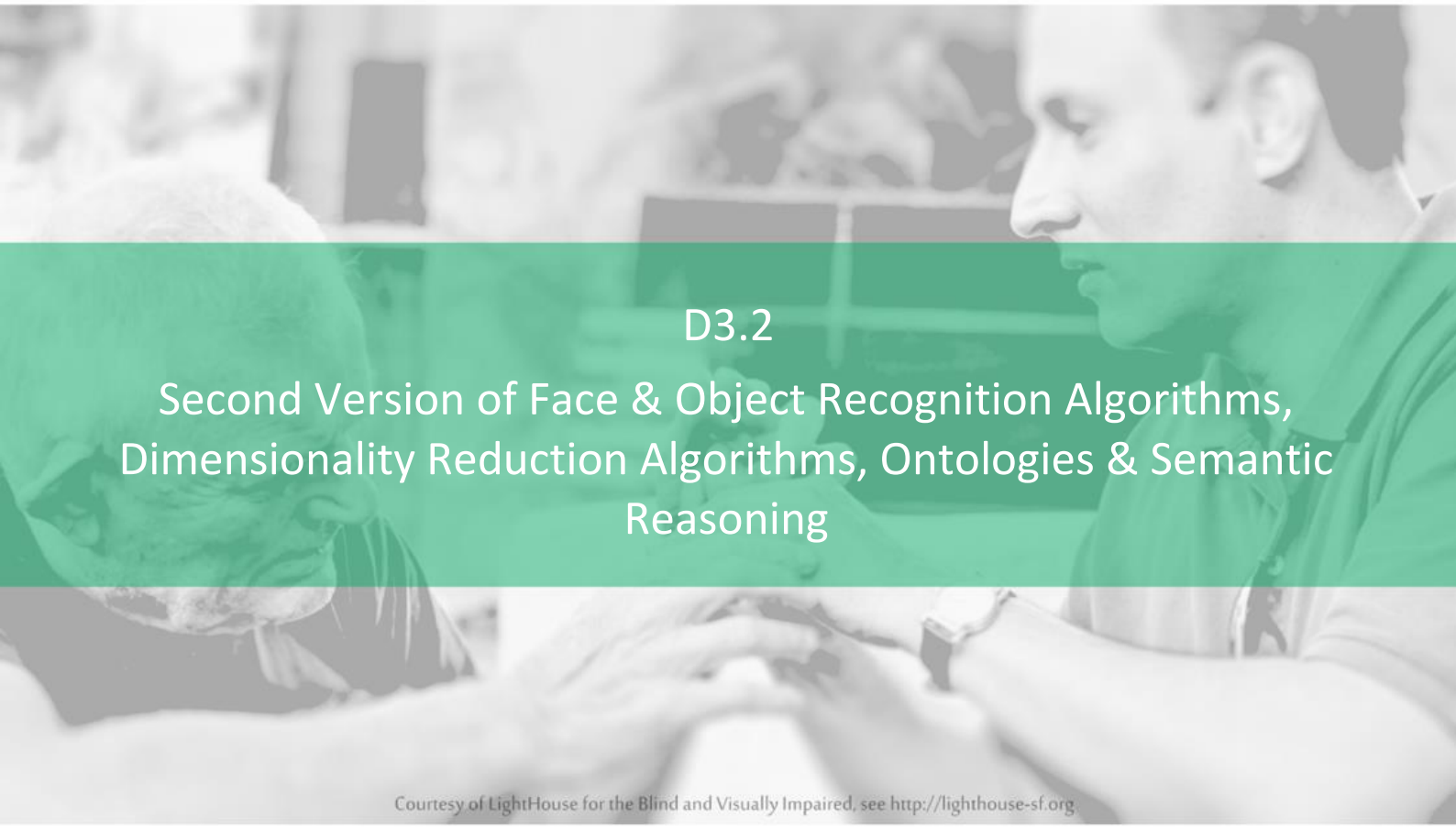
SUITCEYES

1 Jan 2018 - 31 Dec 2020

Smart, User-friendly, Interactive, Tactual, Cognition-Enhancer, that Yields Extended Sensosphere
Appropriating sensor technologies, machine learning, gamification and smart haptic interfaces



D3.2

Second Version of Face & Object Recognition Algorithms, Dimensionality Reduction Algorithms, Ontologies & Semantic Reasoning

Dissemination level		
PU	PUBLIC, fully open, e.g. web	X
CO	CONFIDENTIAL, restricted under conditions set out in Model Grant Agreement	
CI	CLASSIFIED, information as referred to in Commission Decision 2001/844/EC.	

Deliverable Type		
R	Document, report (excluding the periodic and final reports)	X
DEM	Demonstrator, pilot, prototype, plan designs	
DEC	Websites, patents filing, press & media actions, videos, etc.	
OTHER	Software, technical diagram, etc.	

Deliverable Details	
Deliverable number	D3.2
Part of WP	WP3
Lead organisation	CERTH
Lead member	P. Petrantonakis

Revision History			
V#	**Date**	**Description / Reason of change**	**Author / Org.**
v0.1	31-Dec-2019	Structure proposal	P. Petrantonakis / CERTH
v0.2	30-Jan-2020	1st draft for internal review	P. Petrantonakis / CERTH S. Darányi / HB
v0.3	07-Feb-2020	2nd draft addressing review comments	P. Petrantonakis / CERTH S. Darányi / HB
v0.4	14-Feb-2020	Pre-final draft after PC's comments	P. Petrantonakis / CERTH
v1.0	17-Feb-2020	Final draft submitted to the EU	T. Bebis / HB

Authors	
Partner	**Name(s)**
CERTH	P. Petrantonakis
HB	S. Darányi

Contributors		
Partner	**Contribution type**	**Name**
CERTH	Internal revisions to Chapter 2	P. Giannakeris
CERTH	Internal revisions to Chapter 4	M. Riga
ULEEDS	Internal review	R. Holt, Zhengyang Ling

Glossary	
Abbr./ Acronym	**Meaning**
ADL	Activities of Daily Living
CNN	Convolutional Neural Networks
DR	Dimensionality Reduction
GMM	Gaussian Mixture Modelling
HIPI	Haptic Intelligent Personal Interface
HOF	Histograms of optical flow
LBP	Local Binary Patterns
mAP	mean Average Precision
MBH	Motion boundary histogram
OWL	Web Ontology Language
PCA	Principal Component Analysis
SIFT	Scale Invariant Feature Transform
SNE	Stochastic Neighbours Embedding
W3C	World Wide Web Consortium

Executive Summary

This document constitutes SUITCEYES deliverable D3.2 presenting the work conducted during the period M13-26 within WP3 and reports on the following:

(a) The advanced version of visual analysis algorithms evaluated in benchmark datasets for performing object detection and tracking, face detection and tracking, and first-person activity recognition;

(b) A novel algorithm for dimensionality reduction along with the description of haptogram design for perceptible haptic information in low dimensional (4x4) haptic grid;

(c) The second version of the semantic knowledge graphs for semantically integrating the multimodal outputs from the various heterogeneous SUITCEYES sensors and components.

Table of Contents

1 Introduction

In this deliverable we present the advanced versions of the algorithms and methods that were presented in D3.1.

In the second Chapter, we present additions and modifications that have been made to the object detection module with respect to: a) the requirements and capabilities of the camera sensor that has been integrated on the HIPI system and b) the expansion of existing functionalities of the module. Secondly, a new face detection method is presented which is unified with generic object detection by treating human faces the same way as any other object of interest. In addition, an advanced first-person activity recognition algorithm is presented which leads to better results by exploiting Dimensionality Reduction approaches within the implementation of the method.

In the third chapter, a new dimensionality reduction algorithm is presented that exhibits good representations in low dimensional spaces with very low computational burden. Nevertheless, for more robust and reliable haptic information we also describe the design rationale of haptograms, i.e., specific haptic grid representations for different objects and scenes. In addition, dynamic haptograms are combined to form phrases to inform the user with environmental information and concepts.

The fourth chapter presents the second generation of the implemented SUITCEYES ontology and the semantic reasoning framework. The ontology was extended to fit the needs of the users/project requirements (as those were defined in D2.1, D2.2) and to the advanced functionality of the HIPI, focusing mainly on those related to the increase of the context and location awareness of the people with deafblindness. Moreover, the so-called Knowledge Base Service (KBS) was created, which enables the communication and integration of data reported from both sensory- and analytics-components to the knowledgebase. Finally, the Semantic Reasoning Mechanism (SRM) was enriched with ontology-based rules that serve specific queries for inferring the aforementioned context-related information, either in the form of structured content (JSON format) or as natural language phrases.

Finally, section 5 concludes the deliverable.

2 Visual Analysis

2.1 Algorithms and Results

This section discusses the current visual analysis algorithms that have been developed and provides qualitative and quantitative results for each task as a means of evaluation.

2.1.1 Object Detection and tracking

One of the main functions of Visual Analysis in the SUITCEYES project is to detect and recognize the objects that exist in front of the user. It is noteworthy that object identification was also highlighted in D2.1 and D2.2 as an important activity that the HIPI system should embed. In D3.1 we described how pixel coordinates of bounding boxes can be predicted by a deep Convolutional Neural Network (CNN) object detector. We also showed how we can fine-tune the Faster-RCNN-resnet101 architecture on the objects of the Activities of Daily Living (ADL) dataset which has annotated boxes taken from a first-person view camera. Moreover, we have presented a way to speed up object detection in videos using object tracking through individual frames which is a cheaper procedure in terms of computational complexity. We describe in this section the additions and modifications that have been made to the object detection module with respect to: a) the requirements and capabilities of the particular camera sensor that has been integrated on the HIPI and b) the expansion of existing functionalities.

The RGB-D sensor that is attached to the HIPI, as described in D4.1, is the RealSense R200 and is mounted on an Up-Board running the Robot Operating System (ROS). The camera is capable of providing conventional colour images as well as depth maps, which are the equivalent 2D distance images with values measured in millimetres. The Visual Analysis component takes as input the RGB images and the corresponding depth maps that are generated from the Sensor System and are provided at a fixed rate. In the current version the Sensor System supports data upload with an approximate rate of 1 image per second. Object tracking is not required to be applied in this setting, since object detection for an image takes less than a second. In other words, in this setting, the Object Detection module runs faster than the rate of data acquisition even without the use of Object Tracking.

Additionally, depth information from the RGB-D sensor can now be used in order to estimate the distance of the detected objects from the HIPI. The depth maps are transformed to have the same resolution as the input RGB images (480x640 pixels). Bounding box coordinates can then be projected to the appropriate depth map and the mean distance value inside each rectangular region can be calculated for every detected object. There are cases where the sensor cannot capture an accurate distance measurement (e.g. reflective or dark surfaces) and the corresponding pixels take a zero value. In cases where the obtained rectangular region has many zero values, an accurate mean distance cannot be estimated and is therefore set to -1 so as to signal unknown distance.

Until now, the object detector could recognize objects captured from a first-person view camera with a wide field of view (fov). Objects appear distorted in wide fov cameras like the one that was used to capture the images of the ADL dataset, where the first version of the object detector was fine-tuned. Realsense R200 however does not have a wide fov sensor, thus the distortion does not appear in the images. In order to extend the applicability of our object detector to more general-purpose camera settings like the RealSense R200 we chose to adopt the Faster R-CNN architecture, pre-trained in the Open Images V4 (OI4) dataset (Kuznetsova et al., 2018). The OI4 is a collection of images that contain bounding box annotations for 600 object categories. All the images in the dataset are free from wide angle lens distortion, thus leaving the trained model free from the fov bias. From the 600 categories we select only those that match

the object types of the ADL dataset and are generally items used in everyday living. Note however, that all the other categories of the OI4 dataset remain integrated into the model and can be easily added to expand the list of available objects if necessary, at a future version.

Our Object Detection module has been tested and evaluated in the mid-term review of the project. During the demo, the RealSense R200, mounted on the HIPI, was configured to send images with an approximate rate of 1fps. A real time communication protocol was designed and repeatedly tested prior to the demo, so as to establish a reliable functioning communication pipeline.

The protocol is defined as follows:

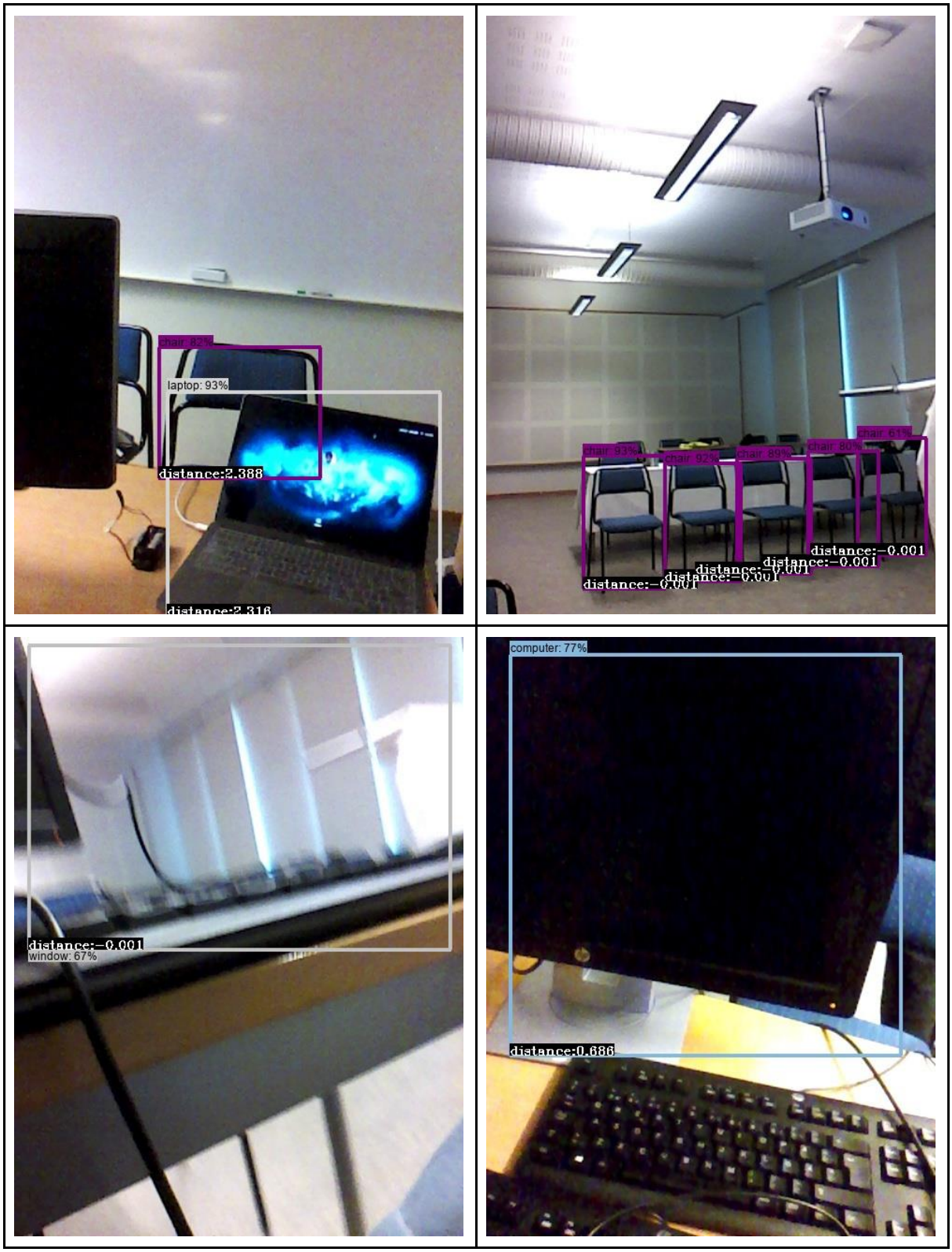
1. As the RGB image is captured from the RealSense, it is stored first locally to the HIPI. At the same time, the depth values are captured and translated to the corresponding depth map image. The capture timestamp is appended to both image filenames.
2. At the other end (the CERTH host that performs the Visual Analysis tasks) an image storage service was set up. The image storage service can accept PHP POST requests with image files as contents, and then store them locally. The service is also password protected and rejects every upload request that doesn't enclose the correct passphrase.
3. As soon as an image is uploaded from the HIPI to the CERTH host, the HIPI also sends a notification to the Visual Analysis API that a new image is ready to be analyzed. The API is responsible for all the communication, queueing and threading functions of the Visual Analysis component. It can accept multiple requests simultaneously and serve back the analysis results to the KB (Knowledge Base) using a (First In First Out) FIFO queue.

Specifically, for the evaluation of the Object Detection module during the demo, we selected object categories that were expected to be present in the demo location, i.e. a classroom. The object categories from OI4 that were integrated are shown in Table 1.

Table 1: Object categories used in the demo.

table	*chair*	*window*
mug	*book*	*cell phone*
laptop	*computer*	

Figure 1 shows qualitative results that were obtained during the demo. Each detection is visualized with a bounding box around the object, a class prediction with a confidence score and an approximate distance measurement. As explained earlier, negative or zero distance values indicate inability to fetch sufficient depth data inside the bounding box region. It can be observed that in a real demonstration environment there are several factors in play that can seriously challenge the robustness of the detector, such as luminosity, obscure viewpoint variations, blurriness induced by camera motion and occlusions. Nevertheless, it can be seen in the images that the majority of the objects that exist in them, have been detected accurately.


chair: 82%
laptop: 93%
distance:2.388
distance:2.316
chair: 93%
chair: 92%
chair: 89%
chair: 80%
chair: 61%
distance:-0.001
distance:-0.001
distance:-0.001
distance:-0.001
distance:-0.001
distance:-0.001
window: 67%
computer: 77%
distance:0.686

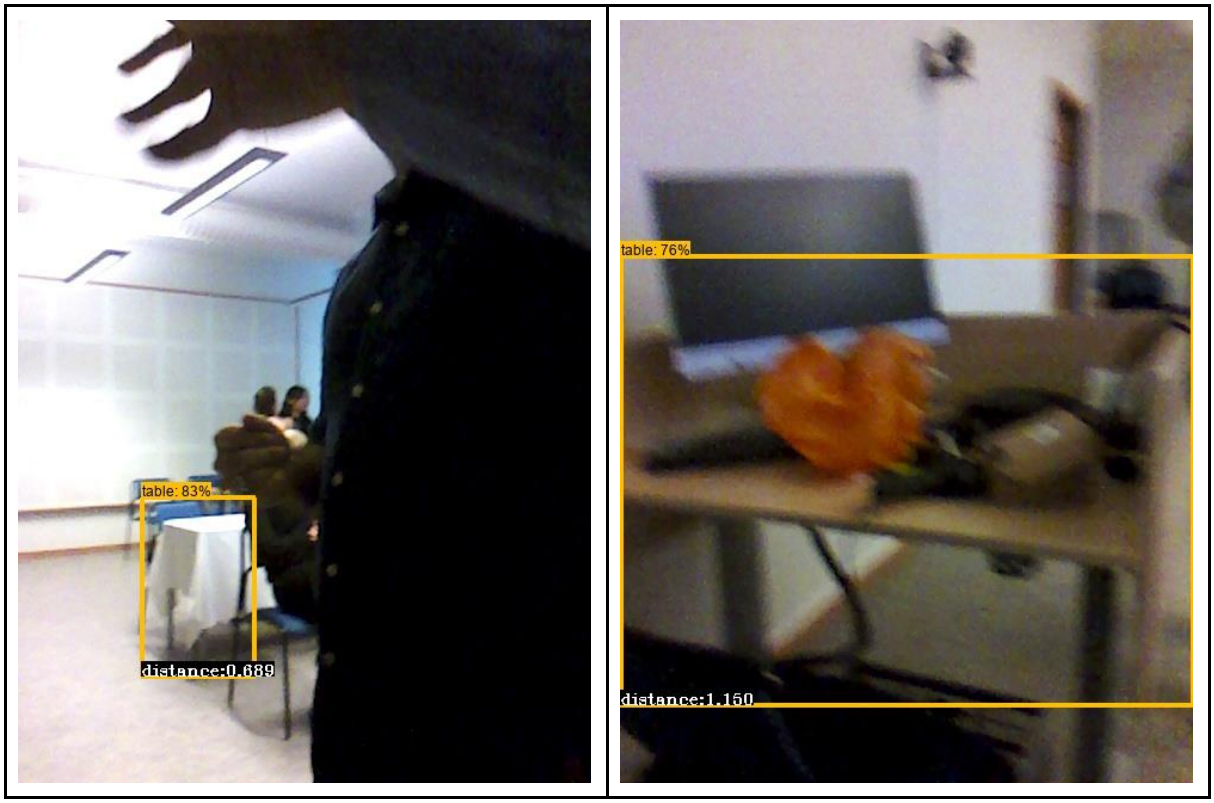


Figure 1: Object Detection qualitative results.

2.1.2 Face Detection

In the previous version of Face Detection module (Deliverable 3.1), we chose to adopt the TinyFaces deep CNN framework (Hu & Ramanan, 2017) due to its distinct advantage in the detection of various face sizes and scales. In order to gain the aforementioned advantage, the input is filtered through the CNN feature extractor multiple times so as to perform forward passes at various scales. This technique generally performs better when the CNN is fed with high resolution images because the processing can be done at numerous scales. However, due to the heavy workload that this technique imposes on the overall image processing cycle it is considered unsuitable for a real-time application. As already mentioned in D3.1, the face detection problem can be effectively tackled by an algorithm for generic object detection that is trained to recognize bounding boxes of faces. This approach can exploit sharing of GPU resources for both the object and face detection tasks so as to enhance interoperability and speed up execution times. With the upgrade of our object detector which can now recognize object categories from the OI4 dataset, face detection becomes possible without an additional model deployment, as the OI4 dataset contains face instances in addition to other objects. Thus, face detection in this version is unified with generic object detection by treating human faces the same way as any other object of interest.

The purpose of face detection is to tell whether or not other people exist in the immediate surroundings of the HIPI user. Although this is an important function that can provide meaningful information to the user, it still remains agnostic towards the identity of the people for whom their faces have been detected. Thus, we aim to tackle the task of face recognition as well. We have never previously dealt with this task

in the SUITCEYES project; therefore, a brief background section follows before we delve deeper into the technical specifics regarding our developed and integrated techniques.

