Open access • Book Chapter • DOI:10.1007/978-3-319-14112-1_20

A Benchmarking Model for Sensors in Smart Environments — Source link

Andreas Braun, Reiner Wichert, Arjan Kuijper, Arjan Kuijper ...+2 more authors

Institutions: Fraunhofer Society, Technische Universität Darmstadt

Published on: 11 Nov 2014 - Ambient Intelligence

Topics: Smart environment and Use case

A Benchmarking Model for Sensors in Smart Environments

Andreas Braun[1], Reiner Wichert[1], Arjan Kuijper[12], Dieter W. Fellner[12]

[1] Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany
{firstname.lastname@igd.fraunhofer.de}
[2]TU Darmstadt, Darmstadt, Germany

Abstract. In smart environments, developers can choose from a large variety of sensors supporting their use case that have specific advantages or disadvantages. In this work we present a benchmarking model that allows estimating the utility of a sensor technology for a use case by calculating a single score, based on a weighting factor for applications and a set of sensor features. This set takes into account the complexity of smart environment systems that are comprised of multiple subsystems and applied in non-static environments. We show how the model can be used to find a suitable sensor for a use case and the inverse option to find suitable use cases for a given set of sensors. Additionally, extensions are presented that normalize differently rated systems and compensate for central tendency bias. The model is verified by estimating technology popularity using a frequency analysis of associated search terms in two scientific databases.

Keywords: Benchmark, smart environments, modeling, sensor technology

1 Introduction

When designing a new application or system for a specific purpose, the parties involved have to make a number of decisions regarding the different components, processes and methods that are to be used. Benchmarking is a method mostly used in business practice to compare the performance of processes, products and market entities against one another. A single or a set of different indicators are used to act as metric or calculate an overarching metric of performance that can be compared to other entities [1]. This tool is widely used for supporting decisions in different domains. Looking at smart environments, a common challenge is to select a specific sensor technology for any given application. While the majority of systems are following a structured approach in the design process, e.g. by ranking available systems or performing an iterative trial & error routine, so far there has been no generic model that would allow to evaluate the expected performance of a system based on a specific sensor technology. This is particular in complex domains, such as Ambient Assisted Living (AAL) that involve constant interaction of actors within the environment, third parties that exchange information with the smart environment, and a variety of different sensor systems measuring certain aspects of the environment. In this work we introduce a formal benchmarking model that allows estimating the performance of applications in smart environments based on a specific sensor technology. Using a set of common features and an adaptive weighting model we are able to cover a high number of different applications in a specific domain and thus support the decision process at an early stage of the application design. We are presenting related works ranging from technology benchmarking to selection of specific metrics before discussing features that are relevant for smart environments. After introducing the model formalism we evaluate the method by performing a popularity analysis within scientific works. For this we use a set of search terms on two scientific databases. Furthermore, we discuss several normalization and compensation techniques that cope with specific effects we observed in the benchmarking process.

2 Related Works

Benchmarking is a tool that is widely used in computing technology [2]. Hardware benchmarks compare the performance of different single systems, often seen for GPUs or CPUs to evaluate both theoretical and real-life performance. Some metrics that are used for theoretical comparison in CPUs are FLOPS (floating point operations per second), e.g. measured by Linpack [3], or MIPS (million instructions per second), e.g. measured by Dhrystone [4]. Regarding GPUs the benchmarks include Texel rate (how many triangles can be processed per second) and Pixel rate (how many pixels are processed per second. Real-life benchmarks for CPUs typically included timing specific tasks on applications that are demanding for certain aspects of the CPU, such as video processing, image processing or audio encoding. For GPUs many PC games provide benchmarking tools that allow evaluating the real-life performance of different graphics cards at different settings, e.g. resolution or detail level. The typical metric here are FPS (frames per second) that denote how often the screen content can be rendered in a second.

System benchmarks are a step up from single component benchmarks and combine the performance measurements of various components in different scenarios to evaluate the estimated behavior in numerous real-life situations. There are several standardized test suites that provide this functionality, such as SPEC [5]. A common single index that is available for all newer Windows machines (Vista and beyond) is the Windows System Assessment tool that calculates the WEI (Windows Experience Index), a combined score of CPU performance, 2D and 3D graphics performance, memory performance and disk performance. For determining the lowest score of all single metrics is chosen to determine a lower bound for expected real-life performance.

If different systems of the same category are compared, technology reviewers often use a single index that is calculated based on various aspects of the system. Smith introduced different potential combined metrics that can be used for this purpose [7]. Three different approaches are mentioned, geometric mean, arithmetic mean and harmonic mean. Additionally varieties with a specific weighting are mentioned.

