ICRA 2022 IEEE International Conference on Robotics and Automation

May 23-27, 2022 Philadelphia (PA), USA

IEEE Robotics & Automation Society IEEE

### Acceptance Decision and Reviews of Submission 456 for ICRA 2022 (January 31, 2022)

Home Access Workspace PIN Refresh Log out Contact the ICRA 2022 organizers Contact Technical Support

All deadlines are 23:59:59 Pacific Time. Current time 13:04:47

Kyle Vedder 219789 (Author). Your current session expires in 56:08

Cancel the page

| Acceptance Decision and Reviews of Submission 456 for ICRA 2022 | |
|---|---|
| Submission number | 456 |
| Authors or proposers | Kyle Vedder*, Eric Eaton |
| Title | Sparse PointPillars: Maintaining and Exploiting Input Sparsity to Improve Runtime on Embedded Systems |
| Scroll down to view the publication decision and reviews when and if available | |

**iThenticate Scan Results**

ICRA 2022 has access to the CrossCheck database. Read more about the CrossCheck initiative to prevent scholarly and professional plagiarism.

Similarity scores and scan reports are generated by an external provider (iThenticate) and not by the conference.

The scan reports are stored on the iThenticate servers and not downloaded to the conference submission system servers. Eventually the reports are deleted from the iThenticate servers at a time determined by conference and provider policy, after which they are no longer available.

Please check the on-line report to understand how the similarity score was determined and which sections of your paper may need changes.

The score and report are provided to assist you in revising your manuscript, if necessary.

The conference will not be able to provide you a score or a report for the revised manuscript.

Download several pages from the iThenticate User Manual with instructions on how to read and interpret the report

| | |
|---|---|
| iThenticate scan report | File 0456_MS.pdf, scanned September 17, 2021, Similarity score 44%.<br>View the report in a new window or tab. You may need to turn off the pop-off blocker of your browser |

| Publication Decision for this Submission | |
|---|---|
| **Timestamp** | January 31, 2022 12:20:48 |
| **Decision** | Rejected |
| **Decisions on attachments** The inclusion of attachments in the conference program is subject to the acceptance of the submission itself | Video Attachment: Received |

**Cover message**

```
To: Dr. Kyle Vedder
Re: ICRA 2022 Contributed paper 456: Sparse PointPillars: Maintaining and Exploiting Input Sparsity to Improve
on Embedded Systems

Dear Kyle,

We regret to inform you that your paper has not been accepted for presentation at the 2022 IEEE International
Conference on Robotics and Automation (ICRA), May 23-27, 2022, Philadelphia (PA), USA. We received 3,313 paper
submissions; from the very large number of high-quality papers we selected 1428 for presentation, which repres
acceptance rate of 43.1%.

You can access the reviews and editorial board comments, as well as the iThenticate report, by logging into yo
PaperPlaza account.

CONFERENCE DECISION PROCESS
Each contributed ICRA paper was reviewed by reviewers, an Associate Editor and an Editor of the Conference Edi
Board (CEB), under the supervision of the CEB Editor-in-Chief. The CEB comprises 24 editors, 458 associate edi
5,854 reviewers. RA-L papers with the ICRA option were reviewed by the RA-L editorial board.

Final decisions about inclusion in the ICRA 2022 program were made by the Senior Program Committee (SPC) on Ja
2022. All decisions are FINAL.

We know that it is disappointing to have a paper rejected. We hope the reviews are useful and that we will see
Philadelphia (PA), USA or virtually at ICRA 2022 for what promises to be a strong and exciting technical progr
conference.

Sincerely,
```

| | |
|---|---|
| | Hadas Kress-Gazit, Aisha Walcott-Bryant, Hanna Kurniawati<br>ICRA 2022 Program Chairs<br><br>Marcia O'Malley<br>ICRA 2022 Editor-in-Chief, ICRA Editorial Board<br><br>George J. Pappas, Vijay Kumar<br>ICRA 2022 General Chairs |
| **Comments to the author** | Comments to author (Associate Editor)<br>=====================================<br><br>The paper improves the run-time performance of existing methods for 3D object detection from point clouds (PointPillars) by exploiting the sparse nature of the input while sacrificing detection performance.<br><br>However, this paper is primarily a combination of several works by other groups [3][5][6], thus there is very limited technical contribution. Also, there is no quantitative and qualitative comparison with other sparse convolution-based detectors like SECOND/SBNet.<br><br>----------------------------------------<br>Comments on Video Attachment:<br><br>slow down the narrative<br>Comments to author (Editor)<br>===============================<br><br>Authors are highly encouraged to address the reviewer comments.<br><br>----------------------------------------<br>Comments on Video Attachment:<br><br>Good complement to the paper. |