The task of face recognition has always been divided into two subtasks in the literature. Specifically, a technique is first employed for face feature extraction (face representation) and another is later used so as to classify a representation to a specific person identity. Face representations on most of the early works were built by low-level face descriptors, such as LBP, SIFT, or CMD variants, and were then combined with shallow models for classification, such as metric learning, discriminative dimensionality reduction, or Fisher encoding (Z. Cao, Yin, Tang, & Sun, 2010; Chen, Cao, Wen, & Sun, 2013; C. Huang, Zhu, & Yu, 2012; Simonyan, Parkhi, Vedaldi, & Zisserman, 2013). Later works achieve significant boost in performance with the aid of deep learning. In this class of algorithms deep feature extractors are used to generate face representations. They are trained with the aim to acquire invariance to pose and illumination from the plethora of the available training data rather than from low-level engineered features. Initial attempts to leverage deep learning involved training a Siamese network for deep metric learning (Chopra, Hadsell, LeCun, & others, 2005). A Siamese network works by extracting features separately from two compared inputs with two identical CNNs, taking the distance between the outputs of the two CNNs as dissimilarity. In (Zhu, Luo, Wang, & Tang, 2014) it was proposed to warp faces from arbitrary pose and transform them to frontal view with normal illumination using a trained deep neural network and then used the last hidden layer to get face representations.

A multi-stage approach was proposed in (Taigman, Yang, Ranzato, & Wolf, 2014) that aligned faces to a general 3D shape model and combined them with a multi-class network for classification. Building several compact networks that processed different face patches simultaneously was also explored in (Sun, Wang, & Tang, 2015). Soon after, significant performance improvement was seen by approaches that focused on deep metric learning. The aim of those approaches is to extract face representation vectors that have smaller maximal intra-class distance than the minimal inter-class distance under the embedding space. This is the essential characteristic that modern approaches strive to accomplish for Face Recognition under the deep metric learning formulation. Therefore, recent works meticulously explore and experiment with several loss functions for CNNs, so as to find the most appropriate for this task (Deng, Guo, Xue, & Zafeiriou, 2019; Liu et al., 2017; Schroff, Kalenichenko, & Philbin, 2015; Wang et al., 2018).

There are a few benchmark databases that exist for Face Recognition evaluation. The most popular database used in the literature is Labeled Faces in the Wild (LFW) (G. B. Huang, Mattar, Berg, & Learned-Miller, 2008). LFW contains close to 13K images that show faces of 5749 people. One of the best performing models at LFW is the FaceNet embeddings (Schroff et al., 2015). The authors propose a deep CNN feature extractor coupled with L2 normalization that leads to an embedding which is trained in an end-to-end manner using a triplet loss function on the output. The triplet loss is designed to minimize the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity. The model can then be trained on a large-scale database so as to learn meaningful face embeddings with the aim to minimize the distances of same identity instances in Euclidean space. Thus, for our Face Recognition module, we harness the FaceNet's discriminative power of embeddings.

Considering that the HIPI may detect several unknown identities to the user throughout his daily life, we shall shield him from constant and redundant information. Besides, even if this wasn't the case, in a real-world application a face recognition system cannot exist without incorporating the concept of the "unknown" identity, since there can only be a finite number of known identities to the system. Therefore, a small-scale database of familiar identities must first be constructed for each user. Each custom-made face database must contain several images per person. Empirically we set the minimum number of images per person to be 20. In order to increase the robustness of our Face Recognition system each portrait

collection must provide images taken under various illumination settings and camera viewpoints. A custom-made database can then be used to train our Face Recognition system.

We first obtain FaceNet Inception ResNet v1 feature extractor pre-trained on the VGGFace2 database (Q. Cao, Shen, Xie, Parkhi, & Zisserman, 2018) which consists of 3.3M faces and 9000 classes. We use the pre-trained embedding layer in order to generate face representations for several identities that exist in the LFW database. We select those for which the dataset provides more than 20 instances. Those instances will serve as a pool of random unknown identities. Then, we generate face representations for all the identities in the custom-made known face database and combine the two groups, known and unknown, together. Finally, we train a multi-class SVM classifier to recognize all the identities. During inference time, a given instance is fed to FaceNet embedding layer first. Then, the resulting vector is passed through the SVM classifier and an identity class is predicted. In the case of a known identity, the embedding layer is trained to compute a representation close to the training samples of the same identity. Therefore, the instance is expected to be classified correctly to the appropriate known identity with high confidence. In the case of an image with a random pedestrian, the embedding layer will produce a representation similar to no known or unknown identity in the training set, but the classifier will forcefully predict one of the identity classes nevertheless. However, the prediction will be made with a low confidence due to the high dissimilarity of the embedding vector with the training samples. Therefore, a confidence threshold is applied so as to properly interpret the SVM's classification result. If a known identity class is predicted and the confidence score surpasses the threshold, we accept the classification result. Otherwise, if the confidence score is low or an unknown identity class is predicted we derive that the query instance belongs to an unknown person.

As in the Object Detector's case, we have evaluated the performance of the Face Detection and Recognition module during August's demonstration. To be precise, as we have explained in this chapter, the Face Detection task is now integrated practically into the Object Detection module, and the Face Recognition module is now applied on top of the faces that have been detected. Figure 2 show several images taken during the demo with detected faces inside white rectangles, and recognized faces inside yellow rectangles. Specifically, in order to evaluate the Face Recognition module in real time conditions, we gathered a small database with some photos of a member of the SUITCEYES project, to act as the known person in our demonstration. It can be seen in the images that the module can recognize the known person (person_PanosPet) from different angles. At the same time, several unknown persons have been detected in the images, even in challenging instances with small scale or occluded faces.

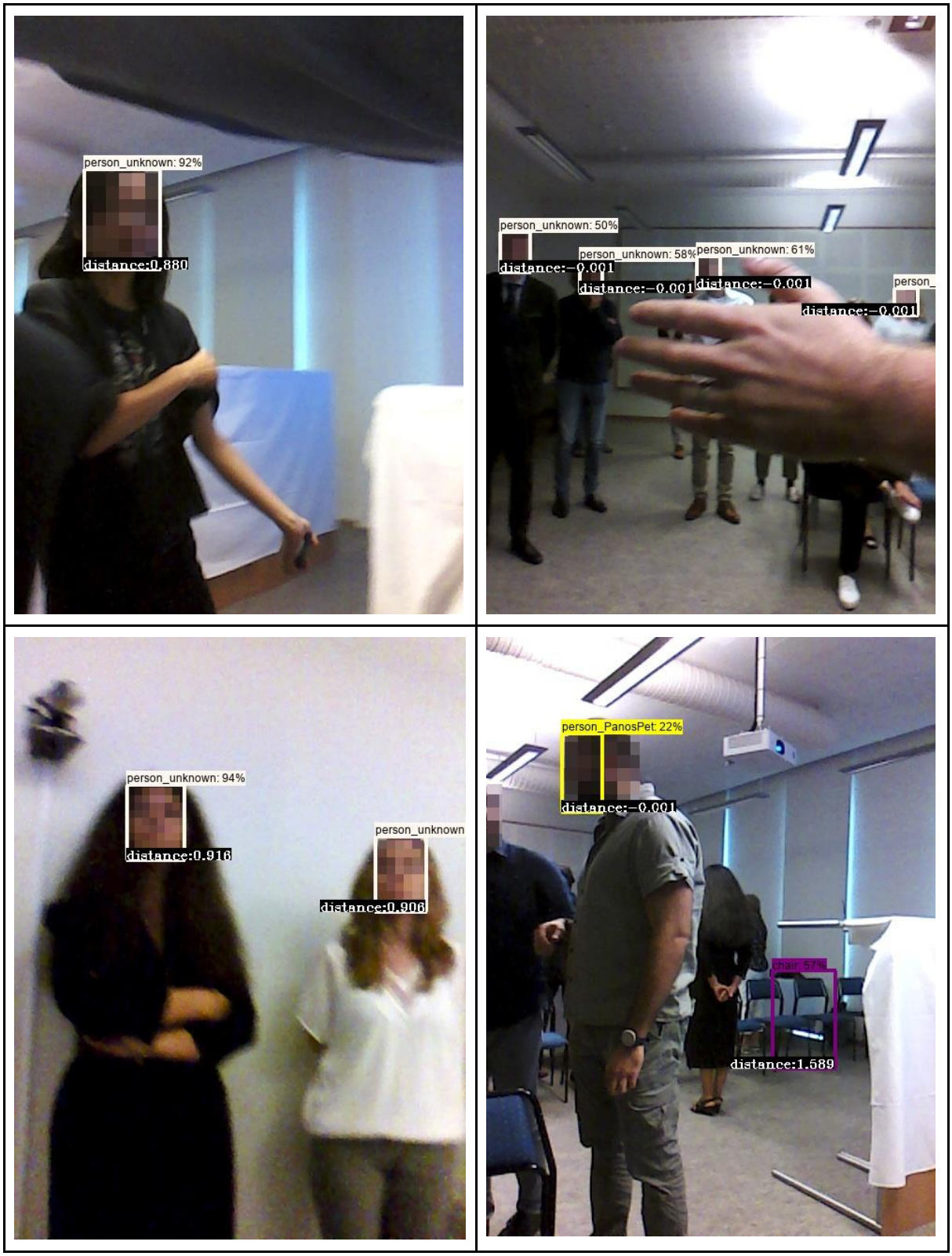


Figure 2: Face recognition qualitative results.

2.1.3 First-Person Activity Recognition

We have presented in D3.1 a novel first-person human activity recognition framework. Our methodology is inspired by the central role object movements have in egocentric activity videos. Using a deep Convolutional Neural Network, we detect objects and develop discriminant object flow histograms in order to represent fine-grained micro-actions during short temporal windows. Our framework is based on the assumption that large scale activities are synthesized by fine-grained micro-actions. We gather the micro-actions and perform Gaussian Mixture Model clustering, so as to build a micro-action vocabulary that is later used in a Fisher encoding scheme. However, the high dimensionality of our micro-action descriptors severely affects the computational burden. In the new version of the algorithm we intended to alleviate this problem through dimensionality reduction approaches. Thus, the research endeavour of this algorithm is equally shared between Task 3.1 and Task 3.2 of WP3 but is presented in this chapter for consistency with D3.1. This research has led to a journal publication (Giannakeris et al., 2020).

For dimensionality reduction two approaches were adopted, i.e., Principal Component Analysis (PCA) and Random Projections (RP).

PCA projects the data onto a lower-dimensional orthogonal subspace that captures as much of the variation of the data as possible. More formally, the PCA technique works as follows: first the covariance matrix $C_X = X^T X$ is computed, where $X \in R^{n \times D}$ being the high-dimensional data (large D). Secondly using the Single Value Decomposition method, the covariance matrix is decomposed, i.e., $C_X = U \Lambda U^T$. U is the eigenvector matrix whereas Λ is a diagonal matrix of the eigenvalues of C_X. Finally, the orthogonal representation is accomplished by a linear mapping, i.e., $Z = XU$ whereas the dimensionality reduction is done when using only the first d components, i.e., $Y = XU_d$, where U_d is a matrix that contains only the d columns of U (sorted in decreasing order of the corresponding eigenvalues) and thus $Y \in R^{n \times d}$. By definition, such a reduction minimizes the $||X - YU^T||_2^2$ error. Nevertheless, the eigenvalue decomposition of the data covariance matrix with size $D \times D$ is expensive to compute. The computational complexity of estimating the PCA is $O(D^2 n) + O(D^3)$.

Due to the fact that the PCA approach is quite expensive to compute for high-dimensional data sets we also investigated a computationally simpler method of dimensionality reduction that does not introduce a significant distortion in the dataset, i.e., the RP approach. In RP the original high-dimensional data, $X \in R^{n \times D}$ is projected onto a lower-dimensional subspace using a random matrix whose rows have unit lengths. More formally, using matrix notation where $X \in R^{n \times D}$ is the original set of n D-dimensional observations, $Y = XR$, where $R \in R^{D \times d}$ is the random matrix and $Y \in R^{n \times d}$ is the projection of the data onto the lower d-dimensional space. The fundamental idea of random projection arises from the well-celebrated Johnson-Lindenstrauss lemma (Johnson & Lindenstrauss, 1984) which states that if point instances in a vector space are projected onto a randomly selected subspace of appropriate dimension, then distances are approximately preserved (Frankl & Maehara, 1988). In this work we use random matrix whose elements are Gaussian distributed with zero mean and unit variance.

Due to its computational simplicity and our sparse feature vectors, RPs are ideal for the dimensionality reduction task in this work. In particular, the aforementioned random projection procedure is of order $O(Ddn)$ and taking account that X is in our case sparse (assuming l nonzero entries per row) the complexity is of order $O(ldn)$ (Papadimitriou, Raghavan, Tamaki, & Vempala, 2000).

We have already discussed our framework's parameters in D3.1, which briefly are: W, the micro-action temporal window duration in seconds, the number of PCA components, the number of GMM clustering words, the Neural Network classifier architecture.

Keeping other model parameters fixed, we experiment upon using different dimensionality reduction schemes, for all the different descriptors and window sizes. The same evaluation scheme is also applied in this section as well, i.e. leave-one-person-out cross validation.

First, we investigate selecting 1000 PCA components instead of only 80 and 256 in order to explore the capabilities of our method for higher dimensional micro-action vectors. When reducing from some thousand components to only 256 it is possible that only a small portion of the dataset variance can be explained, thus the reduction step can become a bottleneck to the overall performance. In Table 2 the results indeed show significant improvement in mAP (mean Average Precision) for all settings ranging from 3% to 5%.

Table 2: Activity recognition performance for various PCA components.

	Performance (mAP%)			
PCA Components	MBH W90	MBH W60	HOF W90	HOF W60
80	52.9%	57.1%	52.4%	50.9%
256	53.1%	57.1%	52.8%	48.1%
1000	58.9%	**60.1%**	**56.8%**	54.9%

However, as discussed earlier PCA is rather expensive to compute mainly during training time for high dimensional data. Especially in our case, the micro-action descriptors depending on the setting are at least 30720-dimensional and up to 92160-dimensional vectors before the reduction step. For those reasons we chose to experiment with RP using 4 different settings: d=1000, d=2500, d=3500 and d=5000. Table 3 shows the results. We can see that with Random Projections close to 3500 components, the method scores either comparably or surpasses PCA's lower component settings (80, 256). Considering, all RP experiments took lesser time to produce there exists a performance/speed trade-off when choosing one or the other. For full performance gain, it is evident that the 1000 PCA component setting is the ideal one.

Table 3: Activity recognition performance for various RP components.

	Performance (mAP%)			
RP Components	MBH W90	MBH W60	HOF W90	HOF W60
1000	51.1%	54.5%	50.3%	49.6%
2500	54.2%	54.9%	52.1%	50.6%
3500	54.3%	55.9%	52.3%	51.8%
5000	55.4%	**56.7%**	**53.7%**	53.1%

2.2 Chapter Summary and Future Work

We have reported in this section the advanced algorithms and methodologies of the Visual Analysis modules. We have evaluated this version's algorithms using either qualitative examples from the review demonstration or by extensive experiments in public datasets. The results indicate that the modules are progressing steadily towards improved performance and applicability. In the future, corrections and

adjustments on the developed algorithms will be made based on the outcomes of the Visual Analysis tools by people with deafblindness under a co-design framework and new on-site (run on raspberry Pi instead of running on a remote server) implementations of the object and face detection methods will be investigated.

3 Dimensionality Reduction and Haptograms

In this chapter we present a novel dimensionality reduction algorithm along with the theory, implementation issues and conceptual paradigms of haptograms' design for perceptible haptic information using either static or dynamic approach. Moreover, new DR approaches for more reliable and fast implementation of First-Person Activity Recognition task were also investigated (for detailed description see section 2.1.3).

3.1 Dimensionality Reduction - A novel method

In D3.1 we implemented and tested multiple dimensionality reduction (DR) algorithms for the direct mapping of high dimensional signals to low dimensional representations. In this version of the DR approaches, we introduce a novel DR method, which is based on Particle Swarm Optimization algorithm, and leads to great low dimensional representations of up to twenty (20) objects with low computational burden and fast performance (this work has been submitted as a conference publication and is currently under review).

In particular, we developed a new DR technique for data visualization based on the Particle Swarm Optimization algorithm (PSO) (Eberhart and Kennedy, 1995; Shi and Eberhart, 1998; Shi and Eberhart, 1999; Shi, 2001). The PSO-based DR approach (PSO-DR) seeks to preserve the high dimensional structure by exploiting the fast and versatile nature of the PSO algorithm. We provide the respective algorithmic approach for PSO-DR and stress out its straightforward expansion to parallel implementation for fast computation. We compare the proposed PSO-DR approach with the current state-of-the-art tSNE method along with other linear and nonlinear DR algorithms. Our approach is competitive with the most of the DR approaches and performs faster than those with the best visualization quality, especially for larger data sets. In addition, PSO-DR allows the mapping of new data points explicitly, in contrast with the majority of the nonlinear techniques where approximate estimation of out-of-sample extension leads to mapping errors of new data points. In general, through its simplicity, versatility, fast computation and straightforward out-of-sample extension PSO-DR constitutes an efficient, general purpose, DR technique.

3.1.1 Background

The problem of DR is to find a function that maps the high dimensional inputs to a lower dimensional space by preserving the inner structure of the data. Particularly, we seek to map a high dimensional data set $X \in \mathbb{R}^N$ to a $n-D$ data set $Y \in \mathbb{R}^n$ in a low dimensional space (usually $n = 2\ or\ 3$). Each low dimensional point $y_i \in Y,\ i = 1, \dots, M$ represents the mapping of a corresponding high dimensional point $x_i \in X,\ i = 1, \dots, M$. The approach followed, e.g., in classical multidimensional scaling (Torgerson, 1952) is to find all those low dimensional points $y_i, i = 1, \dots, M$ that minimize the sum of the differences between the pairwise distances in the high dimensional space with the pairwise distances of the low dimensional one, i.e., to minimize:

$$\Phi(Y) = \sum_{ij}(d_{ij} - \delta_{ij}) \quad (1)$$

where d_{ij} is the distance (dissimilarity measure) between x_i and x_j whereas δ_{ij} is the distance of the corresponding low dimensional points y_i and y_j. In the new DR method, instead of searching for an optimum solution set, Y, we seek for the optimum low dimensional points one by one. The key step of the proposed approach is to define a set of high dimensional beacons, X_b, i.e., certain, reference points in high dimensional space, and the corresponding low dimensional ones, Y_b. Then, map, e.g., a point x_i to a point y_i by comparing the distances of x_i from X_b with the distances of y_i from Y_b. The following two subsections describe briefly the PSO algorithm and the proposed PSO-DR approach, respectively.

3.1.2 Particle Swarm Optimization algorithm

The PSO algorithm was introduced by Kennedy and Eberhart in 1995 (Eberhart and Kennedy, 1995) and it was inspired by the social behavior of groups of, e.g., birds in order to solve optimization problems.

The PSO algorithm searches the space for optimal solution based on the information shared between the particles of a group. Each particle follows a trajectory which is influenced by stochastic and deterministic components. In particular, each particle moves according to its best achieved position, in terms of the optimization problem, and the best position of the group but with a random component. In every iteration (time point t), a random particle of the group changes its position $z_i^t, i = 1, \dots, P$ (P is the population of the particles in the group) according to the new velocity component, u_i^t (Eberhart and Kennedy 1995), (Koziel and Yang 2011) i.e.:

$$u_i^t = \omega u_i^{t-1} + w_p r_1 \left(z_{i,p}^{t-1} - z_i^{t-1}\right) + w_g r_2 (z_g^{t-1} - z_i^{t-1}) \tag{2}$$

$$z_i^t = z_i^t + u_i^t, \tag{3}$$

where $z_{i,p}^{t-1}$ and z_g^{t-1} are the previous best particle and group positions, respectively, ω, w_p, w_g are constant weights and r_1, r_2 are random numbers. Usually, search space and velocity values are bounded whereas the particles are initially distributed randomly in the search space.

3.1.3 The novel PSO-DR method

The proposed approach for DR is based on the PSO algorithm to find optimal positions Y that correspond one by one to the high dimensional instances X. Thus, for the search of an optimal solution y_i, a group of particles with positions $z_i, i = 1, \dots, P$ are moving according to the rule defined by equations 2 and 3. The function that is minimized is the dissimilarity between the distances of a high dimensional $x_i \in X$ from the high dimensional beacons $x_b^j \in X_b,\ j = 1, \dots, J$, and the distances of a low dimensional candidate solution y_i from the corresponding low dimensional beacons $y_b^j \in Y_b,\ j = 1, \dots, J$. In particular, we seek to minimize:

$$\Phi(y_i) = \sqrt{\sum_j \left(d_j - \delta_j\right)^2} \tag{4}$$

where $d_j,\ j = 1, \dots, J$ are the distances between x_i and $x_b^j, j = 1, \dots, J$, and $\delta_j, j = 1, \dots, J$ are the distances between y_i and $y_b^j, j = 1, \dots, J$. In essence, the optimal solution for equation 4, i.e., y_i, is the mapping of the high dimensional instance, x_i, in the low dimensional space.

Thus, as soon as the high and low dimensional beacons are defined, every high dimensional data can be mapped onto the low dimensional space by minimizing 4 using PSO. It should be stressed out that other optimization approaches could also be used but PSO was chosen here due to its simplicity and fast computation.

As high dimensional beacons, X_b, we choose randomly J instances from data set X. In order to estimate the corresponding low dimensional beacons, Y_b, we define as y_b^1 (associated with the x_b^1) a zero vector, i.e., $y_b^1 = [0 \dots 0] \in \mathbb{R}^n$. Then we proceed with the definition of the rest of the beacons $j_b^j, j = 2, \dots, J$ as follows: y_b^2 is estimated by minimizing $\Phi(y_b^2) = \sqrt{\sum_j (d_1 - \delta_1)^2}$ where d_1, is the distance between x_b^2 and x_b^1, and δ_1 is the distance between the candidate y_b^2 and y_b^1. In accordance, y_b^3 is estimated by minimizing $\Phi(y_b^3) = \sqrt{\sum_j (d_j - \delta_j)^2}$ where $d_j, j = 1{,}2$ are the respective distances between x_b^3 and x_b^1, x_b^2 and $\delta_j, j = 1{,}2$ are the respective distances between the candidate y_b^3 and y_b^1, y_b^2. The rest of the Y_b set is estimated with the same procedure. When the whole Y_b set is defined the rest of the X data set is mapped my minimizing equation 4 with respect to the beacons Y_b.

3.1.4 Implementation Issues

The parameters concerning the PSO algorithm, were chosen according to the best parameters list presented in (Pedersen 2010). In particular, for the number of particles (swarm-size, P), number of iterations of the PSO algorithm (stopping criterion), ω (inertia weight), w_p (particle's-best weight), w_g (swarm's-best weight) we used 25, 400, 0.3925, 2.5586, and 1.3358, respectively. These values were used for all experiments presented here.

For comparison reasons, other DR methods were also used. In particular, PCA, tSNE, Isomap, Sammon mapping, LLE, and Laplacian Eigenmaps were compared with PSO-DR. For all these methods the Matlab Toolbox for Dimensionality Reduction by Laurens van der Maaten (Van Der Maaten et al. 2009) was used. For each method the default values of the parameters provided by the toolbox were used.