There has been considerable work in the domain of identifying suitable metrics for a given benchmark. Crolotte argued that the only valid benchmark for decision support systems is the arithmetic mean of different single benchmark streams, as it is valid for normalized and time-relevant benchmarks [8]. Jain and Raj dedicate several chapters of their book to introduce methods and considerations for metric selection in benchmarking computer systems [9].

In smart environments a number of different benchmarks have been proposed that cover aspects similar to our approach. Ranganathan et al. introduced benchmarking methods and a set for pervasive computing systems [10]. They distinguish system metrics, configurability and programmability metrics and human usability metrics. Another example for benchmarking whole systems is the EvAAL competition that aims at evaluating different technologies that are applicable in Ambient Assisted Living [11]. There are various tracks, including indoor localization and activity recognition. Apart from technical metrics, such as precision, a focus of this competition is on a more holistic approach and thus includes metrics like installation time, user acceptance and interoperability of the solution. Santos et al. presented a model to evaluate human-computer interaction in ubiquitous computing applications, based on trustability, resource-limitedness, usability and ubiquity [12]. In order to assess how well ubiquitous computing applications cover privacy aspects, Jafari et al. propose a set of five abstracted metrics that are applied to whole systems [13]. While these are all benchmarking models within smart environments, they are either aiming at evaluating whole systems or singular aspects not directly related towards sensor technology.

3 Sensor Features

One of the most challenging aspects of benchmarking is selecting the appropriate metrics to be included. In order to identify relevant sensor features for technologies to be applied in smart environments we take inspiration from sensor technology overviews [14] and the pervasive model presented by Ranganathan et al. [10]. Accordingly, we can identify three different groups of sensor features: sensor performance characteristics, pervasive metrics and environmental characteristics. These different groups are detailed in the following sections. We are giving an overview of different potential members of the groups, discuss their relevancy for the benchmarking model and create a feature matrix, as a basis for the feature scoring model.

3.1 Sensor Performance Characteristics

This group of sensor features is related to specific technical properties of the given sensing device, as they would be typically put into the datasheet. A first important characteristic is the sensitivity or resolution of a sensor, which is the smallest change of a measured quantity that is still detectable. For example an accelerometer might be able to only detect changes that are above 0.1g. Another important characteristic is the update rate of a sensor. This denotes the number of samples the sensor is able to measure in a certain timeframe. Typically, the number of samples in a second is noted as frequency, thus a sensor may have an update rate of 20 Hz, generating 20 samples in a second. Another factor that is particularly important for embedded systems or wearables is the power consumption of the sensor that may limit the time it can operate on battery, independent of a power source. A last example is the detection range, denoting the maximum distance between the measured object and the sensing device. This can be a significant distance for cameras (e.g. satellite images), whereas we are primarily looking at smaller smart environments, where it is rare that distances of 20 meters are exceed. Other technologies such as capacitive proximity sensors may not work at this distance [15].

3.2 Pervasive Metrics

Pervasive metrics can be identified as features that specify how well a given sensor system will perform in collaboration with smart environments, when networked with other devices and when placed into existing surroundings. An example for the latter is the obtrusiveness of a sensor device. If it is clearly visible when applied, if there are disturbing signals generated, or if certain privacy concerns are associated to the sensor device, the acceptance by the user and thus the applicability is reduced. If the sensor is operating in a larger network of other devices, the bandwidth required to submit signal to an analyzing node should be kept low. Equally, if the processing capabilities are limited, less complex data processing is preferable. The overall system cost is increasing if single sensors are particularly expensive, thus limiting the potential applications. The system and attached sensors should be robust, both in terms of physical design and quality of service. Finally, the sensors are more readily applicable if the systems are interoperable to each other.

3.3 Environmental Characteristics

The third group is the environmental characteristics of a sensor system. Any sensor is affected by a certain disturbance caused by factors in the environment that are similar to the measured quantity, also called noise. For example an optical sensor is influenced by ambient light sources. In this context it is relevant how frequent those influences are in a certain environment and how robust the sensor is against noise. In many cases the presence of noise can be detected and counteracted with a calibration towards the changed environmental factors. The complexity of this calibration is another interesting factor in this regard. Finally, all sensors have some unique limitations, e.g. specific materials that absorb certain wavelengths of the electromagnetic field are difficult to detect for sensors that work in this specific frequency range.

