

2023-03-22

MobRecon Review

증강지능 연구실

황승현

Contents

도입

01

과제 정의, 논문 소개, 용어 정의

접근 방법

02

2D 이미지를 3D로 변환하는 접근 방법(Method)

결과 분석

03

기존 방법과 논문에서 제시한 방법의 성능(정확도, 시간복잡도) 차이 비교

코드 분석

04

Pre-trained 모델 테스트, 실패 원인

정리

05

논문의 결론, 향후 계획



에

[도입]

- HandPoseEstimation
- 논문 소개
- 배경 지식

도입



Hand Pose Estimation



3D Hand Mesh



2D Hand Pose

Hand Pose Estimation

손 포즈 추정

영상에서 손의 관절 KeyPoint를 추출하고, 손의 위치를 2D/3D 좌표상에 나타내는 것.

증강현실 서비스 구현할 때 손의 위치를 정확하게 추정하여야 한다.

맡은 과제

2D 이미지로 3D 좌표 추정

Recon: Mobile-Friendly Hand Mesh Reconstruction from Monocular Image

Xingyu Chen^{1*} Yufeng Liu³ Yajiao Dong¹ Xiong Zhang² Chongyang Ma¹
 Yanmin Xiong¹ Yuan Zhang¹ Xiaoyan Guo¹
¹Y-tech, Kuaishou Technology ²YY Live, Baidu Inc.
³SEU-ALLEN Joint Center, Institute for Brain and Intelligence, Southeast University, China.

Abstract

In this work, we propose a framework for single-view hand mesh reconstruction, which can simultaneously achieve high reconstruction accuracy, fast inference speed, and temporal coherence. Specifically, for 2D encoding, we propose lightweight yet effective stacked structures, and pose lightweight yet effective stacked structures. For 3D decoding, we provide an efficient graph operation, namely depth-separable spiral convolution. Moreover, we present a novel feature lifting module for bridging the gap between 2D and 3D representations. This module consists of a map-based position regression (MapReg) to integrate the merits of both heatmap encoding and regression paradigms for improved 2D accuracy and temporal coherence. Furthermore, MapReg is followed by global pooling and pose-to-vertex lifting approaches, to transform 2D pose encodings to semantic features at vertices. Overall, our hand reconstruction framework, called MobRecon, comprises affordable computational costs and miniature model size, which reaches a high inference speed of 83FPS on Apple A14 CPU. Extensive experiments on popular datasets such as FreiHAND, RHD, and O3Dv2 demonstrate that our MobRecon achieves superior performance on reconstruction accuracy and temporal coherence. Our code is publicly available at <https://github.com/SeanChenxy/HandMesh>.

Introduction

Single-view hand mesh reconstruction has been extensively investigated for years due to its wide range of applications in AR/VR [28, 73], behavior understanding [40, 63], and tremendous research efforts have been made towards this task, including [18, 90, 46, 85], to name a few. The primary focus of typical existing methods is the reconstruction accuracy [50, 51], while real-world applications additionally necessitate inference efficiency and temporal consistency. In particular, 3D hand information

*Corresponding author, chenxingyu@kuaishou.com

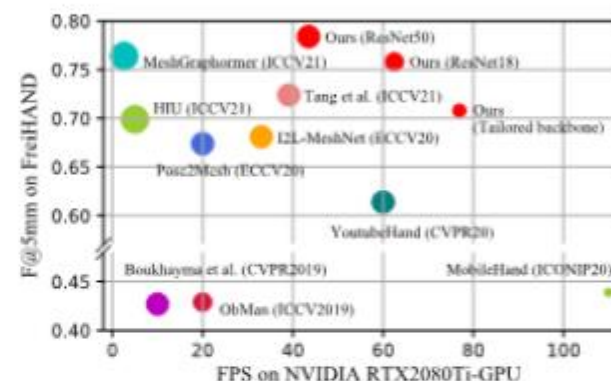


Figure 1. Accuracy vs. inference speed. The marker size is related to the model size. Besides, our tailored method can run at a high speed on mobile CPUs.

is a vital component in mobile applications [73], where the devices comprise limited memory and computational budgets. Thereby, this work aims to explore 3D hand reconstruction for mobile platforms.

A typical pipeline for single-view hand reconstruction includes three phases: 2D encoding, 2D-to-3D mapping, and 3D decoding. In 2D encoding, existing approaches [11, 50, 51] usually adopt computationally intensive networks [30, 75] to handle this highly non-linear task, which are hard to deploy on mobile devices. Instead, if naively leveraging a mature mobile network (e.g., [32]) which is not tailored for our target task, the reconstruction accuracy dramatically degrades [22]. Hence, our motivation is to develop a lightweight 2D encoding structure tailored to balance the inference efficiency and accuracy. Besides, the efficiency of 2D-to-3D mapping and 3D decoding remains relatively unexplored. Thus, we intend to explore a lightweight yet effective lifting method to tackle the 2D-to-3D mapping problem and design an efficient graph operation for processing of 3D mesh data.

