HAL
open science

Towards a versatile framework to integrate and control perception processes for autonomous robots

Andrea de Maio, Simon Lacroix

To cite this version:

Andrea de Maio, Simon Lacroix. Towards a versatile framework to integrate and control perception processes for autonomous robots. 12th national conference on Software & Hardware Architectures for Robots Control & Autonomous CPS (SHARC), Jun 2017, Toulouse, France. 5p. hal-01552360

HAL Id: hal-01552360

https://hal.science/hal-01552360

Submitted on 2 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a versatile framework to integrate and control perception processes for autonomous robots

Andrea De Maio*, Simon Lacroix†
CNRS, Laboratory for Analysis and Architecture of Systems (LAAS)
7 Avenue du Colonel Roche, F-31031, Toulouse, France
*andrea.de-maio@laas.fr, †simon.lacroix@laas.fr

I. Introduction

Perception is at the heart of autonomous robots, as it is the way through which the decision-making processes get information on the environment and the robot itself. In the last years, the availability for different types of sensors in robotics has greatly improved, allowing the integration of a significant variety of sensors on the same platform. Jointly, the state of the art of data processing and data fusion became much richer, offering a broad choice of solutions. Despite this richness, data fusion and perception processes are still tailored, by the robotic engineers, to the task they are needed for, with the sequences of data processes being defined by initially hard-coded scripts. Yet, perception is an active process: deciding which data to acquire and how to process them yields the possibility to optimize the throughput of perception, that is the relevance and quality of the information provided to the decision making processes. It is indeed short sighted to think that the configuration of the perception processes does not have to be changed regardless of the task a robot is carrying out, or the context within which the task is executed. This principle of actively controlling perception becomes even more relevant when thinking of the vast perception possibilities nowadays robots can be endowed with. The fact that perception is active has been early identified by the robotics community (Bajcsy 1988), which has mostly treated the problem as to purposefully change the sensors state parameters according to sensing strategies. Contributions on active perception often turned into active vision, due to the predominant diffusion of visual sensors in robotic platforms (Chen et al. 2011). In the many works published in this field, one particular task related to computer vision is considered, e.g. inspection (Trucco et al. 1997), grasping (Motai and Kosaka 2008), or object modeling (Chen and Li 2005; Pito 1999).

Even though there have been multiple efforts in active perception, there is the lack of a system abstracting from the type of sensor or from the nature of the task. To the best of our knowledge, there exist no work on defining a generic architecture modeling perception process being sensor and task agnostic. In this paper, we present our ongoing work on the definition of an approach to design an architecture controlling a broad perception layer, that allows to dynamically assemble, sequence and control perception processes on board an autonomous robot.

II. Composing perception tasks

In the context of autonomous navigation, Visual Odometry, SLAM and the generation of a Digital Elevation Map (DEM) describe processes which are composed of several signal processes that organized together return the desired output. Let us consider the Visual Odometry example. This process can be considered as a perception compound (PC), composed of atomic signal processing and data fusion functions, refined to a lower level of granularity that can be named Perception Nodes (PN): the PC Visual Odometry can be summarized as the composition of three main PNs: (1) a keypoint extractor, which takes as input a view of the world (e.g. through a camera) and outputs a set of keypoints (KP); (2) a data association algorithm matches two different sets of KP after a small motion is performed; (3) a motion estimator which computes motion thanks to the matches and the absolute 3D position of each KP w.r.t. to a given frame. The final output of the overall process is a 3D transformation between the two time steps at which the features were collected, along with an uncertainty measure of the transform. This "composition approach" is applicable to all the PCs a given robot must be endowed with – note though they do not always follow a pipeline model: some atomic processes can be asynchronously invoked, e.g. a loop closure algorithm, and some feedback sequences can be defined.

