A Software Architecture for Multimodal Semantic Perception Fusion

Luca Buoncompagni[†], Alessandro Carfì[†], and Fulvio Mastrogiovanni

All the authors are affiliated with the Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genoa, Via Opera Pia 13, 16145, Genoa, Italy.
Corresponding authors' e-mails: {luca.buoncompagni@edu, alessandro.carfi@dibris}.unige.it.
[†]These authors contributed equally to this work.

Abstract Robots need advanced perceptive systems to interact with the environment and with humans. Integration of different perception modalities increases the system reliability and provides a richer environmental representation. The article proposes a general-purpose architecture to fuse semantic information, extracted by difference perceptive modules. Therefore, the article describes a mockup implementation of our general-purpose architecture to fuse geometric features, computed from point clouds, and Convolution Neural Network (CNN) classifications, based on images.

Keywords: robot perception, multimodal perception, multimodal fusion, late fusion, semantic perception.

1 Introduction and Background

Multimodal perception gained much attention both for its bioinspired nature and for the benefits that can provide in terms of reliabilities and richness of the information. Indeed, the integration of multiple perception modalities can increase the reliability of shared information while adding to the final representation information exclusive of a particular modality. Robotic systems are an interesting scenario of application for multimodal perception since they typically have different sensors that can be integrated to enhance the robot understanding of the environment.

The multimodal perception paradigm requires a fusion process integrating information from all the modalities, an extensive overview of fusion techniques is presented in [3]. The fusion process can be performed at *feature* level, early fusion, or at *decision* level, late fusion [7]. In early fusion *feature* extracted from the raw data are combined and then analysed as a whole, on the contrary in late fusion outputs from all the perceptive modules are merged to obtain the final output. Both late [2] and early [6] fusion have been used in robotics for multimodal recognition of objects. Late fusion offers particular advantages in terms of modularity, each time a new sensor is installed the module processing

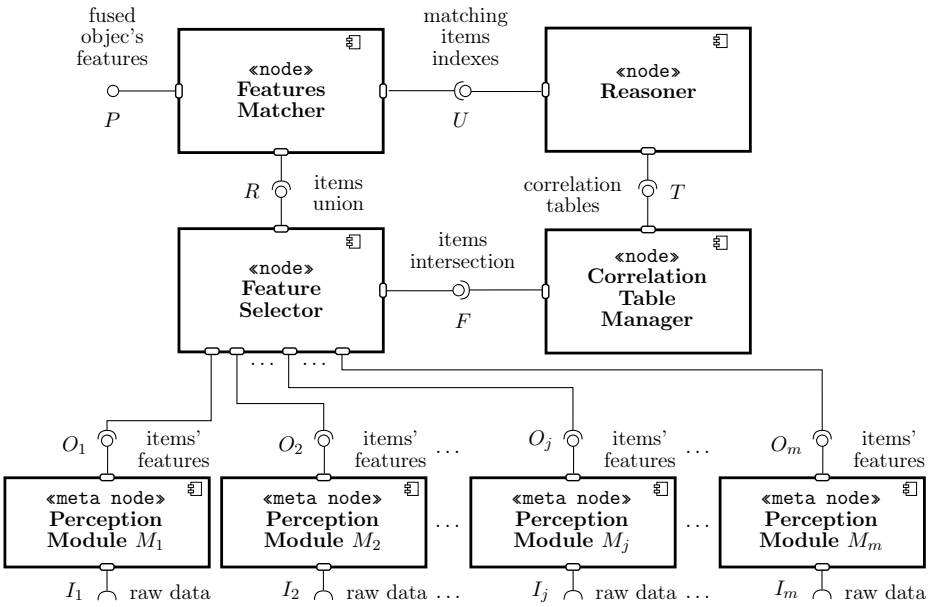


Figure 1: The UML diagram of the proposed architecture with m perception modules.