Although as crucial as accuracy in real-world applications, temporal coherence is usually neglected in the task of 3D hand reconstruction. Previous methods [13, 41, 45] adopt sequential models to incorporate both past and future

MobRecon: Mobile-Friendly Hand Mesh Reconstruction from Monocular Image

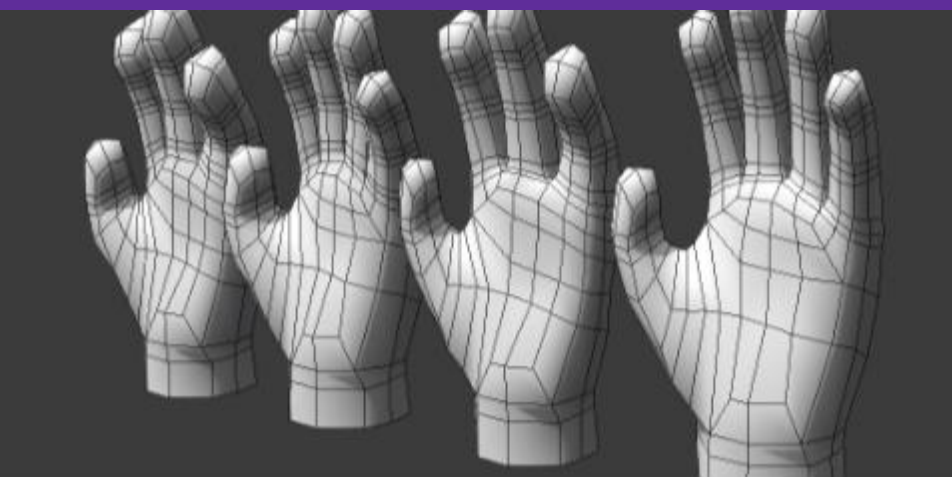
Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, Xiaoyan Guo

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 20544-20554

도입



배경 지식



Hand Mesh

손의 뼈와 관절의 위치와 움직임을 나타내기 위해 사용되는 3차원 모델

기존에는 카메라를 여러대 (Stereo) 사용하여야 구현 가능했음. 3D 좌표를 추정할 때 고성능 컴퓨팅 필요



Monocular Image

Monocular Image

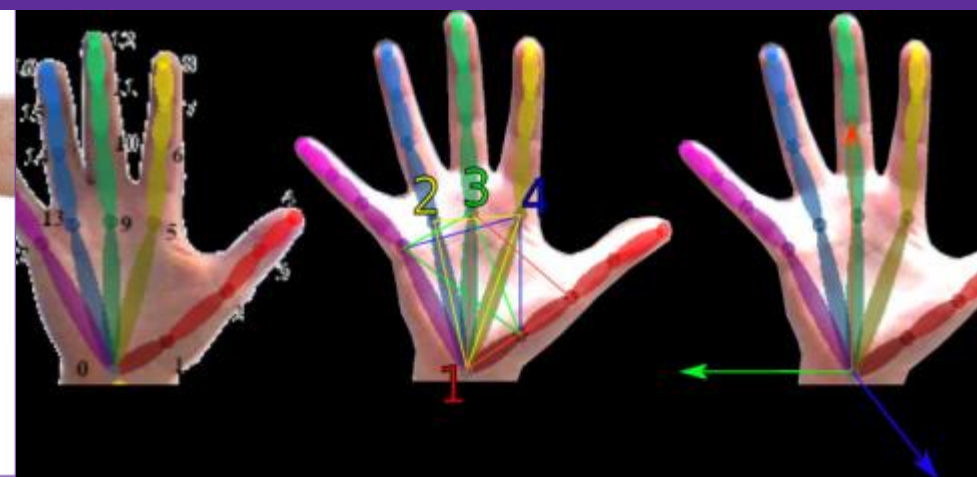
2D image

깊이 정보가 없는 이미지, 일반적인 사진

Stereo Image

3D image

깊이 정보가 있는 이미지, 인간의 두 눈으로 본 이미지.
카메라 2개의 위상차로 3D 구현



Lifting

2D-to-3D Mapping

2D 영상에서 객체의 3D 표현을 생성하는 프로세스

2D 영상에 없는 객체의 깊이 정보를 추정

이 논문에서는 딥러닝을 기반으로 깊이 정보 추정

MPJPE. Much of the literature reports *mean per joint position error*. For a frame f and a skeleton \mathcal{S} , MPJPE is computed as

$$E_{MPJPE}(f, \mathcal{S}) = \frac{1}{N_S} \sum_{i=1}^{N_S} \|m_{f, \mathcal{S}}^{(f)}(i) - m_{gt, \mathcal{S}}^{(f)}(i)\|_2$$

where N_S is the number of joints in skeleton \mathcal{S} . For a set of frames the error is the average over the MPJPE for all frames.