III. Modelling perception activities

In order to allow the dynamic control of perception activities, we need to model them, so as to explicit concepts such as composability, performance and cost, upon which one can defined a principled reasoning formalism. As first step to structure the perception activities, it is important to build a taxonomy of PNs which is as vast as possible in terms of inclusion of signal processes and data fusion functions. The importance of a taxonomy is twofold: on a high level of abstraction it is crucial to understand the nature of perception processes, to classify them under the output they produce and to structure them in a way that eases the use of perception and data fusion processes (PNs) as building blocks of perception-driven tasks (PCs). On a more functional level, such classification will be an element of guidance for the search in the process of autonomously building PCs, given a certain availability of PNs.

The proposed model of a PN x is as follows:

$$x = \langle U, Y, \Sigma, \Theta, \Psi, H \rangle \tag{1}$$

where U and Y are respectively the input(s) and the output(s), Σ is a set of figures of merit related the PN, that qualifies the produced output, Θ is the set of controllable parameters of the PN, Ψ represents the context information, that is the uncontrollable parameters influencing the performance of x (which are not necessarily fully observable), and H is a set of measurements on how x impacts on available resources (CPU, Memory, Power, etc: these are cost information). The proposed model can generally fit any perception processes and can be exploited to compare them. In particular, we aim to taxonomize PNs over the output they produce: this implies that Σ should characterize the quality of the output so to make processes comparable over Y. In perception, performance criteria are mostly related to uncertainty due to the nature of sensors robotic systems: for instance it is common to asses the quality of an estimation process by its covariance matrix (Mihaylova et al. 2005). Whilst estimation PNs and PCs are easily evaluated with such a criterion, there are many more figures of metric which can be applied to other kind of processes. They can either be generic (e.g. Precision/Recall or F1 score for a data association process), or tailored to the process itself. For instance, the quality of the output generated from a feature detector can be described by the number of generated features, possibly weighted by a quality assessment of the features.

IV. Planning perception activities

To select and configure the perception processes to be executed to achieve a given perception task, one shall take into account the possibility to vary Θ in order to increase the quality of the output. One shall also reason on Ψ and explore two main possibilities: (1) given $\langle \Psi, x \rangle$, tune Θ to maximize/minimize Σ, (2) given $\langle \Psi, x_s \rangle$, select an alternative PN x_c so that $\exists\ \sigma_i(x_c) > \sigma_i(x_s)$[1], with $\sigma_i \in \Sigma$. All these choices shall consider the availability of on-board resources (CPU, Memory, Power) and time. The system must be able to sacrifice performances in order to satisfy external constraints dictated by H. This impacts on both cases shown before: the change of Θ may influence H as well as the selection of an alternate PN. One of the crucial challenges is going to be the formalization of Ψ, which more than H shall be the driving factor of the search. The information extrapolated from the perceivable world can be vary and hardly representable but the system shall be able to identify the knowledge usable by the reasoning module and to formalize it.

V. A taxonomy of perception nodes

As introduced in Sec.III, a taxonomy of perception nodes is fundamental to obtain a deep understanding of the perception layer. Our aim is to classify and characterize a broad set of data fusion and signal processes, addressing the problem in the vastest possible manner. One of the necessities originating from this approach involves the modeling of sensors. In fact, to enable autonomous composition and reconfiguration of PCs, it is necessary to treat sensors as real PNs outputting data structures. By doing so the only thing that matters in terms of composability is the produced data format, while the sensor type loses importance. For instance, in some PCs a Time-of-Flight camera and a Stereo Camera (or a LiDAR if color is not necessary) are completely interchangeable as they are both able to produce point clouds. Sensors shall often represent the start node for any PC. A large set of relevant sensory data structures have been identified: *Image Data, Depth Image Data, Depth Data, Planar Depth Data, Force Data, Torque Data, Angular Data, Angular Magnetic Data, Angular Velocity Data, Angle of Acceleration Data*. Every data type is produced by one or more sensors which can be considered as PN requiring "0 inputs". Perception processes having only one input are taxonomized as *signal processing* nodes. These PNs usually can either produce output having the same semantic as its input, e.g. filters such as a pass-through filter or a stastical outlier removal filter; or output different data types by just having one input. This is the case of feature detectors which can be further subdivided in several categories (edge detection, feature extraction, corner detection). Finally there are the PNs dealing with more than two inputs. These processes fall into the *data fusion* category and are classified through the relation lying between their input and output. Three meta-categories have been identified: Same-Input Same-Output, Same-Input Different-Output and Different-Input Different-Output. The first one features two main categories of data fusion techniques. *Data accretion* concerns the increase in volume of same type of data, like map/image stitching algorithms; while *estimation filters* normally fuse two different data sources to reduce the information uncertainty relative to the data. A typical example of estimation filters is represented by the Extended Kalman Filter. Same-Input Different-Output and Different-Input Different-Output nodes are respectively represented by data association, like feature matchers, and data cooperation techniques. A partial tree view of this taxonomy which does not show instances but only categories is present in Fig.1.

