

인접 픽셀 정보를 이용한 Shift-Convolution 기반의 3D LiDAR 깊이 완성

유 병 준¹⁾ · 기 석 철²⁾

충북대학교 스마트자동차공학¹⁾ · 충북대학교 지능로봇공학²⁾

3D LiDAR Depth Completion Based on Shift-Convolution Using Adjacent Pixel Information

Byeong-Jun Yu¹⁾ · Seok-Cheol Kee^{*2)}

¹⁾Department of Smart Car Engineering, Chungbuk National University, Chungbuk 28644, Korea

²⁾Department of Intelligent Systems and Robotics, Chungbuk National University, Chungbuk 28644, Korea

(Received 9 March 2021 / Revised 5 May 2021 / Accepted 8 May 2021)

Abstract : In an automated driving system, recognizing range information is essential in order to understand the surrounding environment. As a result, we proposed depth completion method, filling the area with depth information of point cloud, which is projected onto the image plane, and high resolution color data from the image. The projected point cloud is placed into the shift-convolution network which expands received sparse LiDAR data to the pixel level, and then it is inserted synchronously into the convolutional neuron network(CNN) with image. Fully completed ground truth is formed by using max and median filters sequentially, and it is taken as input of shift-convolution to make an expanded point cloud that focuses more on completing an empty area than section the contour off. Finally, CNN uses point cloud to get the exact depth information and image for separating objects along the outline. The system that uses expanded point cloud has approved almost 9 % more than the system that does not.

Key words : Heterogeneous sensor calibration(이종 센서 캘리브레이션), CNN(콘볼루션 뉴런 네트워크), Depth completion(깊이 완성), Autonomous driving(자율주행), LiDAR(라이다)

Subscripts

- P : point cloud
- I : image
- GT : ground truth
- Shift-Conv = shift-convolution
- E-Point = expanded point cloud

1. 서론

1.1 연구 배경

현재 ADAS(Advanced Driver Assistance System) 및 자율주행 기술은 수많은 딥러닝 기반의 인지 관련 연구¹⁻⁴⁾가 진행되고 우수한 연구 결과를 보고하고 있다. 특히, 자율주행 기술에서는 안전한 주행을 보장하기 위해서

정확한 깊이를 얻어 주변 환경을 더 정밀하게 이해할 수 있게 하는 깊이 추정 연구가 최근 중요해지고 있다. 이러한 연구들은 신뢰성이 있는 깊이정보를 얻기 위해 카메라, LiDAR(Light Detection And Ranging), Radar(Radio Detection and Ranging) 등과 같은 다양한 센서의 깊이 데이터를 활용한다. 하지만 깊이 센서의 데이터 밀도는 매우 낮아 채널의 개수 등과 같은 한계점이 명확하다. 그에 비해, 카메라와 같은 비전 센서는 해상도가 상대적으로 높지만, 단안 카메라는 깊이 측정이 불가하고, 스테레오 시스템은 거리가 멀어질수록 깊이 정확도가 급격히 낮아진다. 이러한 한계점을 극복하기 위해 깊이 정보를 제공하는 LiDAR 센서와 풍부한 정보를 제공하는 카메라 센서의 캘리브레이션을 활용하여 데이터를 융합하고 딥러닝 시스템을 적용하여 LiDAR의 부족한 깊이 정보를

*Corresponding author, E-mail: sckee@chungbuk.ac.kr

^{*}This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium provided the original work is properly cited.

이미지의 픽셀레벨로 복원하는 깊이 완성 알고리즘이 활발하게 연구된다. 기존의 연구는 LiDAR를 이미지 도메인으로 투영시킨 것과 이미지의 데이터를 원형 그대로 CNN의 입력으로 활용한다. LiDAR의 Point cloud 데이터는 이미지 도메인의 픽셀에 비하면 희박한 데이터 구성을 갖고 있고 커널 단위로 연산을 수행하는 단순한 컨볼루션 방식으로는 비효율적이다. 이런 문제를 해결하기 위해 LiDAR에 적용하는 딥러닝 기법을 컨볼루션과 같이 커널 레벨의 마스크 연산기반이 아닌 주변 데이터와의 연관성을 고려하는 새로운 딥러닝 기법을 제안하였다.

