Seminar Research Center for Technology and Art

"One Shot 3D Photography "

https://s2020.siggraph.org/smile-photos-converted-into-3d-from-any-mobile-device/
https://arxiv.org/abs/2008.12298

IPHD YuanFu Yang

Outline

01 Background

02 Art Statement

03 Method

04 Connection/Demo

Background

Johannes Kopf

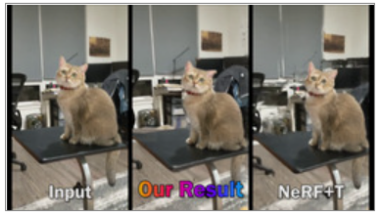
A research scientist at Facebook, where he support a group working on computational photography research. Before joining Facebook, he worked at Microsoft Research. He received the EUROGRAPHICS Young Researcher Award in 2013 and the SIGGRAPH Significant Researcher Award in 2015.

His group works on cutting-edge research projects at the intersection of computer vision, graphics, and machine learning. They also like to productize Their work. A favorite recent example is 3D photos.

Publications

Dynamic View Synthesis from Dynamic Monocular Video,
Chen Gao, Ayush Saraf, Johannes Kopf, Jia-Bin Huang,
arXiv 2021
Project page / arXiv / Bibtex

Robust Consistent Video Depth Estimation,
Johannes Kopf, Xuejian Rong, Jia-Bin Huang,
CVPR (Oral Presentation) 2021
Project page / arXiv / Code / Colab / Bibtex



Space-time Neural Irradiance Fields for Free-Viewpoint Video,
Wenqi Xian, Jia-Bin Huang, Johannes Kopf, Changil Kim,
CVPR 2021
Project page / arXiv / Bibtex

⋮

Solid Texture Synthesis from 2D Exemplars,
Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, Oliver Deussen,
Dani Lischinski, Tien-Tsin Wong,
SIGGRAPH 2007
Project page / Bibtex

Capturing and Viewing Gigapixel Images,
Johannes Kopf, Matt Uyttendaele, Oliver Deussen, Michael F. Cohen,
SIGGRAPH 2007
Project page / Bibtex

Joint Bilateral Upsampling,
Johannes Kopf, Michael F. Cohen, Dani Lischinski, Matt Uyttendaele,
SIGGRAPH 2007
Project page / Bibtex

Art Statement

Art Statement

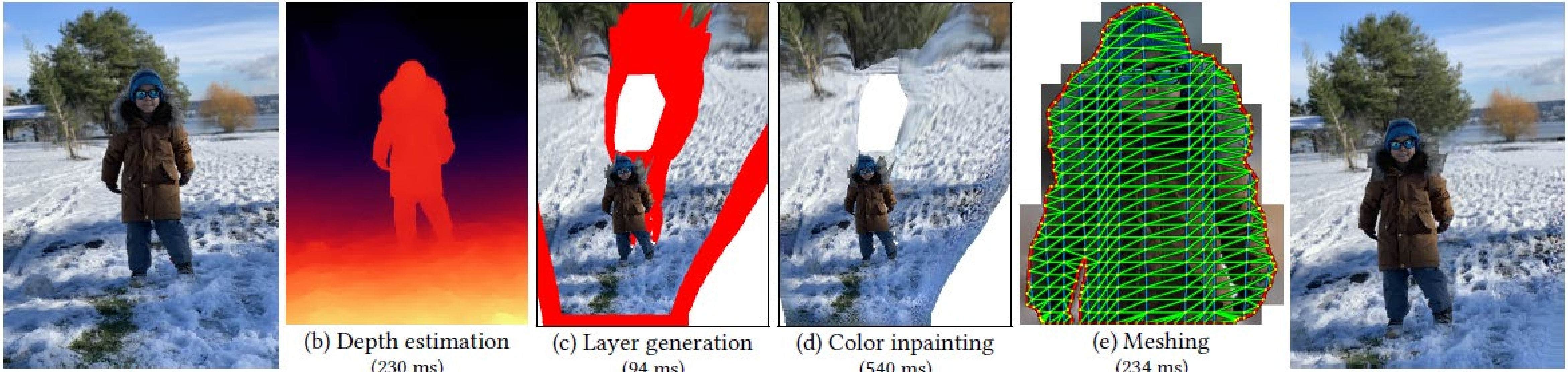
Challenges:

- Generate Depth Information
- Fill in the image of the occluded area
- Low computing cost for mobile device.

(a) Depth-warping (holes) (b) Depth-warping (stretching) (c) Facebook 3D photo (d) Our result

Method

- Depth Estimation
- Layer Generation
- Color Inpainting
- Meshing



(a) Input
(b) Depth estimation (230 ms)
(c) Layer generation (94 ms)
(d) Color inpainting (540 ms)
(e) Meshing (234 ms)
(f) Novel view (real-time)

Processing: 1,098ms on a mobile phone (iPhone 11 Pro)

Depth Estimation

New Depth Estimation Neural Networks - *Tiefenrausch*

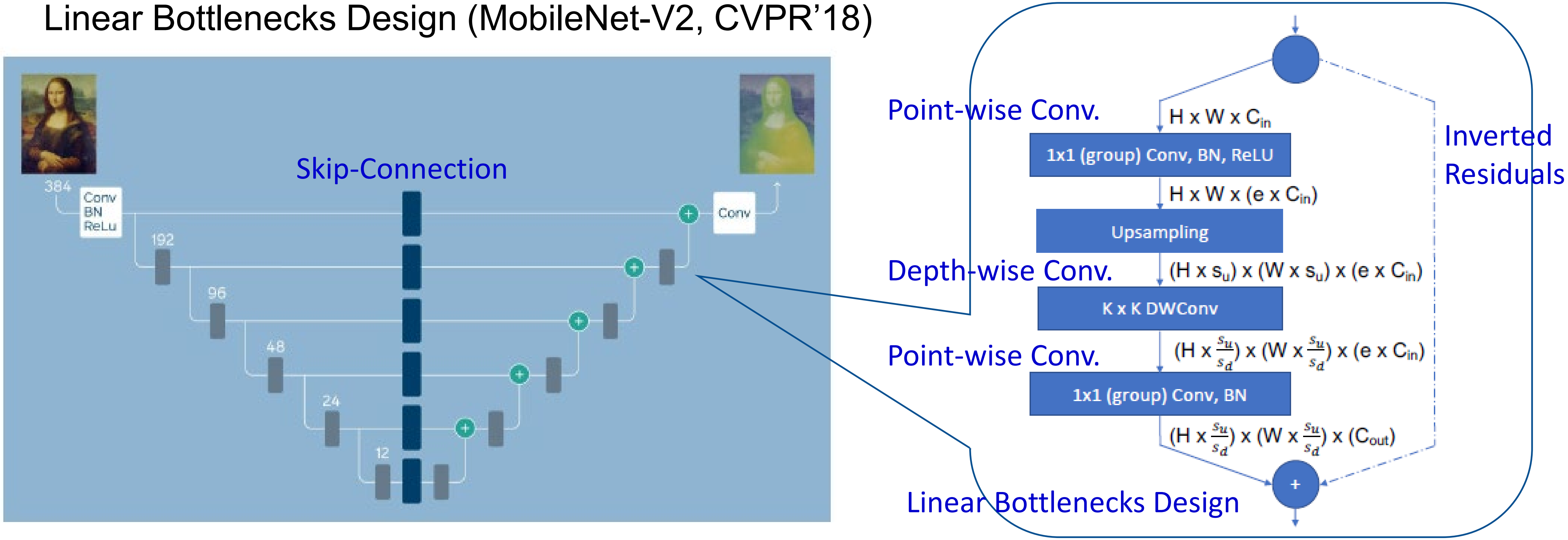
- Efficient Block Structure
- Neural Architecture Search
- 8-bit Quantization