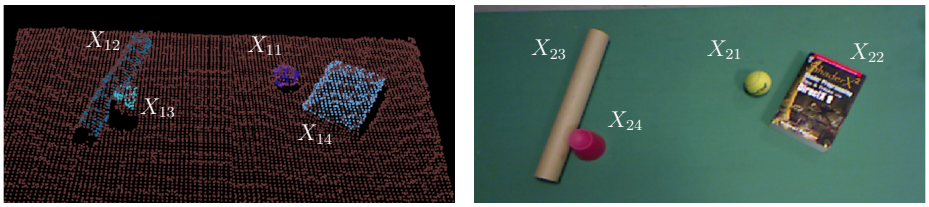
its data can be easily integrated into the system. Furthermore, this approach encourages reusability and when a well-known technique to extract information from a sensor is available can be easily adapted to the particular use case.

To enhance modularity and reusability of code in robotic, we propose an architecture for multimodal perception using late fusion. Late fusion requires a common representation to be shared among all the module outputs. Because of its intuitiveness, we have designed a semantic representation in which each *item*, detected by the perception modules, is associated with a list of semantic characteristics, which in the paper will be simply named *features*. The architecture uses features shared between different modalities to correlate *items*.

2 A Modular Software Architecture Overview

The proposed architecture[1], shown in Figure 1, performs a late fusion of distinct perception *modules* resulting in a structure P, provided as output. The perceptive modules $\{M_i, \forall i \in [0 \dots m]\}$ have an unconstrained input interface I_i and a well defined output structure O_i. In particular, M_i generates a set of semantic *items* $X_{ij} \subseteq O_i$ described by *features* through a map $\langle v_{ij} \rangle^s$ that relates semantic *key* ($s \in S_i$) to a value (v_{ij}^s) (as shown in Table 1). Remarkably, we assume that in all key-values maps, the keys are unique and we define the set

[1] an implementation is available at:
`https://github.com/EmaroLab/mmodal_perception_fusion`



(a) A point cloud, I_1, used by M_1 to extract the features of the X_{1j} items.

(b) An RGB image, I_2, (RGB image) used by M_2 to extract the features of the X_{2j} items.

Figure 2: An example of input and extracted features obtained from two perception modules of the architecture shown in Figure 1.

containing the semantic key of the whole system as $S = \bigcup_{i=1}^{m} S_i$. The features describing an item X_{ij} span in a subset of S, note that it might be possible $\nexists\, v_{ij}^{s}$. Finally, the output P has the same structure of O_i, but while the latter contains key-value maps generated from a single module, P is created by the merging process possibly using features from all the perception modalities.

The key-value structure is expressive, flexible and suitable as input for further symbolic reasoning, such as Ontology Web Language (OWL) compatible with the Robotic Operative System (ROS), *e.g.* through a bridge presented in [4]. Indeed, each feature of a perceived item is represented with a semantic key, that belongs to the symbolic domain (*i.e.* is encoded as a *string*), and a value, which can be a boolean signal, a real or natural number, as well as another symbol, *e.g.* $X_{ij} = \{\langle \texttt{radius}, 0.3\rangle, \langle \texttt{cluttered}, \textit{true}\rangle, \langle \texttt{color}, \texttt{red}\rangle\}$.

The architecture interfaces with the perception modules through the Features Selector, which manages the synchronisation of the incoming data and generated R and F. Where R is the *union* of all the perceived items and F is a structure containing only the values with shared keys. The Correlation Table Manager computes the *correlation tables* T as a function of the features distance while considering only the features contained in F. This map is used by the Reasoner to identify lists of items that can be merged, and corresponding item indexes are stored in U. Finally, the Feature Matcher uses indexes store in U to fuse correlated items and provides as output a set of new items P.

3 Software Interfaces for Multimodal Perception Fusion

As describe in Section 3 the proposed architecture is designed to work with *modules* that provide outputs through the O_i interface, which is formally defined as $O_i = \{X_{ij}, \forall j \in [1 \dots \eta(i)]\}$, where $\eta(i)$ represents the number of *items* perceived by the i-th module at some instant of time, and each item is represented with a map of *features* $X_{ij} = \langle v_{ij}\rangle^{s}$. Given some output O_i from different i-th modules, we define their *union* as the concatenation of all the items perceived

v^s_{ij}		**semantic features** (s)					
		time [h:m:s.ms]	*position* [m]	*shape*	*radius* [m]	*label*	...
perceived items (ij)	X_{11}	09:37:45.92	(.42, .13, .04)	`sphere`	.04		
	X_{12}	09:37:46.03	(.37, -.21, .02)	`cylinder`	.03		
	X_{13}	09:37:46.85	(.31, -.22, .03)				
	X_{14}	09:37:47.35	(.17, .34, .04)	`plane`			
	X_{21}	09:37:46.20	(.45, .11, .05)			`ball`	
	X_{22}	09:37:46.31	(.21, .33, .03)			`book`	
	X_{23}	09:37:46.37	(.34, -.19, .02)				
	X_{24}	09:37:46.42	(.31, -.22, .03)			`glass`	