1.2 관련 연구

Fig. 1은 딥러닝을 이용한 Semantic,⁵⁾ Instance Segmentation⁶⁾의 예시이다. 최근 이미지 데이터를 이해하여 물체의 종류와 각 물체, 배경을 구분하는 Segmentation과 관련해 많은 연구가 나오고 있고 그 성능 또한 훌륭하게 나오고 있다. 영상에 단순히 하나의 물체만 존재하고 그것을 배



Fig. 1 (a) DeepLabV3, Examples of Semantic Segmentation,⁵⁾ (b) Mask R-CNN, Instance Segmentation⁶⁾

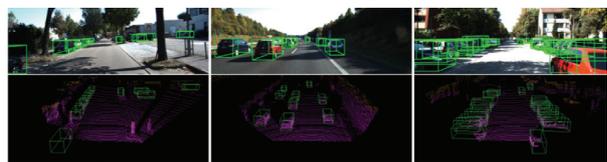
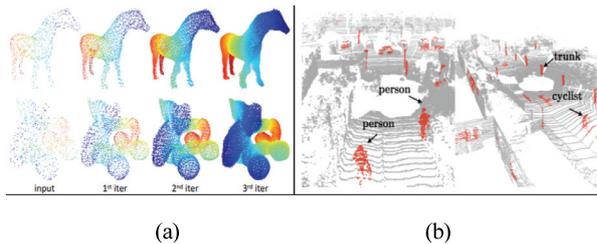


Fig. 2 (a) Example of PU(Point Upsampling)-Net⁸⁾, (b) 3D Semantic Segmentation⁹⁾, (c) 3D Object Detection¹⁰⁾

경과 구분하는 것을 넘어서 여러 물체가 겹쳐 있거나 물체를 구분하기 어려운 매우 복잡한 장면의 성능까지 만족할 수준으로 연구가 진행되었다. 이러한 알고리즘들은 단안 카메라를 입력으로 받고 CNN 레이어를 이용하여 결과를 산출한다. 고해상도의 표현을 출력하기에 물체의 윤곽, 작은 물체 감지 등 고 난이도의 문제를 해결한다.⁷⁾

이미지 기반 딥러닝의 연구가 활발해짐과 동시에 항상 카메라 센서와 비교되는 LiDAR에 관련된 연구도 활발해졌다. Fig. 2는 실 좌표계에서 LiDAR Super Resolution, 3D Semantic Segmentation, 3D Object Detection 문제를 해결하려고 한 연구들이다. 이러한 연구들은 LiDAR 데이터 자체만을 이용하여 단순한 CNN이 아닌 일정 볼륨을 하나로 합치는 Voxel,⁹⁾ 포인트 별로 하나씩 접근하는 MLP(Multi Layer Perceptron)⁸⁾와 같이 빈공간을 참조하는 컨볼루션 연산의 비효율을 타파하려는 연구들이 있다.

2. Point Cloud 데이터 구성

이번 장에서는 비정형 데이터인 Point cloud를 이미지 평면으로 변환하는 과정에서 생긴 캘리브레이션 오차와 카메라-LiDAR 센서의 싱크 문제로 인한 오차 그리고 GT와 데이터 밀도가 희박한 Point cloud의 데이터 특성에 대한 내용을 다루고 3장에서 소개할 Shift-Conv이라는 새로운 방법을 학습하기 위한 데이터 구성을 설명한다.

모든 데이터는 KITTI dataset¹¹⁾을 활용하였다.

2.1 Point Cloud의 이미지 평면 투영

Point cloud 데이터를 이미지 평면으로 투영하는 데카르트 변환식은 식 (1)과 같다.

$$I_{u,v} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} L_{x,y,z} \quad (1)$$

여기서 I는 이미지 좌표계의 점 (u, v)를 의미하고 L은 데카르트 좌표계의 점 (x, y, z)를 의미하고 r은 변환 파라미터를 의미한다. 카메라와 LiDAR의 포즈를 추정하는 과정에서 오차가 발생하여 변환 행렬을 구성할 때 오차가 더해진다. 부정확한 센서 외부 파라미터 추정에 더해 LiDAR 센서가 데이터를 출력하기 위해 구좌표계에서 데카르트 좌표계로의 변환이 이뤄지는 과정에서의 소수점 누락으로 인한 오차, 센서간 싱크 문제로 인한 오차 등이 누적된다. 이러한 센서 캘리브레이션 문제 때문에 물체 간의 외곽선 구분이 어려워지는 결과가 발생하게



Fig. 3 Occlusion problem of the projected point cloud

된다. 하지만 본 연구에서는 딥러닝 레이어의 데이터 처리 과정에서 이미지의 색상정보를 이용하여 부정확한 캘리브레이션 오차를 해결하였다.