Depth Estimation

Efficient Block Structure:

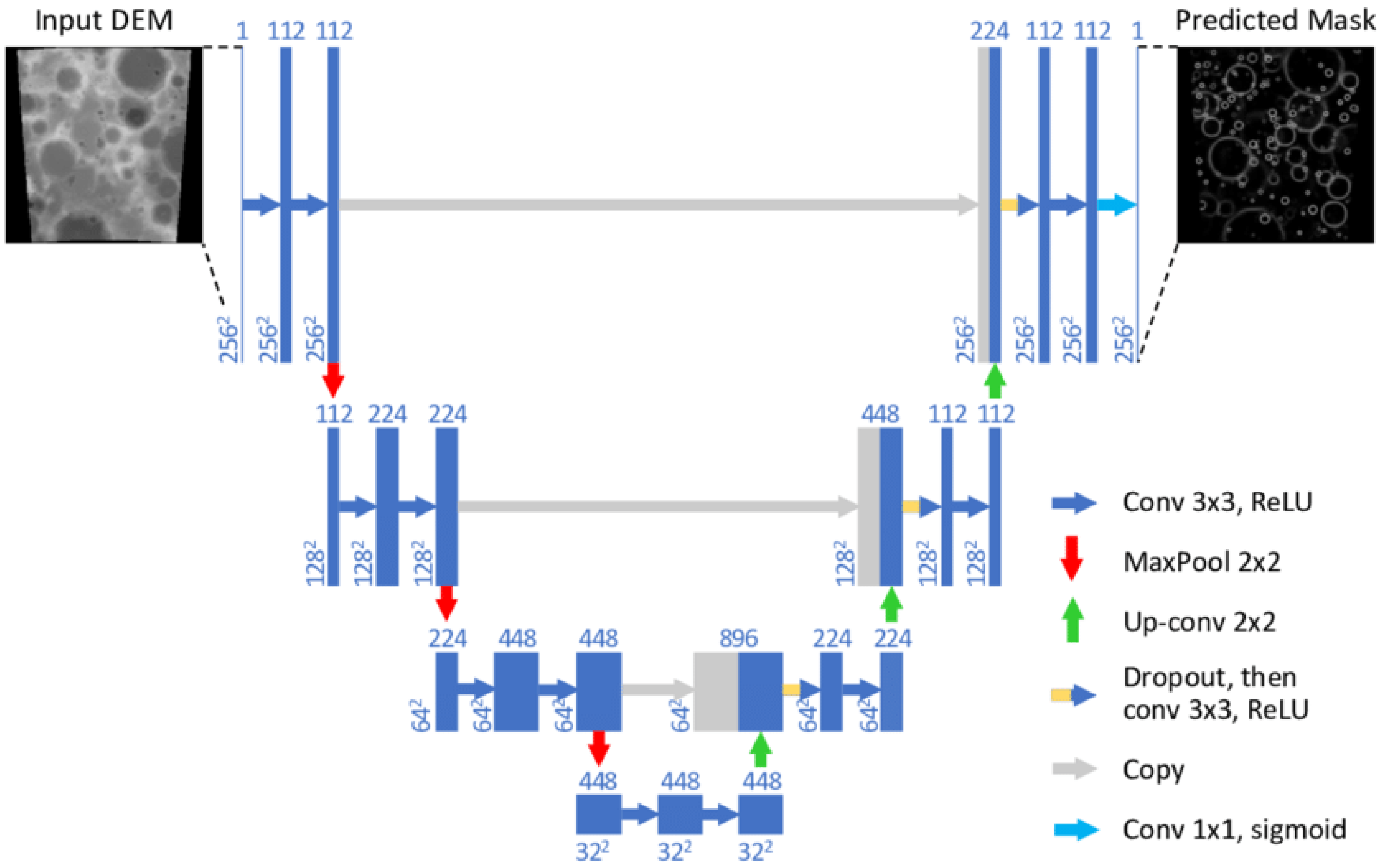
- Skip-Connection (U-Net, MICCAI’15)
- Depthwise Separable Convolution (MobileNet-V1, arXiv’17)
- Inverted Residuals (MobileNet-V2, CVPR’18)
- Linear Bottlenecks Design (MobileNet-V2, CVPR’18)

Efficient Block Structure



Depth Estimation

- U-Net, MICCAI'15 :

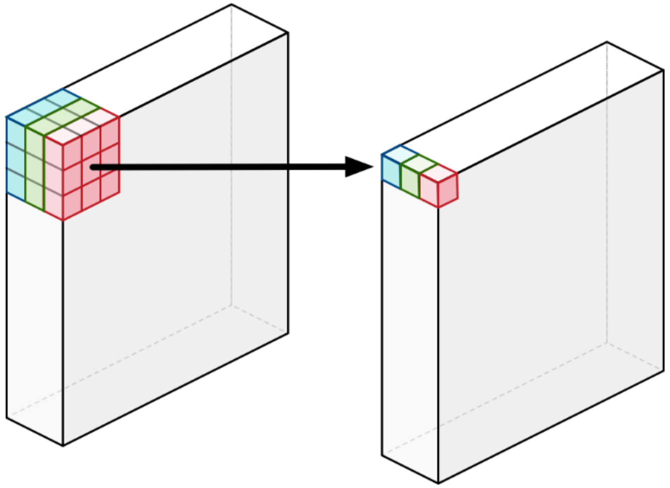


O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", Lecture Notes in Computer Science Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pp. 234-241, 2015.