The number of beacons for each data set was defined to be a quarter of the number of instances x_i in each data set X, i.e., $J = \frac{1}{4}M$ except for the cases where J was larger than 1000; then we set $J = 1000$ irrespectively of the data set size.

3.1.5 Experiments

We first describe the data sets used for DR and subsequently elaborate on the experimental setup. Next, we present the respective results.

Data Sets

Four different data sets were used to evaluate the performance of the PSO-DR algorithm. In particular, the MNIST data set (The MNIST data set is publicly available from http://yann.lecun.com/exdb/mnist/index.html), the COIL-20 data set (Nene et al. 1996), the FMNIST data set (Xiao et al. 2017) and the Swiss Roll data set with $M = 3000$.

The MNIST data set contains 60, 000, 28 x 28-pixel (i.e., $N = 784$), grayscale images of hand written digits $(0 \dots 9)$. In this work we choose randomly $M = 6000$ images (600 per class) for computation reasons. The FMNIST data set has the same format as MNIST except for the classes where each class represents fashion items. Again, for FMNIST we choose randomly $M = 6000$ images (600 per class). The COIL-20 data set, contains, 32 x 32 (i.e., $N = 1024$) images of 20 different objects which are viewed from 72 orientations (angles), i.e., resulting in $M = 1440$ images

Experimental Setup

For the MNIST, COIL-20, and FMNIST data sets we use the PSO-DR, PCA, tSNE, Isomap, and Sammon mapping techniques to transform the high dimensional representations to a two-dimensional ($n = 2$) map. For the Swiss Roll data set we use the PSO-DR, PCA, Laplcian Eigenmaps, Isomap, and LLE techniques to map to the 2-D space. We substituted tSNE and Sammon mapping with Laplacian Eigenmaps and LLE as the latter performed better with Swiss Roll data set.

The resulting maps in each one of the DR task are shown as scatter plots. The coloring in the scatter plots is used to provide a way of evaluation for the performance of the DR techniques. Moreover, for each one of the DR methods the time needed to map the respective data set is depicted.

For the proposed PSO-DR method, parallel computation was utilized. Due to its straightforward parallelization we used multiple of the CPU to estimate the mapping in parallel, accelerating that way the PSO-DR computation. After the estimation of the beacons Y_b is finished, the conversion of, e.g., a high dimensional input x_i is independent of the conversion of any other input $x_i, j \neq i,$ thus, it is possible to map simultaneously multiple inputs.

For the data sets MNIST, COIL-20, and FMINST, d_i and δ_i measures correspond to the euclidean distance whereas for Swiss Roll data set δ_i is the euclidean distance and d_i corresponds to the geodesic distance estimated as in the Isomap method.

Results

Figure 3 shows the results of the application of PSO-DR, PCA, tSNE, Isomap, and Sammon mapping on the MNIST dataset. It is revealed that PSO-DR algorithm is much faster than all the rest methods except for PCA. Moreover, the visualisation quality of the proposed approach is comparable or better that the majority of the DR methods apart from tSNE which is currently the best DR method for data visualization.

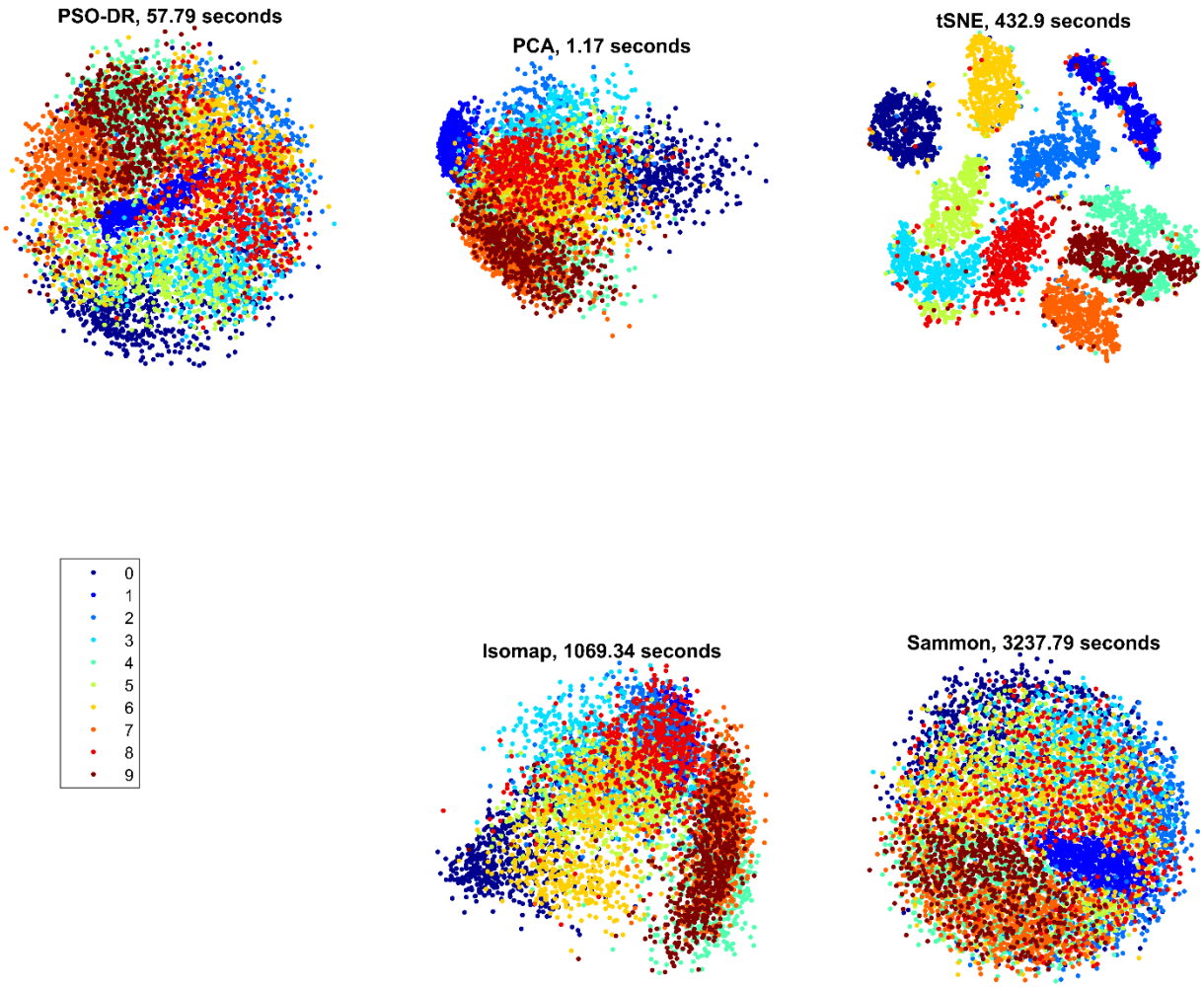


Figure 3: Visualization of the MNIST data set (6000 digits) using the PSO-DR, PCA, tSNE, Isomap, and Sammon mapping.

It is noteworthy that PSO-DR and Sammon mapping is constructing a similar ball with PSO-DR exhibiting a somewhat better discrimination between the different classes. The mapping of PCA and Isomap exhibit more extensive overlap between the classes.

Figure 4 presents the respective results for COIL-20 data set (labels 1-20 refer to each one of the 20 objects). Again PSO-DR is faster than tSNE and Sammon mapping. Nevertheless, the time differences are not of the same magnitude and, in addition, PSO-DR is now slightly slower than Isomap. This is due to the fact that COIL-20 has much fewer high dimensional points. Thus, PSO-DR's superiority in terms of fast computation is mostly revealed with very big data sets where the parallel way of computation for PSO-DR leads to fast estimation. Moreover, the similarity of PSO-DR and Sammon mapping is again observed. Nevertheless, Sammon mapping is almost 5 times slower than the PSO-DR approach.

Fig. 5 shows the results of the same DR methods on FMNIST data set. Similarly, PSO-DR is significantly faster from all the other approaches, except for PCA, with better quality of visualization from PCA and

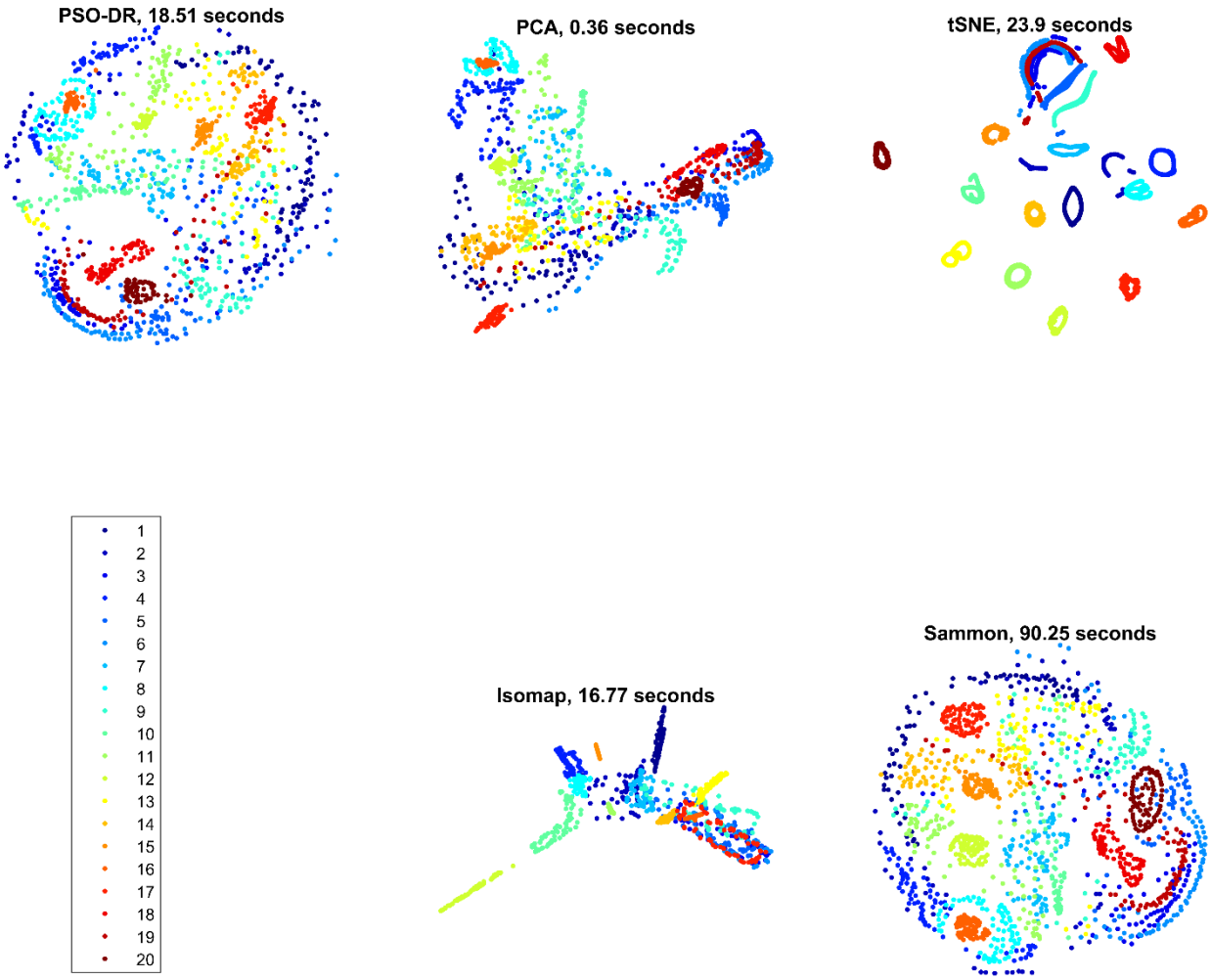


Figure 4: Visualization of the COIL-20 data set using the PSO-DR, PCA, tSNE, Isomap, and Sammon mapping (labels 1-20 refer to each one of the 20 objects).

Isomap, similar quality with Sammon mapping, and comparable visualization quality with the tSNE approach.

In Figure 6 the mappings of the Swiss Roll data set using the PSO-DR, PCA, Laplacian Eigenmap, Isomap, and LLE methods are presented. In this implementation of the PSO-DR method, instead of using the Euclidean distance as a measure for d_i, the geodesic distance as estimated also in Isomap approach is used. Thus, it is revealed that PSO-DR results in almost identical representation with Isomap whereas the rest of the state of the art approaches have poorer performance. The PSO-DR approach, though, is slightly slower than Isomap. Nevertheless, this is the case if only 6 cores are used for parallel implementation of PSO-DR. For the cases where more cores are available, it is possible to estimate PSO-DR much faster.

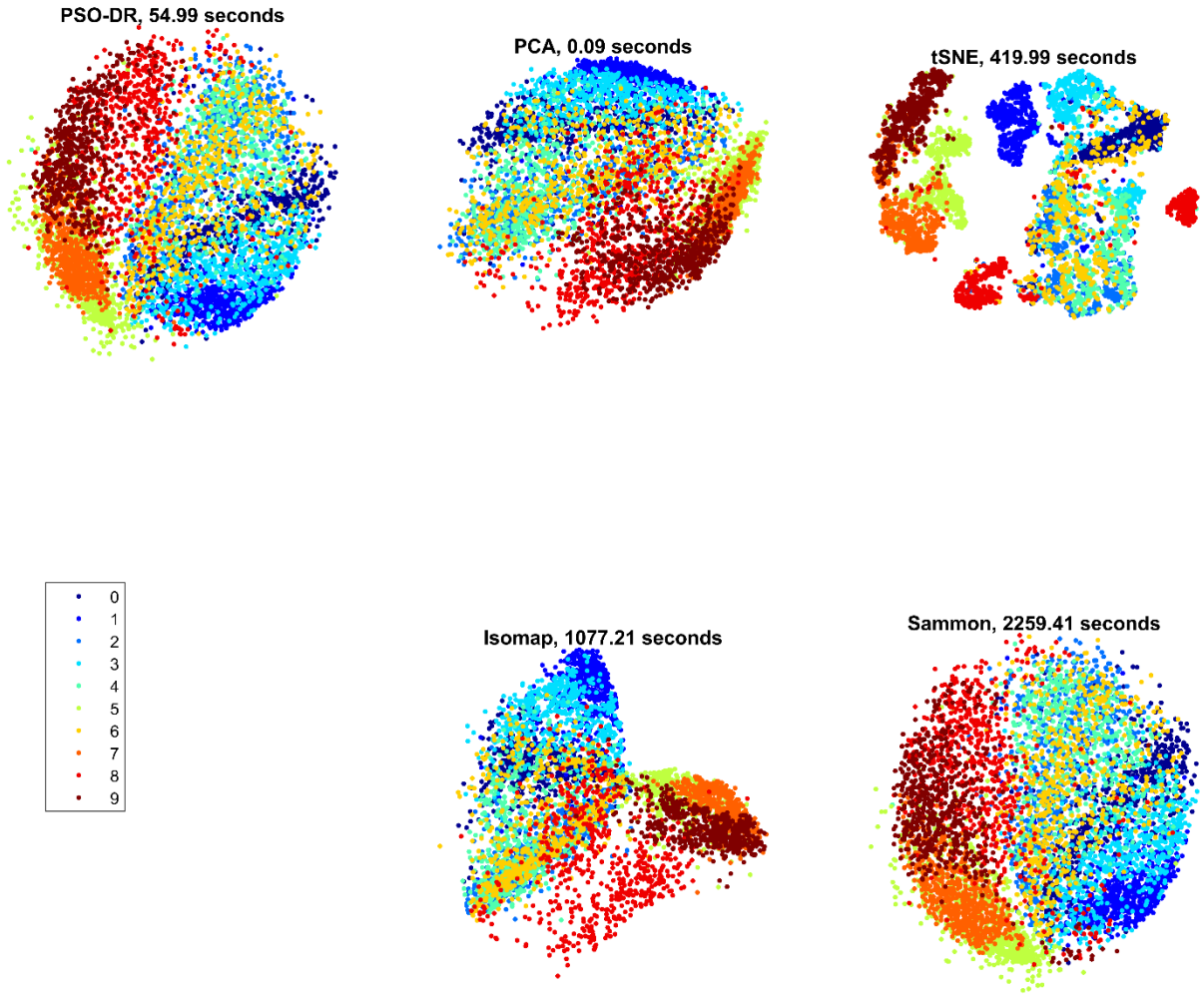


Figure 5: Visualization of the FMNIST data set (6,000 fashion items) using the PSO-DR, PCA, tSNE, Isomap, and Sammon mapping. (labeling was used for convenience, Labels 0 to 9 correspond to the 10 different fashion items).

It should be stressed out that if euclidean distance is used for the d_i measure for the case of the Swiss Roll data set, the performance of the PSO-DR approach is poorer. Nevertheless, the versatility of the proposed approach makes it possible to easily adjust it to the needs of the data set under consideration and select the dissimilarity measures of the input and output spaces accordingly.

3.1.6 Discussion and Future work

The experiments presented here demonstrate that PSO-DR is a simple, fast and versatile algorithm for DR for data visualization where multiple choices for distance measures both for d_i and δ_i are possible in a simple and straightforward way.

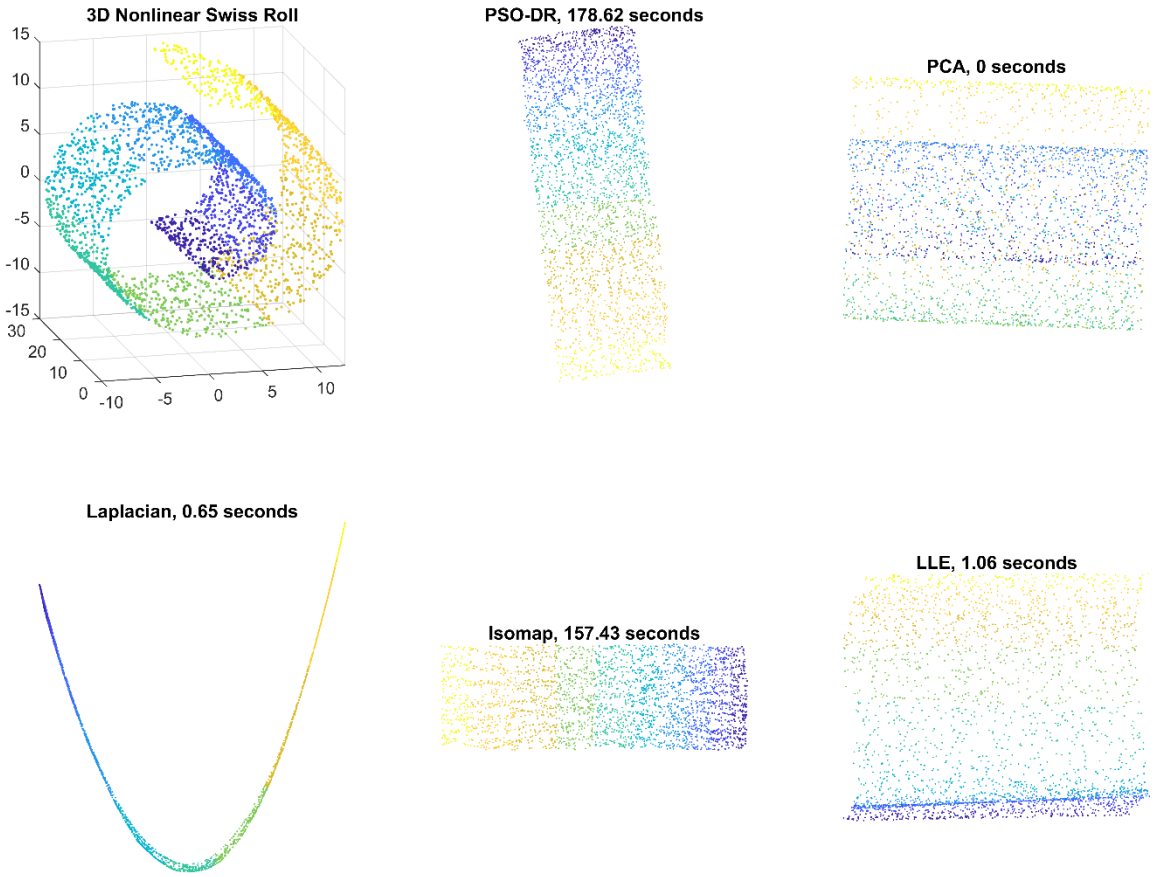


Figure 6: Visualization of the nonlinear Swiss Roll data set in 3D space and by using the PSO-DR, PCA, Laplacian Eigenmaps, Isomap, and LLE.

PSO-DR exhibits comparative or better visualization quality with the majority of the state of the art approaches that it was compared to. In essence, apart from tSNE, PSO-DR outperforms the rest of the approaches in terms of visualization quality. Nevertheless, tSNE is much slower than PSO-DR especially for large data sets. Moreover, for tSNE, as for many other non-linear DR approaches, out-of-sample extension is not straightforward (Van Der Maaten et al. 2009). On the contrary, the out-of-sample extension in PSO-DR is inherent in its functionality, as any new input can be mapped directly by comparing it with a fixed set of reference beacon-points.

In addition, the independence of the mapping of each high dimensional input from other inputs can favour the straight-forward parallel implementation of the algorithm. Thus, its computation time depends on and scales in inverse proportion with the number of parallel computation units (cores). Furthermore, the design of the PSO-DR algorithm makes it a good choice for ever increasing data sets, even with live streamed data points as it uses a specified set of beacons to compare the new data points and map them in the low dimensional space.

In essence, the only part of the PSO-DR algorithm where the mapping of an input depends on the previous inputs is the definition of the beacon set Y_b. As soon as this step is completed, PSO-DR mapping can be performed independently with regard to the different inputs. The alternative approach where all mappings depend on the previous inputs, thus, no beacons are defined but all x_i in X are mapped depending on the distance from all the previous inputs would result in slower computation without better visualization quality. Figure 7 shows the low dimensional mapping of MNIST data set based on such an approach. The result is comparable with the one shown in Fig. 1 whereas the time needed is multiple times larger than the one needed with the beacons approach.