3.4 Discussion of Feature Selection

We want to select the three most relevant features of each category. This allows a more manageable overall model, however, requires a selection of the presented features. In this work the selection is based on the authors' analysis of the related works. In future it is advisable to use more sophisticated methods, such as surveying AmI experts and calculating inter-rater reliability [16]. Of the sensor performance characteristics group we will select resolution, update rate and detection range. Resolution is a critical feature in any application, determining precise any detection is and if particular objects may be detected at all. Update rate is equally important if fast objects are to be detected and if we want to have reactive systems that respond in real-time. The importance of detection range correlates with the size of the environment and may lead to a reduction of required sensors. Of the mentioned features we omit power consumption. The actual power consumption of a whole system is a more interesting metric but very difficult to predict from the energy usage of a single sensor [17]. Of the pervasive metrics group we select unobtrusiveness, processing complexity and robustness. Unobtrusiveness of the sensor device is a desired feature in many different scenarios, where it should not impede the environment.

Table 1. Feature matrix denoting capabilities required for a certain rating

Feature	--	-	*o*	+	++
Resolution (res)	very coarse	coarse	normal	fine	very fine
Update Rate (upd)	< once per second	slower real-time	real-time	faster real-time	> 100 times per second
Detection Range (det)	touch	less than one meter	less than 5 meters	less than 20 meters	more than 20 meters
Unobtrusiveness (unob)	open large system	open small system	hidden, large exposure	hidden, noticeable exposure	invisible
Processing Complexity (proc)	single sensor CPU	10+ sensors CPU	single sensor embedded chip	10+ sensors by single chip	no further processing
Robustness (robu)	single point of failure	error detection	quality of service	self-recovery	fully redundant
Disturbance Frequency (disfr)	very frequent	frequent	average	unlikely	highly unlikely
Calibration Complexity (calco)	very hard	hard	normal	easy	very easy
Unique Limitations (uniql)	very critical	critical	average	not critical	none

While microprocessors are becoming ever faster processing complexity is still crucial if the number of sensors is increasing. A dedicated chip will require a more complex architecture and lead to more cost, higher energy usage and more potential points of failure, leading to the final chosen feature of robustness, both against physical abuse, but also in terms of system design, where it should be resilient towards failure of single components. We omitted the required bandwidth, as this metric is not important for many sensors, as they have low bandwidth requirements in general, but also the available bandwidth in wired and wireless systems is increasing continuously. In the last group of environmental characteristics we choose frequency of the disturbing factor, calibration complexity and unique limitations. If the disturbing factor occurs only rarely it is not critical and therefore should be part of the benchmark. Calibration complexity combines both the processing complexity and time that is required to recalibrate the system. This is highly important in real-time systems that have to monitor the environment continuously. Finally, unique limitations are a rather broad metric that is difficult to quantify. However, in many scenarios it is obvious that a specific limitation might arise, e.g. if the smart environment is in an area with a lot of human noise, microphones could be regularly disturbed. Including this metric allows modeling those applications into the benchmark with a strong weight penalizing unsuited sensors.

3.5 Feature Matrix

From the selected metrics we want to create a feature matrix that allows us to associate specific capabilities to a specific rating that is used later in the scoring process of the benchmark model. Each feature is mapped to five different ratings on an ordinal rating scale comprised of the items “least favorable” (--), “not favorable” (-), “average” (o), “favorable” (+) and “very favorable” (++). This leads to the feature matrix shown in Table 1, which will be discussed briefly.

- ***Resolution*** is ranging from “very coarse” to “very fine”. We are using this unspecific rating, as the range may vary strongly between different sensor types. A mapping to actual should depend on the application and object that has to be detected. If the object is large a sensor that would be ranked “coarse” for smaller objects can be ranked as “fine”.
- ***Update Rate*** is rated around real-time performance that is often rated at around 20 samples per second. Slower sensors might miss various events, while faster sensors allow detecting highly dynamic events. It should be noted that for certain sensor categories that measure fast events can require considerably faster update rates.
- ***Detection Range*** is rated around the 5m distance mark, that is typically enough to cover the entirety of a single apartment room. For larger rooms sensors with a higher detection distance are favorable, many sensors only react to touch.
- ***Unobtrusiveness*** is ranging from exposed systems placed in the environment (one example would be the Microsoft Kinect) to invisible systems that integrate seamlessly into the environment.

- ***Processing Complexity*** has a range from dedicated CPUs that are required to process the data of a single sensor to smart sensors that require no further processing, which allows to apply numerous sensors without adding additional processing capabilities to the environment.
- ***Robustness*** is following criteria for quality of service. The least favorable system fails, if only a single node is present and failing. The preferred system is fully redundant.
- ***Disturbance Frequency*** is ranging from frequently occurring disturbing signals, to highly unlikely disturbing signals, resulting in a better rating.
- ***Calibration Complexity*** is a combined metric including the calibration time, the required processing capabilities and if external aid is required in the calibration process, leading to a rating from “very hard” to “very easy”.
- ***Unique Limitations*** should be ranked according to their criticality, as previously explained they may be suitable to penalize certain sensors or emphasize the prevalence of a disturbing factor in a noisy environment.