All these techniques are generically combined together so as to form PCs. At the end of such compounds we usually find dedicated data structures which are usable by high-level decision making functions. These are the final products of every PC, a global path planner will make use of a digital elevation map, while a local path planner may need to read pose information incoming from visual odometry.

[1] If we are trying to maximize Σ. $\sigma_i(x_c) < \sigma_i(x_s)$ otherwise.

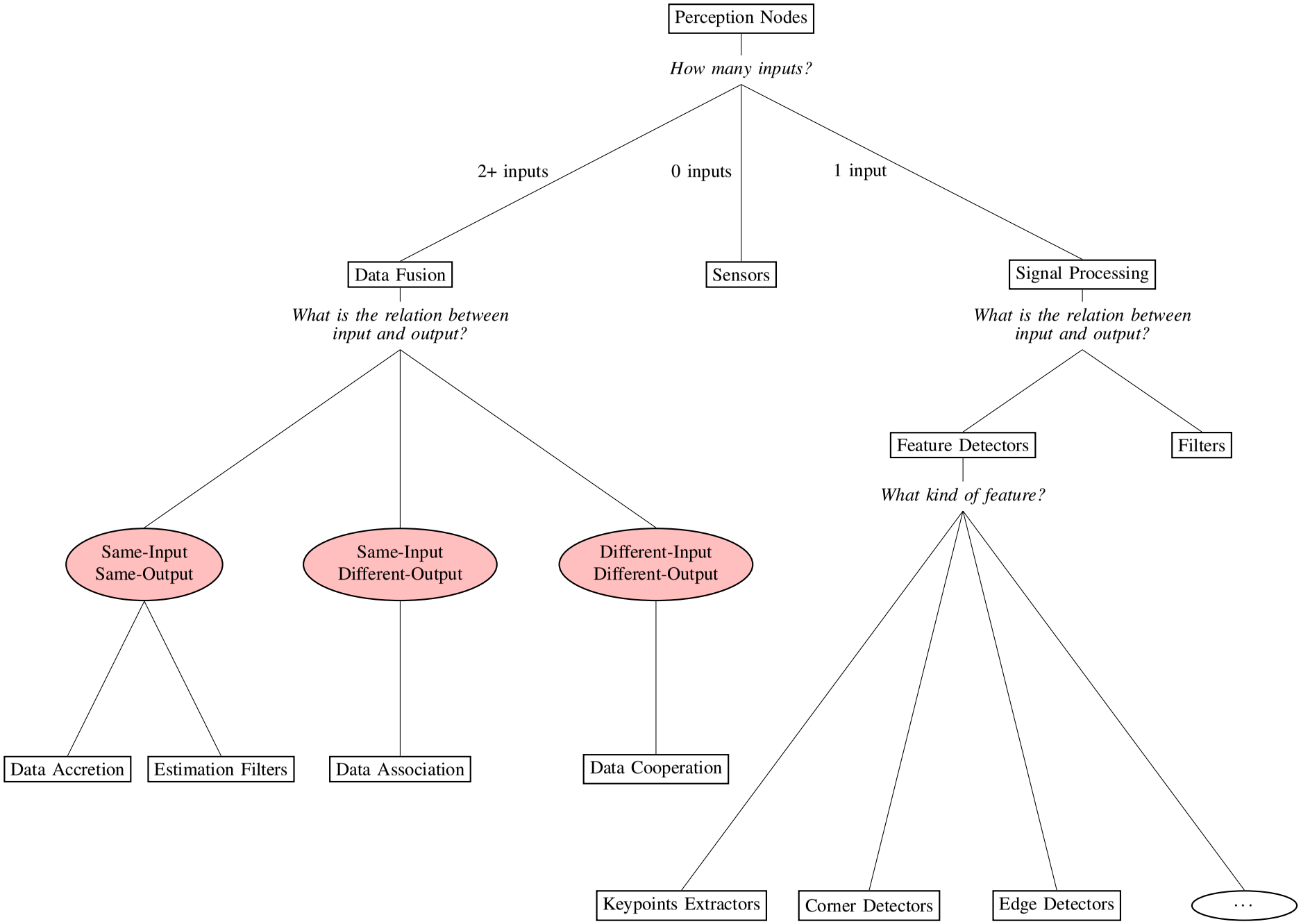

Fig. 1. A tree representing the proposed taxonomy of perception nodes *(work in progress)*.

VI. A USE CASE

Let us instantiate a PC into the proposed model. For this example we will consider again the Visual Odometry case. We already introduced in Sec.II its nature and its processing steps but it is now possible to instantiate the PNs composing it, in the context of the proposed taxonomy. Moreover, as mentioned in Sec. V sensors (and their data processing algorithms) are PNs too. A stereo computation algorithm, which we will consider as step 0, has to be added to the PC in case. Therefore, we will consider 4 PNs which are here depicted and instantiated. Note some part of their model are not definite (as shown by the numerous "..."), these 4 instanciations being preliminary: the purpose of presenting them is to better grab the generic PN model definition.