**Reviews of this Submission**

| Reviewer number | Review ID | Comments to the author | Attachment to the review | Comment on the Video Attachment |
|---|---|---|---|---|
| 8 | 14795 | Summary:<br>The paper improves the run-time performance of an existing method for 3D object detection from pointclouds (PointPillars) by exploiting the sparse nature of the input while sacrificing detection performance.<br><br>The main contribution is the modification of the Feature Net and Backbone of the PointPillars using existing methods that exploit input data sparsity. In particular, in the Feature Net, rather than forming a dense pseudoimage from the feature vectors, a tensor format from prior work (coordinate format) is used to construct a sparse pseudoimage. In the Backbone, 3x3 stride-2 convolutions are replaced by 2x2 stride-2 convolutions, and sub-manifold convolutions (from prior work) are used instead of full convolutions. A combination of the tensor format used and the type of convolutions used results in the maintainance of the input sparsity in the final output of the Backbone with fewer computations.<br>The method is evaluated on two datasets - the KITTI dataset and a dataset constructed from Matterport3D for chair detection. The results show that the author's method has a lower runtime and a lower object detection performance. The runtime decreases even more significantly when inference is on a CPU.<br><br>Overall impression:<br>The paper addresses a very relevant problem in robotics, namely, being able to run state-of-the-art machine learning models on limited hardware resources while still maintaining a reasonable performance. The novelty is limited, however, since the authors use existing methods for the base model and the modifications. The descriptions of the modifications performed are good, but could do with more details (described below). The experimental evaluation is valuable since it clearly shows the benefits of the authors' modifications, particularly when used on typical hardware available on mobile robots. It is also very helpful that all materials (code, models, weights etc.) are provided to reproduce the results.<br><br>Some detailed comments:<br>Fig 2: If it can be incorporated without too much clutter, the type of modifications done at each stage would be a useful addition to the figure (e.g. a label for sparse psuedo image, and sparse conv/SubM conv in the backbone)<br><br>III A: it is a bit unclear what the terms "gather" and | | The video provides a useful summary of the paper. it appears that the audio has been sped up, which difficult to follow unless slowed down. I suggest down the content so that the audio is of normal s |

| | | | | |
|---|---|---|---|---|
| | | "scatter" mean in this context. They may be terms used in the original PointPillars paper, but should be clarified here.<br><br>III A: it would be helpful to shortly describe what the COO format is<br><br>III B: "uses sparse convolutions [27] and ..SubM convolutions [6]" -- the way in which SubM convolutions are used is clear, but it is not clear where/how sparse convolutions are used. Is it the replacement of 3x3 stride-2 convolutions with 2x2 stride-2 convolutions? What exactly is a sparse convolution?<br><br>Fig 4: the caption mentions that BatchNorm is only used on non-zero entries but this is not mentioned in the main text. Is this a feature of the SubM convolutions, or is it an additional step? Fig 3. also mentions BatchNorm being removed. Is this only for illustration purposes, or is this performed in the actual implementation of the original PointPillars? In general a few sentences addressing BatchNorm should be mentioned in the main text.<br><br>Fig 5: this illustration is very helpful in understanding SubM convolutions<br><br>IV: are the densities mentioned associated with the selected pillar and pseudoimage sizes mentioned in V?<br><br>Table 1: the table lists several expressions for the number of convolutions performed in both versions, but it is not clear how these expressions were arrived at. IV mainly discusses the implications of the input density on Sparse Pillars. A description or derivation of at least one of the terms in the table would be helpful in understanding the process by which they were derived.<br><br>Table 1/Fig 6: Conv^T are presumably deconvolutions - this can be explicitly mentioned in the text or captions<br>Fig 6: the size of each of the Conv^T filters can be mentioned here, since they are mentioned in Table 1<br><br>V A: What was the intended purpose and expected result of the two abalations?<br><br>V A: How does the 9x9 Conv simulate smearing if the SubM conv only convolves when centered on non-zero entries? Does this depend on the density of non-zero entries?<br><br>V A: Given that for the Matterport dataset, the sparse Backbone runs faster that the baseline on a CPU (compared to slower on a GPU), it would be useful to see a similar result with the KITTI dataset.<br><br>Code: A pointer in the README to the particular files where the contributions of your work can be seen can be added (e.g. where is the pseudoimage represented with the COO format, or where are SubM convolutions used)<br><br>Minor comment regarding phrasing: the term 'maintain and exploit' is perhaps used too often in the first few sections of the paper (including twice in the abstract). It should probably be emphasized again in the conclusions though. | | |
| 9 | 39665 | This paper proposes to use sparse convolution to accelerate the inference of PointPillar for efficient 3d object detection, which is desirable in robotics and autonomous driving. Overall, the results seem good, and the presentation is good. However, this paper is mainly just a combination of several works: [3][5][6], thus there is nearly no technical contribution. Also, there is no quantitative and qualitative comparison with other sparse convolution-based detector like SECOND/SBNet in the experiments.<br><br>Strongness:<br>[1] The results seem good, and the presentation is good.<br>[2] The code is released, and the video seems good.<br>[3] The experiments are good in terms of efficiency, and this is really useful for the hardware implement in the real-world scenario.<br><br>Weakness:<br>[1] The novelty is limited, this paper is mainly just a combination of several works: [3][5][6], thus there is nearly no technical contribution.<br>[2] There is no quantitative and qualitative comparison with other sparse convolution-based detector like SECOND/SBNet in the experiments. What's your key advantages compared to the existing sparse detectors?<br>[3] The figure is copied from the original PointPillar paper  without any modifications, which makes me suprised.<br><br>Conclusion:<br>The novelty is limited and the experiments are not comprehensive. Therefore, the authors are suggested to | | good |

| | | | | |
|---|---|---|---|---|
| | | improve their work in terms of technical novelty and experiments to make it publishable in ICRA. | | |

Cancel