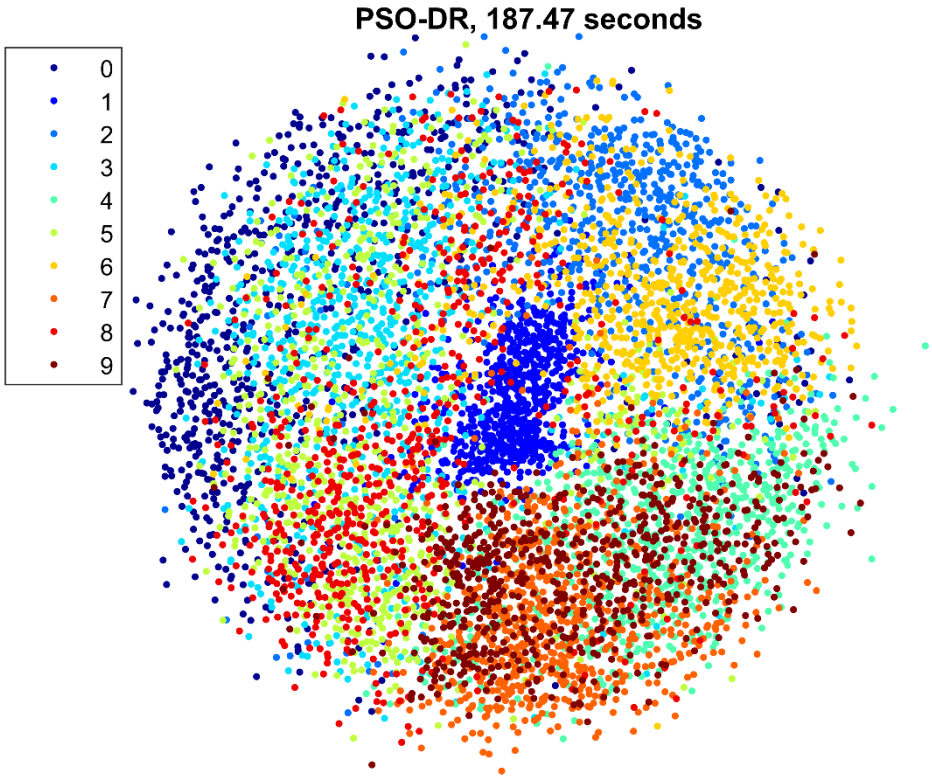


Figure 7: Visualization of the nonlinear Swiss Roll data set in 3D space and by using the PSO-DR, PCA, Laplacian Eigenmaps, Isomap, and LLE.

3.1.7 Conclusion

In sum, a new approach for dimensionality reduction is presented. The proposed approach leads to great representation results with low computational burden. Nevertheless, perceptible representation in a low dimensional haptic space demands more sophisticated discrimination methods. In order to, further enhance the haptic information that is conveyed to the HIPI user we also adopted a haptogram-based vocabulary approach, which is presented in the next section.

3.2 Perceptible haptic information

The HIPI is intended to convey perceptible signals to the users. Various constraints have been considered in the related design and development decisions. For example, the symbols conveyed by the HIPI must: i) be easily distinguished and perceived; ii) represent a simple language whose vocabulary is either similar to established vocabularies familiar to the user, or is intuitive and simple enough to learn; iii) enable sentence building; and iv) correspond with the messages that are relevant for HIPI communication. Thereby object and person recognition and situational awareness (WP3), as much as landmark identification (WP4) and gamification (WP7), were to be bridged by the effort reported below. In terms of simple haptic language use, SUITCEYES has been inspired by the Social Haptic Communication (SHC) initiative (e.g., Lahtinen 2008; Lahtinen *et al*. 2010), and accordingly, communication with the HIPI was informed both by WP6 constraints and existing SHC conventions (e.g., Danish Association of the Deafblind 2012; National Knowledge Center for Deafblind Issues 2020). With this background we started to cooperate on the co-design of haptograms among others with Riitta Lahtinen and Russ Palmer, based on their theoretical and practical work of 20 years in SHC signalling development.

Next we briefly discuss communication design considerations underlying proof-of-concept development; the role and kinds of semantics that were found to be relevant for haptogram design; and the aspect of incoming vs. outgoing messages. Our work was presented at the "SyMpATHY: SeMAntic Technologies for Healthcare and Accessibility Applications" workshop SUITCEYES organized at SEMAPRO-19[1] (Darányi *et al.* 2019).

3.2.1 Communication design considerations

Using the skin as a medium of communication is an actively explored interdisciplinary research topic.[2] While working out our own approach, we had to coordinate it with progress in ontology construction (WP3, see Section 4), itself delimited by multivariate algorithms underlying visual analysis (see sections 2 and 3); the specifics of actuators in testing (WP4); the technical limitations of smart textiles (WP5); and the psychophysical constraints to interpret signals on body parts as screens with different sensitivity levels, subject to parametrization according to spatial and temporal constraints (WP6). Further, project goals like navigation (WP4) and gamification (WP7) were to be integrated into the solution. With these in mind, we opted for haptograms at the receiver's end, and precoded questions (PQ) vs. precoded short messages (PSMS) coupled with haptograms for the sender's convenience. The haptogram kit in development is able to transmit visual analysis results labelled by the ontology, i.e. whatever concepts and relations it contains, plus sequences (statements) that can be generated therefrom. Below we first explain how haptograms and their sequences implement word and sentence meaning, then give specifics of the design process with examples.

1 https://www.iaria.org/conferences2019/ProgramSEMAPRO19.html

2 https://2020.hapticssymposium.org/program/ccc-theme-2-communication/

3.2.2 Semantic theories underlying haptogram design

To pin down word semantic vs. sentence semantic theories which can help one identify kinds of meaning implementable for deafblind communication, we took inspiration from Chinese *logograms* representing concepts, instead of characters that stand for speech sounds. Thereby a sequence of haptograms with symbolic content can establish sentence meaning.

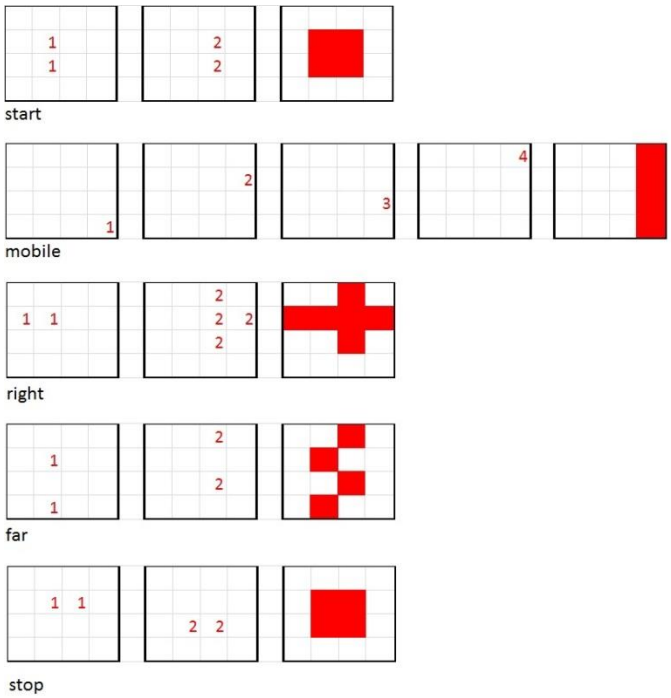
To conceive a set of actuator patterns, which correspond to units in a mental vocabulary, partly overlapping with ontology labels, Wierbiczka's *semantic primitives* (1996) and *semantic universals* (1997) were a relevant point of departure. These include elementary, archetypical concepts such as substantives (*you, I; someone, people; something*), mental predicates (*think, know, want, feel, see, hear*), descriptors (*big, small*), temporality (*when, after, before, a long time, a short time, now*), etc., while their concatenations can be related to Fodor's *Language of Thought hypothesis* (1975), a theory which describes the nature of thought as possessing "language-like" or compositional structure, with simple concepts combined in systematic ways (akin to the rules of grammar in language), so that in its most basic form, thought, like language, has syntax. Another aspect to connect semantic primitives as components of concepts with *haptemes* as components of touch (Lahtinen, 2008) originates in structural linguistics (Trubetzkoy, 1939), where we fell back on the interpretation that *phonemes* constitute an abstract underlying representation for segments of words. *Distributional semantics* (Harris, 1970; Firth, 1957; Wittgenstein, 1967) - responsible for meaningfulness in dimensional reduction methods and feeding forward to the ontology - adds another relevant semantic theory to the aforementioned, with statements derived from ontology labels by semantic reasoning going back to Carnap's *logical semantics* (1934). As an extended context, the very idea of *conceptual languages* has a history as well, to mention only Leibniz's *characteristica universalis* dating back to 1676 (Hintikka, 1997), or Frege's *Begriffschrift* (1879). Another umbrella term for the above in a deafblind context is the research field of *sign language semantics* (Zucchi, 2012). A series of relevant PhD theses augmented our resources to relate semantics mapped to the body in a sensory deprivation context (Ojala, 2011; McDaniel, 2012; Caporusso, 2012; Edwards, 2014).

3.2.3 Haptograms in a nutshell with examples

Coupled with ongoing ontology development to bridge visual and sensor analytics for situational awareness and navigation with semantic labelling of environmental cues, we designed a set of dynamic haptograms to represent entities and relations for communication between users with and without deafblindness and the HIPI, respectively. A haptogram corresponds to a tactile symbol drawn over a touchscreen, its dynamic nature referring to the act of writing or drawing, where the touchscreen can take several forms, including a smart textile garment designated for specific areas on the body. First test results with haptograms over a 3 x 3 grid were presented in D6.3. With other body parts as potential "screens" also in mind, our current haptogram set is generated over a 4 x 4 matrix of cells and is displayed

on the back of the user, to be tested for robustness and perceivability at the receiver's end.[3] The concepts and concept sequences stand for simple questions and answers to enable discourse generation.[4]

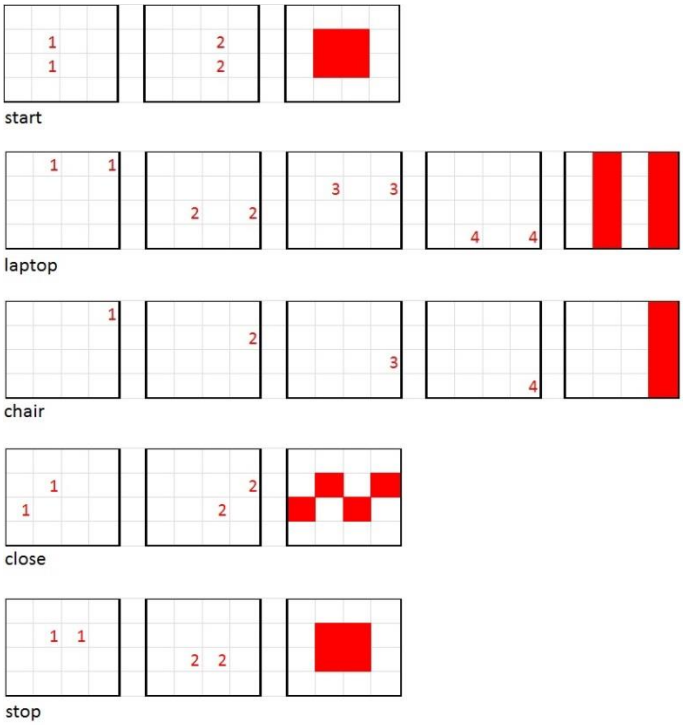
We focused on patterns by vibrational actuators only, so that the signs were composed of "tactile phonemes" to overlap with *haptemes* by Lahtinen (2008). Their list was as follows: location on body; single vs. multiple actuators activated, corresponding to single vs. multiple taps at the same time; movement; direction; speed; rhythm; duration; gap length between taps; gap length between haptograms; repetition; intensity of actuation. The above were used to compose a sample of 104 dynamic haptograms, and covered the following grammatical categories: nouns, verbs, adjectives, time and place adverbs, diacritics, sentence start/stop signals, sentence type markers, question words, alert signal, affirmation/negation for short responses to indicate agreement/disagreement, and personal pronouns. These were necessary to match ontology capabilities, and are describing 15 detected scenes, with persons known vs. unknown, and spatial relations like 'left/right', 'in front of you/not in front of you', or 'immediately close to/close to/far from you'. Figure 8 shows two system responses to predefined user queries (see also in Table 7 in Section 4), with patterns plus their unfolding sequences in red. The sign <> in the caption indicates variables with multiple values. In the Appendix more sample haptograms (for different objects and scenes), and example phrases are listed.



"Your <mobile> is located on your <right>, <far from> you."

[3] We dropped static haptograms because psychophysical tests in WP6 indicated that they are difficult to recognize on the back.

[4] Our use of the haptogram idea is different and independent from Haptogram as a trade mark (Korres and Eid 2016), and goes back to the use of tactons for visual content transmission (Jones and Ray 2008; Jones *et al*. 2009).



“A <laptop> and a <chair> have been found <very close> to you.”

Figure 8: System responses to predefined user queries

The encoding of haptograms into actuator patterns for a 4 x 4 grid is in progress. Typically, one size does not fit all: e.g. for the message, “There are <3> persons <close to> you”, the smallest grid where digital numbers 0-19 can be displayed is a 5 x 5 one (Figure 9). This means that with an expanding vocabulary, the grid size would be expected to increase as well.

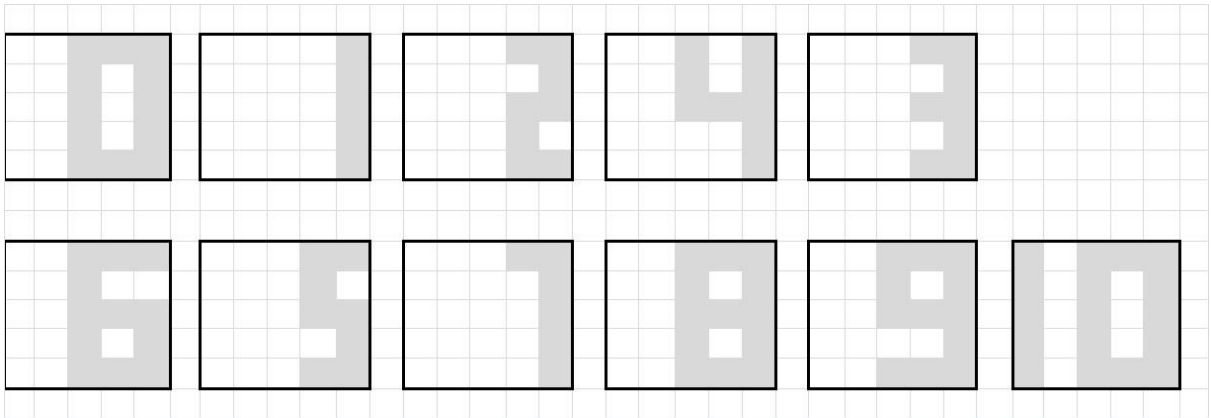


Figure 9: Standard digital numbers over a 5 x 5 grid

3.3 Chapter Summary and Future Work

A new DR algorithm and a haptogram-based approach for perceptible haptic information are presented in this chapter. In the next period of the project, we plan the following developments:

- Investigate further enhancement of the Visual Analysis tools by exploiting Dimensionality reduction algorithms.
- Based on feedback about the recognition rate of static, pulsating static and dynamic haptograms, WP6 will inform the co-design work with DB users to integrate their experiences in the updated version of the haptogram vocabulary. During this, we anneal our haptogram vocabulary to existing touch patterns in Social Haptic Communication by Lahtinen et al. (2010);
- The next version of the vocabulary, over a 5 x 5 actuator grid, will be drawn up and encoded;
- Experiments to implement precoded queries and short messages with integrated haptogram selection will be conducted.

4 Semantic Knowledge Representation and Reasoning

In the current section, we aim to describe the 2nd version (v2) of the implemented SUITCEYES ontology and the semantic reasoning framework. The ontology (KB, Knowledge Base) was extended to fit the needs of the users/project requirements (as those were defined in D2.1, D2.2) and to the advanced functionality of the HIPI, focusing mainly on those related to the increase of the context and location awareness of the DB-people Section 4.1. Moreover, the so called Knowledge Base Service (KBS) was created, which enables the communication and integration of data reported from both sensory- and analytics-components to the KB (Section 4.2). Finally, the Semantic Reasoning Mechanism (SRM) was enriched with ontology-based rules that serve specific queries for inferring the aforementioned context-related information, either in the form of structured content (JSON format) or as natural language phrases (Section 4.3). The latter formulated the basis for conveying dynamic haptograms to the end-user, as previously described in Section 3.2.

4.1 Additions to the SUITCEYES Ontology

The definitions of the second version of the SUITCEYES ontology target to integrate heterogeneous, multimodal input from different sensors in a formal and semantically enriched basis, and thus to combine user's context-related information so as to provide enhanced situational awareness that can potentially augment users' navigation and communication capabilities. The initial ontological schema documented in D3.1 was extended on the basis of two main directions: (a) to **augment its semantic interoperability** with other ontologies that exist in the domains of interest, (b) to **enrich its semantic representation** capabilities for covering the additional concepts and functional requirements that emerged throughout the advance of the overall HIPI system.

More specifically, regarding the adoption of third-party vocabularies, a summary of all imported ontologies is presented in Table 4. Both the Dem@care and the SEAS Building Ontology were utilized in order to enrich the context and spatial reference of entities existing around the HIPI user. The notion of `Object` was mainly adopted and extended from the Dem@care ontology, representing various objects that could be of interest for daily use (e.g. mug, plate, toothbrush, furniture, window, door, etc.), all inspired by real case scenarios. Moreover, the notion of Room was adopted and extended from both the Dem@care and the SEAS Building Ontology, representing space, buildings and rooms, for referencing the HIPI user within an area of interest. The hierarchy of concepts related to rooms and space, adopted from both imported ontologies, are visualized in Figure 10. They latter set of concepts is grouped under the notion of `sot:SemanticSpace`, where "`sot`" is the prefix for the SUITCEYES ontology.

Table 4: A list of third-party ontologies utilised and extended within the SUITCEYES ontology

prefix	Ontology	URL	Concepts	Imported in SUITCEYES version
foaf	Friend-Of-A-Friend	http://xmlns.com/foaf/spec/	`Person` and its asserted properties	v1
sosa	Semantic Sensor Network	https://www.w3.org/TR/vocab-ssn/	`Sensor` and its asserted properties	v1

dem	Dem@Care	http://www.demcare.eu/results/ontologies	Activity Object Room and their asserted properties	v1 v1 v2
seas	Smart Energy Aware Systems Building Ontology	https://ci.mines-stetienne.fr/seas/BuildingOntology-1.0	BuildingSpace Room and their asserted properties	v2 v2

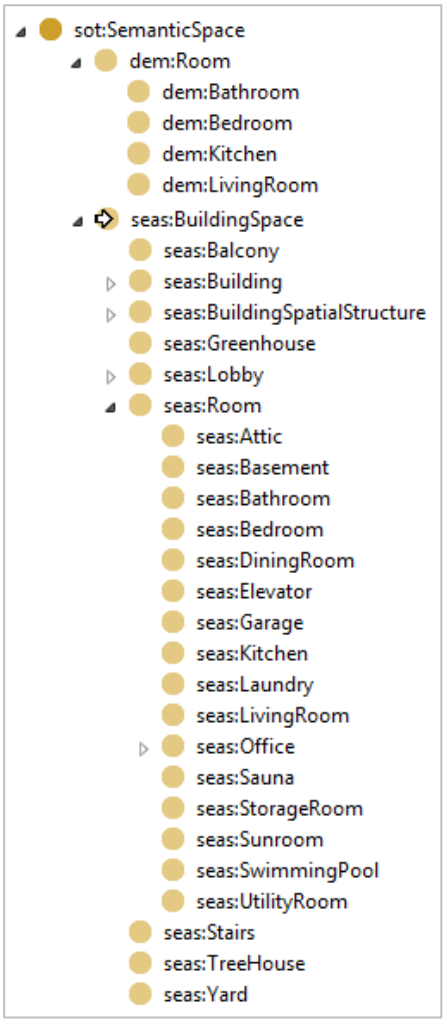


Figure 10: Hierarchy of concepts related to semantic space, which were adopted in the SUITCEYES Ontology from the Dem@Care and the SEAS Building Ontologies

Regarding the enrichment of semantic representation capabilities, we extended the SUITCEYES ontology with additional classes and properties, for handling both the **data integration** and **reasoning** tasks. As already presented in D2.1 and D2.2, the HIPI must be capable of delivering semantic content to the user, with respect to his/her physical surroundings. In the current implementation of the overall SUTICEYES system, the raw measurements as well as high-level analytics about detected entities are derived from the sensor system (D4.1) and the visual analysis (VA) component (Section 2) respectively. The aforementioned data are semantically annotated in the KB, in order to be fused properly, producing next a higher level interpretation of the combined information.

Thus, in order to define the *representational* and *technical* background within the ontology that cover the efficient communication and interpretation of data sourced from the different SUITCEYES components, we have extended the SUITCEYES ontology as presented in the following subsections. It should be noted that the relevant diagrams are based on the Grafoo ontology visualization notation (Falco et al., 2014), on the basis of which the yellow rectangles represent classes, while the green ones represent data properties (i.e., properties that take a raw data value, like, e.g., integers and strings). The prefixes in front of some of the class names indicate the namespace of the respective third-party ontologies, as mentioned above. Classes and properties that have no prefix belong to the core SUITCEYES ontology.

4.1.1 Representing detections

The class `Detection` is fundamental within the context of the SUITCEYES ontology and refers to environmental cues (detected by the sensors) that have been instantiated in the KB. An instance of type `Detection` may be associated with the relevant sensor(s) that provide data to the ontology, via the property `providedBy`. Currently, there are two specific categorizations of the class `Sensor` (i.e., `Camera` and `iBeacon`) which are related to the relevant operational sensors attached to the HIPI and provide data to the KB. On the basis of the sensors' data, an instance of `Detection` class can be associated with one or more instances of type `Person` (known or unknown persons), `Object`, `Activity`, and/or `SemanticSpace` (i.e., rooms, building spaces, as previously described); the relevant assertion is achieved via the property `detects`. On the basis of the semantic reasoning mechanism (details in Section 4.3), higher-level results that combine incoming data from all sensors, are produced and represented in the ontology via the class `Output`. A detailed representation of the aforementioned concepts and relations are visualized in Figure 11.