Table 1: An example of item's features perceived through the inputs in figures 2a (provided in the O_1 interface shown in Figure 1) and 2b (provided in the O_2 interface). Perceived items are shown by row, while semantic keys by columns.

by all the modules, *i.e.*

$$R \doteq \bigcup_{i=1}^{m} O_i = \{X_{ij},\, \forall i \in [1 \dots m],\, j \in [1 \dots \eta(i)]\}\,.$$

On the other hand, we define the *intersection* operator as the collection of pairs of items X_{hq} and X_{kp} where all the features related to not common keys are removed. And the remaining values referring to the common keys, v^z_{hq} and v^z_{kp} where

$$z \in Z_{hq,kp} = \Big\{ s : \forall s \in S,\, \exists\, v^s_{hq}, v^s_{kp} \in R,\, h \neq k \Big\} \subset S,$$

are structured as $H^z_{hq,kp} = \big\{\langle v_{hq}\rangle^z, \langle v_{kp}\rangle^z\big\}$. Finally the intersection is defined as

$$F \doteq \bigcap_{i=1}^{m} O_i = \Big\{ H^z_{hq,kp} : \forall z \in Z_{hq,kp}, k, h \in [1 \dots m], \\ q \in [1 \dots \eta(h)],\, p \in [1 \dots \eta(k)] \Big\}.$$

Remarkably, our architecture correlates items perceived from different modules based on feature with common semantic key. In particular, if $H^z_{hq,kp} = \emptyset$ the hq-th and kp-th items can not be directly correlated and, if $F = \emptyset$ all the items can not be correlated.

Let $\Phi = \{\varphi^z,\, \forall z \in Z_{hq,kp}\}$ be a set of φ^z distance functions associated to the hq-th and kp-th items; thus, each distance can be computed as $\varphi^z\left(v^z_{hq}, v^z_{kp}\right) = d^z_{hq,kp} \in [0, \inf)$. We define the correlation score between the hq-th and kp-th items as

$$f_{hq,kp} = \tanh\left(-\frac{\sum_z d^z_{hq,kp}}{w}\right) + 1 \in [0, 1],$$

in this way low distances values are mapped to high-level of correlation scores, and w is a parameter that can be tuned for modulate the mapping function

behaviour. Through the computation of $f_{hq,kp}$ for all the pairs of perceived items in F, we obtain a set of tables $T = \{T_{hk}, \forall h, k \in [1 \dots m], h \neq k\}$ (thus T collects $m(m-1)/2$ tables), where T_{hk} is a table of size $\eta(h) \times \eta(k)$.

The system uses the correlation tables T as a grounded representation to reason on the best matching among the X_{ij} items. Such a reasoning generates a set $U = \{U_e, \forall e \in [1 \dots g]\}$, where g is the number of objects perceived by the architecture (*i.e.* real objects), and U_e is a list of indexes ij-th associated to the l-th items that can be merged to describe the e-th real object, *i.e.* $U_e = \langle i, j \rangle^l$. From R we extract all the l-th items $\{X_{ij}, \forall i, j \in U_e\}$ which have z-th shared and y-th unique features. Fusing the l-th items generates $P_e = \langle v_e \rangle^z \cap \langle v_e \rangle^y$, where a function δ is used to compute $v_e^z = \delta\left(v_{ij}^z \, \forall i, j \in U_e\right)$ and $v_e^y = \left\{v_{ij}^y, \forall i, j \in U_e\right\}$. Finally, the architecture output is $P = \{P_e, \forall e \in [1 \dots g]\}$.