MPJPE MPVPE

mean per joint position error. (MPJPE)

mean per vertex position error (MPVPE)

3D Hand 오차 측정법

3D Hand Pose Estimation 알고리즘의 평가 지표
지수가 적을 수록 좋은 알고리즘



02

「 접근방법 」

- 논문의 Method
- 각 Method 설명

접근방법



논문의 Method

01

2D 좌표 추출

Stacked Encoding Network

lightweight stacked structures for
2D encoding

02

Lifting

2D-to-3D mapping

MapReg, pose pooling, and PVL
DSConv

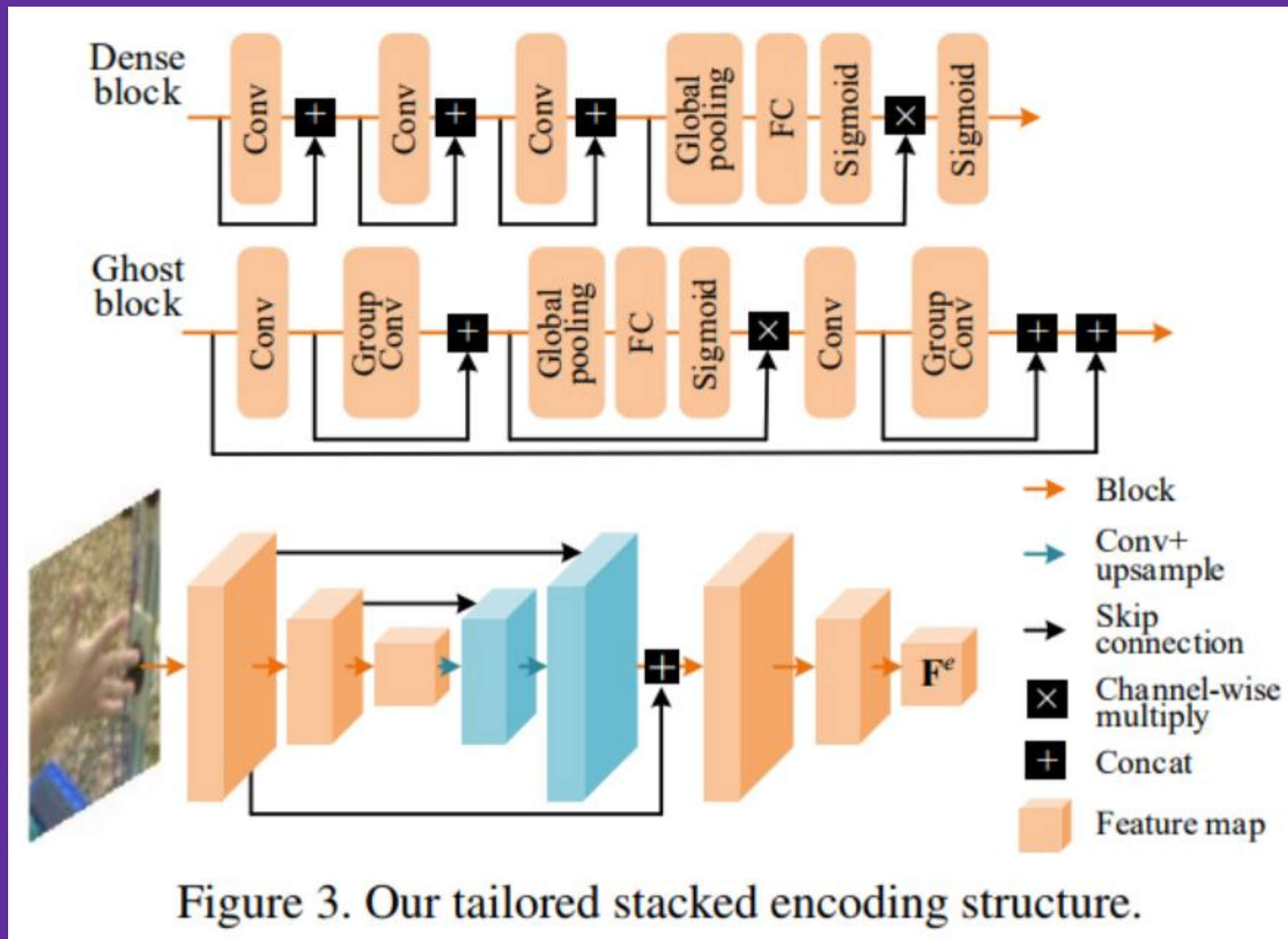
03

테스트

3D 손 데이터셋에서, 깊이 정보를 제외한 데
이터로 학습, 오차 측정

접근 방법

Stacked Encoding Network



2D Keypoint

DenseStack

Dense Block : Upsampling

input: 128×128 single-view image,

