5. This results in a weighting mask $W(x, y) = C[\phi(x), \theta(y)]$, whereas $\phi(x)$ and $\theta(y)$ are given by

$$\phi(x) = \arctan\left[\frac{(x - (N + 1)/2) \cdot l}{l}\right] \quad (7)$$

$$\theta(y) = \arctan\left[\frac{(y - (N + 1)/2) \cdot l}{l}\right]. \quad (8)$$

Here, l equals the side length of a pixel and N the number of pixels in horizontal and vertical direction.

Due to the diffuse radiation characteristics of objects, it makes sense to describe their behaviour based on radiant exitance M as well as the received flux by the irradiance E_r , since both are given in W/m^2 . Formula 6 can be written as

$$U_i \approx K \cdot \frac{\sum_{N \times N} (E_r(x, y) - E_a) \cdot W(x, y) \cdot P(x, y)}{\sum_{N \times N} W_i(x, y) \cdot P(x, y)}, \quad (9)$$

whereas the denominator serves as normalization factor. $P(x, y)$ is the so called projection correction that considers the fact that a sensor cannot be modelled by a pinhole camera directly. The derivation of $P(x, y)$ is described in the next section.

5.1.2 Projection Correction

As stated in Sect. 3.2, the flux transfer between two objects can be calculated based on formula 5. However, in case of a simulated sensor this formula can be reduced to

$$\Phi = A_E \cdot \int_{A_S} L \cdot \frac{\cos^2(\gamma_1)}{r^2} dA_S, \quad (10)$$

since the image plane and the active area of the simulated sensor are in parallel ($\gamma_1 = \gamma_2$). Moreover, the integration over A_R can be neglected as its size is low (point sensor).

Regarding the working principle of a pinhole camera, as shown in figure 6, it becomes obvious that the image size of an object on the focal plane only depends on the distance g between the object and the plane of aperture

$$\frac{G_1}{B_1} = \frac{G_2}{B_2}, \quad (11)$$

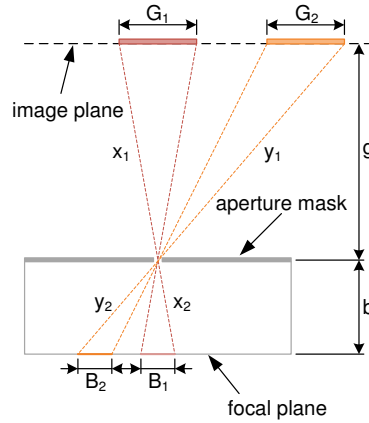


Figure 6: Working principle of a pinhole cam

which can be followed by the intercept theorems. When calculating the sensor output neither the real distance between the sensor and the current patch nor Lambert's cosine law are considered.

Therefore, a projection correction is required which realizes an appropriate weighting of the pixels with

$$P(x,y) = \underbrace{\cos^2(\gamma(x,y))}_{\text{attenuation of distance}} \cdot \underbrace{\cos(\gamma(x,y))}_{\text{Lambert (source)}}. \quad (12)$$

Here, it is assumed, that the distance between the sensor and the image plane is one and Lambert's cosine law for the sensor is already considered by its characteristic. Furthermore, $\gamma(x,y)$ is given by

$$\gamma = \text{atan} \left(l \sqrt{\left(x - \frac{N+1}{2}\right)^2 + \left(y - \frac{N+1}{2}\right)^2} \right), \quad (13)$$

where x and y are the corresponding pixel indices and N is the number of pixel per line and column.

5.1.3 Efficient Calculation of Sensor Values

In order to realize an accurate and efficient way of calculating the sensor outputs based on an infrared image, the following aspects have to be considered:

- **Coding** - In OpenGL, typically the four channel RGBA color system is used to code the rendered pictures, whereas every channel has a resolution of eight bit. To represent the irradiance one channel with a higher resolution would be much more convenient. Therefore, a special format is applied that allows 32 bit floating-point coding for every channel.
- **Efficient rendering for array sensor** - As already explained, the pixels of an array sensor have only slightly different orientations. Hence, it would be very inefficient to render a completely new image for every pixel. Thus, the field of vision of the camera is chosen in such a way that the field of view of every sensor pixel is included in the resulting image. The selection of the relevant part for the particular pixel is in turn realized by its characteristic implicitly.
- **Efficient use of shaders** - To calculate the sensor values based on the rendered image, the projection correction matrix which includes the attenuation for every pixel, as well as the weighting mask have to be stored as textures, before the simulation is started. Based on these textures a shader has to calculate the following equation:

$$E(x,y) = (E_r(x,y) - E_a) \cdot W(x,y) \cdot P(x,y) \quad (14)$$

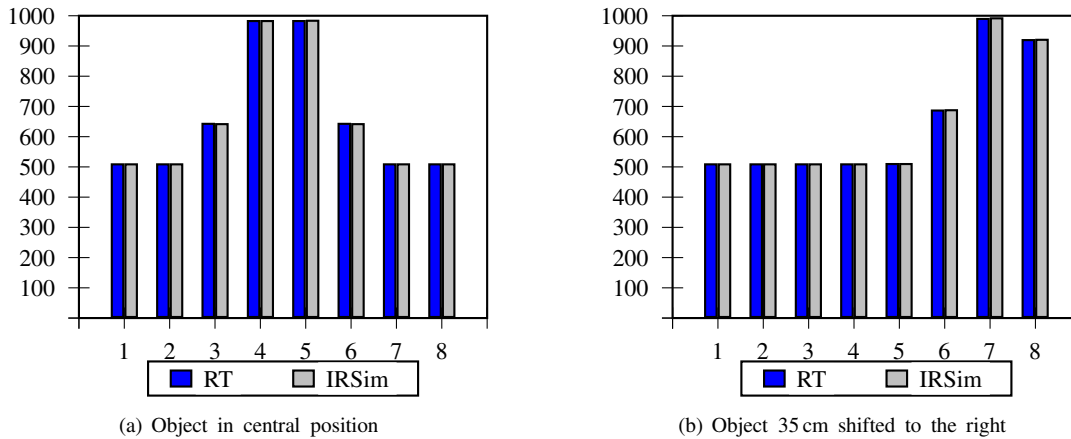


Figure 7: Comparison between raytracer and simulator

The results are again stored in a texture, so that afterwards the required averaging can be processed in parallel, whereas every shader sums four pixels. Also in this case a texture serves as result matrix, since the averaging has to be repeated iteratively. Finally, the overall sum is divided by the sum of all pixel weights which yields the average irradiance.

- **Parallel pixel processing** - As the representation of an infrared image requires only one 32 bit channel, the four 32 bit channels can be used for parallel processing of sensor pixels, when simulating array sensors.

6 EVALUATION

In this section, the presented simulator is evaluated with respect to performance and accuracy. Hence, some simulation results of the simulator are compared to that of a raytracer developed by [Iske et al. \(2004\)](#). That includes the overall accuracy of the approach as well as the repercussion due to negligence of diffuse reflections. Furthermore, a comparison with the results of a real sensor – we use a Perkin Elmer TPL08A 3.9 – is presented. In all simulations a line sensor with eight pixels is emulated. The conversion factor of the sensor is set to $0.006 \text{ Vm}^2/\text{W}$ and the wavelength range is limited to $7.5 \mu\text{m}..13.5 \mu\text{m}$.

Regarding the presented results, it should be noted that the sensor output values are not given as voltage but as integers corresponding to a ten bit AD conversion output, whereas 512 equals 0 V.

6.1 Evaluation of Projection Correction

In order to show the general correctness of the sensor emulation by a pinhole camera, a simple scene composed of a heating plate (75°C) of $30 \times 30 \text{ cm}^2$ is used. This plate is first placed of one meter in front of the sensor and then shifted 35 cm to the right.

Figure 7 illustrates that the the simulator and the raytracer results are almost the same, whereas the comparison is done for every pixel. Further configurations with different positions of the plate exhibit similar results. Regarding all simulations, the maximum difference that occurred between the results of the simulator and the raytracer was less than 0.1 % of the measurement value.

6.2 Comparison to a Real Sensor

To test the ability of the simulator to emulate a real sensor, the characteristic of a *TPL08A 3.9* was approximated three-dimensionally in the simulator, as shown for the horizontal direction in Fig. 8. It was assumed that the pixel characteristic in vertical direction equals the horizontal one.

Moreover, the testbed, shown in Fig. 9, was used for the evaluation. The heating plate that consists of a coated copper plate ($31 \times 31 \text{ cm}^2$) held by a metallic frame, had a temperature of 70°C and an emissivity of 0.95. With this testbed, several measurements where processed by shifting the plate in front of the sensor in steps of 15 cm at 1.5 m distance from the left to the right.