Fig. 3은 이미지 평면으로 투영된 Point cloud가 이미지가 제대로 정돈되지 않은 문제와 Point cloud의 서로 다른 물체위에 놓인 점들이 겹쳐 외각선을 흐릿하게 만드는 문제를 보여준다.

2.2 Point Cloud와 GT의 데이터 구성

이미지와 LiDAR 포인트들의 캘리브레이션 문제와 별개로 Point cloud 데이터 자체의 희박한 밀도 특성이 존재한다.

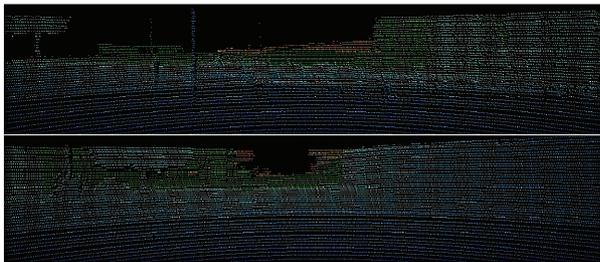


Fig. 4 Examples of the projected point cloud

Table 1 Distribution table of the projected point cloud

	Number of pixels	Ratio
Empty pixel	286,515.1	93.48 %
Valid pixel	19,984.9	6.52 %

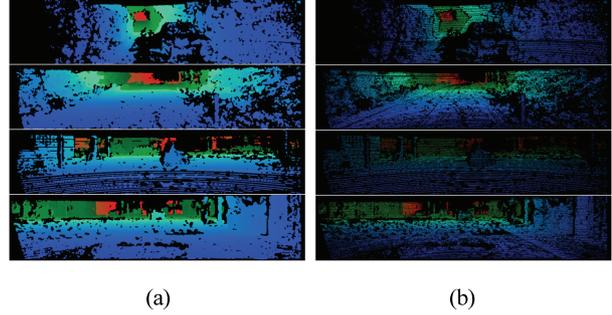


Fig. 5 Examples of (a) expanded GT and (b) original GT

Fig. 4는 이미지 평면에 투영된 LiDAR 포인트들의 예시이고 Table 1은 이미지 평면으로 투영된 Point cloud들의 데이터 분포도이다. 깊이정보가 채워져 있는 픽셀의 비율은 약 7%로 이러한 희박한 밀도 특성은 네트워크 안의 컨볼루션 연산을 비효율적으로 만든다.

우리가 제안하는 네트워크를 학습시키기 위한 GT 데이터는 스테레오 카메라와 LiDAR를 이용하여 구성하였다.

Fig. 5는 영상처리 기법이 적용되어 팽창된 GT와 원본 GT의 예시이다. 원본 GT는 빛이 강하거나 그림자가 많이 드리운 부분, 이미지의 좌우 끝에는 정보가 존재하지 않고 LiDAR에 비해 밀도가 높지만 여전히 상대적으로 이미지에 비해 데이터 밀도가 열다. 팽창된 GT는 원본 Point cloud 데이터를 입력 받아 깊이정보가 빈 곳 없이 채워진 이미지를 만드는 Shift-Conv를 학습시키기 위해 만들어졌다. 원본 GT의 밀도를 높이기 위해 최댓값 필터(팽창연산)를 씌워 빈 곳을 채우고 잡음 제거와 스무딩을 위해 중간값 필터를 씌워 팽창된 GT를 만든다.

3. 제안하는 알고리즘

이번 장에서는 데이터가 드물게 존재하는 LiDAR Point cloud 데이터의 밀도를 키우는 Shift-Conv와 깊이 완성에 사용되는 제안하는 CNN 구조에 대해 설명한다.