A stereo camera requires no data in input and produces an image and a point cloud. Recalling the model described in Eq.1, its instantiation is as follows:

Type: Sensor
Instance: Stereo camera

$$\begin{cases} U = \emptyset \\ Y = \{B/W\ Image, PointCloud\} \\ \Sigma = \{\#Points, \Sigma(error)\} \\ \Theta = \{Exposure, WhiteBalancing, Gamma, ...\} \\ \Psi = \{Illumination,\ Textures, ...\} \\ H = CPU\ Time,\ Power \end{cases}$$

For the first computation step, a *keypoint extractor* is used. These kind of nodes compute descriptors for selected regions of interest which are reliably recognizable under changes in scale, light and distortion. One of the possible choice could be the *SIFT descriptor*.

Type: Keypoint extractor
Instance: SIFT descriptor

$$\begin{cases} U = B/W\ Image \\ Y = \{KP_t^1,\ \ldots,\ KP_t^n\} \\ \Sigma = \{\#Features, \frac{\#AcceptableFeatures}{\#Features}^2\} \\ \Theta = \{PeakThreshold, EdgeThreshold, ...\} \\ \Psi = \{Illumination,\ Textures, ...\} \\ H = CPU\ Time, Memory \end{cases}$$

This process takes in input a black and white image and produces a set of keypoints at a given time instant *t*. The KPs are expressed in the sensor reference frame. In the case of SIFT, parameters are the *peak threshold* and the *edge threshold*.

The second step involves a data association algorithm. Its task is to match KPs extracted at different time steps.