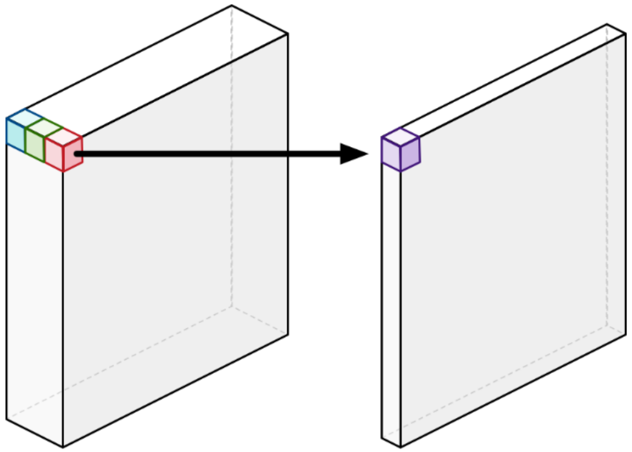
Depth Estimation

Depthwise Separable Convolution

= Depthwise Convolution + Pointwise Convolution



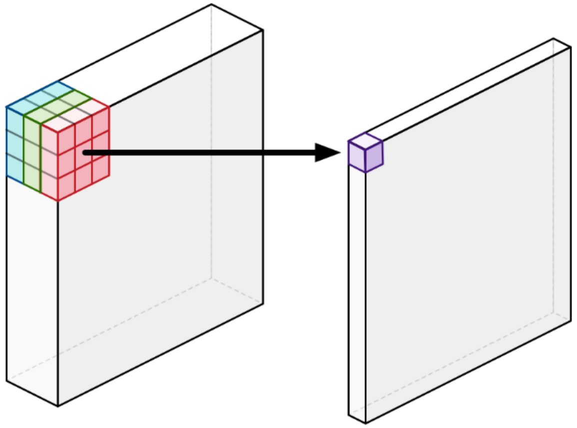
在高維度擷取特徵



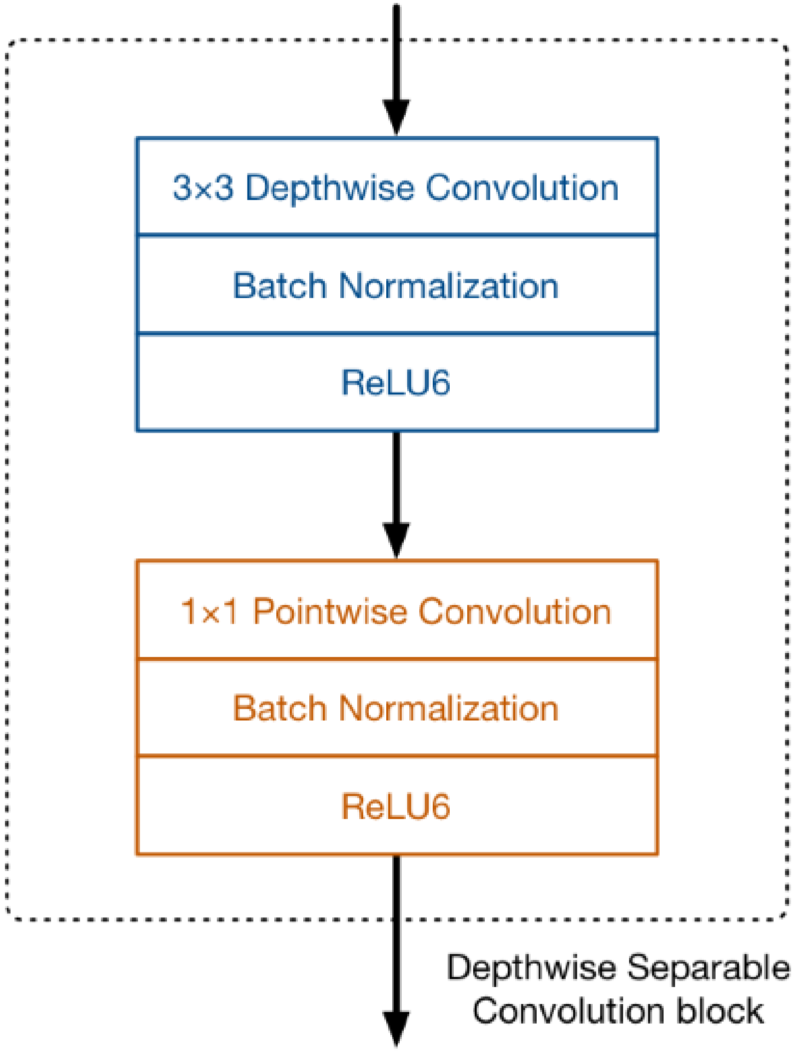
在低維度傳遞資料

A regular convolutional:

在高維度擷取特徵
在高維度傳遞資料



Depthwise Separable Convolution Block



Depth Estimation

Depthwise Separable Convolution

- ✓ Depthwise convolution is the channel-wise DK×DK spatial convolution. Suppose in the figure above, we have 5 channels, then we will have 5 DK×DK spatial convolution.
- ✓ Pointwise convolution actually is the 1×1 convolution to change the dimension.

the operation cost of DW Separable Convolution is:

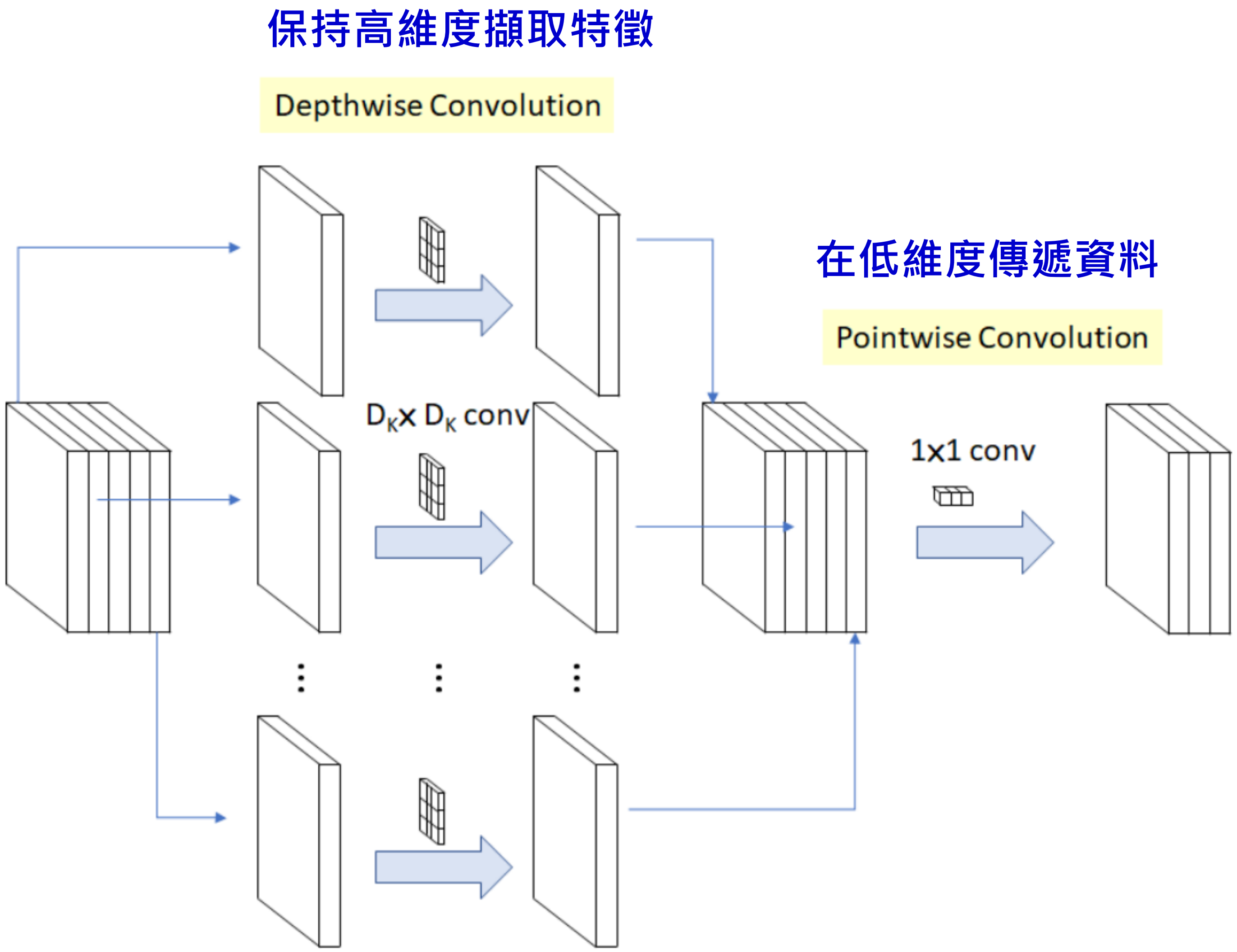
$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

the operation cost of standard Convolution is:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$$

Thus, the computation reduction is:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F}$$
$$= \frac{1}{N} + \frac{1}{D_K^2}$$



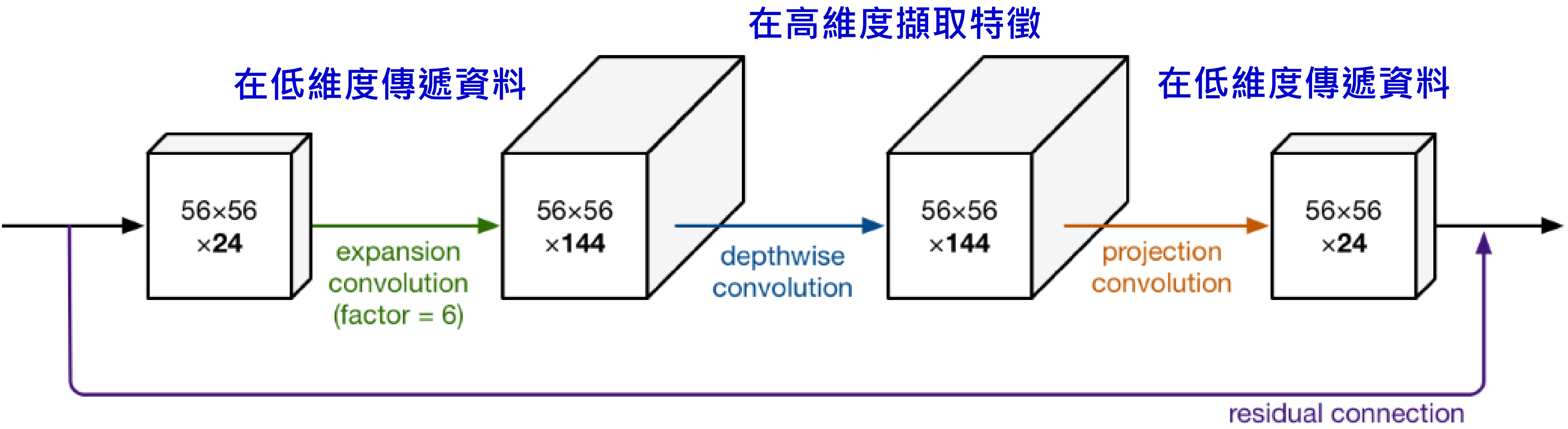
When DK×DK is 3×3, 8 to 9 times less computation can be achieved, but with only small reduction in accuracy.

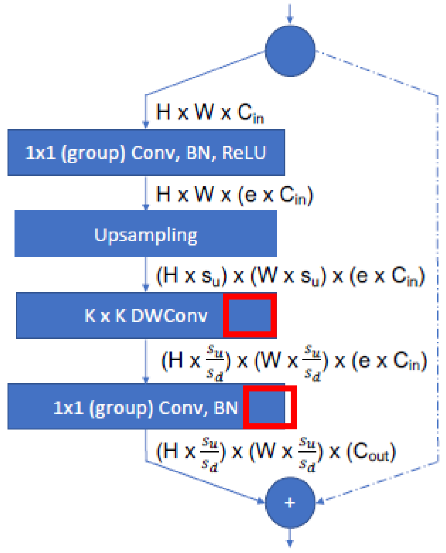
Depth Estimation

Inverted Residuals/Linear Bottlenecks Design

- ✓ There are 3 layers for both types of blocks.
- ✓ The first layer is 1×1 convolution with ReLU6.
- ✓ The second layer is the depthwise convolution.
- ✓ The third layer is another 1×1 convolution but without any non-linearity.
- ✓ And there is an expansion factor t. And t=6 for all main experiments. If the input got 24 channels, the internal output would get 24×t=24×6=144 channels.