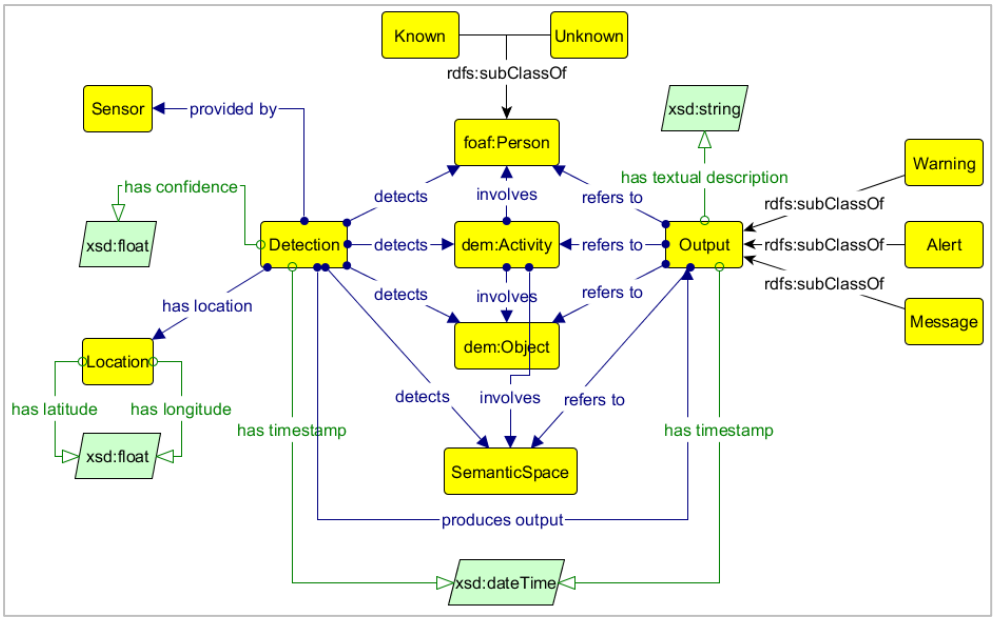


Figure 11: Overview of the core classes of the SUITCEYES ontology v2

4.1.2 Representing spatial relations

Topological relations of geometric objects have been widely described in literature and are generally utilized for navigation-, location- and context-based services. More specifically, the *Egenhofer relations* (Egenhofer, 1989) or the *DE-9IM* topological model (Clementini et al., 1994) can be used to specify how an object is located in space in relation to some reference object. For any two spatial objects, which can be points, lines and/or polygonal areas (represented by the definition of a bounding box), there are 9 relations derived from the model, which are: *equals*, *disjoint*, *intersects*, *touches*, *contains*, *convers*, *covered* by, and *within*. Distinct specializations of topological relations also exist, such as the so called **alignment relations** (*horizontally aligned* or *vertically aligned*) and **orientation relations** (*left of*, *right of*, *top of* and *bottom of*); these are considered in literature as mereology and parthood relations, as described in detail in (Varzi, 2007) and visualized in Figure 12 from different perspectives (object-centered vs observer-centered).

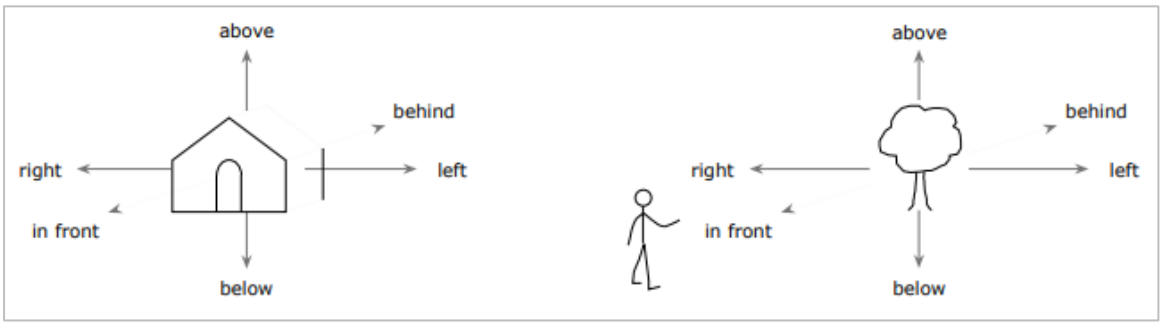


Figure 12: Object-centered (left) and observer-centered (right) frames of reference

Driven by the nature of the reported data from the involved sensors in the SUITCEYES framework, and more specifically from the two main components that are operational in the current HIPI installation, i.e., the **Visual Analysis** (VA) module and the **iBeacon** sensors, we focus on specific spatial relations that have to do with the orientation (left/right), existence (in a room) and the distance (far/close/immediate). Thus, in the SUITCEYES ontology, an entity that occupies space (e.g., persons, objects) is considered as a `SpatialEntity` and the occupied space (e.g., a room or a location) belongs to the `SemanticSpace` representation. These two aspects formulate the respective entity's `Spatial Context`, which provides information regarding the entity's relationship to the semantic space it is located in. Examples include: `in`, on, `left`, `right`, `far`, `close`, etc. The aforementioned concepts are depicted in Figure 13.

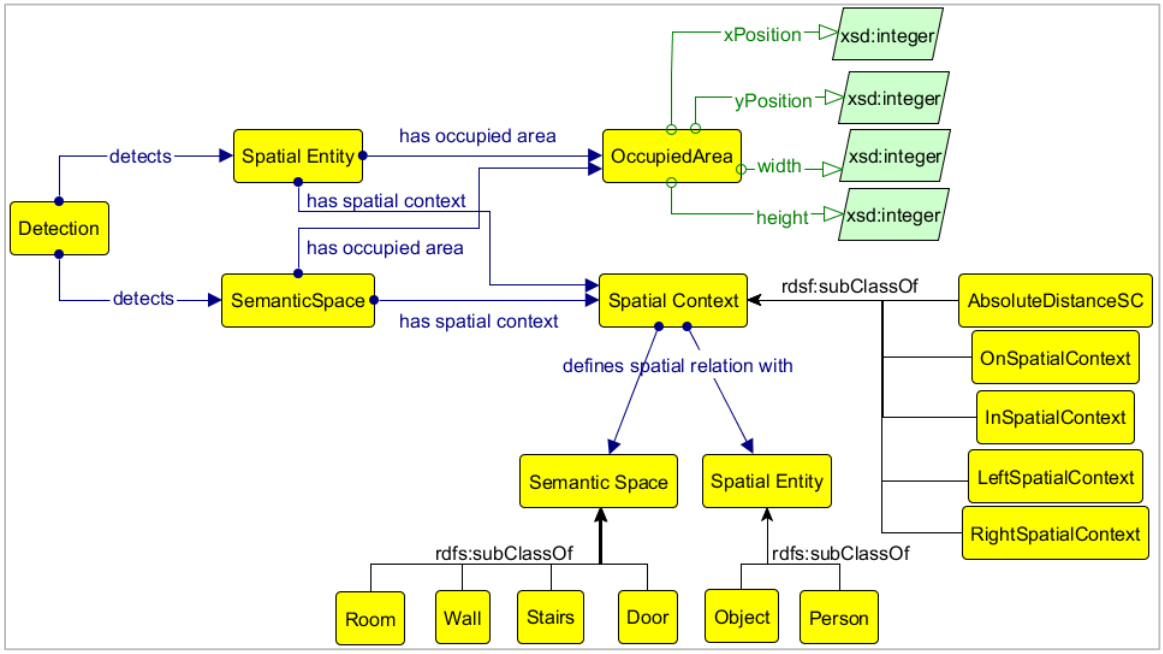


Figure 13: Semantic spaces and spatial contexts in the SUITCEYES ontology

These definitions play a key-role in the semantic reasoning mechanism described in Section 4.3, as they form the basis for inferring spatially-related information to the user, like for example, which objects are close to or far from the user, what short of entities are located in the room where the user is, etc.

4.1.3 Sample instantiation

Based on the ontological concepts already presented, Figure 14 illustrates a sample instantiation resembling an activity detected by the system's camera. The activity involves two people speaking to each other, one of them is known to the user (i.e., `john`) and the other is unknown. Moreover, these two people are currently located in the kitchen (i.e., `in_room_spatial_context`), and the respective message (`output_1`)[5] is communicated to the user.

[5] The message is related to the relevant rule that is triggered by the HIPI system. Details about the output messages are given in Section 4.3.

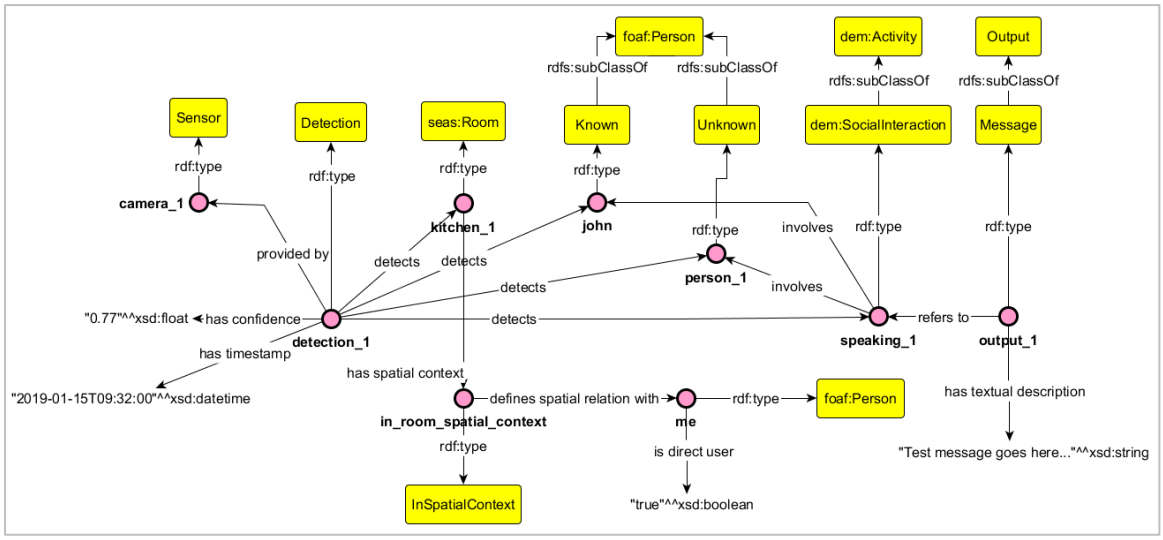


Figure 14: Sample instantiation of `Detection` class, representing the detection of two people discussing in the kitchen

4.2 Semantic Integration

Semantic integration is the process of interrelating information from diverse sources. Within the context of the project, semantic integration is critical in order to integrate the valuable information from the different sensors attached to the HIPI and to infer implicit knowledge to the DB-user, on the basis of the asserted facts. As already mentioned, raw data originate from the different sensors attached to the HIPI and from the different components communicated for further processing. The different SUITCEYES sensors/components that are linked to the reporting, analysis and transmission of data interact with the KB via the Knowledge Base Service (KBS). The latter can be conceived as the interface to the ontology that handles the incoming messages and integrates properly the information to the semantic model. The KBS also receives output from the semantic reasoning process (inference) running on top of the KB and forwards the inferred, high-level knowledge back to other interested system modules, like for example to the actuators. The interaction between the KB and the KBS is established with the use of proper ontology-based queries (SPARQL/SPIN), which can insert/delete/fetch relevant data from the components to the KB and vice versa. The communication between the KB, the KBS and the different components and sensors is depicted in Figure 15.

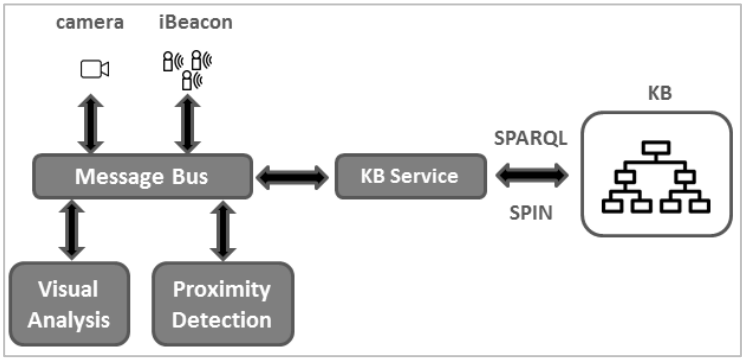


Figure 15: Interaction of KB, KBS and the different HIPI modules

Thus, during the reporting period, we proceeded with the implementation of the communication framework between the ontology and the operational SUITCEYES components (i.e., *Visual Analysis* and *Proximity Detection*) through the implementation of the relevant Knowledge Base Service (KBS). The latter, was built by utilising state-of-the-art semantic web technologies and tools, as described below:

- **GraphDB**[6] - a popular graph database (triplestore) for hosting the operational ontology and serving queries as a SPARQL endpoint. It can be considered as the functional base of the KBS.
- **Java 1.7 EE**[7] and **RDF4j**[8] - the well-known programming language for implementing the service, and a Java framework for processing, handling and querying RDF data.
- **SPARQL** (Harris & Seaborne, 2013) and **SPIN** (Knublauch et al., 2011) - the semantic query language for submitting (insert/delete/update/fetch) queries to the ontology and running rules on top of the model.
- **Realtime.co**[9] - a messaging framework for exchanging real-time messages between the KBS and the aforementioned components.
- **JSON** - a text-based data interchange format that represents its content as simple key-value pairs, giving emphasis in both the content and the structuring of data.
- **json-simple** - a java toolkit for parsing (encoding/decoding) JSON text.

The implemented KBS has the following **functional characteristics**:

- Communication has been already established between the following operational components: (i) from the VA to the KBS, (ii) from the iBeacon to the KBS, and (iii) from a set of 6 actuators attached to the HIPI for experimental reasons.
- The incoming messages to the KBS sourced from the different components, have a **predefined structure** (examples are given in Table 5). This enables the straightforward mapping of the fields and values given in the JSON messages as triples in the ontology. An example instance of Detection type as populated in the ontology, representing a specific detection reported from the VA tool, is visualised in Figure 16, as exported from the GraphDB repository.
- The KBS creates a pool of incoming messages, as reports from the different components may arrive **simultaneously**, even at the time that another message is served (populated in the ontology). This ensures that all messages are handled without losing any content, keeping at the same time their initial order, on the basis of the **FIFO** (first-in-first-out) method.
- The population process (i.e., the creation of relevant instances in the ontology for representing data reported in the incoming messages) is **near real-time**. In a set of experiments conducted for the preparation of the first Demonstration of the SUITCEYES platform to the reviewers (M18 of the project), the typical response times of the KBS for the population process ranged approximately 150-220ms for detection(s) reported in each message provided by the VA component, and 100-150ms for detection(s) reported in each message provided by the iBeacon component.

[6] http://graphdb.ontotext.com/
[7] https://www.oracle.com/java/technologies/java-ee-sdk-7-download.html
[8] https://rdf4j.eclipse.org/
[9] https://framework.realtime.co/messaging/

Table 5: JSON structure of incoming messages from different components

(a) incoming message from the VA component


```
{
  "header": {
    "timestamp": "2019-08-22T11:42:31.154",
    "message_id": "VA_5642560d17dc424da4043a428aff0bb8",
    "recipients": [
      "KBS"
    ],
    "sender": "VA"
  },
  "body": {
    "data":
"http://160.40.50.243:8008/upload/storage/VA_5642560d17dc424da4043a428aff0bb8.json"
  }
}
```


(b) message structuring analysed data from the VA, stored in the provided URL


```
{
  "image": {
    "target": [
      {
        "top": 429,
        "distance": -1,
        "type": "chair",
        "confidence": 0.925,
        "width": 187,
        "left": 224,
        "height": 200
      },
      {
        "top": 424,
        "distance": -1,
        "type": "table",
        "confidence": 0.578,
        "width": 189,
        "left": 2,
        "height": 213
      }
    ],
    "scene_type": "kitchen",
    "scene_score": 0.939,
    "timestamp": "2019-08-22T11:42:28.146",
    "width": 480,
    "name": "22_08_19_11_42_28_146",
    "height": 640
  }
}
```


(c) incoming message from the iBeacon sensor


```
{
  "timestamp": "22-08-19 11:42:30",
  "data": [
    {
      "distance": 6390.0,
      "positionX": 0.0,
      "positionY": 0.0,
      "name": "chair",
      "rssi": 187,
      "message": "Near",
      "id": 5
    },
    {
      "distance": 4582.0,
```

```
      "positionX": 0.0,
      "positionY": 0.0,
      "name": "phone",
      "rssi": 191,
      "message": "Near",
      "id": 1
    },
    {
      "distance": 2837.0,
      "positionX": 0.0,
      "positionY": 0.0,
      "name": "laptop",
      "rssi": 197,
      "message": "Immediate",
      "id": 3
    },
    {
      "distance": 2322.0,
      "positionX": 0.0,
      "positionY": 0.0,
      "name": "table",
      "rssi": 200,
      "message": "Immediate",
      "id": 2
    }
  ]
}
```


	subject	predicate	object
1	sot:detection_898f1fa0-9b68-43a2-bd31-b2c0dfa8af9e	sot:detectsObject	sot:computer_b97c193c-c670-4ca8-bf42-12b451ef2108
2	sot:detection_898f1fa0-9b68-43a2-bd31-b2c0dfa8af9e	sot:detectsObject	sot:table_91faf16a-6468-4014-8ba8-7afd8c3ed431
3	sot:detection_898f1fa0-9b68-43a2-bd31-b2c0dfa8af9e	sot:detectsSemanticSpace	sot:scene_unknown
4	sot:detection_898f1fa0-9b68-43a2-bd31-b2c0dfa8af9e	sot:hasConfidence	"0.0"^^xsd:float
5	sot:detection_898f1fa0-9b68-43a2-bd31-b2c0dfa8af9e	sot:hasTimestamp	"2019-12-13T12:45:50.619"^^xsd:dateTime
6	sot:detection_898f1fa0-9b68-43a2-bd31-b2c0dfa8af9e	sot:locatedInGridSpace	sot:detection_grid_space_ed44be67-a75d-460b-8e30-664104e53080
7	sot:detection_898f1fa0-9b68-43a2-bd31-b2c0dfa8af9e	sot:providedBy	sot:camera
8	sot:detection_898f1fa0-9b68-43a2-bd31-b2c0dfa8af9e	rdf:type	sot:Detection

Figure 16: An example set of triples populated for a detection result provided by the VA component

4.3 Semantic Reasoning

As already presented in D3.1, the term semantic reasoning refers to the process of deriving facts that are not explicitly expressed in an ontology. A semantic reasoner is a piece of software able to infer logical

consequences from a set of asserted facts, axioms or rules that are integrated or aligned with the ontology. In the 2nd version of the SUITCEYES ontology, we have implemented ontological **rules** and **queries**, which support the conversion of the available low-level definitions (data originating from the operational SUITCEYES components, i.e. the Visual Analysis and the iBeacon) to higher-level implicit knowledge (messages transmitted to the HIPI user through the actuators, i.e., the Haptic Signal Generator). These inferences were designed on the basis of specific scenarios derived from the user and technical requirements. The semantic reasoning scenarios that the ontology may currently address can be divided in two main types: continuous search and static search, which are described in detail in the following subsections (4.3.1 and 4.3.2).

4.3.1 Continuous search queries

This type of reasoning regards a **set of ontological queries** that run as a thread (continuously) and monitor/report details about the detection of a specific entity of interest. The user may trigger for example a search for an object or a known person and the thread continuously reports the latest detections from the two operational components of the HIPI. Reporting is *differential*, which means that information is inferred to the user only if a value changes, when compared to the previously reported value. Information that is reported regards *TRUE/FALSE* values of the corresponding detection status of each component, and also *distance* categorical values (far, close, immediate). The scenario requires the user to move around the space so that the state of the detections changes (i.e., VA *not found*, iBeacon *found* the entity of interest) and the user can support his/her decision in which direction to follow next so as to reach the entity of interest. The thread runs until the entity is found from both involved components (Visual Analysis, iBeacon). Example queries (in free text) can be considered the following: "*Can you find me my <mug>*[10]*?*". The aforementioned type of search involves the implementation and triggering of proper **SPARQL queries**[11] **at the KBS level**, with the use of Java programming language and with the utilisation of GraphDB platform for populating, querying and fetching the relevant information stored in the ontology. An indicative SPARQL query implemented in Java is presented in Figure 17.

[10] the symbols "<>" are used to define that the content included within is dynamic and may be referred to any other related entity (object, person) that is be supported by the detection/sensor components.
[11] https://www.w3.org/TR/rdf-sparql-query/

```
String queryString =
        "PREFIX sot: <" + Vocabulary.NAMESPACE_TBOX + "> "
    +   "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> "
    +   "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "
    +   "PREFIX dem: <http://www.demcare.eu/ontologies/event.owl#> "
    +   "SELECT DISTINCT ?detection ?timestamp ?entity ?distance ?xGridPosition ?semanticLabel "
    +   "WHERE {"
    +       "?detection a sot:Detection . "
    +       "?detection sot:hasTimestamp ?timestamp . "
    +       "?detection sot:providedBy ?sensor . "
    +       "?sensor rdf:type sot:Camera . "

    +       "?detection sot:detectsObject ?entity . "
    +       "?entity rdf:type sot:Object . "
    +       "?entity rdf:type " + "<" + Vocabulary.getClassURIOfObjectType(entityName) + "> . " // e.g. "?object rdf:type dem:Cup .

    +       "?entity sot:hasSpatialContext ?spatialContext . "
    +       "?spatialContext sot:definesAbsoluteDistance ?distance . "
    +       "?entity sot:hasXGridSpace ?xGridSpace . "
    +       "?xGridSpace sot:positionedInXGrid ?xGridPosition . "

    +       "?detection sot:detectsSemanticSpace ?semanticSpace . "
    +       "?semanticSpace rdf:type sot:SemanticSpace . "
    +       "?semanticSpace rdfs:label ?semanticLabel . "
    +   "} "
    +   "ORDER BY DESC(?timestamp) "
    +   "LIMIT 1 "
    ;
```

Figure 17: Indicative SPARQL query implemented in Java, for retrieving the relevant details for a specific entity detection provided by the VA component.

For visualising the continuous process followed within the aforementioned type of inference, we present in Figure 18 the flowchart that describes the steps followed in every single iteration of the search loop. The process starts when a message arrives at the message bus for the KBS. Data are populated within the ontology, by initialising the relevant concepts in the ontology. Then, the reasoning mechanism continuously searches for the latest detections from the two involved SUITCEYES reporting components, the VA and the iBeacons. The process compares their reported values with respect to the latest reported values to the user and acts accordingly; if, for example, the user was informed in a previous iteration of the loop that the VA did not detect the requested entity, then the status remains the same and no new info is delivered to the user. On the basis of the possible TRUE/FALSE combinations, a relevant result is produced for the user, as seen in the bottom of the flowchart for an example case where the user is searching for a specific person.