Now that the feature matrix is complete, the next step is presenting the formalized benchmarking model and how we can use the presented features and their rating to calculate a benchmark score that allows us to compare different sensor categories with regard to different applications.

4 Benchmarking Model

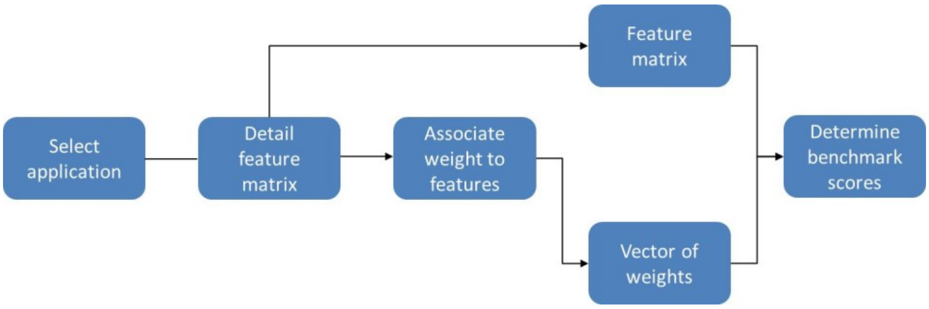

Figure 1. Benchmarking process

In this section we will describe a formal model that will allow us to determine a benchmark score for a given application and a given sensor technology. As previously explained the different applications are distinguished by applying a different set of weights to the known features. We will begin by discussing the process of this feature weighting and giving some examples about proper application. Afterwards, we will introduce a formal model that deduces a single score benchmark for any sensor technology and any application. The overall process is shown in Figure 1 and will be detailed in the following sections, including an example.

4.1 Feature Score and Weighting

The presented feature matrix has some ratings that need detailing in order to be quantifiable in the specific application. The ordinal measurements of the feature matrix should be assigned a quantifiable measure. Taking “Unobtrusiveness” the open system can be detailed as “visible by users” and “large system” as size larger than 100 x 100 x 100 mm. Similar levels of detail can be applied to the other features, leading to the application-specific detailed feature matrix that is used in the scoring process. The different ratings are assigned different numeric values, namely 0.00 (--), 0.25 (-), 0.5 (o), 0.75 (+) and 1.00 (++). The weight of the features for the specific application is also rated on a 5-point ordinal scale, denoted as “not important” (numeric value 0.0), “less important” (0.25), “moderately important” (0.5), “important” (0.75) and “very important” (1.00). Thus for each application we have a distinct detailed feature matrix and a vector of associated weights that can be applied to a set of sensor technologies in order to calculate the benchmark score. We will now introduce the formal modeling that allows us to determine the calculation for this scoring process.

4.2 Modeling

The model is supposed to formalize a benchmark for any application and any sensor technology in any domain. We will start with the following definitions:

- Set of n domains $D = \{d_1, \dots , \ d_i , \dots \ d_n\}$
- Set of m applications $A = \{a_1, \dots , \ a_j , \dots \ a_m\}$
- Set of o features $F = \{f_1, \dots , \ f_k , \dots \ f_o\}$
- Set of p sensor technologies $S = \{s_1, \dots , \ s_l , \dots \ s_p\}$

In any domain d_i we have a set of potential applications $A_{d_i} \subseteq A$ and a set of relevant features $F_{d_i} \subseteq F$. For each feature f_{k,d_i} there is the associated feature score $r_{F_{d_i}}$ as explained in the previous section. Each sensor technology s_l has a specific feature score $r_{s_l,F_{d_i}} \in [0,1]$. The combined feature scores result in the following vector.

$$\overrightarrow{r_{s_l,F_{d_i}}} = \begin{pmatrix} r_{s_l,f_{1,d_i}} \\ \vdots \\ r_{s_l,f_{o,d_i}} \end{pmatrix} \tag{1}$$

The weights $w_{f_o} \in [0,1]$ associated to a specific application a_j in a domain d_i have the same cardinality $|w|$ as the vector of feature scores $\left|\overrightarrow{r_{s_l,F_{d_i}}}\right|$.

$$\overrightarrow{w_{a_j}} = \begin{pmatrix} w_{f_1,a_j} \\ \vdots \\ w_{f_o,a_j} \end{pmatrix} \tag{2}$$