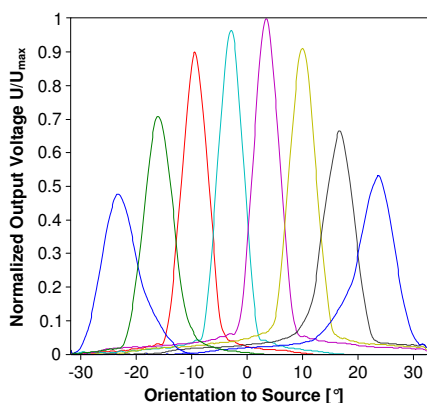


Figure 8: Characteristic of a sensor in horizontal direction

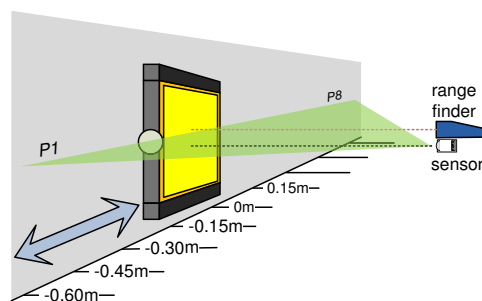


Figure 9: Testbed for comparison with real sensor

Refl.	P1	P2	P3	P4	P5	P6	angle
no	524	572	514	512	512	512	-18,8°
yes	524	572	514	513	514	514	-18,8°

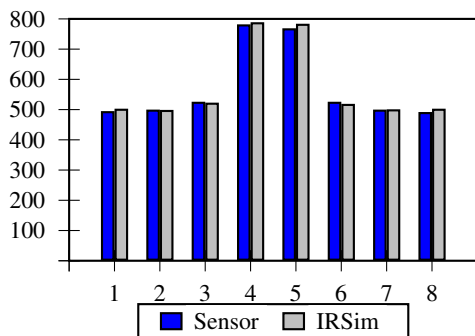
Table 1: Results for the first arrangement

Refl.	P1	P2	P3	P4	P5	P6	angle
no	513	576	565	512	512	512	-14,7°
yes	513	576	573	522	514	513	-14,4°

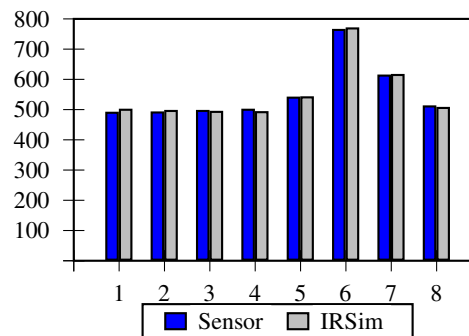
Table 2: Results for the second arrangement

Figure 10 exemplary shows the results for the heating plate in central positions and 30 cm shifted to the right. It is obvious, that the simulated sensor is not exact. The reasons for differences between simulated and real sensor are explained by influences of the used sensor characteristics approximation, component tolerance and measurement noise.

Nevertheless, the typical behaviour of the sensor is reproduced well by the simulation, which is also confirmed by the other measurements.



(a) Plate in central position



(b) Plate 30 cm shifted to the right

Figure 10: Comparison between raytracer and simulator

6.3 Repercussion of Diffuse Reflections Negligence

Another evaluation was processed in order to determine the impact of diffuse reflections on the sensor output. Two different situations were simulated with the raytracer, as shown in Fig. 11. In the first arrangement the reflection is located near the real object and in the second one not, so that the object and its reflection are seen by none adjacent pixels.

The simulation results of the relevant pixels are shown in Tab. 1 and 2 as well as the corresponding angles calculated based on these values.

It can be seen that the difference between considering and not considering the diffuse reflection are small regarding the resulting angle. That approves the assumption, that diffuse reflection can be neglected, especially, since the chosen reflectivity of 0.3 was quite high, as walls typically have an emissivity greater than 0.8.



Figure 11: Used arrangement for reflection tests

6.4 Performance tests

In order to test the performance of the simulator a typical scene with a human and several pieces of furniture was processed for a line sensor. To this end a 3 GHz dual core computer with an NVIDIA Geforce 9800GTX+ was used. The resulting processing times are presented in Tab. 3. It can be seen that for a typical resolution of 512×512

Resolution	Rendering	$\Sigma_{N \times N}$ PC	$\Sigma_{N \times N}$ 9800GTX+
128×128	9,5 ms	5.5 ms	2,8 ms
256×256	9,5 ms	24.4 ms	3,7 ms
512×512	10 ms	97,9 ms	4,8 ms

Table 3: Processing times for a lines sensor

one sensor can be processed in 15 ms. This includes rendering and averaging, whereas the latter took twenty times longer on the PC. Hence, as assumed, the accumulation of pixels values by the graphic card is much more efficient. If we do not only simulated one sensor, but eight, as typically needed (Kemper, Linde, and Walter 2008), 80 ms for rendering and 40 ms for averaging are required. That results in a frame rate of approximately 8 Hz. Consequently, the proposed real-time nearly requirement is fulfilled, if we assume an sensor update rate of 10Hz for the localization.