3.1 Shift-Convolution

이 절에서는 원본 LiDAR를 입력 받아 팽창시킨 E-Point를 만드는 목적을 가진 Shift-Conv 구조에 관해서 설명한다.

Fig. 6은 주변 픽셀 정보를 하나씩 순차적으로 받아들여 Shift-Conv 연산을 진행하는 구조를 도식화한 그림이다. 그림은 설명을 위해 한 픽셀만을 예시로 설명하였지만 Shift-Conv 연산은 모든 픽셀에서 진행된다. 데이터가 무효한 빈공간을 깊이 값으로 채우기 위해 입력 특징맵의 채널을 9등분하고 그 중 8개의 채널을 8방향으로 Shift

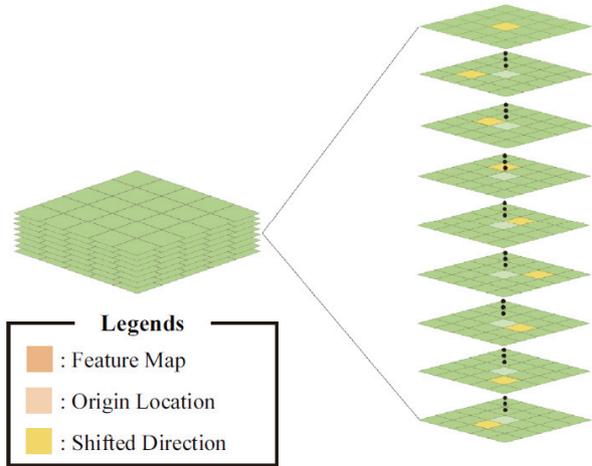


Fig. 6 Architecture of Shift-Conv. The green boxes mean feature map, the light green pixels mean origin location, and the yellow pixels mean shifted direction

시킨다. 그 후, (1×1) 컨볼루션을 수행하여 원본 특징과 주변 특징을 모두 비교하여 빈 공간 없이 깊이 값으로 가득 찬 데이터를 얻는다.

$$f\left(g\left(T_0(X_{9i}), T_1(X_{9i+1}), T_2(X_{9i+2}), \dots, T_8(X_{9i+8})\right)\right), i = 0, 1, 2, \dots, \frac{C-8}{9} \quad (2)$$

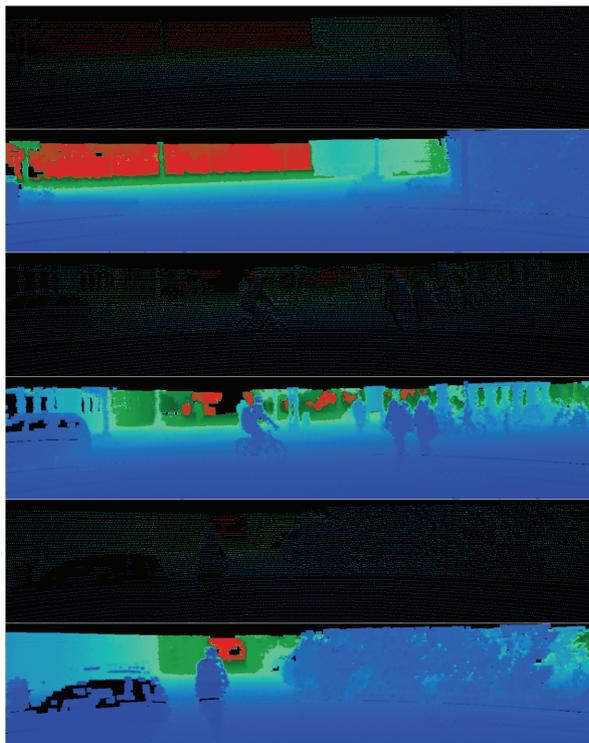


Fig. 7 Examples of the projected point cloud and E-point

식 (2)는 Shift-Conv 모듈을 수식으로 표현한 것이다. 여기서 f 와 g 는 각각 (1×1) 컨볼루션, 레이어를 이어 붙이는 연결(Concatenation)을 의미하고 T_d 는 원본을 포함한 8방향으로 특징맵을 Shift 시키는 9가지의 Translation 연산을 의미한다. X_c 는 입력벡터 X 의 c 번째 채널을 의미하고 C 는 입력벡터의 전체 채널의 개수이다.