output: encoding feature

373.0M Mult-Adds and 6.6M parameters

GhostStack

모델 크기를 더 줄이기 위해 고스트 특징 생성

96.2M Mult-Adds and 5.0M parameters

ResNet18 기반 Stacked Network

2391.3M Mult-Adds and 25.2M parameters

모바일 애플리케이션에서 사용불가

접근 방법

Feature Lifting Module

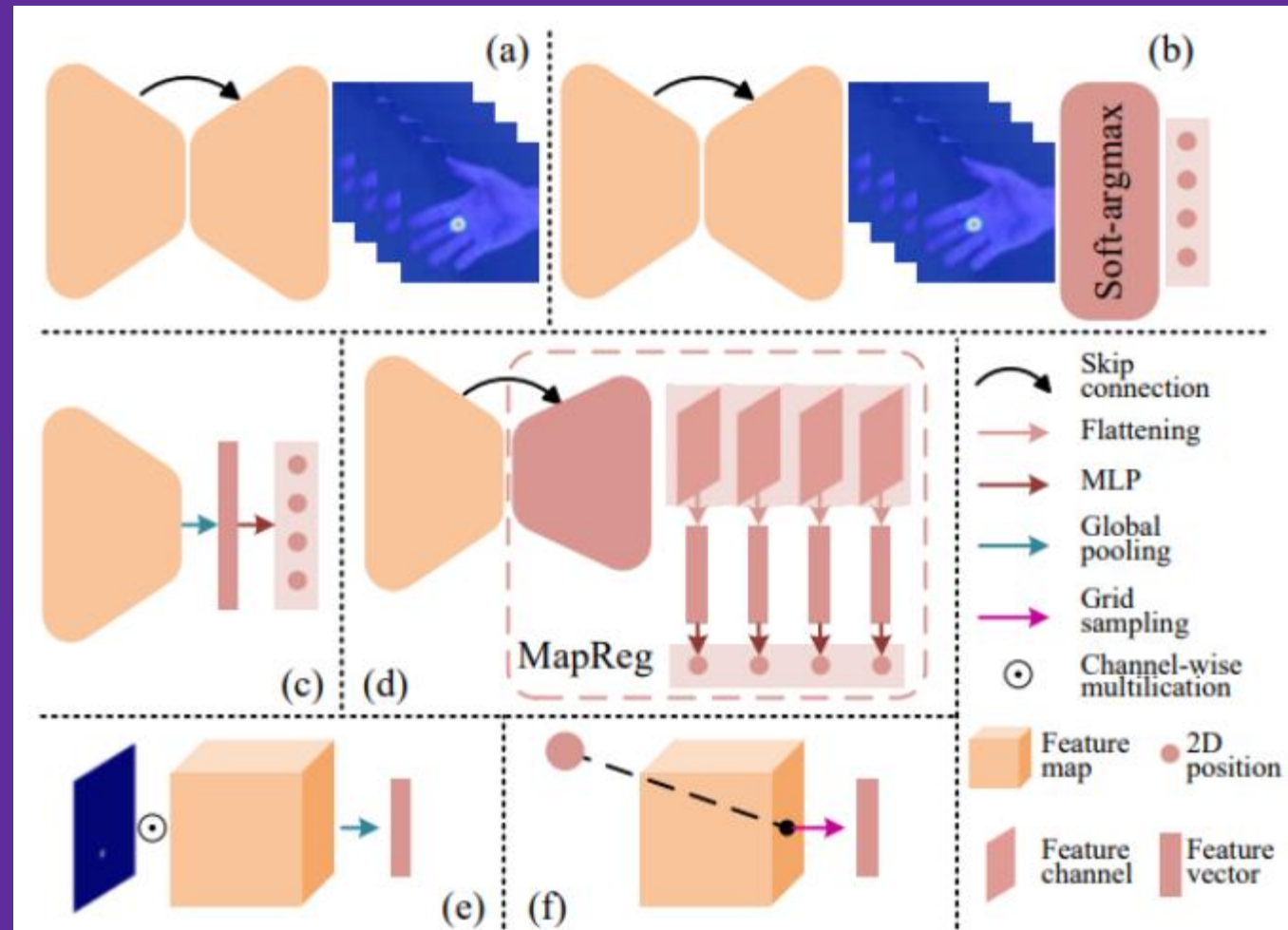


Figure 4. Comparisons of 2D representations and pose pooling methods. (a) heatmap, (b) heatmap + soft-argmax, (c) regression-based position, (d) MapReg-based position, (e) joint-wise pooling with a heatmap, (f) grid sampling with a 2D position. For better visualization, only 4 landmarks are illustrated.