The feature scores and associated weights allow us to determine a benchmark score b_{s_l} for a specific sensor technology s_l for any application a_j by using the scalar product of feature score and respective weight and apply normalization with regard to the weight.

$$b_{s_l} = \frac{\overrightarrow{r_{s_l,F_{d_\imath}}} \cdot \overrightarrow{w_{a_j}}}{\sum_{k=1}^{o} w_{f_{\mathrm{k}},a_j}} \tag{3}$$

We can now compare different sensor technologies by calculating and comparing the different benchmark scores for a given set of sensor technologies $S_p \subseteq S$ and receive a set B_{S_p} with $t = |S_p|$.

$$B_{S_p} = \{b_{s_l,1}, \dots, b_{s_l,t}\} \tag{4}$$

Thus in order to determine the optimal (chosen) sensor technology b_c for an application a_j and given the prerequisites regarding non-negativity of weights and feature scores, we can evaluate the set for the maximum element.

$$b_c = \max(B_{S_p}) \tag{5}$$

4.3 Feature Score Normalization

With regards to actual benchmarking the problem of bias towards a specific technology may occur. If the average features ratings are different between two technologies the calculated benchmark score will increase. In many instances this might be beneficial, yet if comparing numerous technologies to a set of different applications a trend might be more important than absolute scores. Thus, we provide an optional step of calculating the normalized feature vector $r_{s_l,F_{d_i},norm}$with regard to the average associated value of 0.5, using the following equation.

$$\overrightarrow{r_{s_l,F_{d_\imath},norm}} = \begin{pmatrix} r_{s_l,f_{1,d_i}} \\ \vdots \\ r_{s_l,f_{o,d_i}} \end{pmatrix} \cdot \frac{\mathrm{o} \cdot 0.5}{\sum_{p=1}^{o} r_{s_l,f_{\mathrm{p},d_i}}} \tag{6}$$

The feature-normalized benchmark score is accordingly determined with the following equation.

$$b_{s_l,norm} = \frac{\overrightarrow{r_{s_l,F_{d_\imath},norm}} \cdot \overrightarrow{w_{a_j}}}{\sum_{k=1}^{o} w_{f_{\mathrm{k}},a_j}} \tag{7}$$

4.4 Benchmark Scoring

Now with the formal model and the available set of feature matrix and weights we are able to calculate the benchmarking score for a set of sensor technologies. As an example we are choosing the application indoor localization in a public shopping area to monitor customer behavior. As a first step the feature matrix has to be detailed according to the specific requirements of the application. These include a tracking accuracy of 50 cm, with a large area to cover and potentially fast moving persons. Thus the importance ratings for performance characteristics are moderately important for resolution, important for update rate and very important for detection range. The system can also be used for security purposes, thus unobtrusiveness is less important. There can be dedicated servers, so processing complexity is not important, but the system should be difficult to disturb, thus robustness is important. Disturbance frequency is moderately important, as a large number of persons is monitored, leading to statistically significant results, even if single measurements are disturbed. The environment is fairly static, thus calibration complexity is less important. It is possible that a crowded shop produces a lot of acoustic noise, therefore no unique limitations towards acoustic disturbances should be present and this is moderately important. The resulting vector of weights is:

$$\overrightarrow{w_a} = (0.50\ 0.75\ 1.00\ 0.25\ 0.00\ 0.75\ 0.50\ 0.25\ 0.50)^{\mathrm{T}} \tag{8}$$

This vector of weights is static for the benchmarking of this specific application. As a next step it is necessary to determine the feature scores for a sensor technology. In this case, we assume a selection based on previous experiences and best practice for this application and choose a system based on numerous stationary cameras. The system has high resolution cameras, with an update rate of 30 samples per second and a high detection range of more than 20 meters. The cameras are external, not hidden from view but attached on the ceiling. The processing complexity is very high, requiring a dedicated CPU per camera. Since they are out of reach they are robust towards human intervention and independent from each other. In the given setting visual disturbance is unlikely, calibration is difficult but not required regularly and the system is not disturbed by acoustic noise. This results in the following rating vector:

$$\overrightarrow{r_{s,f}} = (1.00\ 0.75\ 1.00\ 0.25\ 0.00\ 0.50\ 0.75\ 0.25\ 1.00)^{\mathrm{T}} \tag{9}$$

Using those two vectors we can calculate the final scoring for this sensor system using the equations of the previous section, leading to $b_s \approx 0.76$ and a feature-normalized score of $b_{s,norm} \approx 0.63$. Determining the feature rating vector for other technologies is possible in a similar fashion, whereas the optimal technology has the highest score b_s or $b_{s,norm}$.