Fig. 7은 이미지 평면으로 투영된 Point cloud와 Shift-Conv를 이용하여 팽창시킨 E-Point의 예시이다. 국소적인 주변부만의 정보만을 가져와 추론하므로 외각선이 뚜렷한 대신, 잡음에 약한 모습을 보여준다. 이러한 E-Point를 이용하여 성능을 개선하였다.

3.2 제안하는 네트워크 구조

팽창된 E-point와 원본 Point cloud, 이미지를 네트워크의 입력으로 사용하여 픽셀단위의 깊이를 복원한다.

Fig. 8은 제안하는 시스템의 알고리즘 흐름도이다. 제안하는 시스템은 Point cloud의 밀도를 팽창시키는 Shift-Conv와 깊이 완성을 수행하는 CNN을 End-to-end 방식으로 학습시키고 깊이를 추론한다.

Fig. 9는 제안하는 네트워크의 구조이다. 연산량이 매우 많은 HRNet¹²⁾의 레이어의 개수를 줄여 연산량을 줄인 네트워크로 모델을 구성하였다. 기존의 HRNet은 높은 단계의 표현 레이어 진행 횟수와 낮은 단계의 표현 레이어의 진행 횟수가 같다. 한정된 컴퓨터 자원과 연산 속도를 고려하여 가장 낮은 단계의 표현 레이어의 개수를 절반으로 줄였다. 인코딩 레이어의 입력으로 활용하기 위해 E-point, Point cloud, 이미지를 원본 데이터를 바로 입력하지 않고, 컨볼루션 레이어를 한번 수행한 후 입력으로 사용하였다. 위로 진행되는 화살표는 최근방