2D to 3D

Lifting: 2D-to-3D Mapping

과정

2D feature 수집 -> 3D 영역에 매핑

Map-based position regression

MapReg: heatmap + position-based

Pose pooling

how to retrieve pixel-aligned features after obtaining 2D representations

Pose-to-vertex lifting

PVL: 3D 공간 매핑을 위한 선형 연산자

접근 방법

Depth-Separable SpiralConv

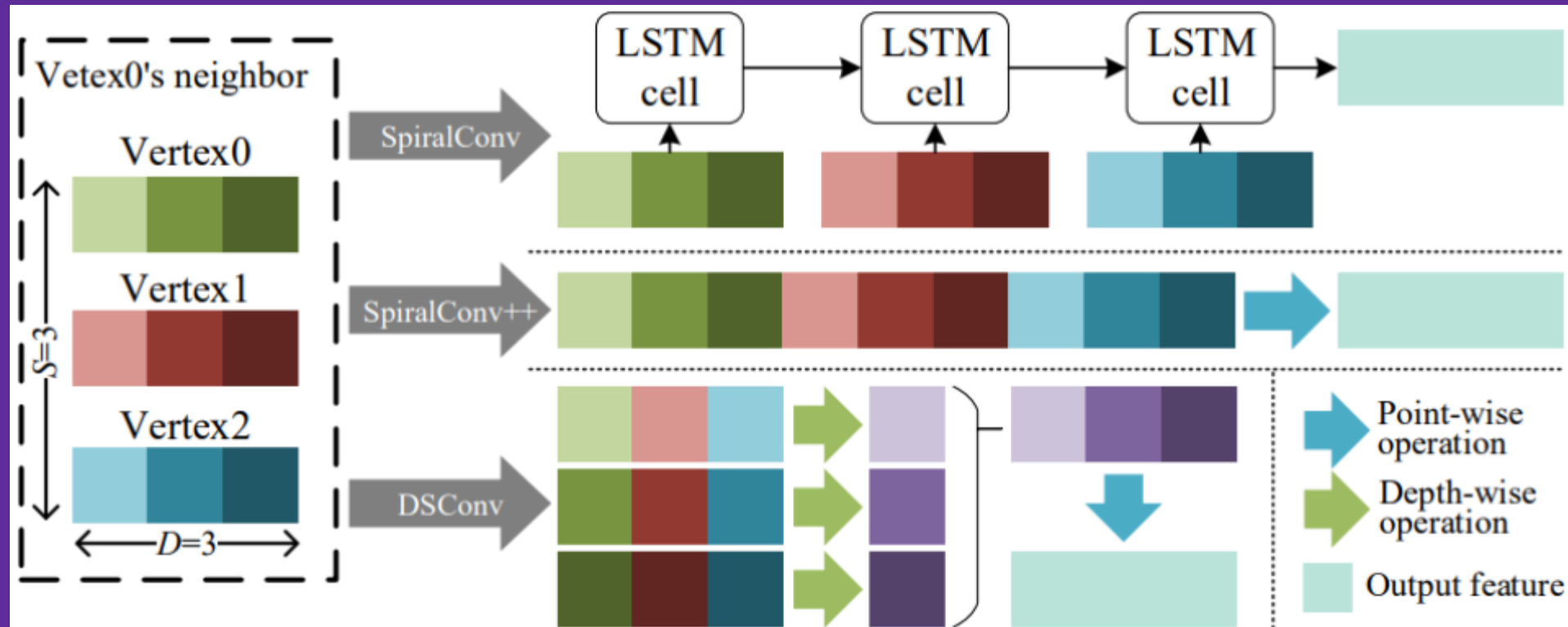


Figure 5. Comparison of SpiralConv, SpiralConv++, and DSConv. For better visualization, a case with $S = D = 3$ is shown.