5 Evaluation

In order to evaluate the method we propose a discussion based on previous successful works in the domain of smart environments. We will select three different application areas and for each benchmark three different sensor technologies. In order to estimate how popular a certain technology is for a given application we will be using the ACM Digital Library[1] (from now on referred to as ACM DL) to query scientific publications with respective author keywords. This method is limited, as the

[1] http://dl.acm.org

chosen keywords may not catch all relevant publications. Therefor we will slightly increase the focus by using multiple associated search terms for each application and technology. Additionally we will also perform respective searches using the Google Scholar[2] (referred to as GS) database that has a much broader scope-The advantages of the latter are the huge collection of scientific resources and no strong selection bias. However, there are various associated issues that may affect the method. The search results vary on the search term, additionally there will be results that mention the search term but do not necessarily rely on the technology for their respective system. Therefore, the results should be considered as an indicator for popularity in the research community. Similar to the ACM DL search we are also looking for synonyms and calculate an average between the search results.

As applications we choose hand gesture interaction, a marker-based identification system and obstacle avoidance for an autonomous system. The technologies are camera systems, radio-based systems, depth or stereo cameras and ultrasound devices.

5.1 Scoring

Table 2. The importance weighting of different applications, based on the features.

	res	*upd*	*det*	*unob*	*proc*	*robu*	*disfr*	*calco*	*uniql*
Hand Gesture	++	++	-	+	o	o	+	+	-
Identification	--	-	++	++	o	++	+	-	+
Obstacle Avoidance	-	+	-	o	+	+	++	++	+

At first we determine the weights of the different applications with regards to the features. The results are shown in Table 2. For the tables in this section we are using short notation of the features in order of appearance in Section 3.4.

Table 3. Feature rating of the different sensor technologies

	res	*upd*	*det*	*unob*	*proc*	*robu*	*disfr*	*calco*	*uniql*
Camera	++	o	+	-	o	o	o	-	o
Radio	-	+	++	+	o	o	o	o	-
Depth camera	+	o	o	-	-	o	-	o	o
Ultrasound	-	+	o	o	+	o	+	o	o

The rating of the different technologies and the resulting score is shown in Table 3. Here it is possible to follow different strategies regarding the rating. In terms of unbiased comparison looking at the equations it would be necessary that all technologies have the same average feature rating. The second strategy is to apply an absolute ranking to all technologies, independent of the given application. This might lead to certain technologies being unsuited for a given task, or technologies that have the best benchmark score regardless of application. In this specific case the average rating

[2] http://scholar.google.com

according to equation (iii) is 0.53 for cameras, 0.58 for radio, 0.44 for depth cameras and 0.56 for ultrasound devices. The importance weights and feature ratings are translated to numerical values, as shown in equations (ix) and (x). Table 4 displays the different calculated benchmark scores for the combinations between applications and technologies. As we are comparing numerous technologies and applications the feature-normalized benchmark score (equation (viii)) is also included.

Table 4. Regular and normalized benchmark score matrix of different applications and technologies

		Camera	*Radio*	*Depth Camera*	*Ultrasound*
Hand Gesture	b_{s_l}	0.53	0.57	0.46	0.55
	$b_{s_l,norm}$	0.50	0.48	0.51	0.50
Identification	b_{s_l}	0.49	0.64	0.40	0.57
	$b_{s_l,norm}$	0.46	0.55	0.45	0.51
Obstacle Avoidance	b_{s_l}	0.47	0.56	0.42	0.59
	$b_{s_l,norm}$	0.44	0.48	0.47	0.53

The effect of the normalization is easily visible. Particularly radio has a high feature rating and is negatively affected by the normalization. The only example with a negative average feature rating is the depth camera. After applying the normalization it becomes competitive in some applications.

Finally, Table 5 shows the search results regarding the different technologies and applications. Particularly the ACM DL keyword search can generate empty results if the search terms are too specific. Thus, the search terms we were using are “gesture“, “identification and “obstacle” in this regard and add synonyms for the different technologies. For each sensor category we allowed the following synonyms. “Camera” and “video” for the first technology, “radio”, “rf” and “wifi” for the second, “depth camera”, “stereo camera” and “Kinect” for the third and “ultrasound” as well as “ultrasonic” for the last one. All search results were averaged according to the number of synonyms used For the Google Scholar search we used more specific terms, “hand gesture”, “user identification” and “obstacle avoidance” with the same synonyms to prevent an excessive number of search results and prevented inclusion of patents and citations. All searches were performed on January 30th, 2014.