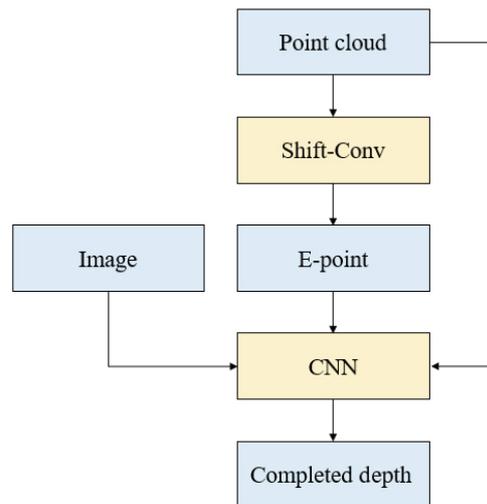


Fig. 8 Flow chart of the proposed system

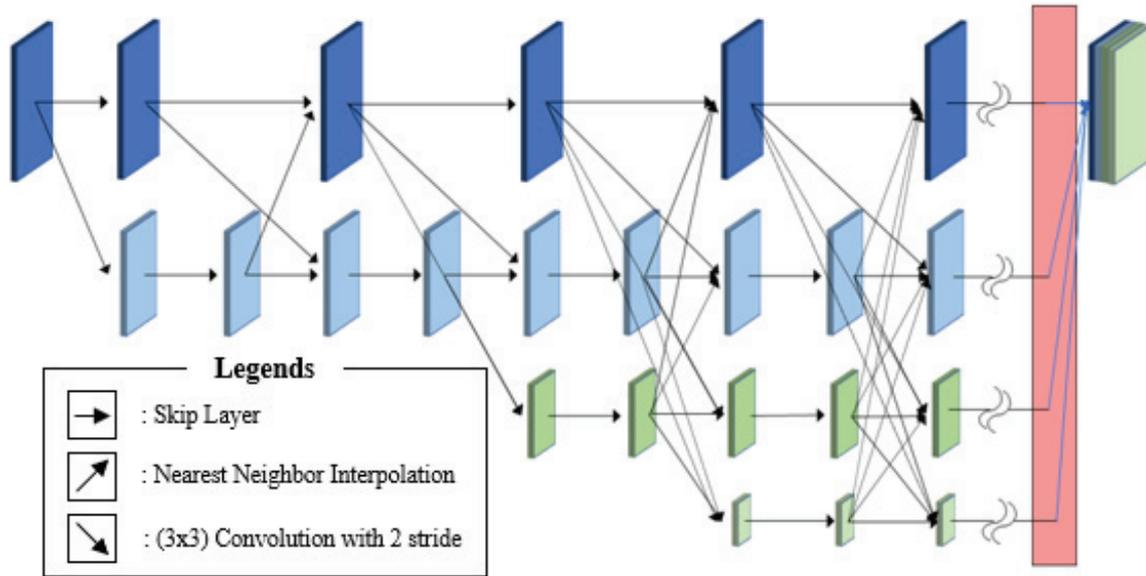


Fig. 9 Structure of the proposed network

이웃 보간법을 이용한 업샘플링을 의미하고, 수평으로 진행되는 화살표는 BottleNeck¹³⁾의 Skip-Layer, 아래로 진행되는 화살표는 Stride가 2로 설정되어 특징맵의 크기가 절반으로 줄어드는 하나의 (3×3) 컨볼루션 레이어를 이용한 다운샘플링을 의미한다. 데이터를 합칠 때는 합연산을 이용하고 최대 1/8 크기까지 줄어드는 다운샘플링 이후에 반복되는 레이어 구조는 물결표시를 이용하여 생략하였다. 물결 생략을 끝으로 인코딩 과정을 마친 후 복호 레이어의 입력으로 넣기 전에 원본 사이즈로 모든 특징맵을 키우고 연결하여 데이터를 구성한다.

복호 레이어는 컨볼루션 레이어 하나를 이용해, 픽셀 단위의 1채널 깊이 데이터를 출력한다.

4. 실험방법 및 결과

4.1 실험방법

본 연구의 실험은 Intel core i9-7900X 3.3 GHz CPU와 우분투 18.04 OS, NVIDIA GTX TITAN Xp GPU를 이용하여 실험을 진행했고 알고리즘을 평가하기 위해 KITTI dataset의 Depth completion 부분의 Evaluation 데이터셋을 사용하였다. 성능을 평가하기 위한 데이터셋은 학습에 사용되지 않은 데이터이며 학습용 데이터셋과 마찬가지로 모두 크기가 (1216×256)가 될 수 있도록 정규화하였다. 학습에 활용한 최적화 함수는 Adam¹⁴⁾을 사용하였고 학습율은 1e-5, Weight decay는 1e-4로 설정했다. Shift-Conv를 보조 학습시키기 위해 팽창된 GT를 사용했고 CNN을 학습시키기 위해 원본 GT를 사용했다. 사용

한 손실 함수도 사용하는 모듈마다 다르게 적용하였다. Shift-Conv는 EMD(Earth Mover's Distance)를, CNN에는 MSE(Mean Squared Error)와 RMSE(Root Mean Squared Error)를 활용하였다.

여기서 활용한 EMD 함수는 하나의 벡터에서 목표 벡터까지 옮기는데 드는 최소한의 비용을 계산하는 함수이다. 두 이미지간 거리 측정을 예로 들면,¹⁵⁾ $P = \{(p_m, w_{pm}), m = 1, 2, \dots, M\}$, $Q = \{(q_n, w_{qn}), n = 1, 2, \dots, N\}$, p 와 q 는 색, w 는 색의 비율인 두 이미지가 존재할 때 EMD는 P 와 Q 의 색상간 거리 측정을 하기 위한 최소비용을 계산한다. 최소비용을 계산하기 위해 p 에서 q 로 옮겨진 색의 양과 옮겨진 색의 거리의 곱을 구한다.

4.2 실험 결과

정량적 평가를 진행하는 평가 지표로써 가장 널리 쓰이는 RMSE와 동작속도를 이용하여 비교하였다. 데이터의 총 수는 1,000장이고 오직 GT에서 정답이 존재하는 유효한 픽셀에 대해서만 평가하였다.

Fig. 10은 네트워크 최종 결과물의 예시이다. 입력 이미지, Point cloud, E-point를 깊이를 완성하는 네트워크에 입력하여 출력 결과를 만든다. E-point를 이용해 먼 거리의 작은 물체도 정확하게 깊이를 측정하고 명확한 윤곽선의 출력을 만들었다. 하지만, 상대적으로 밀도가 희박한 부분의 결과는 물체와 배경의 경계가 모호해지거나, 외형이 제대로 형성되지 않는 결과가 만들어진다.