4 Implementation

To provide an application example, we have built an implementation that uses images and point clouds to detect objects in a tabletop scenario (as shown in Figures 2). The architecture have been implemented using the ROS middleware, specifically for two perception modules (*i.e.* $m = 2$): M_1 and M_2. The point clouds are processed by M_1 with a stack of RANSAC simulations to segment the objects laying on the table [5]. Each j-th item perceived by M_1 can be described by one or more of the features contained in $S_1 = \{$`time`, `shape`, `position`, `orientation`, `radius`, `high`, `vertex`$\}$. On the other hand, M_2 exploits a Convolution Neural Network (CNN) from the tensorflow tutorial [1] to detect objects and assign them a describing label. Each j-th item perceived by M_2 can be described by one or more of the features contained in $S_2 = \{$ `time`, `label`, `position`$\}$. Therefore, common features of object detected by the two methods are contained in $Z_{1p,2q} = \{$`time`, `position`$\}$.

The correlation table T_{12} have been computed as described in Section 3, while the two φ^z functions have been defined as Euclidean distance. To finally merge information from M_1 and M_2 we have used an algorithm that explores T_{12} to find the row and column indexes of cells which contains a high correlation score. The algorithm ensures that each index cannot occur twice in U_r (*i.e.* each object detected from M_1 is associate at maximum to one object detected by M_2) and conflicts are addressed to prioritise higher correlation scores. Finally, to merge all the objects we have defined the δ function for `time` and `position` as the geometric mean.

5 Discussions and Conclusions

The paper proposed a general-purpose architecture for late semantic fusion. Indeed, it can accommodate an arbitrary set of perception modules that process different data sources, but they have to generate a specific type of outcomes,

defined through the semantic item's features. Nevertheless, these semantic structures are flexible, and the architecture uses them to correlate items perceived by different modules, providing a fused representation as output.

The architecture relies on the distance between shared features, computes the correlation between items, requires a reasoner for items matching, and a function for item fusing. We deeply analysed how to orchestrate such elements in a general scenario and we present a simple implementation based on RANSAC and CNNs.

We argued that for a general case, it is required a further investigation of the distance functions between complex features, (*e.g.* `color`, `shape`, etc.), as well as regarding the types of reasoning to be performed with the computed correlation tables. On the other hand, such tables are expressive, allowing to achieve complex decisions for the item fusion. For example, they contain all the information to merge objects with partially shared features, through transitivity properties. Future developments of this work will include a wider integration of perceptive modules and an experimental evaluation of the architecture.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: a system for large-scale machine learning. In: OSDI. vol. 16, pp. 265–283 (2016)
2. Aldoma, A., Tombari, F., Prankl, J., Richtsfeld, A., Di Stefano, L., Vincze, M.: Multimodal cue integration through hypotheses verification for rgb-d object recognition and 6dof pose estimation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). pp. 2104–2111. IEEE, Karlsruhe, Germany (May 2013)
3. Atrey, P.K., Hossain, M.A., El Saddik, A., Kankanhalli, M.S.: Multimodal fusion for multimedia analysis: A survey. Multimedia Systems 16(6) (2010)
4. Buoncompagni, L., Capitanelli, A., Mastrogiovanni, F.: A ROS multi-ontology references services: OWL reasoners and application prototyping issues. In: Proceedings of the 5th Italian Workshop on Artificial Intelligence and Robotics (AIRO) A workshop of the XVII International Conference of the Italian Association for Artificial Intelligence. CEUR-WS, Trento, Italy (2018)
5. Buoncompagni, L., Mastrogiovanni, F.: A software architecture for object perception and semantic representation. In: Proceedings of the 2nd Italian Workshop on Artificial Intelligence and Robotics (AIRO) A workshop of the XIV International Conference of the Italian Association for Artificial Intelligence. vol. 1544, pp. 116–124. CEUR-WS, Ferrara, Italy (2015)
6. Eitel, A., Springenberg, J.T., Spinello, L., Riedmiller, M., Burgard, W.: Multimodal deep learning for robust rgb-d object recognition. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 681–687. IEEE, La Jolla, California, USA (October 2015)
7. Snoek, C.G., Worring, M., Smeulders, A.W.: Early versus late fusion in semantic video analysis. In: Proceedings of the 13th annual ACM international conference on Multimedia. pp. 399–402. ACM, Singapore (November 2005)