DSConv

graph operator: 3D 포인트 클라우드 생성

SpiralConv with LSTM

Spiral Sampling: 3D 포인트 클라우드를 CNN에서 사용할 수 있게 선형으로 변경

SpiralConv++

graph convolutional neural network (GCN)

모델 크기 크게 증가

DSConv

기존 gcn기반 연산보다 효율적임

depth-wise operation, point-wise operation



03

[결과 분석]

- 기존 방법과의 차이 분석

결과 분석

기존 방법과의 차이 분석

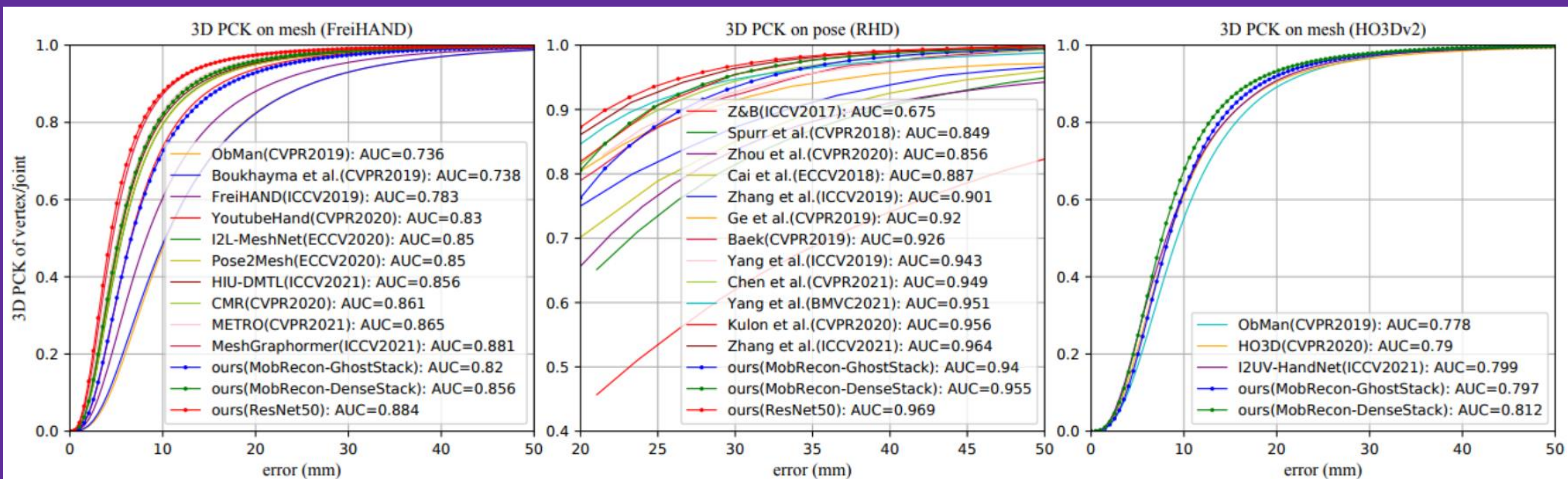


Figure 9. 3D PCK vs. error thresholds.

3D decoding	Mult-Adds	#Param	PJ↓	Acc↓	FPS↑
<i>w/ GhostStack</i>					
SpiralConv++	159.0/263.1M	1.0/6.2M	8.63	2.31/7.07	77
DSConv (ours)	19.5/123.5M	0.1/5.3M	8.76	2.30/6.98	83
<i>w/ DenseStack</i>					
SpiralConv++	159.0/579.4M	1.0/9.0M	6.85	1.98/4.75	59
DSConv (ours)	19.5/439.9M	0.1/8.1M	6.87	1.92/4.73	67

Table 4. Ablation study of 3D decoding. Mult-Adds and #Param are *w.r.t.* the 3D decoder/overall model; 2D/3D Acc is presented; FPS is tested on Apple A14 CPU; Accuracy and temporal performance are tested on FreiHAND and HO3Dv2, respectively.

As shown in Table 4, DSConv dramatically decreases the Mult-Adds and #Param of the 3D decoder and obtains on par, sometimes even better, performance compared with SpiralConv++. Overall, our MobRecon with DenseStack/GhostStack can reach 67/83 FPS on Apple A14 CPU.

Discussion. MobRecon has a limitation that the DSConv increases memory access cost, so some engineering optimization should be involved for higher inference speed.

결과 분석

기존 방법과의 차이 분석

모바일 환경

DSConv

SpiralConv++(기존 방법)에 비해 파라미터 줄여 필요한 연산 줄임. 기존 방법과 정확도는 비슷하거나, 더 좋음

모바일 환경에 적합함.