Table 2는 KITTI dataset을 이용하여 제안하는 방법의 성능을 평가한 표이다. 제안하는 방법은, 기존의 알고리

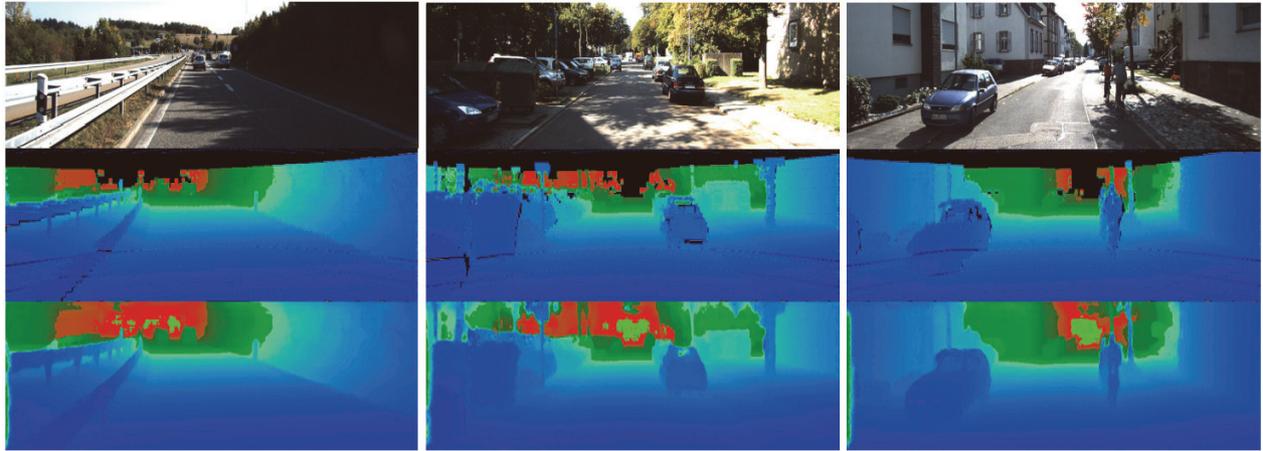


Fig. 10 Example of the results. (Top) is the input image, (Middle) shows E-point, and (Bottom) represents final result of the proposed system

Table 2 KITTI depth completion benchmark

	RMSE (mm)	Runtime (s)
CSPN ¹⁶⁾	1,019.64	1
DCd_3 ¹⁷⁾	1,109.04	0.1
DFuseNet ¹⁸⁾	1,206.66	0.08
Proposed Network	1,102.11	0.06

Table 3 Evaluations for ablation study

	w/ Shift-Conv	w/o Shift-Conv
RMSE	1,102.11	1,206.93

증들에 비해 약 16배 빠른 동작속도를 갖고 성능은 2.7% 가량 상승했다.

또한, 제안하는 Shift-Conv 모듈의 효율성을 입증하기 위해 Ablation 실험을 진행하였다. Shift-Conv를 제거한 시스템은 기존의 사용하던 CNN과 동일한 구조에 이미지와 Point cloud만을 입력으로 활용하여 깊이 완성을 진행하였다.

Table 3은 Shift-Conv의 성능을 보여주기 위해 Shift-Conv를 포함한 모델과 포함하지 않은 모델의 성능을 비교한 표이다. Shift-Conv를 이용하여 만든 E-point는 작은 물체의 적은 포인트 수, LiDAR의 Occlusion 문제, 적은 밀도 문제를 해결하였고, 이러한 특성 덕분에 Shift-Conv를 포함한 모델은 포함하지 않은 모델에 비해 8.7%의 성능향상을 가졌다.

5. 결론

본 논문은 단순히 이미지 평면에 투영시킨 Point cloud와 이미지 데이터를 딥러닝 시스템에 넣어 활용하는 것이 아닌, 데이터의 단점을 개선할 수 있는 전처리 딥러닝 모듈인 Shift-Conv를 연구하고 기존의 성능은 좋지만 파라미터가 많은 네트워크의 동작속도를 개선하는 연구를 진행하였다. Shift-Conv를 이용하여 무효한 픽셀이 많은 투영된 Point cloud의 깊이 값을 픽셀단위로 완성시키고 이미지 데이터와 원본 Point cloud로 보조하는 방법을 제안하였다.