Table 5. Search result frequency given specific applications, sensor technologies and synonyms for ACM Digital Library (DL) and Google Scholar (GS)

	Camera		*Radio*		*Depth Camera*		*Ultrasound*	
	DL	GS	DL	GS	DL	GS	DL	GS
Hand Gesture	66	14100	27	7350	32	6850	3	1660
Identification	81	5590	162	4920	10	3957	5	599
Obstacle Avoidance	8	24000	1	13017	17	12278	8	14500

5.2 Discussion of Benchmarking Strategy

In this evaluation we included both benchmark score types to outline their differences. “Camera”, “radio” and “ultrasound” have a feature rating above average, whereas “depth cameras” had a lower than average rating. The feature-normalized benchmark score is thus adapted accordingly. Regarding the application of “hand-gesture recognition” this leads to “depth cameras” being considered the optimal technology as opposed to “cameras” that had a higher score before normalization. For the other applications there is no change in optimal technology. The preferred strategy for applying feature-normalized or non-feature-normalized benchmark scoring should depend on the specific benchmarking. If we are comparing numerous technologies and applications at once, the feature-normalization might be helpful to get a tendency regarding the optimal system. However, if the application is very specific it might be preferred to get a clear ranking and penalize unsuited technologies, regardless of their average feature weight. Accordingly, it is possible to refrain from normalization.

5.3 Discussion of Search Results

Looking at the search results we can draw several conclusions. The prevalence is unequally distributed between the different technologies. Both in keywords and general occurrence cameras are the most commonly occurring sensor device, with radio and depth camera ranked behind. Ultrasound on the other hand is less frequently occurring. This may be explained by the higher versatility of the other options. Regarding the “hand gesture” application, cameras have both the highest benchmark scores and most results in the database searches. The benchmark score for “user identification” and “radio” are matched for the ACM DL. However, there are more GS results for “camera”. As already mentioned cameras are more commonly used, yet, the difference in keyword search results is significant. “Obstacle avoidance” is least common in the ACM DL, however quite popular in GS. Accordingly, “ultrasound” sensors are significantly more common in both searches, as opposed to the previous applications. Nonetheless, “stereo cameras” are the most common sensor device for this application. They are commonly used in automotive scenarios, where the detection range of ultrasound is insufficient, as the objects are moving fast [18]. Therefore, the application scenario might have to be redefined for fast-moving object detection in open areas as opposed to obstacle avoidance for robots in home scenarios.

5.4 Querying Scientific Databases

We additionally have to discuss the method of using database searches for verifying the benchmarking method, as opposed to expert opinion. Surveys of a specific application or certain technologies are common in scientific literature. However, while they might be comprehensive and cite several hundred different applications, the ACM DL database covers more than 2.2 million entries and GS searches can lead to more than 9.7 million results. Therefore, the index searches are preferable in terms of broadness. The search for keywords in ACM DL results in few hits compared to the database size. As they are chosen by the authors there is a large variety in word choice, spelling or number of keywords. While extending the number of different searches might lead to more results overall, it may also lead to additional overshoot, including work that does not cover the desired topics. The GS searches are very prone to overshooting, and should be preferably used to discover trends in data, as opposed to narrowly clustered results. The presented approach is just a first attempt on using these databases to evaluate the popularity of different research topics. Potential extensions to the presented approach could use automated querying of similar search terms, a specific weighting of keyword or creating yearly queries to discover more recent trends. Additionally, one could consider preferring frequently cited articles, thus including the scientific impact of certain works into the results. While the ACM DL is more focused on computer science and has a well-defined database, GS provides an open and fast search that can be more easily fed using scripts. Therefore, it is suited for more complex searches.

5.5 Central Tendency Bias

We want to briefly discuss the tendency of the benchmark scores to crowd around 0.5. While the benchmark score has a range between 0 and 1 the two normalization processes the average is close to 0.5. Thus, even smaller differences close to this average may have a higher significance. This effect is called central tendency bias and is a common occurrence on Likert-scale questionnaires and rating systems [19]. Experts scoring technologies, just like survey respondents have a tendency to avoid extreme responses to a question. While experience of the person executing the benchmarking process might avoid this problem, it is also possible to use a corrective term in the calculation of the final benchmarking score. The primary purpose of this corrective term is to make the comparison between different scores easier to the reader. The following equations can be used to fix either regular or normalized benchmarking scores, resulting in the modified benchmarking score m_b, respectively $m_{b,norm}$:

$$m_b = (b_{s_l} + 0.5)^a \qquad m_{b,norm} = (b_{s_l,norm} + 0.5)^a \tag{10}$$

The exponent a should be a value higher 1 and chosen according to the level of adjustment that is desired. As an example, Table 6 shows adaptations of b_{s_l} and $b_{s_l,norm}$ for cameras and ultrasound from Table 4. The different values for a are 1, 5 and 10.