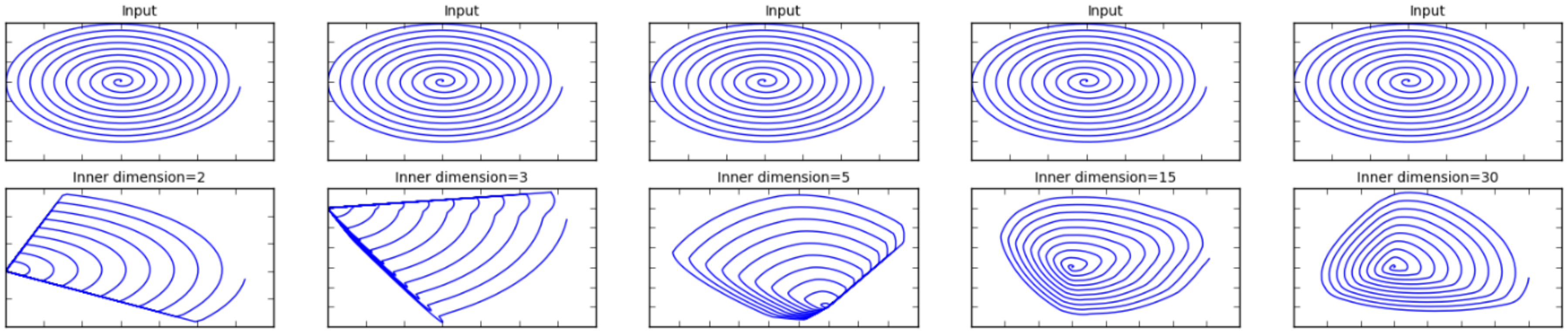
Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$



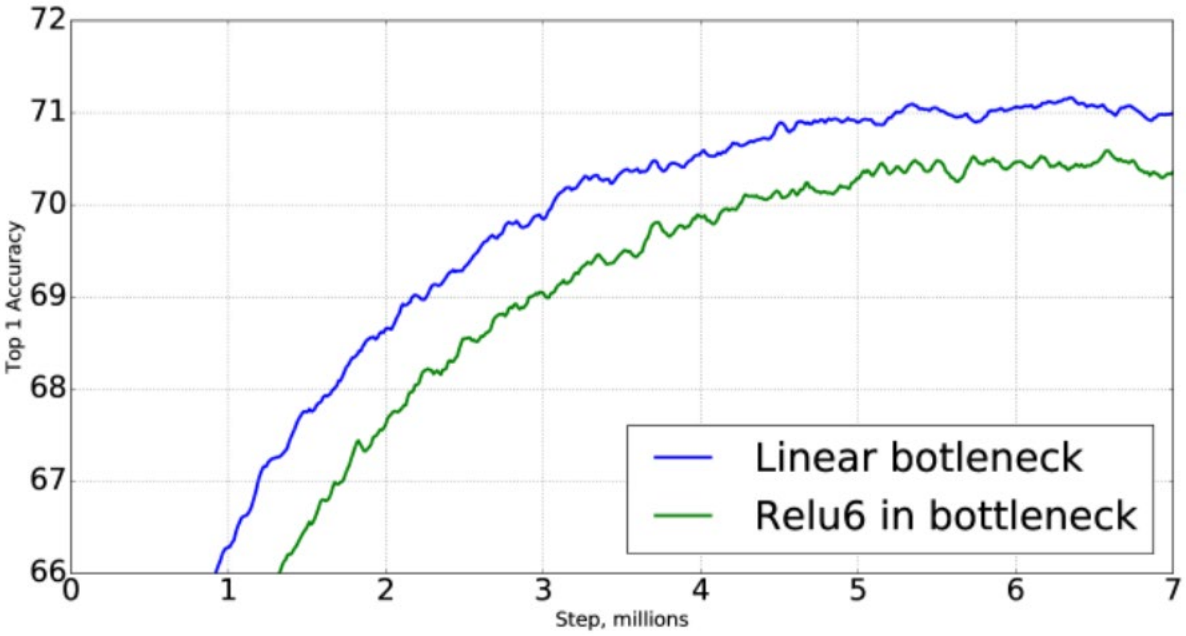


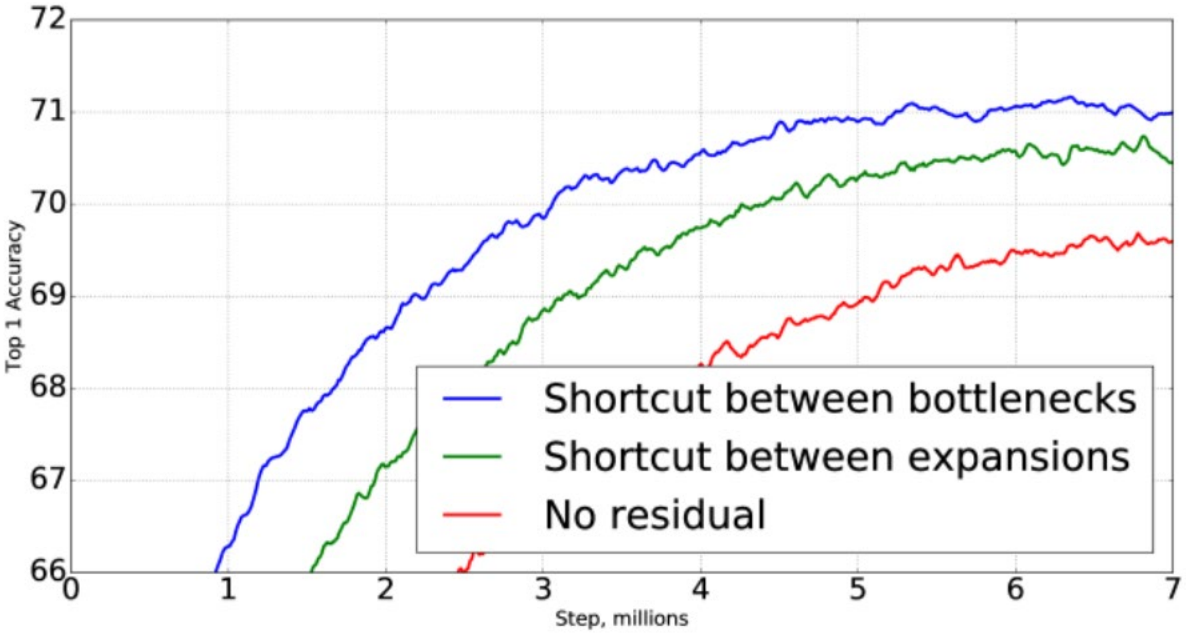
Depth Estimation

- Non-Linear Bottlenecks: Tensor Collapse



- The impact of non-linearities and various types of shortcut (residual) connections.