7 CONCLUSION

In this paper we presented a real-time simulator for infrared array sensors that allows the simulation of dynamic environments. To realize this real-time processing, a hardware accelerated approach based on OpenGL is used. Evaluations have shown that the disadvantages of OpenGL in contrast to raytracing and radiosity can be neglected, since the achieved accuracy is sufficient in the context of indoor localization.

Future work will be concentrated on setting up a comprehensive database by investigating typical indoor furniture and devices with respect to their infrared radiation behaviour. Therefore, efficient ways to determine and describe such behaviour and to map it to geometrical models have to be found. Finally, this simulator shall be used to ease the development of filter and localization algorithms for passive infrared localization.

REFERENCES

- Arnaud, R., and M. C. Barnes. 2006. *Collada - Sailing the Gulf of 3D Digital Content Creation*. A K Peters, Ltd.
- Fraden, J. 2004. *Handbook of Modern Sensors - Physics, Designs and Applications*. 3. Edition ed. Springer-Verlag.
- Gfeller, F., and U. Bapst. 1979, Nov.. Wireless in-house data communication via diffuse infrared radiation. *Proceedings of the IEEE* 67 (11): 1474–1486.
- Gustav Robert Kirchhoff 1859. über den zusammenhang zwischen emission und absorption von licht und wärme. *Monatsberichte der Königlich Preuß. Akademie der Wissenschaften*.
- Hightower, J., and G. Borriello. 2001, August. Location Systems for Ubiquitous Computing. *Computer* 34 (8): 57–66.
- Infrared, E. 2005. Physics of electro-optic detectors.
- Iske, B., B. Jäger, and U. Rückert. 2004, April. A Ray-Tracing Approach for Simulating Recognition Abilities of Active Infrared Sensors Arrays. In *IEEE Sensors Journal*, Volume 4, 237–247.

- Kemper, J., and H. Linde. 2008, March. Challenges of Passive Infrared Indoor Localization. In *5th Workshop on Positioning, Navigation and Communication (WPNC)*.
- Kemper, J., H. Linde, and M. Walter. 2008, August. Human-Assisted Calibration of an Angulation based Indoor Location System. In *2nd International Conference on Sensor Technologies and Applications (SENSORCOMM)*.
- Photonics, H. 2004. Technical informasd-12 – characteristics and use of infrared detectors.
- Schmittler, J., I. Wald, and P. Slusallek. 2002. SaarCOR - A Hardware Architecture for Ray Tracing. In *Graphics Hardware*, ed. T. Ertl, W. Heidrich, and M. Doggett, 1–11. Computer Graphics Group, Saarland University, Germany.
- Tauber, J. A. 2002. Indoor Location Systems for Pervasive Computing. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA.

AUTHOR BIOGRAPHIES

DANIEL HAUSCHILDT received the diploma in computer science from TU Dortmund University (TUD) in 2009. He is currently working towards his PHD in electrical engineering at the Robotics Research Institute (RRI) of the TUD. His research interests include stochastic filtering and multi-target tracking. His email is daniel.hauschildt@udo.edu

BENEDICT JURETKO received his diploma in electrical engineering from the TUD in 2008. His email is benedict.juretko@udo.edu

JÜRGEN KEMPER received the diploma in information technology from the TUD in 2004. From 2004 to 2009 he worked as a research assistant at the Robotics Research Institute. Since 2009 Jürgen Kemper is employed as software developer for ABB. His research interests include indoor localization and home automation. His email is juergen.kemper@udo.edu

NICOLAJ KIRCHHOF received the diploma in electrical engineering from the TUD in 2009. He is currently working towards his PHD in electrical engineering with the RRI. His research interests include blind source separation and indoor localization. His email is nicolaj.kirchhof@udo.edu

HOLGER LINDE received his Ph.D. in electrical engineering from the TUD in 2007. Since 2008 Holger Linde is employed as software developer for ABB. His research interests include indoor localization and home automation. His email is holger.linde@udo.edu