Method	Backbone	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
MobileHand [22]	MobileNet	—	13.1	0.439	0.902
FreiHAND [93]	ResNet50	11.0	10.9	0.516	0.934
YoutubeHand [46]	ResNet50	8.4	8.6	0.614	0.966
I2L-MeshNet [57]	ResNet50*	7.4	7.6	0.681	0.973
HIU-DMTL [85]	Customized*	7.1	7.3	0.699	0.974
CMR [11]	ResNet50*	6.9	7.0	0.715	0.977
I2UV-HandNet [10]	ResNet50	6.7	6.9	0.707	0.977
METRO [50]	HRNet	6.7	6.8	0.717	0.981
Tang <i>et al.</i> [73]	ResNet50	6.7	6.7	0.724	0.981
MeshGraphormer [51]	HRNet	5.9	6.0	0.765	0.987
MobRecon (ours)	GhostStack*	8.8	9.1	0.597	0.960
MobRecon (ours)	DenseStack*	6.9	7.2	0.694	0.979
ours [‡]	ResNet18*	6.7	6.8	0.727	0.979
ours [†]	ResNet18*	6.1	6.3	0.758	0.983
ours [‡]	ResNet50*	6.1	6.2	0.760	0.984
ours [†]	ResNet50*	5.7	5.8	0.784	0.986

Table 5. Results on the FreiHAND dataset. *: stacked structure; †: These models are based on ImageNet pre-trained backbone and mixed fine-tuning data; ‡: These models are totally unrelated to our complement data.

Method	Backbone	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
ObMan [29]	ResNet18	11.0	11.0	0.464	0.939
HO3D [25]	CPM [77]	10.7	10.6	0.506	0.942
I2UV-HandNet [10]	ResNet50	9.9	10.1	0.500	0.943
MobRecon (ours)	GhostStack	10.0	10.2	0.488	0.948
MobRecon (ours)	DenseStack	9.2	9.4	0.538	0.957

Table 6. Results on the HO3Dv2 dataset.

결과 분석

기존 방법과의 차이 분석

결과 분석

FreiHAND dataset

superior performance, superior inference speed on 3D PCK
3D AUC of 0.955 and 0.940, outperforming most compared approaches.

HO3Dv2 dataset

MobRecon outperforms existing methods

new state of the art

we surpass previous methods with ResNet50, leading to a new state of the art




04

[코드 분석]

- 코드 개요
- 실패 원인

https://github.com/SeanChenxy/HandMesh


 Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

SeanChenxy / HandMesh Public Watch 10 Fork 60 Starred 253

<> Code Issues 3 Pull requests Actions Projects Security Insights

main 1 branch 0 tags

Go to file Add file <> Code

 SeanChenxy Baidu cloud link dbdca0d on Aug 27, 2022 72 commits


cmr	retrain mobrecon	last year
conv	add some comments	10 months ago
data	add some comments	last year
mobrecon	fix issue #51 and #46	8 months ago
options	add mobrecon	last year
smplpytorch	retrain mobrecon	last year
template	add mobrecon	last year
utils	fix issue #51 and #46	8 months ago
.gitignore	retrain mobrecon	last year
LICENSE	add hguman3.6m	2 years ago
README.md	Baidu cloud link	7 months ago
complement_data.md	complement data for training	10 months ago
requirements.txt	complement data for training	10 months ago


⋮ README.md


About


No description or website provided.


freihand cvpr2021 cvpr2022

 Readme

 MIT license

 253 stars

 10 watching

 60 forks


Releases


No releases published

Packages

No packages published

Contributors 2

 SeanChenxy Xingyu Chen

 vra Yunfeng Wang



Perceiving Systems Mesh Package

This package contains core functions for manipulating meshes and visualisation supported on Linux and macOS operating systems.

The screenshot shows a GitHub issue page for the repository `MPI-IS/mesh`. The issue title is "Installing on Windows 10 #46". The issue was opened by `NahomKidane` on August 13, 2020, and has 2 comments. The issue is currently open.

The first comment by `NahomKidane` (Aug 13, 2020) asks: "Can we install it on windows 10? Any suggestions." and includes a "thanks" list item. It has 12 upvotes.

The second comment by `Metarefektor` (Mar 27, 2021) responds to `@NahomKidane`, stating: "I was able to install it by checking out this branch (https://github.com/johnbanq/mesh/tree/fix/MSVC_compilation) and running the command like its stated it the readme under "Windows". You need to install boost in addition." It has 1 upvote.

The third comment by `jcpassy` (Apr 23, 2021) is partially visible, starting with "Hi @NahomKidane @Metarefektor".

On the right side of the issue, there are sections for "Assignees" (No one assigned), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), "Development" (No branches or pull requests), "Notifications" (Subscribe button), and "3 participants".



