Workload Analysis of Autonomous Driving System: Apollo, Baidu Inc.

1st Alessandro Toschi
Shanghai Jiao Tong University
Shanghai, China
aleto_95@sjtu.edu.cn

2nd Jingwen Leng
Shanghai Jiao Tong University
Shanghai, China
leng-jw@sjtu.edu.cn

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

***Abstract*—Autonomous driving is a field that gathers many interest from the academics world and from industry leaders. The software of an autonomous driving systems (ADS) incorporates the state-of-the-art from many disciplines, such as computer vision, robotics, geo-localization. Although the high level architecture of an autonomous driving system and the main algorithms used are known, the complete analysis of a real ADS is still difficult, especially for what concerns the modules interdependencies, interactions, either software and hardware, and pre- and post-processing. In this paper, we want to extract those point of views and quantify them according different architectural aspects: response times, memory movements, computational complexity and CPU-GPU relationship. The analysis is based on the open-source Apollo ADS developed by Baidu and is focused on the most important modules: perception, prediction and planning.**

I. Introduction

Autonomous driving system has several design constraints [1] to be met in order to produce a safe and reliable output.

Response time [1] is crucial for the predictability and accuracy of the system, especially when multiple sensors and components are present, each of them with a processing routine associated. The maximum response time, which has been adopted as standard in the field of autonomous driving, is 100 ms and should ensure a proper and safe reaction to any possible situation. Several processing routines use time deltas to perform corrections and projections of input and if those time-deltas are exceeding context-related thresholds then the input is discarded, losing some potential useful information, thus limiting the response time will affect also the accuracy of the system.

Apollo is a modular data-driven ADS, containing several modules, each of them pursues an high level feature of autonomous driving, such as perception, prediction, planning, control, localization. Modules can be treated as black boxes and described in terms of input/output relationships. This characterization enables the analysis of modules independently, provided that the inputs fed are feasible. Modules are further expressed in terms of set of components, which represent lower level tasks. Each component follows the same paradigm of modules, which means interactions within a module are based on input/output relationships through a publisher and subscriber architecture.

Cyber is the Apollo's runtime framework that implements the communication among components. The publisher and subscriber communication adopted by Cyber is based on channels and messages. Messages are serialized objects, using the Google's Protocol Buffer, which then are broadcast on channel(s). Each component can be a reader or writer of multiple channels at the same time.

Apollo supports multiple sensors, different prediction evaluators and several scenario-based planners. This rich environment carries out the need of having the right hardware equipment to support each task. CPU should be able to sustain many multi-threaded algorithms and provides enough cores to execute several processes concurrently. GPU is required for the execution of convolutional neural networks (CNN), vector and matrix operations, which are especially encountered in the perception module.

A schematic overview of the Apollo software architecture is presented in Figure 1. Although the communication is data-driven, is reasonable to describe the architecture starting from sensors. Apollo supports two Full-HD cameras with two different focal lengths, 6mm and 12mm respectively, which are polled at 20 fps. Two types of lidars are used in Apollo: 128L and 16L. The 128L lidar is set to be the master sensor in the architecture. GPS and IMU data are provided to the localization module in order to estimate the car's position and pose. The perception detects obstacles present in the environment perceived from cameras and lidars and then the prediction module predicts obstacle trajectories using detections, localization estimations and trajectories chosen by the planner. Finally, the planning module selects the best trajectory, coherent to the route requested and obstacle trajectories, containing the path and speed profile to adopt. Many modules rely on HDMap to get an enriched understanding of the surrounding environment, being able to query road signs, junctions, lanes and many more.

In the next sections, we will describe the perception, prediction and planning modules. The description will be composed by an high level overview of the module and the analysis of each component within the module. The analysis is focused on response times, computational complexity, data dependencies, CPU-GPU relationship and memory movements. In the last section, we will detail the simulation environment, settings and datasets used in this work.