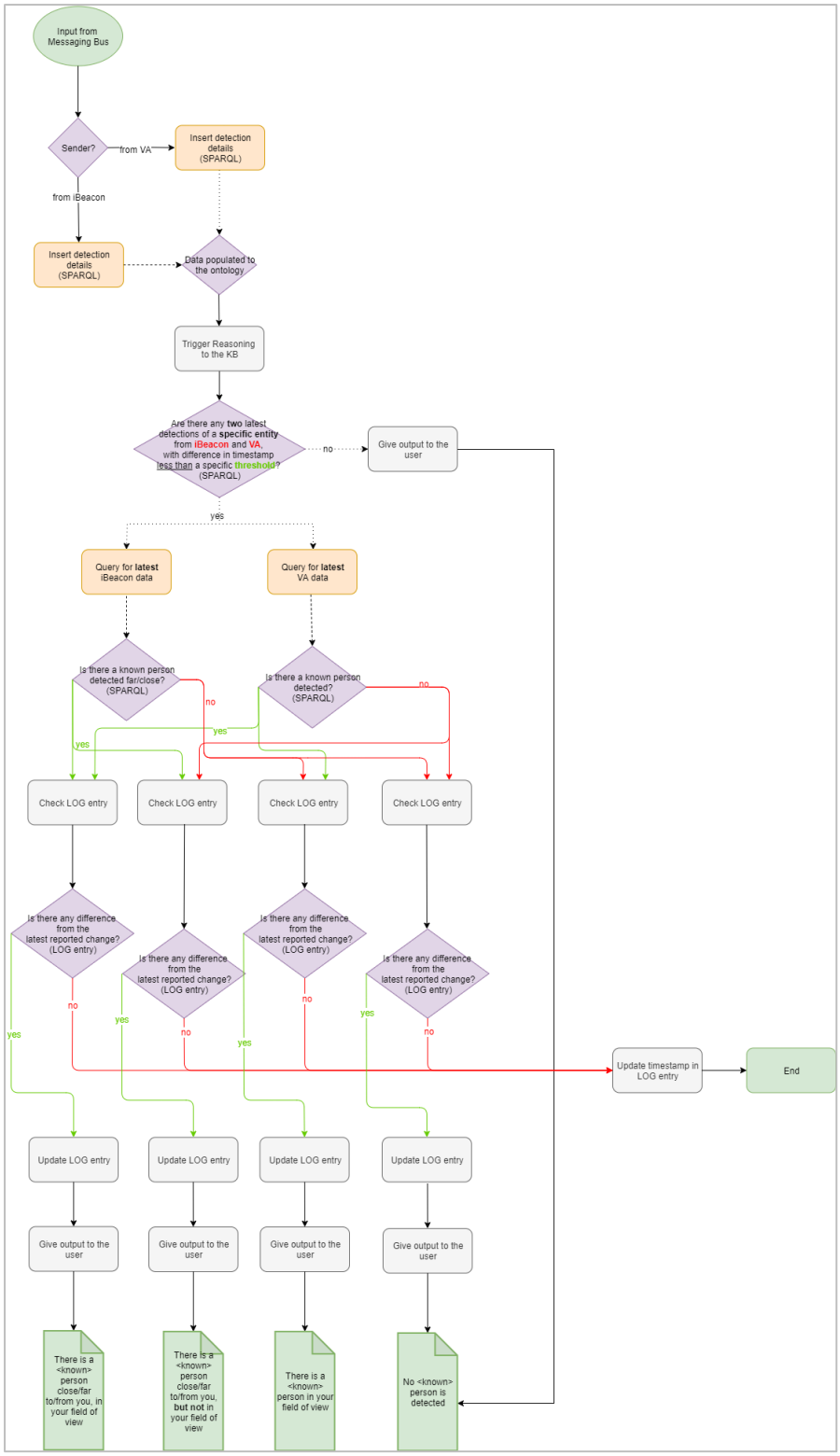


Figure 18: Flowchart of the process followed when a message for the KBS arrives at the message bus

4.3.2 Static search queries

This type of reasoning regards a **set of ontological rules** that run on top of the ontology, whenever a specific query is triggered by the user. There is no continuous nature in this type of search; on the contrary, the inference is completed at once, and the answer is given on the basis of the current status reported by the detection/sensor modules. Within the context of this scenario, we have created a list of indicative queries that can dynamically change on specific aspects, i.e., to cover different entities of interest, different spatial relations (with respect to the distance or position-left/right), etc. Example queries (in free text) can be considered the following: "*Which objects are observed on my left hand side?*", "*How many <persons> have been detected close to me?*", "*When (timestamp) and where (room) was my <phone> last detected?*", "*In which room am I now located?*", and other similar questions. Inferences with respect to the orientation (left/right) side are based on the bounding boxes of detected entities, as positioned within the overall image by the VA tool, while inferences that are related to the notion of distance are based on reports from both the VA and iBeacon components. The aforementioned type of search involves the implementation of proper **SPIN rules**[12] **at the KB level**, with the utilisation of GraphDB platform for inferring the new knowledge derived from the rule-based reasoning mechanism. Examples SPIN rules and results are presented in detail in Section 4.3.3.

4.3.3 Defining the output format

The developed reasoning mechanism can produce the derived inferences as an output in specific format, with respect to the needs and the current implementations of the project. Inferences are firstly produced within the ontology in the form of triples; once the relevant queries are served, results are populated in the ontology as a set of statements in the form of <*subject predicate object*>. This output can be then adjusted and transposed according to the requirements of the final receivers. A schematic representation of the different types of the delivered output is depicted in Figure 19.

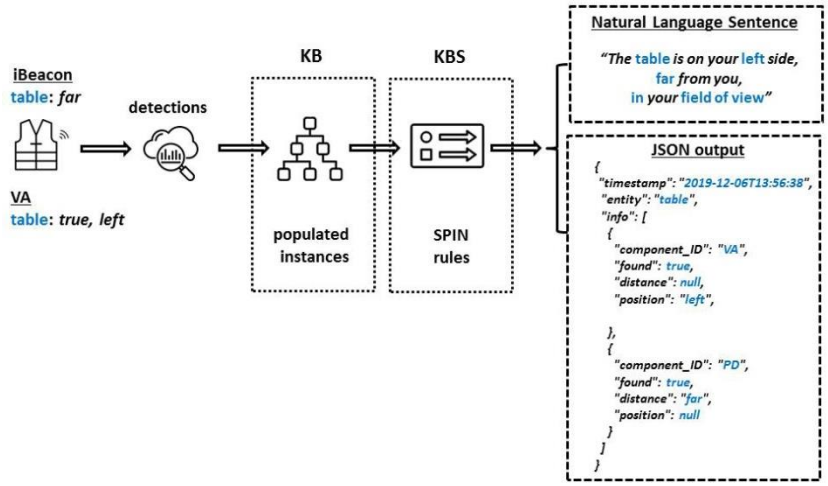


Figure 19: Schematic representation of the KB/KBS input and output provided

[12] https://spinrdf.org/

More specifically, firstly, for demonstration purposes, the need was to deliver the derived content to a custom vibro-tactile belt, which has been implemented as an example of haptic feedback interface within the scope of WP4 (D4.2). Communication between the ontology and the actuators attached to the vibro-tactile was made on the basis of the commonly agreed JSON format, via the messaging bus. As an example, in Table 6, a **JSON message** that reports that a *chair* was detected from the iBeacon[13] component and the distance was estimated as *immediate*, while the VA component could not find it. This content is then properly interpreted by the relevant haptic feedback interface and a proper direction (*move forward, turn around, stop – found, continue – not found*) is communicated to the user, on the basis of its implementation. This process is described in detail in D4.2 (Section 3.7). For the aforementioned example, the actuators will vibrate in such a way in order to direct the user to move around, since the PD reports that the entity of interest is very close to the user, but he/she is still not facing it (not in the field of view of the camera).

Table 6: JSON structure of outsourced messages from the KB to the haptic feedback interface

JSON message from the KB component

```
{
  "timestamp": "2019-08-09T13:56:38",
  "entity": "chair",
  "info": [
    {
      "component_ID": "VA",
      "found": false,
      "distance": null,
      "position": "null",

    },
    {
      "component_ID": "PD",
      "found": true,
      "distance": "immediate",
      "position": null
    }
  ]
}
```

Additionally, as already reported in D2.1, there is a requirement that the HIPI must be capable of delivering semantic content to the user through haptic media. This requires that information delivered from the ontology is structured into a message, as a **natural language phrase** - rather than delivering a simple alert, or indication. This information will be then communicated to the DB-user via haptograms (already described in Section 3.2) by aligning each word or set of words included in the sentence with the relevant haptogram representation. In order to convert the inferred triples to natural language phrases, we have implemented a set of SPIN rules that may serve queries like the indicative ones presented in table below.

Table 7: Example queries, for which the ontology can produce results in natural language text

#	Query	Answer
Q1	Where is my <phone> now?	Your <phone> is located on your <left>, <close to> you.
Q2	In which room am I now located?	You are located at the <kitchen>.
Q3	When and where was my <mug> detected for the last time?	Your <mug> was found at the <living room>, <10 seconds> ago.

[13] iBeacon sensors are considered as part of the PD-Proximity Detection component.

Q4	Which <entity> is observed on my <left> side?	A <table> is on your <left> side.
Q5	Which are the objects I am <closer to>?	A <laptop> and a <chair> have been found <very close> to you.
Q6	How many persons have been detected <close to> me?	There are <3> persons <close to> you.
Q7	Are there any <known> persons detected?	<John> is detected <far from> you.

Two were the most important SPIN modules (W3C, 2014) that were adopted for the aforementioned task:

a) **SPIN rules** - The built-in class named `spin:rule`[14] can be used to specify inference rules using SPARQL CONSTRUCT and DELETE/INSERT, in order to create new triples in the ontology and thus enrich the knowledge stored in the schema. Each implemented SPIN rule defines an inference rule that describes how additional triples can be inferred from what is stated in the WHERE clause of the rule.

b) **SPIN functions** - The built-in class named `spin:Function` can be used to isolate common SPARQL patterns, facilitating this way the decongestion, reuse or extension of SPARQL blocks. For more flexibility, they can be parameterised, by specifying input arguments (parameters/variables). In order to use the SPIN Functions module, we create subclasses of the class `spin:Function`.

In practice, we present in Table 8a set of specific SPIN modules combined in order to report with a dynamically created sentence, the *latest scene detection* populated in the ontology, serving queries similar to Q2 (Table 7). In Table 8a, we present the SPARQL SELECT query called so as to order (from latest to oldest) all populated instances of type `Detection` which are associated to detections of semantic spaces (`seas:Room`) and return only the latest one; next, in Table 8b we present a SPARQL CONSTRUCT query for creating an instance of `Output` type, which is asserted to the latest detection of a semantic space. For the aforementioned instance, a textual description is produced dynamically, for any detected semantic space, for which a relevant space label is defined in the ontology.

Table 8: SPIN function and rule for inferring knowledge relevant to the latest scene detection

(a) SPIN function: infer latest space detection

```
SELECT DISTINCT ?detection
WHERE {
   ?detection a sot:Detection .
   ?detection sot:hasTimestamp ?timestamp .
   ?detection sot:detectsSemanticSpace ?space .
   ?space a seas:Room .
}
ORDER BY DESC (?timestamp)
LIMIT 1
```

(b) SPIN rule: create output for latest space detection

[14] spin: is the prefix of URI http://spinrdf.org/spin#

```
CONSTRUCT {
   ?detection sot:producesOutput ?output .
   ?output sot:refersToDetection ?detection .
   ?output a sot:Output .
   ?output sot:hasTextualDescription ?description .
}
WHERE {
   BIND (sospin:Function_SelectLatestSceneDetection() AS ?detection) .
   ?detection a sot:Detection .
   ?detection sot:detectsSemanticSpace ?space .
   ?space a seas:Room .
   ?space rdfs:label ?space_label .
   BIND (BNODE() AS ?output) .
   BIND (CONCAT("You are located at the ", ?space_label) AS ?description) .
}
```

Similarly, in Table 9 we combine a different set of functions and rule to report the *latest known person* detected on the left or right side of the user (Q4 in Table 9). Output results are inferred dynamically, whenever the conditions of the examined rules are satisfied. The same rules may produce results regardless the left/right context or the name of the known person; for the latter, a relevant label should be given to the associated instance in the ontology, a step that is done when populating instances in the ontology from the reported components (Section 4.1.1).

Table 9: SPIN functions and rule for inferring knowledge relevant to the latest known detected person

(a) SPIN function 1: infer latest detection of known person(s)

```
SELECT DISTINCT ?detection
WHERE {
   ?detection a sot:Detection .
   ?detection sot:hasTimestamp ?timestamp .
   ?detection sot:detectsPerson ?person .
   ?person a ?person_type .
   FILTER (?person_type = sot:KnownPerson) .
}
ORDER BY DESC (?timestamp)
LIMIT 1
```

(b) SPIN function 2: infer text for proper left/right annotation

```
SELECT ?leftright_annotation
WHERE {
   ?spatialContext a ?spatialContextType .
   FILTER ((?spatialContextType = sot:LeftSpatialContext) || (?spatialContextType = sot:RightSpatialContext)) .
   ?spatialContextType sot:leftrightSpatialContext ?leftright_annotation .
}
```

(c) SPIN rule: create output for latest detection of known person, with additional reference to his/her left/right spatial relation with the user

```
CONSTRUCT {
   ?detection sot:producesOutput ?output .
   ?output sot:refersToDetection ?detection .
   ?output a sot:Output .
   ?output sot:hasTextualDescription ?description .
}
WHERE {
   BIND (sospin:Function_SelectLatestPersonDetection() AS ?detection) .
   ?detection a sot:Detection .
   ?detection sot:detectsPerson ?person .
   ?person a ?person_type .
   FILTER (?person_type = sot:KnownPerson) .
   ?person rdfs:label ?personName .
   ?person sot:hasSpatialContext ?spatialContext .
   ?spatialContext a ?spatialContextType .
   ?spatialContextType rdfs:subClassOf sot:SpatialContext .
   FILTER ((?spatialContextType = sot:RightSpatialContext) || (?spatialContextType = sot:LeftSpatialContext)) .
   BIND (sospin:Function_LeftRightContext(?spatialContext) AS ?leftright_annotation) .
   BIND (BNODE() AS ?output) .
   BIND (CONCAT(?personName, " is on your ", ?leftright_annotation, " side.") AS ?description) .
}
```

4.4 Chapter Summary and Future Work

This chapter presented the second iteration (v2) of the SUITCEYES ontology that corresponds to the user needs, the requirements of the HIPI and the technical aspects of the involved components. First, we presented the additional conceptualisations of the ontology, which extend the representational capabilities of the first version in the following aspects: (i) we cover more adequately details related to the **detections** reported from the different operational components (sensors, analytics), and (ii) we introduce completely new concepts that semantically define information related to **spatial relations** between the user and the (detected) spatial entities (objects/persons). Next, we described the integration framework on the basis of which the communication and exchange of information between the KB service and the different SUITCEYES operational components was achieved. Finally, we concluded with the extended version of the semantic reasoning framework and rule set, which elaborates complex rules for providing meaningful information to the user. We support different forms of the output of the reasoning mechanism in order to fit to the needs of the HIPI and the available SUITCEYES components that will be used for providing haptic feedback to the user.

The following directions for improvements are foreseen for the next iteration of the ontology:

- Extension of the communication framework for handling the connection and exchange of messages between the KBS and the module that will transpose those messages to dynamic haptograms. The creation of a new channel in the message bus, potential adjustments to the format of the message and the triggering of the rule-based reasoning mechanism should be further investigated for proper integration of the involved components.
- Extension of the KB reasoning mechanism, for efficiently handling the integration of similar content within the context of a produced natural language phrase. The current version of the KB is capable of handling *single entity output reports*; this means that if, for example, more than one result is inferred from the reasoning mechanism[15], then the current version of rules produce one

[15] as for example in Q5 example answer in Table 7

output sentence per detected entity, i.e., "*A <laptop> has been found <very close> to you. A <chair> has been found <very close> to you.*". Our intention is to integrate in future version of the KB such inferences in one sentence.

- Elaborate more on the efficient manipulation of incoming data from different components to the KB. The KBS should handle the similar reports arriving every second in the KB, and populate only those that report a different situation than the previously reported one. This way, we will avoid populating duplicate information in the ontology, something that is critical for the KBS response time.

5 Conclusions

This deliverable presents the second version of the tools within WP3 for capturing, translating and semantically representing environmental cues. More specifically, we presented the second versions of the tools for visual analysis (chapter 2), dimensionality reduction and perceptible haptic information (chapter 3) and semantic knowledge representation and reasoning (chapter 4). In the future, VA tools will be further tested and adjusted to meet the needs of people with DB. Moreover, perceptible haptic information on low dimensional haptic grid will be further extended in conjunction with semantic reasoning task to allow for a more pragmatic and reliable integration of multiple sources and output informative phrases and content to the user.

References

Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In Advances in neural information processing systems (pp. 585-591).

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation, 15(6), 1373-1396.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). Vggface2: A dataset for recognising faces across pose and age. 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 67–74.

Cao, Z., Yin, Q., Tang, X., & Sun, J. (2010). Face recognition with learning-based descriptor. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2707–2714.

Caporusso, N. (2012). Issues, Challenges and Practices in Advancing Pervasive Human-Computer Interaction for People with Combined Hearing and Vision Impairments. PhD dissertation. IMT Institute for Advanced Studies, Lucca.

Carnap, R. (1937). The Logical Syntax of Language. Humanities Press, New York.

Chen, D., Cao, X., Wen, F., & Sun, J. (2013). Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3025–3032.

Chopra, S., Hadsell, R., LeCun, Y., & others. (2005). Learning a similarity metric discriminatively, with application to face verification. CVPR (1), 539–546.

Clementini, E., Sharma, J. & Egenhofer, M.J. (1994). Modelling topological spatial relations: Strategies for query processing. *Computers and Graphics*, 18(6), pp. 815-822.

Danish Association of the Deafblind. (2012). 103 Haptic Signals – a reference book. ISBN 978-87-989299-4-9. At http://wasli.org/wp-content/uploads/2013/07/103-Haptic-Signals-English.pdf (20-01-08).

Darányi, S., Olson, N., Riga, M., Kontopoulos, S., Kompatsiaris, I. (2019). Static and Dynamic Haptograms to Communicate Semantic Content Towards Enabling Face-to-Face Communication for People with Deafblindness. Paper presented at the "SyMpATHY: SeMAntic Technologies for Healthcare and Accessibility Applications" Workshop at SEMAPRO-19. Porto, 24 September 2019.

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4690–4699.

Eberhart, R., & Kennedy, J. (1995, November). Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks (Vol. 4, pp. 1942-1948). Citeseer.

Edwards, T. (2014). Language Emergence in the Seattle DeafBlind Community. PhD dissertation. UNiversity of California, Berkeley.

Egenhofer, M.J. (1989). A formal definition of binary topological relationships. In Litwin, W., Schek, H.J. (eds) *Foundations of Data Organization and Algorithms* (FODO 1989). Lecture Notes in Computer Science, Vol. 367, Springer, Berlin, Heidelber.

Falco, R., Gangemi, A., Peroni, S., Shotton, D., & Vitali, F. (2014, May). Modelling OWL ontologies with Graffoo. In *European Semantic Web Conference*, (pp. 320-325). Springer, Cham.

Firth, J.R. (1957). A synopsis of linguistic theory 1930-1955. Studies in Linguistic Analysis, pp. 1-32. Reprinted in F.R. Palmer, ed. (1968). Selected Papers of J.R. Firth 1952-1959. London, Longman.

Fodor, J. A. (1975). The Language Of Thought. Harvard University Press, Cambridge, Mass.

Frankl, P., & Maehara, H. (1988). The Johnson-Lindenstrauss lemma and the sphericity of some graphs. Journal of Combinatorial Theory, Series B, 44(3), 355–362.

Frege, G. (1879). Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle.

Giannakeris, P., Petrantonakis, P.C., Avgerinakis, K., Vrochidis, S., & Kompatsiaris, I. (2020). First-Person Activity Recognition from Micro-Action Representations using Convolutional Neural Networks and Object Flow Histograms. Multimedia Tools and Applications, Springer, in press.

Hadsell, R., Chopra, S., & LeCun, Y. (2006, June). Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 2, pp. 1735-1742). IEEE.

Harris, S., Seaborne, A., & Prud'hommeaux, E. (2013). SPARQL 1.1 query language. *W3C recommendation*.

Harris, Z. (1970). Distributional structure. In: Papers in structural and transformational linguistics. pp. 775-794. Humanities Press, New York.

Hintikka, J. (1997). Lingua Universalis vs. Calculus Ratiocinator. An ultimate presupposition of Twentieth-century philosophy. Kluwer.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. Journal of educational psychology, 24(6), 417.

Hu, P., & Ramanan, D. (2017). Finding tiny faces. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 951–959.

Huang, C., Zhu, S., & Yu, K. (2012). Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. ArXiv Preprint ArXiv:1212.6094.

Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments.

Johnson, W. B., & Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. Contemporary Mathematics, 26(189–206), 1.

Jones, L.A., Kunkel, J., Piateski, E. (2009). Vibrotactile pattern recognition on the arm and back. Perception, vol. 38, pp. 52-68.

Jones, L.A., Ray, K. (2008). Localization and Pattern Recognition with Tactile Displays. In IEEE Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems, 13-14 March 2008, Reno, Nevada, USA. pp. 33-39.

Knublauch, H., Hendler, J.A., & Idehen, K. (2011). SPIN - overview and motivation. *W3C Member Submission*.

Korres, G., Eid, M. (2016). Haptogram: Ultrasonic Point-cloud Tactile Stimulation. IEEE Access, vol 4, pp. 7758-7769.

Koziel, S., & Yang, X. S. (Eds.). (2011). Computational optimization, methods and algorithms (Vol. 356). Springer.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., … others. (2018). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. ArXiv Preprint ArXiv:1811.00982.

Lahtinen, R.M. (2008). Haptices and Haptemes: A Case Study of Developmental Process in Social-haptic Communication of Acquired Deafblind People. PhD dissertation, University of Helsinki.

Lahtinen, R.M., Palmer, R., Lahtinen, M. (2010). Environmental Description: For Visually and Dual Sensory Impaired People. A1 Management, UK.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 212–220.

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), 2579-2605.

McDaniel, T.L. (2012). Somatic ABC's: A Theoretical Framework for Designing, Developing and Evaluating the Building Blocks of Touch-Based Information Delivery. PhD dissertation, Arizona State University.

National Knowledge Center for Deafblind Issues. Social haptic signals. At: https://socialhaptisk.nkcdb.se/signaler/ (20-01-08)

Nene, S. A., Nayar, S. K., & Murase, H. (1996). Columbia object image library (coil-20).

Ojala, S. (2011). Towards an Integrative Information Society: Studies on Individuality in Speech and Sign. TUCS Dissertations No 135. Turku Centre for Computer Science, Turku.

Papadimitriou, C. H., Raghavan, P., Tamaki, H., & Vempala, S. (2000). Latent semantic indexing: A probabilistic analysis. Journal of Computer and System Sciences, 61(2), 217–235.