05

「정리」

- 모더니즘이란 무엇인가
- 모더니즘의 역사적 배경

정리



논문의 결론

「 State-of-the-Art 」

new state-of-the-art (최첨단)

기존에 쓰이던 2D to 3D Point와 성능은 비슷하나,
더 적은 리소스. 모바일 환경에서 사용 가능

아이폰에 실제 구현되어있음

정리



코드 분석 계획

「 GPU 서버 개발 환경 구축 / 변형된 Mesh 」

원활한 코드 분석을 위하여,

1. Windows에서 사용가능하게 변형된 Mesh 라이브러리 사용.
2. GPU 서버(Linux Mint)에 모델 돌릴 수 있는 가상환경 구축.

정리



향후 계획

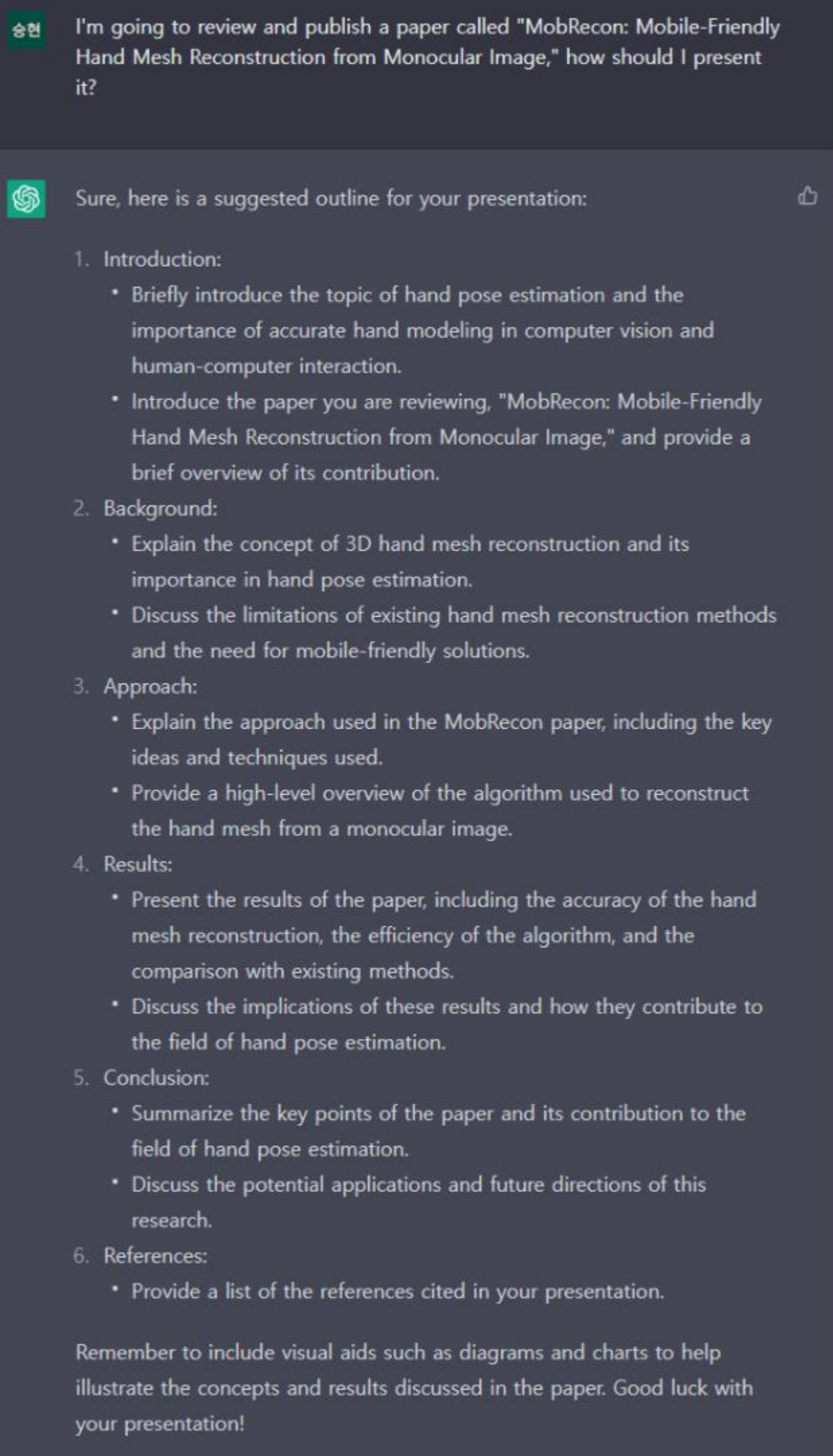


MediaPipe & MobRecon



MobRecon
Stacked Encoding Network + 아이폰,

안드로이드 환경, 미디어파이프로 2D 좌표 추출



정리

참고자료

ChatGPT

발표의 아웃라인 설정

세부 내용, 이미지: 논문 내에서 발취



QnA

질문이 있다면 말씀해주세요.