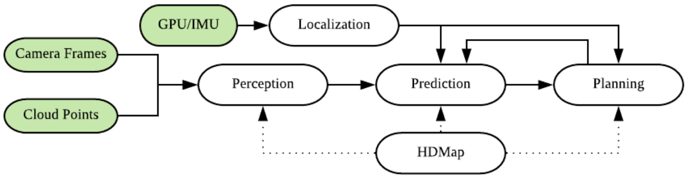


Fig. 1: Apollo Software Architecture

II. Perception

The perception module perceives the environment, through the data coming from cameras and lidars, by detecting obstacles. In the Figure 2a is presented the internal representation of the perception modules, highlighting the module's components and flow. Due to the different nature of the inputs, camera frames and cloud points require different detection algorithm, pre- and post-processing. The Fusion Camera Detection is a unique component shared from both 6mm and 12mm camera frames and processes them using first-in, first-served policy through an Obstacle Camera Pipeline. The Lidar Segmentation is bounded to a specific Lidar because different lidars have different parameters that affect the segmentation. Thus, cloud points are published into distinct Cyber's channels according to the lidars and the segmentation components are driven by those channels concurrently. The ouput of lidar segmentation doens't depends anymore on the lidar of origin, the recognition component indeed is unique and shared. Intermediate detections converge on the Fusion component, which will provide coherent and combined detections among sensors, publishing them as the output of the whole module.

A. Fusion Camera Detection

The Fusion Camera Detection applies a sequence of tasks, called Obstacle Camera Pipeline to camera frames, displayed in Figure 2a. The pipeline is composed by lanes detection and post-processing, obstacles tracking and obstacles detection and post-processing. The pipeline is not applied equally to each frame; lanes detection and post-processing are only applied to camera frames coming from the 6mm camera, due to the wider perspective.

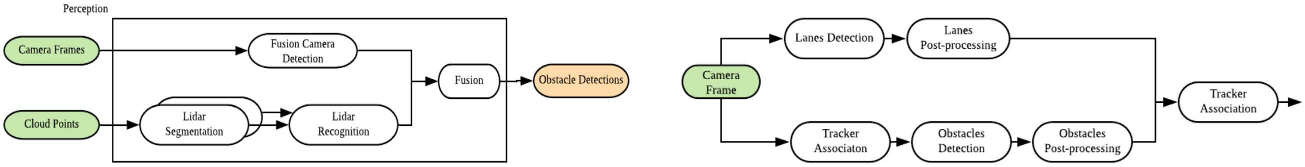
Lanes detection is performed using SCNN [2], spatial convolutional neural network, which exploits spatial relationship among pixels in order to identify straight shaped obstacles, such as lanes. To fit the SCNN input size, the camera frame is resized from 1920x1080 to 640x480 and then forwarded through the network. The network detects at most 13 lanes, the output indeed can be interpreted as 13 frames of size 640x480 each of which highlights a lane if present. This sparse output is then fused to a single 640x480 frame, in which different lanes are highlighted by different identifiers.

Lanes post-processing needs to be applied because lanes' representation is not yet meaningful for the subsequent modules in the ADS. The lanes found so far belong to the camera plane and need to be projected onto car plane and ground plane. For each lane, the third-order polynomial coefficients are estimated from the ground plane projection using the Ransac-Fitting algorithm. Finally a further projection is executed, converting the lane points from the two bi-dimensional planes onto the three-dimensional plane, represented in world coordinates.

Obstacles are detected using 'YoloNET' convolutional neural network which is able to identify 2D bounding boxes of obstacles and their classes. Likewise the lanes detection, the camera frame is resized to fit the size of the neural network. After the inference, the 2D bounding boxes are projected onto the 3D by the obstacle post-processor.

The obstacles tracker adapts the detections of the previous iteration to the current one by estimating the new bouding boxes position according an Adaptive Kalman Filter, car pose and time-deltas. Afterwards, the obstacles tracker matches new obstacles detected to the previous ones, generates hypothesys and deletes the duplicates. Finally, the obstacles detected are propagated towards the Fusion component.

Complexity: The computational complexity of the whole pipeline is dominated by CNN inferences. Lanes detection, in addition, depends on image resizing and the fusion of detected lanes into one frame. Lanes post-processing performs several operations on lane points of each lanes detected and the iterative RANSAC algorithm to find the polynomial coefficients. The tracker prediction relies only to the past obstacles detected and the update of the Adaptive Kalman Filter. The obstacles



(a) Perception Software Architecture

(b) Obstacle Camera Pipeline

Fig. 2

detection, similarly to the lane detection, resizes the image and performs some refinements on the obstacles candidates to refine their bounding boxes. The obstacles post-processing and tracking association involve the comparison between past obstacles and current obstacles.