Depth Estimation

Efficient Block Structure - Neural Architecture Search:

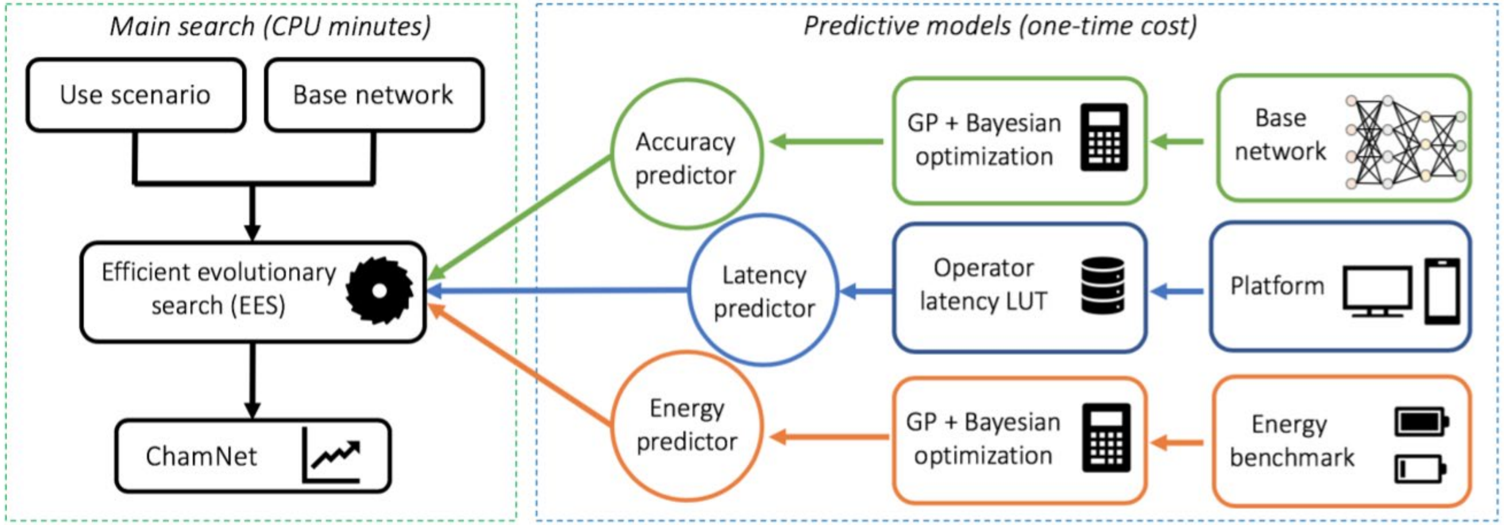
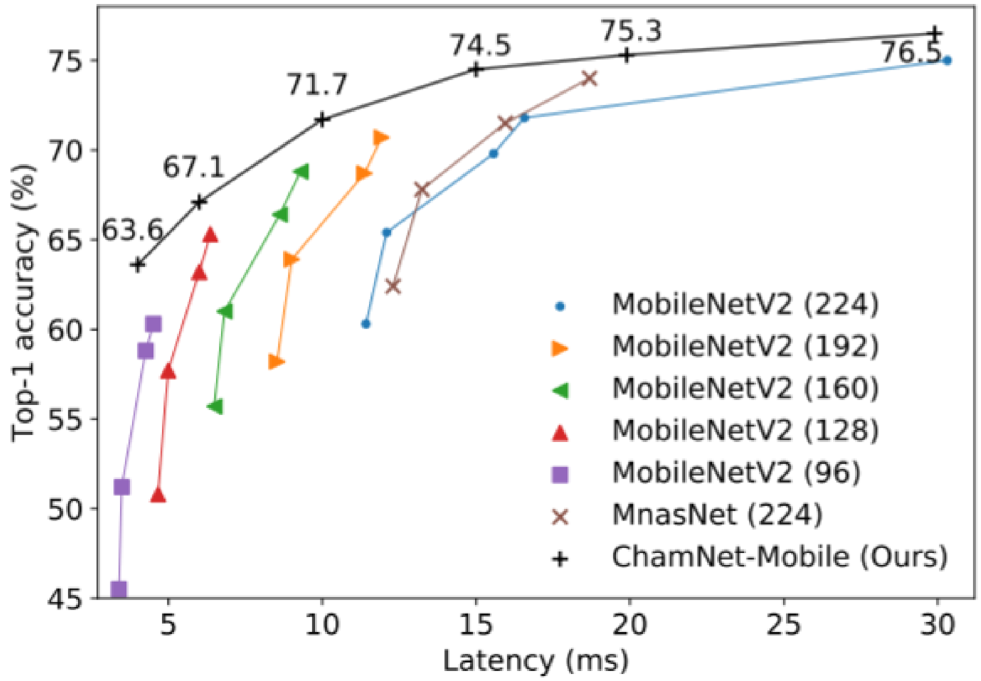


Figure 1. An illustration of the Chameleon adaptation framework

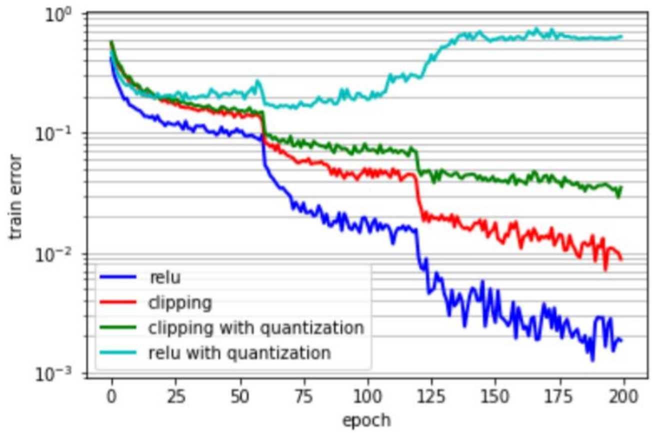


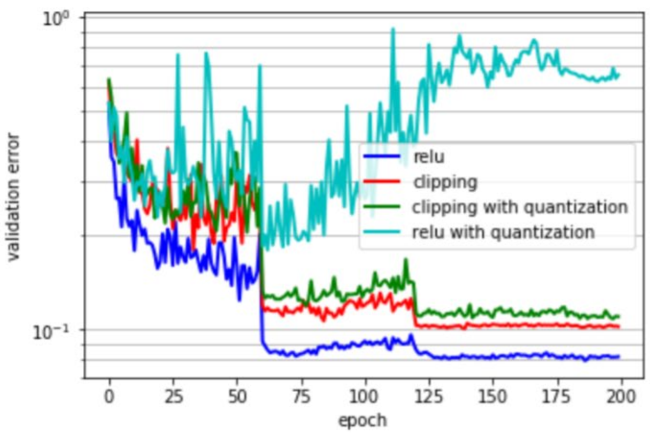
Dai et al., " ChamNet: Towards Efficient Network Design through Platform-Aware Model Adaptation ", CVPR'19

Depth Estimation

Efficient Block Structure - 8-bit Quantization:

Method	Training data	Quality (MegaDepth)			Quality (ReDWeb)			Performance			Model footprint		
		$\delta < 1.25 \uparrow$	Abs rel$\downarrow$	RMSE$\downarrow$	$\delta < 1.25 \uparrow$	Abs rel$\downarrow$	RMSE$\downarrow$	FLOPs$\downarrow$	Runtime$\downarrow$	Peak mem.$\downarrow$	float32	int8	Size$\downarrow$
Midas (v1)	RW, MD, MV	0.955	0.068	0.027	-	-	-	33.2 G	1.11 s	453.7 MiB	37.3 M	-	142.4 MiB
Midas (v2)	RW, DL, MV, MD, WSVD	**0.965**	**0.058**	**0.022**	-	-	-	72.3 G	-	-	104.0 M	-	396.6 MiB
Monodepth2	K	0.845	0.145	0.049	0.350	4.368	0.176	6.7 G	0.26 s	**194.1 MiB**	14.3 M	-	54.6 MiB
SharpNet	PBRS → NYUv2	0.839	0.146	0.051	0.308	6.616	0.196	54.9 G	-	-	114.1 M	-	435.1 MiB
MegaDepth	DIW → MD	0.929	0.086	0.033	0.434	2.270	**0.137**	63.2 G	-	-	5.3 M	-	20.4 MiB
Ken Burns	MD, NYUv2, KB	0.948	0.070	0.026	**0.438**	2.968	0.140	59.4 G	-	-	99.9 M	-	381.0 MiB
PyD-Net	CS → K	0.836	0.148	0.052	0.310	5.218	0.198	-	-	-	2.0 M	-	7.9 MiB
Tiefenrausch (baseline)	MD	0.942	0.078	0.031	0.383	1.961	0.156	18.9 G	-	-	3.0 M	-	11.4 MiB
Tiefenrausch (AS + no-quant)	MD	0.940	0.080	0.031	0.378	1.987	0.157	**6.4 G**	-	-	3.5 M	-	13.4 MiB
Tiefenrausch (AS + quant)	MD	0.941	0.079	0.031	0.382	1.950	0.156	**6.4 G**	**0.23 s**	196.1 MiB	-	3.5 M	**3.3 MiB**
Tiefenrausch (AS + quant)	MD, 3DP	0.925	0.090	0.035	0.407	**1.541**	0.142	**6.4 G**	**0.23 s**	196.1 MiB	-	3.5 M	**3.3 MiB**

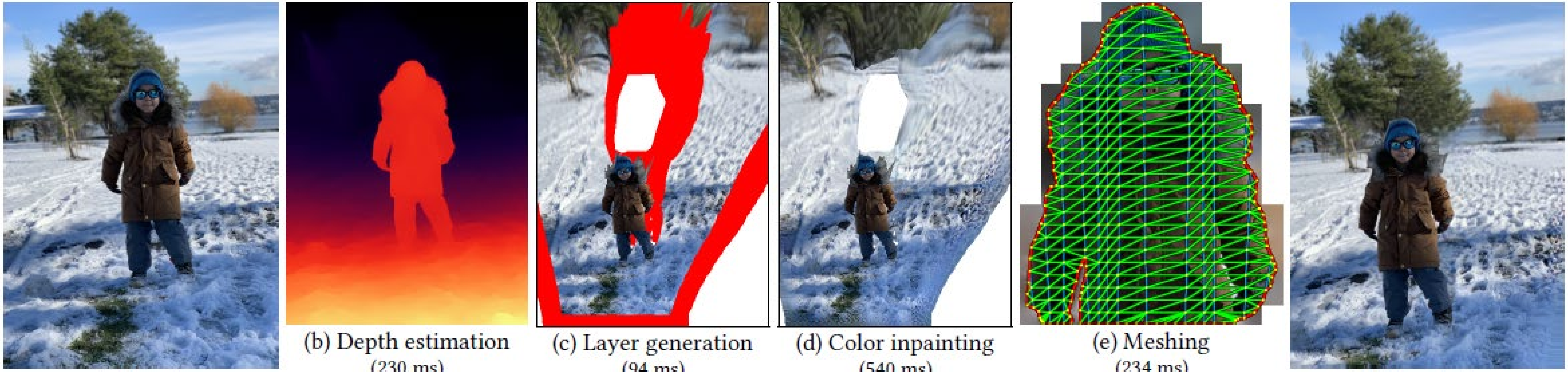




J. Choi, et al., " Pact: Parameterized clipping activation for quantized neural networks," arXiv'18.

Method

- Depth Estimation
- Layer Generation
- Color Inpainting
- Meshing



(a) Input

(b) Depth estimation (230 ms)

(c) Layer generation (94 ms)

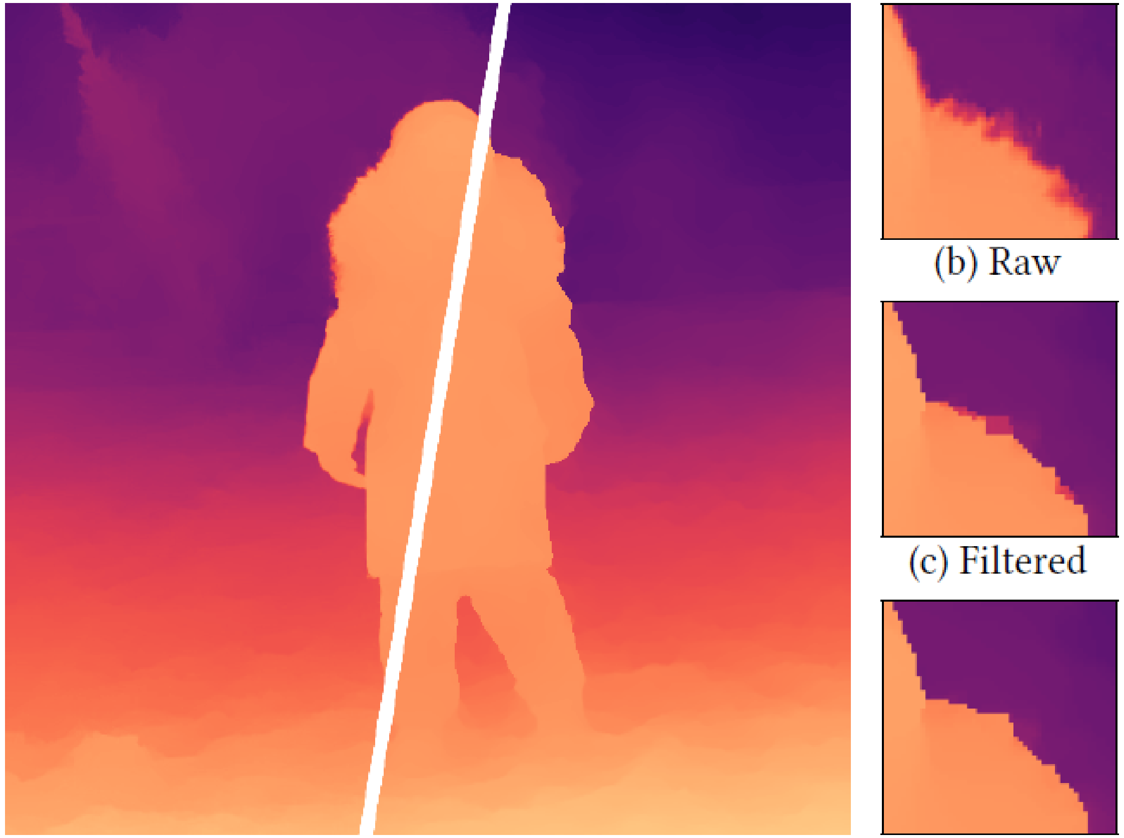
(d) Color inpainting (540 ms)