Pedersen, M. E. H. (2010). Good parameters for particle swarm optimization. Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001, 1551-3203.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. IEEE Transactions on computers, 100(5), 401-409.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 815–823.

Shi, Y. (2001, May). Particle swarm optimization: developments, applications and resources. In Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546) (Vol. 1, pp. 81-86). IEEE.

Shi, Y., & Eberhart, R. C. (1998, March). Parameter selection in particle swarm optimization. In International conference on evolutionary programming (pp. 591-600). Springer, Berlin, Heidelberg.

Shi, Y., & Eberhart, R. C. (1999, July). Empirical study of particle swarm optimization. In Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 3, pp. 1945-1950). IEEE.

Simonyan, K., Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2013). Fisher vector faces in the wild. BMVC, 2(3), 4.

Sun, Y., Wang, X., & Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2892–2900.

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1701–1708.

Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. science, 290(5500), 2319-2323.

Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. Psychometrika, 17(4), 401-419.

Trubetzkoy, N.S. (1939). Grundzüge der Phonologie (Travaux du Cercle Linguistique de Prague, 7). Prague.

Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: a comparative. J Mach Learn Res, 10(66-71), 13.

Varzi, A.C. (2007). Spatial reasoning and ontology: Parts, wholes, and locations. In *Handbook of spatial logics*, (pp. 945-1038). Springer, Dordrecht.

W3C Recommendation (2014). SPIN – Modeling Vocabulary. [online] Available at: http://spinrdf.org/spin.html [Accessed 27.01.2020]

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., … Liu, W. (2018). Cosface: Large margin cosine loss for deep face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5265–5274.

Wierzbicka, A. (1996). Semantics. Primes and Universals. Oxford University Press, New York.

Wierzbicka, A. (1997). Understanding Cultures Through Their Key Words. English, Russian, Polish, German, and Japanese. Oxford University Press, New York.

Wittgenstein, L. (1967). Philosophical Investigations. Blackwell, Oxford.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Zhu, Z., Luo, P., Wang, X., & Tang, X. (2014). Recover canonical-view faces in the wild with deep neural networks. ArXiv Preprint ArXiv:1404.3543.

Zucchi, S. (2012). Formal Semantics of Sign Languages. Language and Linguistics Compass 6/11: 719-734.

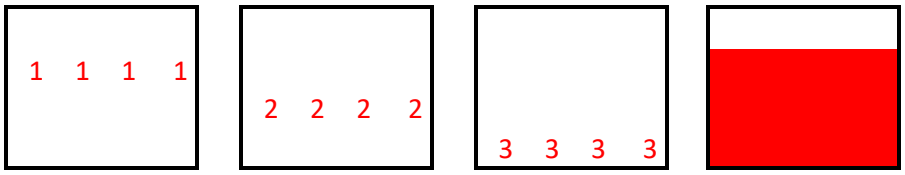
Appendix

ONTOLOGY COMPLIANT PART OF THE HAPTOGRAM VOCABULARY WITH EXAMPLES FOR INCOMING MESSAGE GENERATION.

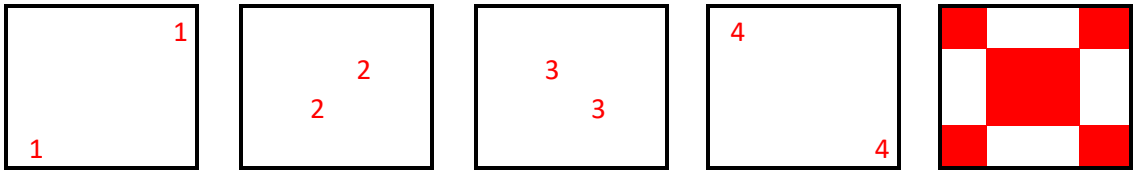
The shape of dynamic haptograms to the right is generated by their preceding grid activation sequences to the left. Actuator distances in the grid, and temporal gaps between firing actuators are subject to parametrization. It must be stressed out that these designs are just examples to demonstrate the idea while the actual final designs are yet to be discussed and tested with users before finalization.

Haptograms for scenes recognized:

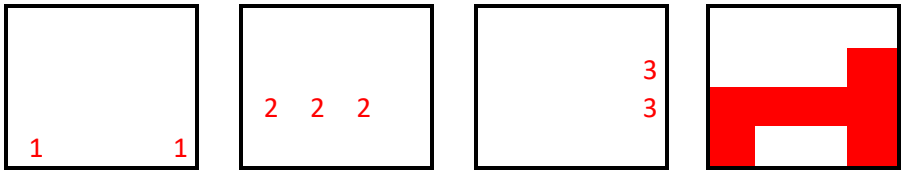
Basement:



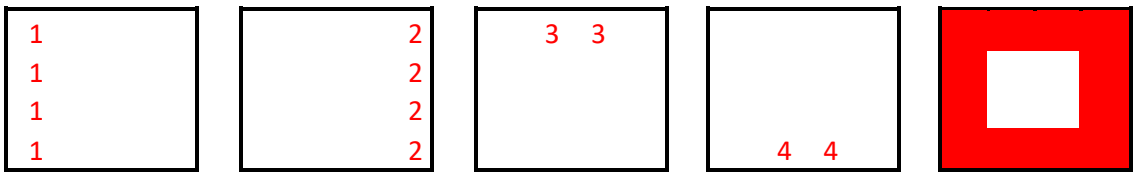
Bathroom:



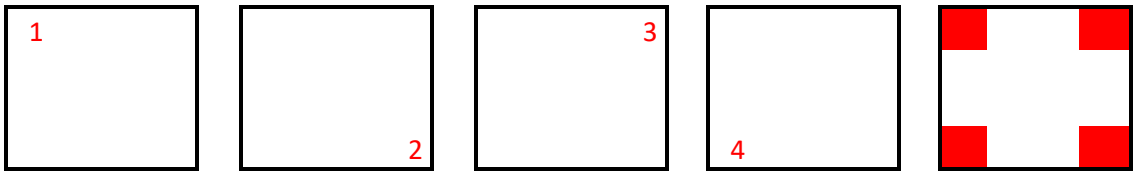
Bedroom:



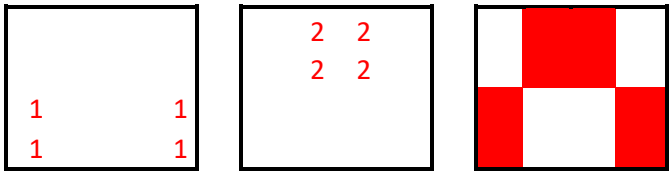
Closet:



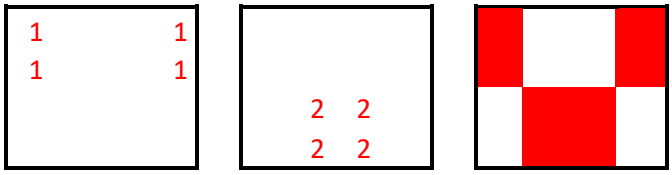
Child's room:



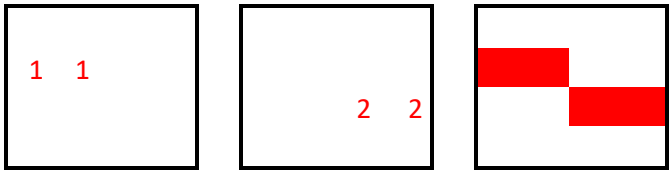
Living room:



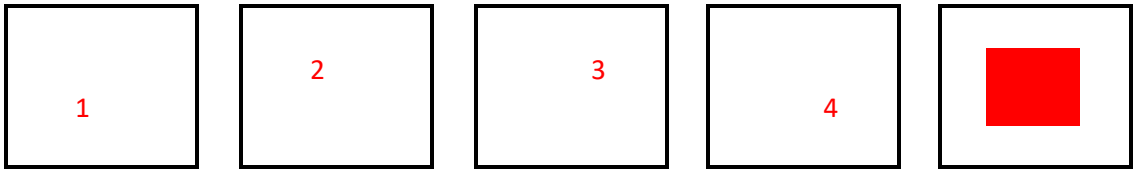
Dining room:



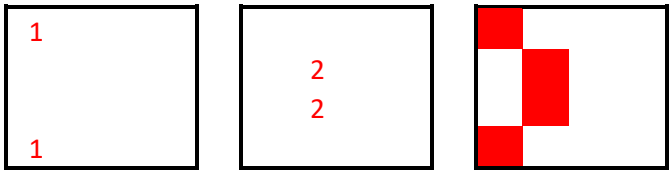
Pantry:



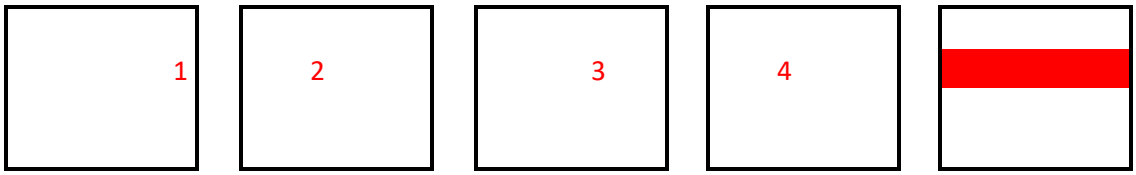
Office:



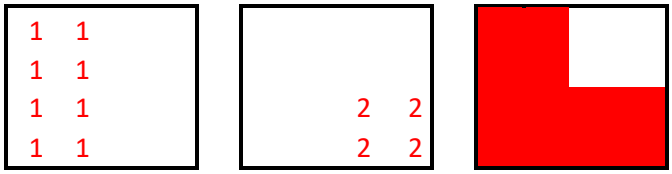
Shower:



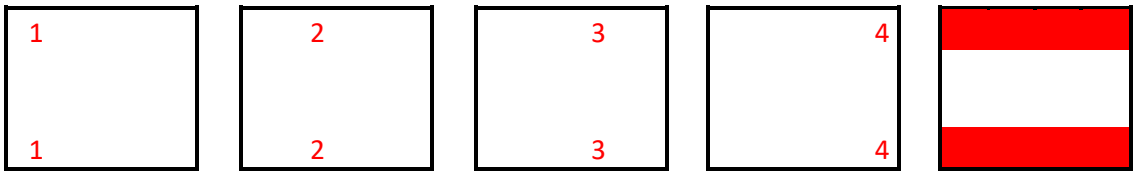
Kitchen:



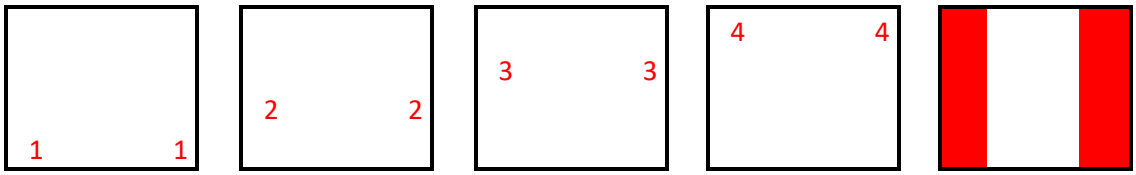
Laundromat:



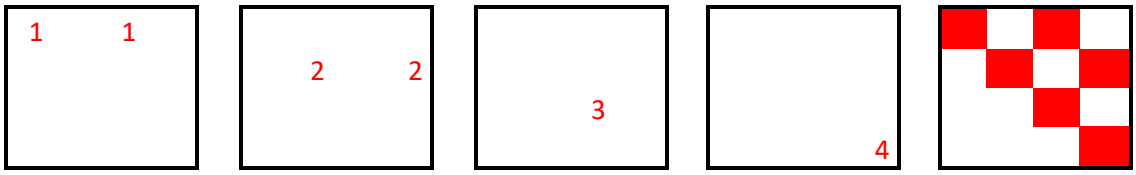
Corridor:



Elevator:

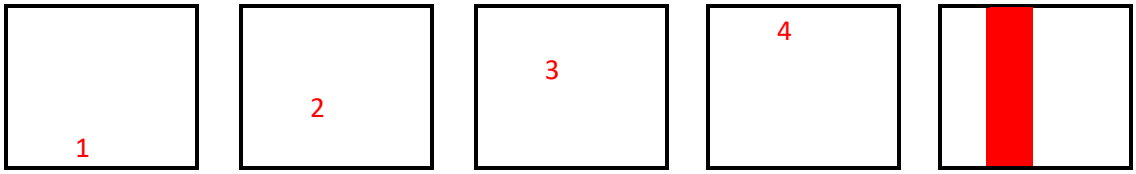


Staircase:

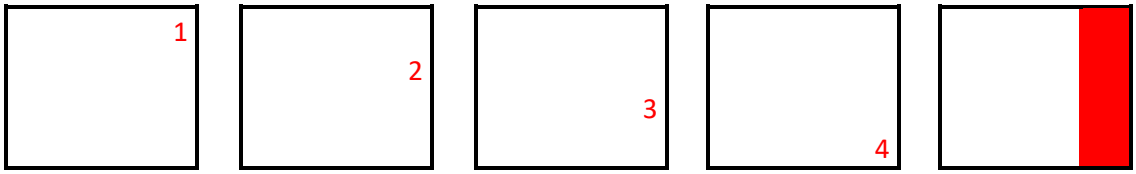


Haptograms for entities (objects recognized):

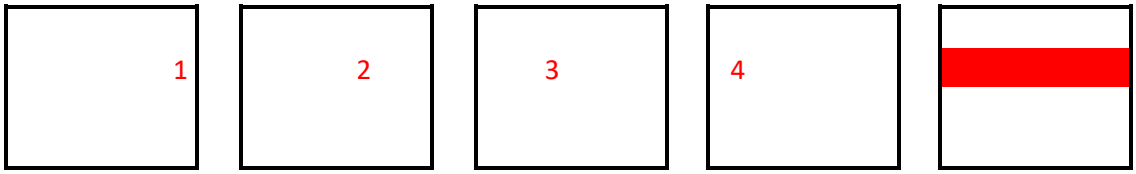
Table:



Chair:

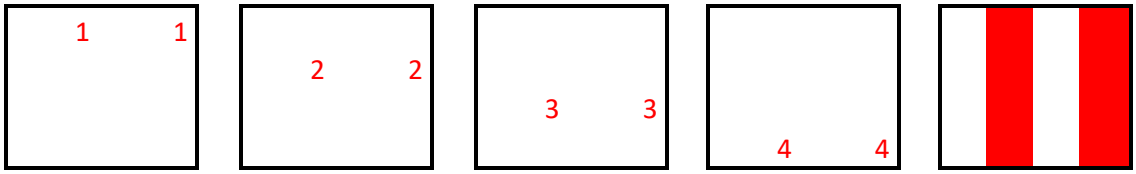


Window:

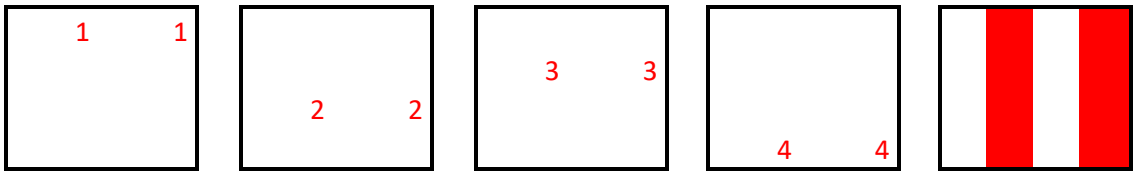


Mug/cup:

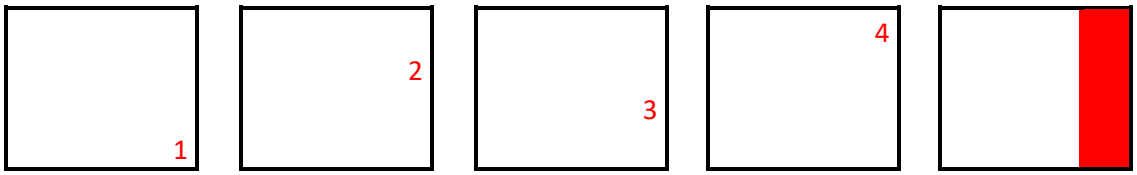
Book:



Laptop:

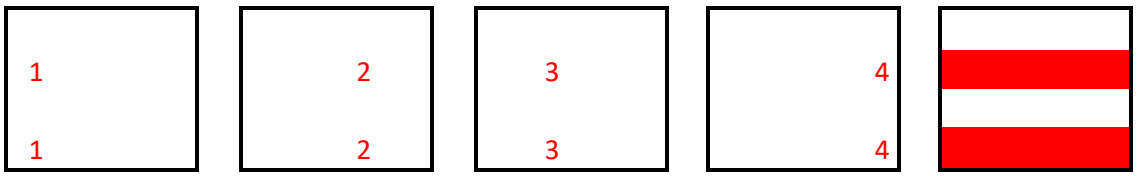


Cell phone/mobile:

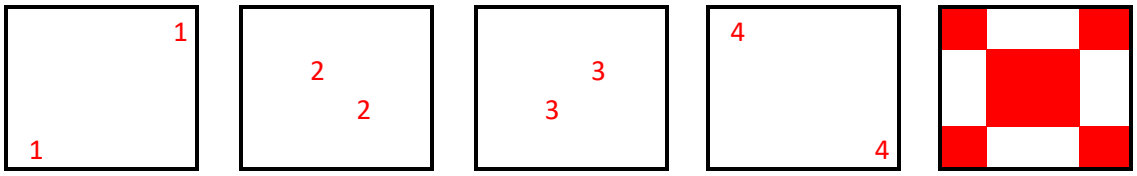


Door:

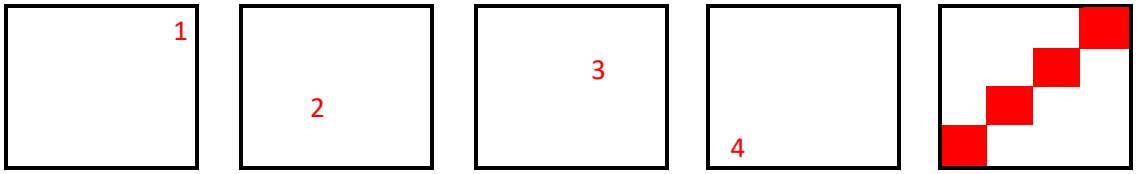
Towel (kitchen):



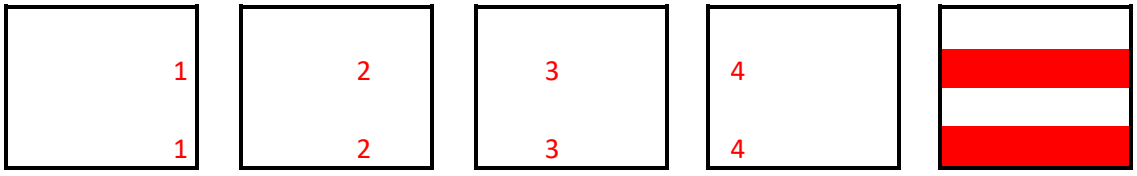
Towel (bathroom):



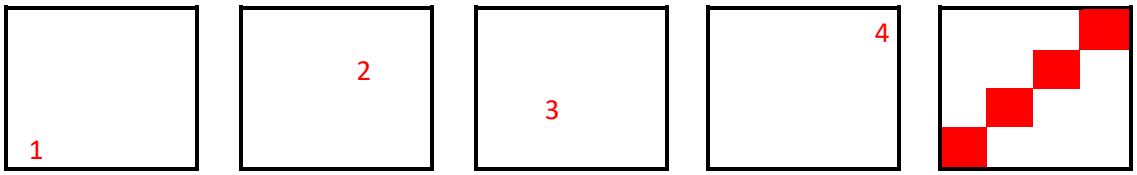
Pan:



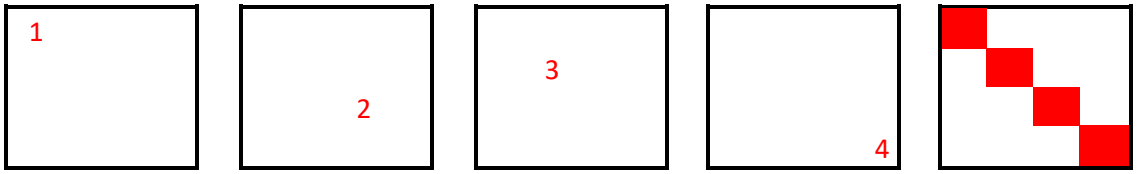
Pitcher:



Kettle:



Knife/spoon/fork/cutlery:



Bottle:

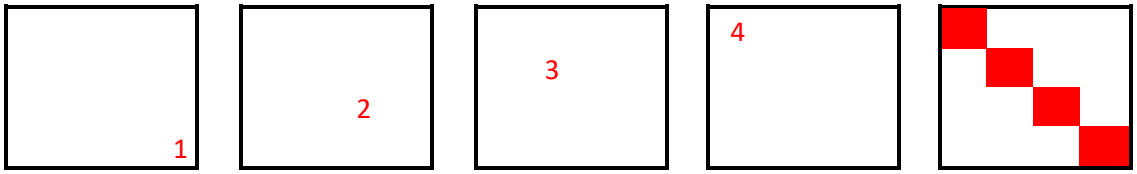
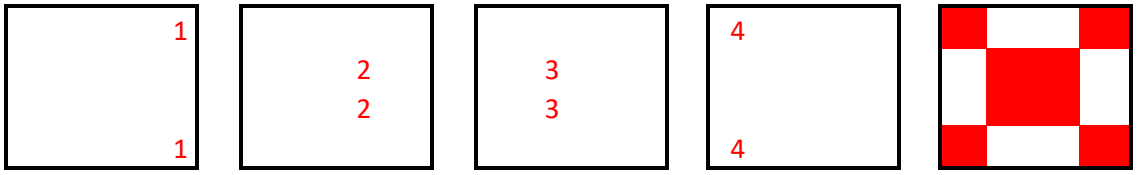
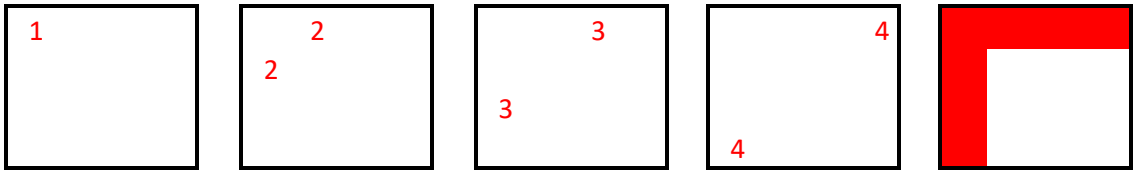