Task	**Complexity**
Lanes Detection	CNN inference, image resizing and single lanes frame
Lanes Post-processing	RANSAC algorithm, lane points x lanes detected
Tracker Prediction	Past obstacles detected
Obstacles Detection	CNN inference, image resizing, obstacles candidates, obstacles detected
Obstacles Post-processing	Obstacles detected
Tracker Association	Obstacles detected x past obstacles

TABLE I: Fusion Camera Detection Complexity

Response time: The average response time of the component in our simulation is 49.76 ms and the 95.86% of time is spent on CNN inferences. This dominance of CNN inferences lead to make the computation independent from the number of obstacles detected, even if the pre- and post-processing operations are needed to generate a correct detection. This component heavily relies on GPU for the CNN inferences, image resizing, the extraction of the obstacles bounding boxes from the YoloNet output and the similarities between obstacles and frames in the tracking phase.

Task	**Response Time**	
	ms	*%*
Lanes Detection	38.14	76.65
Lanes Post-processing	0.72	1.45
Tracker Prediction	0.01	0.02
Obstacles Detection	9.56	19.21
Obstacles Post-processing	0.6	1.2
Tracker Association	0.73	1.47

TABLE II: Fusion Camera Detection Response Time

B. Lidar Segmentation Component

1) What it does and how: Explain which are the input of module, output and tasks. Explain what each task does.

2) Complexity: Explain the complexity of the tasks and their dependencies

3) Response time: Response time analysis and on which device each task runs. Table or graph about response times.

C. Lidar Recognition Component

1) What it does and how: Explain which are the input of module, output and tasks. Explain what each task does.

2) Complexity: Explain the complexity of the tasks and their dependencies

3) Response time: Response time analysis and on which device each task runs. Table or graph about response times.

D. Fusion Component

1) What it does and how: Explain which are the input of module, output and tasks. Explain what each task does.

2) Complexity: Explain the complexity of the tasks and their dependencies

3) Response time: Response time analysis and on which device each task runs. Table or graph about response times.

III. Prediction

IV. Planning

V. Memory Throughput Simulation

Analyze the impact of accelerating, through a PCI device, the inference of CNN in terms of memory movements from the GPU/CPU to PCI Device.

VI. Simulation Details

Datasets, gpu, cpu, software used and so on

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

VII. Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections VII-A–VII-E below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LATEX will do that for you.

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: “Wb/m^2” or “webers per square meter”, not “webers/m^2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.

- Use a zero before decimal points: “0.25”, not “.25”. Use “cm^3”, not “cc”.)

C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{1}$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

D. LaTeX-Specific Advice

Please use “soft” (e.g., `\eqref{Eq}`) cross references instead of “hard” references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don’t use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you’ve discovered a new method of counting.

BibTeX does not work by magic. It doesn’t get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can’t read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it’s supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won’t be any anyway) and it might stop a wanted equation number in the surrounding equation.

E. Some Common Mistakes

- The word “data” is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
- Do not use the word “essentially” to mean “approximately” or “effectively”.
- In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
- Do not confuse “imply” and “infer”.
- The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the “et” in the Latin abbreviation “et al.”.
- The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

G. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

H. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. **??**", even at the beginning of a sentence.

TABLE III: Table Type Styles

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a] Sample of a Table footnote.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

Acknowledgment

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] Lin, S. C., Zhang, Y., Hsu, C. H., Skach, M., Haque, M. E., Tang, L., & Mars, J. (2018, March). The architectural implications of autonomous driving: Constraints and acceleration. In ACM SIGPLAN Notices (Vol. 53, No. 2, pp. 751-766). ACM.

[2] Pan, X., Shi, J., Luo, P., Wang, X., & Tang, X. (2018, April). Spatial as deep: Spatial cnn for traffic scene understanding. In Thirty-Second AAAI Conference on Artificial Intelligence.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.