밀도가 높아지는 E-point의 연구는 성능을 약 9% 개선하는 결과를 보여줬다. 하지만, LiDAR 자체의 Occlusion 문제, 카메라와 LiDAR, 이종센서 간의 부정확한 캘리브레이션과 싱크 문제를 해결하고 이미지 데이터의 보조로 명확한 윤곽선 구분과 정확한 깊이 값을 완성하는 네트워크 성능의 부족으로 더 많은 성능 연구가 필요하다.

후 기

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성(Grand ICT연구센터) 사업의 연구결과로 수행되었음(IITP-2021-2020-0-01462).

References

- 1) J. Seo, S. Oh and Y. K. Kim, "A Study of Curved Lane Detection Based on Dual Sensor Monitoring of LiDAR and Camera," Transactions of KSAE, Vol.29, No.2, pp.197-204, 2021.
- 2) D. Kang, M. Kim, H. Sun and J. Kim, "Feature Mapping with Machine Learning Based Object

- Classification,” Transactions of KSAE, Vol.28, No.6, pp.419-425, 2020.
- 3) J. Kim, D. Lee, B. Yu and S. C. Kee, “A Study on the Parking Point Searching for Wireless Charging of Electric Vehicles,” Transactions of KSAE, Vol.28, No.7, pp.483-489, 2020.
 - 4) G. Park and S. C. Kee, “Forward Vehicle Following Using Artificial Landmark Tracking of Image Sequences,” Transactions of KSAE, Vol.28, No.2, pp.133-142, 2020.
 - 5) L. C. Chen, G. Papandreou, F. Schroff and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation,” arXiv preprint arXiv: 1706.05587, 2017.
 - 6) K. He, G. Gkioxari, P. Dollár and R. Girshick, “Mask R-CNN,” International Conference on Computer Vision(ICCV), pp.2961-2969, 2017.
 - 7) S. Choi, J. T. Kim and J. Choo, “Cars Can’t Fly Up in the Sky: Improving Urban-Scene Segmentation via Height-driven Attention Networks,” IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp.9373-9383, 2020
 - 8) L. Yu, X. Li, C. Fu, D. Cohen-Or and P. A. Heng, “PU-Net: Point Cloud Upsampling Network,” IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp.2790-2799, 2018.
 - 9) C. He, H. Zeng, J. Huang, X. S. Hua and L. Zhang, “Structure Aware Single-stage 3D Object Detection from Point Cloud,” IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp.11873-11882, 2020.
 - 10) M. Simony, S. Milzy, K. Amendey and H. M. Gross, “Complex-YOLO: An Euler-Region-Proposal for Real-time 3D Object Detection on Point Clouds,” European Conference on Computer Vision(ECCV), 2018.
 - 11) A. Geiger, P. Lenz, C. Stiller and R. Urtasun, “Vision Meets Robotics: The KITTI Dataset,” The International Journal of Robotics Research, Vol.32, No.11, pp.1231-1237, 2013.
 - 12) J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu and B. Xiao, “Deep High-Resolution Representation Learning for Visual Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.
 - 13) K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” IEEE Conference on Computer Vision and Pattern Recognition, pp.770-778, 2016.
 - 14) D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” arXiv preprint arXiv:1412.6980, 2015.
 - 15) W. J. Lee, S. D. Ha, K. S. Jung, M. H. Jang and S. W. Kim, “A Performance Analysis on the M-tree with the Earth Mover’s Distance,” Korea Information Processing Society(KIPS) Fall Conference, pp.1232-1233, 2011.
 - 16) X. Cheng, P. Wang and R. Yang, “Depth Estimation via Affinity Learned with Convolutional Spatial Propagation Network,” European Conference on Computer Vision(ECCV), pp.103-119, 2018.
 - 17) S. Imran, Y. Long, X. Liu and D. Morris, “Depth Coefficients for Depth Completion,” IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), pp.12438-12447, 2019.
 - 18) S. S. Shivakumar, T. Nguyen, I. D. Miller, S. W. Chen, V. Kumar and C. J. Taylor, “DFuseNet: Deep Fusion of RGB and Sparse Depth Information for Image Guided Dense Depth Completion,” arXiv preprint arXiv:1902.00761, 2019.