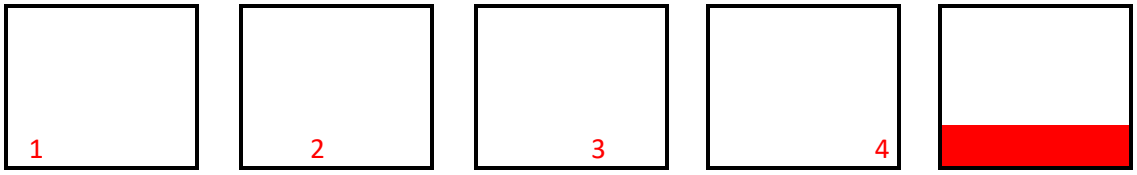
Plate:



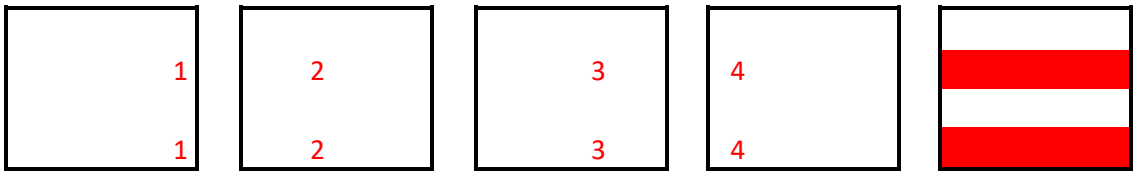
Container:



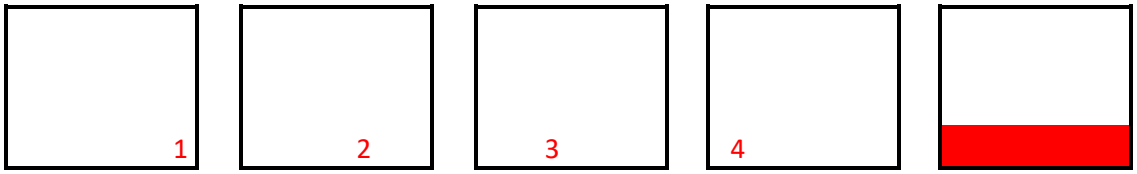
Oven/stove:



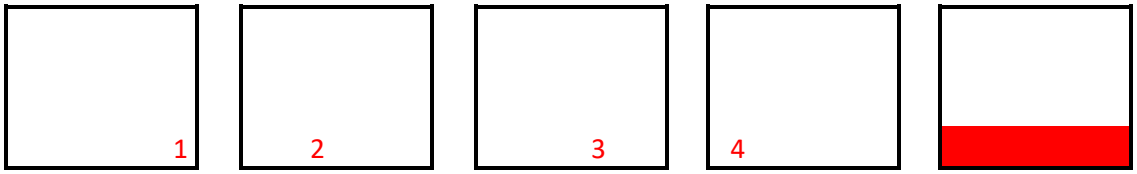
Trash can:



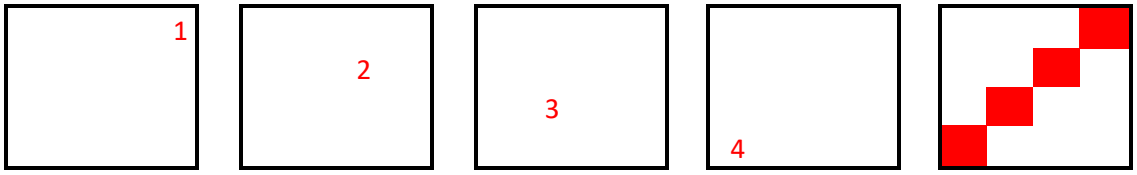
Water tap:



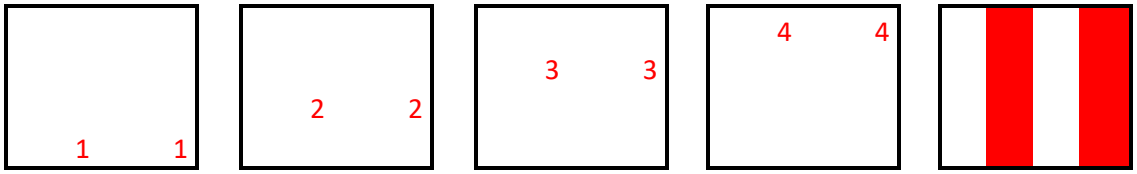
Refrigerator/fridge:



Microwave:

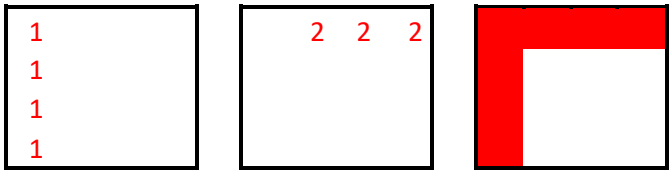


TV:

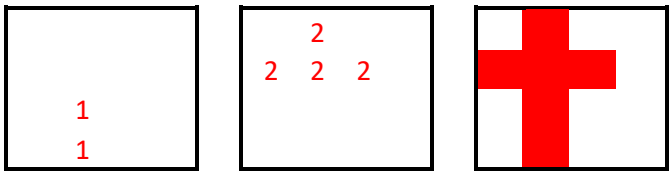


Relation components:

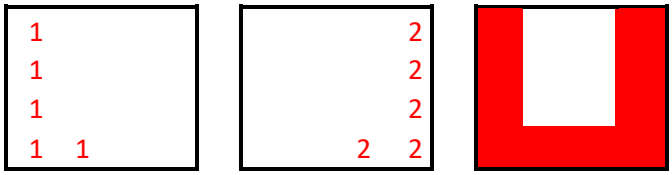
In:



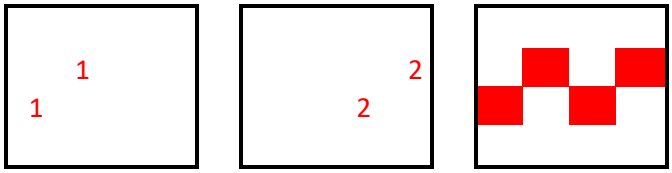
On:



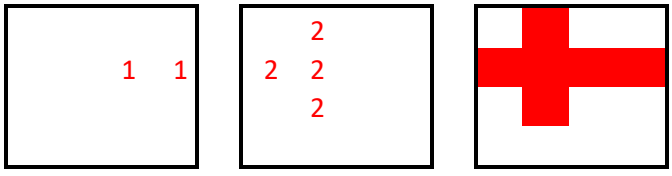
At/by:



Close:

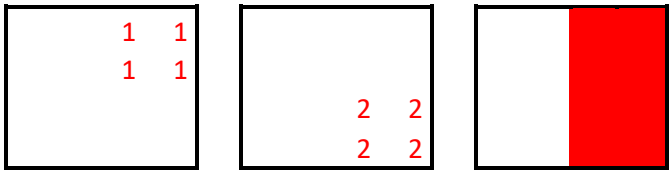


Left:



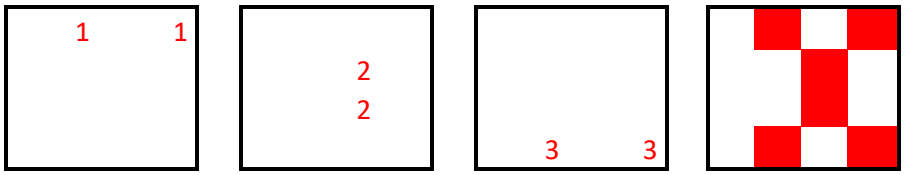
Personal pronoun:

You:



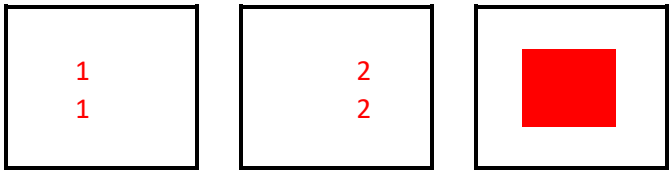
Person:

Unknown person:

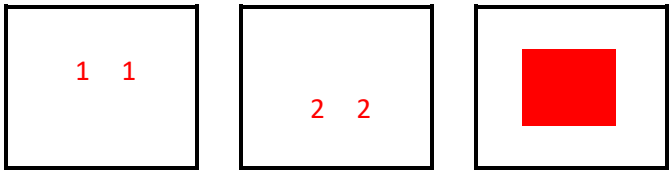


Sentence start/stop signals:

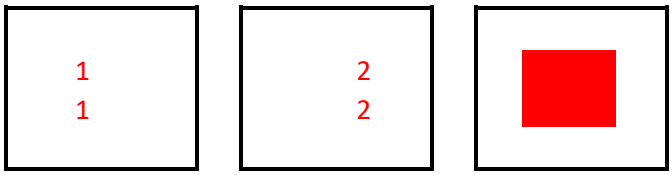
Start:



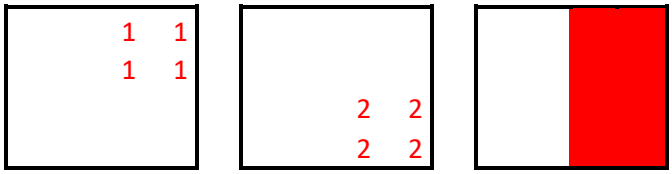
Stop:



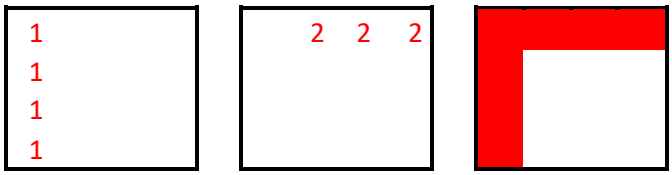
Example 1: “You are in the kitchen.”



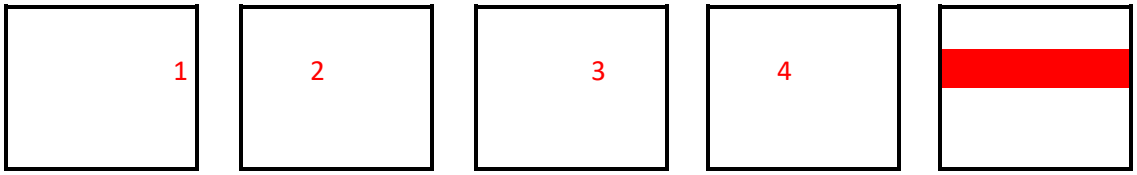
start



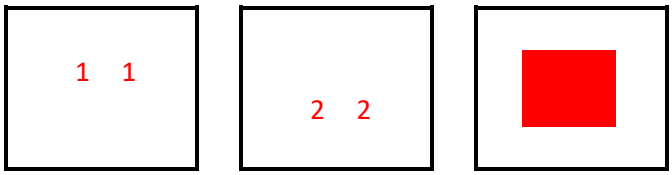
you



in

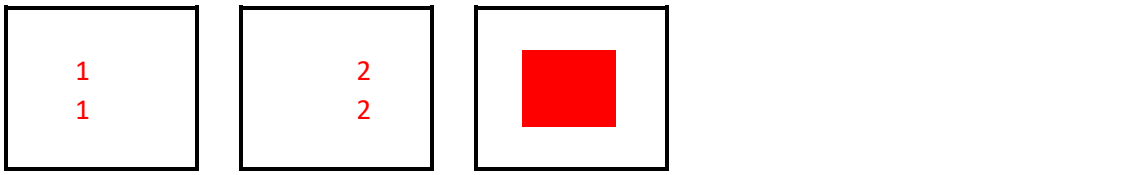


kitchen

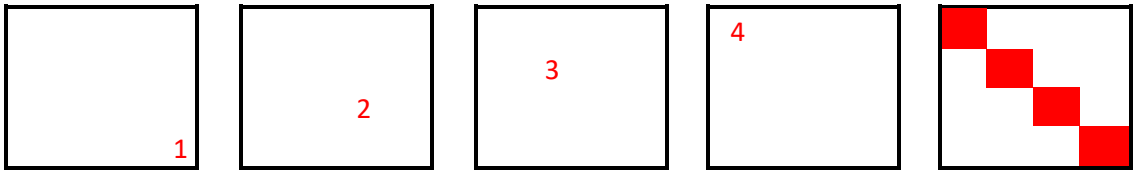


stop

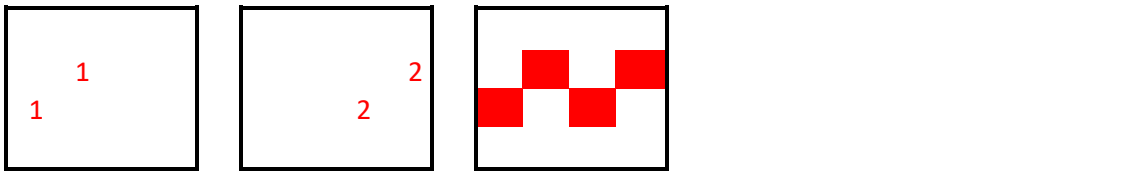
Example 2: “The bottle is close to the left on the table.”



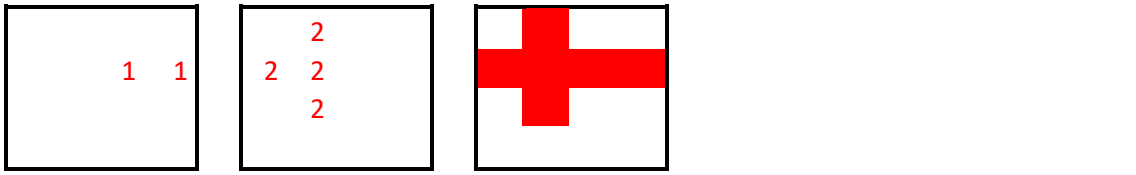
start



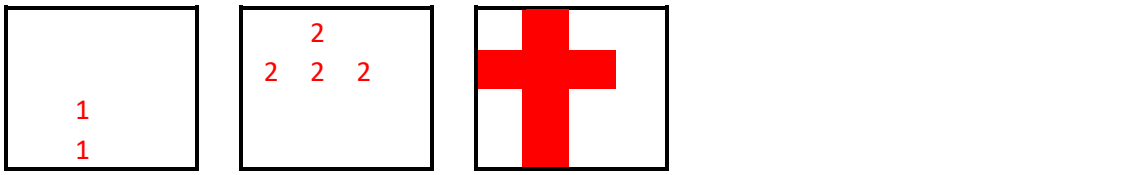
bottle



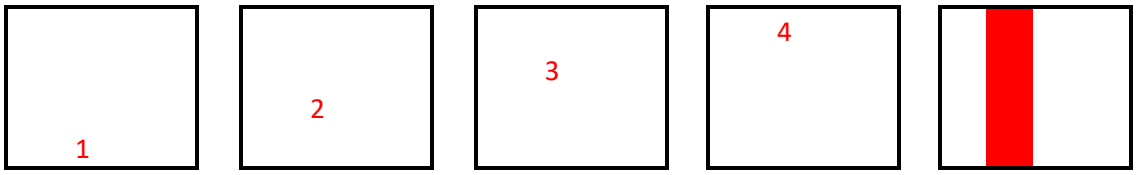
close



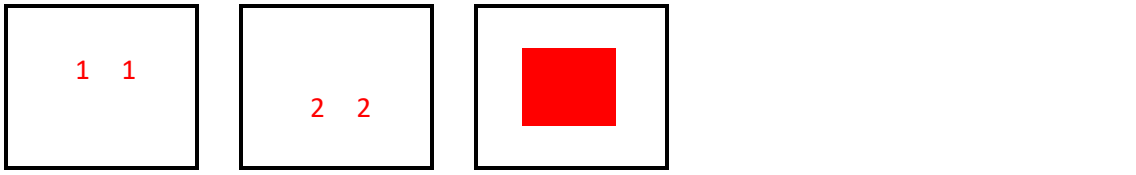
left



on

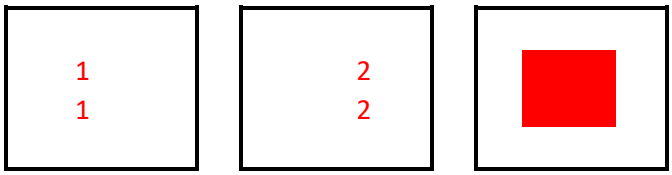


table

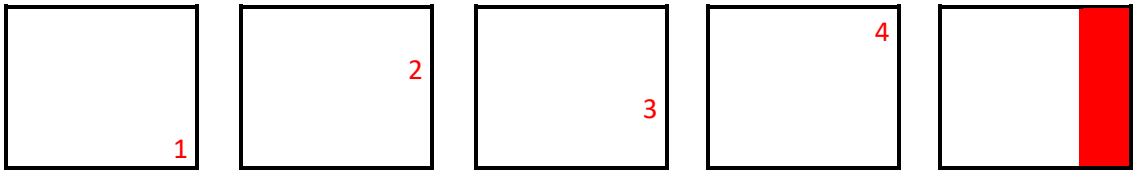


stop

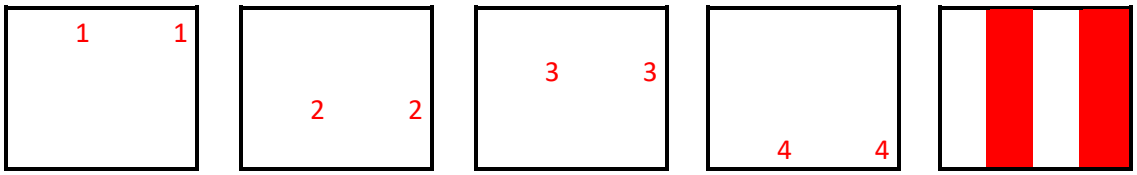
Example 3: “The cell phone and the laptop are by the window in the dining room”.



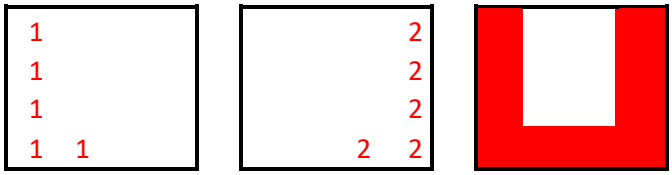
start



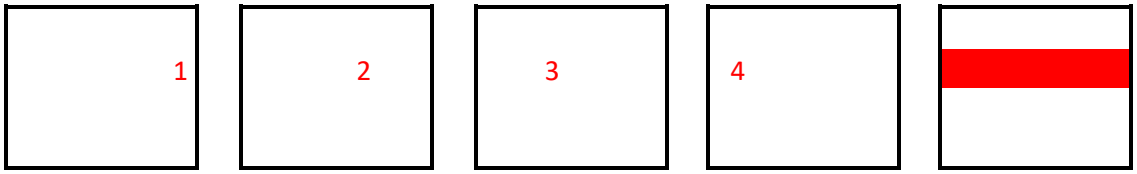
cell phone/mobile



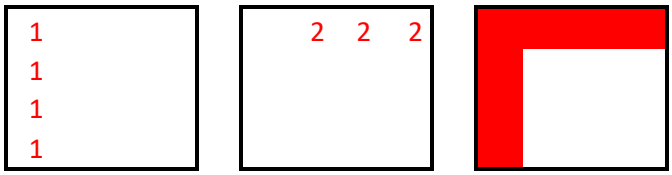
laptop



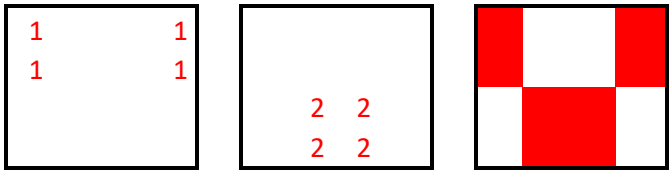
at/by



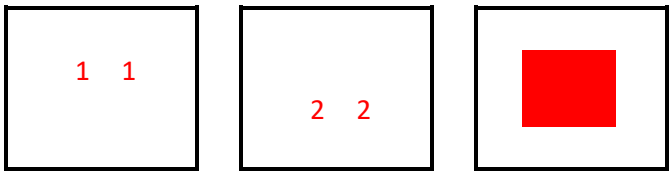
window



in

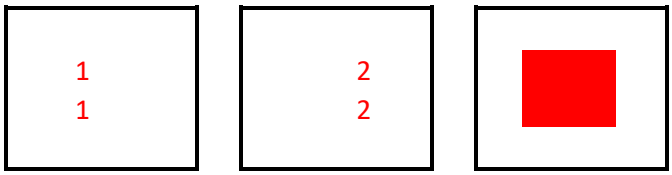


dining room

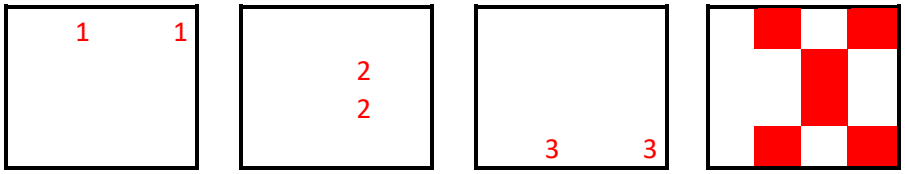


stop

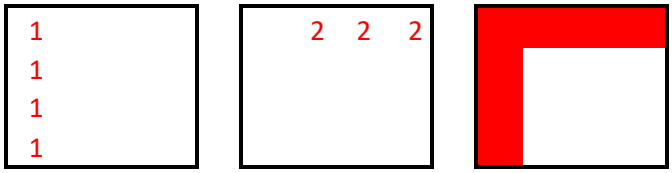
Example 4: “There is an unknown person in the hallway.”



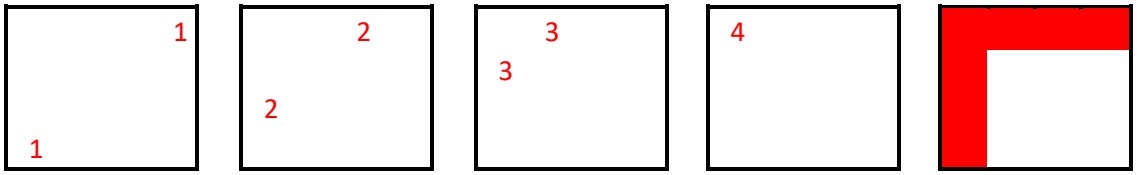
start



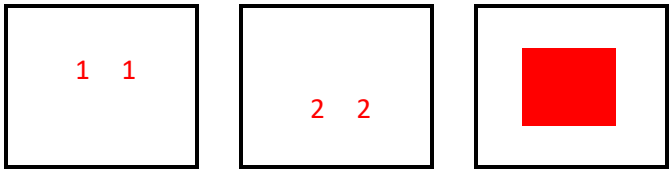
unknown person



in



hallway



stop