Table 6. Central tendency bias correction for different exponents a

		Camera			*Ultrasound*		
		a=1	a=5	a=10	a=1	a=5	a=10
Hand Gesture	m_b	1.03	1.16	1.34	1.05	1.28	1.63
	$m_{b,norm}$	1.00	1.00	1.00	1.00	1.00	1.00
Identification	m_b	0.99	0.95	0.90	1.07	1.40	1.97
	$m_{b,norm}$	0.96	0.82	0.66	1.01	1.05	1.10
Obstacle Avoidance	m_b	0.97	0.86	0.74	1.09	1.54	2.37
	$m_{b,norm}$	0.94	0.73	0.54	1.03	1.16	1.34

6 Conclusion and Future Work

On the previous pages we have introduced the benchmarking model that calculates a benchmark score as an indicator for the suitability of a sensor technology for a certain application. Additionally, it is possible to use the inverse option and benchmark a single sensor technology for a number of applications. The model was derived based on a set of common features for sensor technologies and a weighting factor determining their importance for smart environment systems. It was tested using a frequency analysis of related search terms in the ACM DL and GS scientific databases. Furthermore, we have discussed the effects of different normalization and bias compensation techniques on the benchmarking score.

As future work we want to improve our verification by using survey data to determine a more definite set of sensor features. We are planning to use the benchmarking model for actual validation of different sensor technologies within smart environments. Using this on a large set of potential application domains lets us verify existing applications or identify novel use cases, if a good score is calculated for a domain where the sensor technology has not been used yet. A good candidate is capacitive proximity sensing that our group worked with extensively in the past.

References

1. Camp, R.C.: Benchmarking: the search for industry best practices that lead to superior performance. Quality Press Milwaukee (1989).
2. Lewis, B.C., Crews, A.E.: The evolution of benchmarking as a computer performance evaluation technique. MIS Q. 7–16 (1985).
3. Dongarra, J.J., Luszczek, P., Petitet, A.: The LINPACK benchmark: past, present and future. Concurr. Comput. Pract. Exp. 15, 803–820 (2003).
4. Weicker, R.P.: Dhrystone: a synthetic systems programming benchmark. Commun. ACM. 27, 1013–1030 (1984).
5. Henning, J.L.: SPEC CPU2000: Measuring CPU performance in the new millennium. Computer (Long. Beach. Calif). 33, 28–35 (2000).
6. Smith, J.E.: Characterizing computer performance with a single number. Commun. ACM. 31, 1202–1206 (1988).
7. Crolotte, A.: Issues in benchmark metric selection. TPC Technology Conference. pp. 146–152. Springer (2009).
8. Jain, R.: The art of computer systems performance analysis. John Wiley & Sons Chichester (1991).
9. Ranganathan, A., Al-Muhtadi, J., Biehl, J., Ziebart, B., Campbell, R.H., Bailey, B.: Towards a pervasive computing benchmark. Proceedings PerCom. pp. 194–198 (2005).
10. Barsocchi, P., Chessa, S., Furfari, F., Potorti, F.: Evaluating Ambient Assisted Living Solutions: The Localization Competition. IEEE Pervasive Comput. 12, 72–79 (2013).
11. Santos, R.M., Oliveira, K.M. de, Andrade, R.M.C., Santos, I.S., Lima, E.R.: A Quality Model for Human-Computer Interaction Evaluation in Ubiquitous Systems. Latin American Conference on Human Computer Interaction. pp. 63–70 (2013).
12. Jafari, S., Mtenzi, F., O'Driscoll, C., Fitzpatrick, R., O'Shea, B.: Measuring Privacy in Ubiquitous Computing Applications. Int. J. Digit. Soc. 2, 547–550 (2011).

13. Wilson, J.S.: Sensor technology handbook. Elsevier (2004).
14. Braun, A., Wichert, R., Kuijper, A., Fellner, D.W.: Capacitive proximity sensing in smart environments. J. Ambient Intell. Smart Environ. (in press).
15. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychol. Bull. 76, 378 (1971).
16. Landsiedel, O., Wehrle, K., Götz, S.: Accurate prediction of power consumption in sensor networks. Proceedings Workshop on Embedded Networked Sensors (2005).
17. Bertozzi, M., Broggi, A.: GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Trans. Image Process. 7, 62–81 (1998).
18. Crawford, L.E., Huttenlocher, J., Engebretson, P.H.: Category effects on estimates of stimuli: Perception or reconstruction? Psychol. Sci. 11, 280–284 (2000).