[2]An acceptable feature is a feature that has been evaluated under some utility/quality terms, see for instance (Hartmann et al. 2014) for feature matchability

Type: Data association
Instance: SIFT Matcher

$$\begin{cases} U = \{KP_t^1, \ldots,\ KP_t^n\},\ \{KP_{t+1}^1, \ldots,\ KP_{t+1}^n\} \\ Y = \{\ [KP_t^1,\ KP_{t+1}^1],\ \ldots\ ,[KP_t^n,\ KP_{t+1}^n]\ \} \\ \Sigma = \{Recall, 1 - precision^3, F1 - score, ...\} \\ \Theta = \{MatchThreshold, ...\} \\ \Psi = \{\Sigma(KeypointExtractor)\} \\ H = CPU\ Time \end{cases}$$

Since we used the SIFT descriptor to extract keypoints from the images, we will use an algorithm that matches SIFT features which can be called *SIFT Matcher*. The algorithm receives in input two different sets of KPs extracted at time *t* and *t+1*, then proceeds to find all the possible matches in these two sets. The output is a set of pair of KPs where every pair represent the same 3D point in a world reference frame. Finally, the last step of visual odometry requires a motion estimation algorithm to estimate the motion between the two considered time instants.

Type: Motion estimation
Instance: Least Square Method

$$\begin{cases} U = \{\{KP_t^1,\ KP_{t+1}^1\}, \ldots\ ,\{KP_t^n,\ KP_{t+1}^n\}\} \\ Y = \{T_{t\to t+1}, \sigma\} \\ \Sigma = \{P, \%error\}^4 \\ \Theta = \{\% - outliers, ...\} \\ \Psi = N/A \\ H = CPU\ Time \end{cases}$$

Least Square Method takes in input the set of matches computed by the chosen matcher and produces two pieces of output: an homogeneous transformation minimizing the reprojection error and an uncertainty measure expressed as a covariance matrix (Howard 2008).

VII. CONCLUSIONS

We have presented on-goig work on an approach to model and taxonomize perception processes. We have shown how task oriented perception processes can be subdivided in several sub-processes that we refer to as perception nodes. The composition of such nodes can be generalized and is driven by the matching of inputs and outputs. Although this is only a formalization, this modeling will enable autonomous reconfigurability of perception processes thanks to the use of a decision making framework, for instance planning approaches: planning generally works well with well defined rules which permit to abstract from the specific domain of the problem. This is reflected in the abstraction from the nature of the sensors or of the perception nodes. Other decisional frameworks may be considerd, such as decision theory, constraint-based or resource-based programming, optimisation...

[3]See (Mikolajczyk and Schmid 2005)

[4]See (Mihaylova et al. 2005; Nistér et al. 2006)

The proposed model is still preliminary, and in particular lacks the definition of the geometric dimensions of some perception nodes (in particular for nodes of Sensor type, for which the notion of field of view and resolution must be explicited).

Finally, a major challenge in the configuration of perception compounds will be to have a rich representation of the world. The information modeled shall be the main driving factor for tuning parameters relative to algorithms and for on-board reconfiguration of perception compounds.

REFERENCES

R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76 (8):966–1005, 1988.

S. Chen and Y. Li. Vision sensor planning for 3-d model acquisition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):894–904, 2005.

S. Chen, Y. Li, and N. M. Kwok. Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343–1377, 2011.

W. Hartmann, M. Havlena, and K. Schindler. Predicting matchability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9–16, 2014.

A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3946–3952. IEEE, 2008.

L. Mihaylova, T. Lefebvre, H. Bruyninckx, and J. De Schutter. Active robotic sensing as decision making with statistical methods. *Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management*, 198:129, 2005.

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

Y. Motai and A. Kosaka. Hand–eye calibration applied to viewpoint selection for robotic vision. *IEEE Transactions on Industrial Electronics*, 55(10):3731–3741, 2008.

D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23 (1):3–20, 2006.

R. Pito. A solution to the next best view problem for automated surface acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1016–1030, 1999.

E. Trucco, M. Umasuthan, A. M. Wallace, and V. Roberto. Model-based planning of optimal sensor placements for inspection. *IEEE Transactions on Robotics and Automation*, 13(2):182–194, 1997.