(e) Meshing (234 ms)

(f) Novel view (real-time)

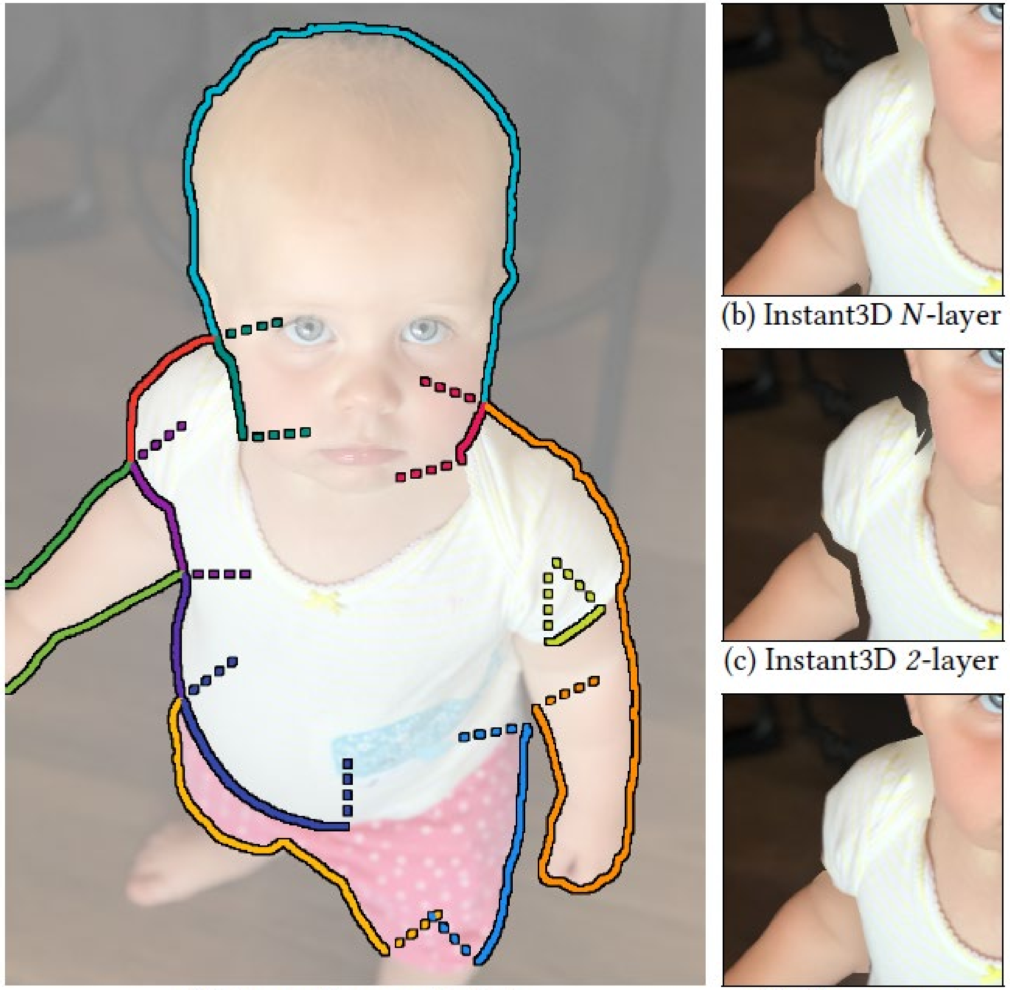
Processing: 1,098ms on a mobile phone (iPhone 11 Pro)

Layer Generation



(a) Raw / cleaned depth image (d) Cleaned

Fig. 4. Depth image before and after cleaning (a). Discontinuities are initially smoothed out over multiple pixels. Weighted median filter sharpens them successfully in most places (c). We fix remaining isolated features at middle-values using connected component analysis (d).

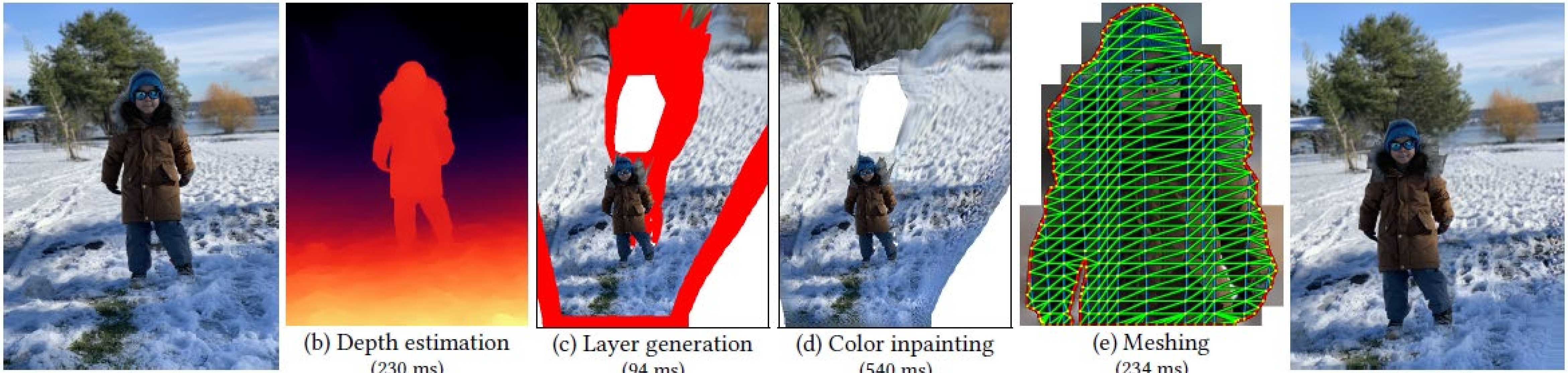


(a) Expansion constraints (d) Our method

Fig. 5. Expanding geometry on the back-side of discontinuities into occluded parts of the scene. Previous work [Hedman and Kopf 2018] produces artifacts at T-junctions: either extraneous geometry if left unconstrained (b) or cracked surfaces when using their suggested fix (c). We improve this by grouping discontinuities into curve-like features (color-coded), and inferring spatial constraints to better shape their growth (dashed lines).

Method

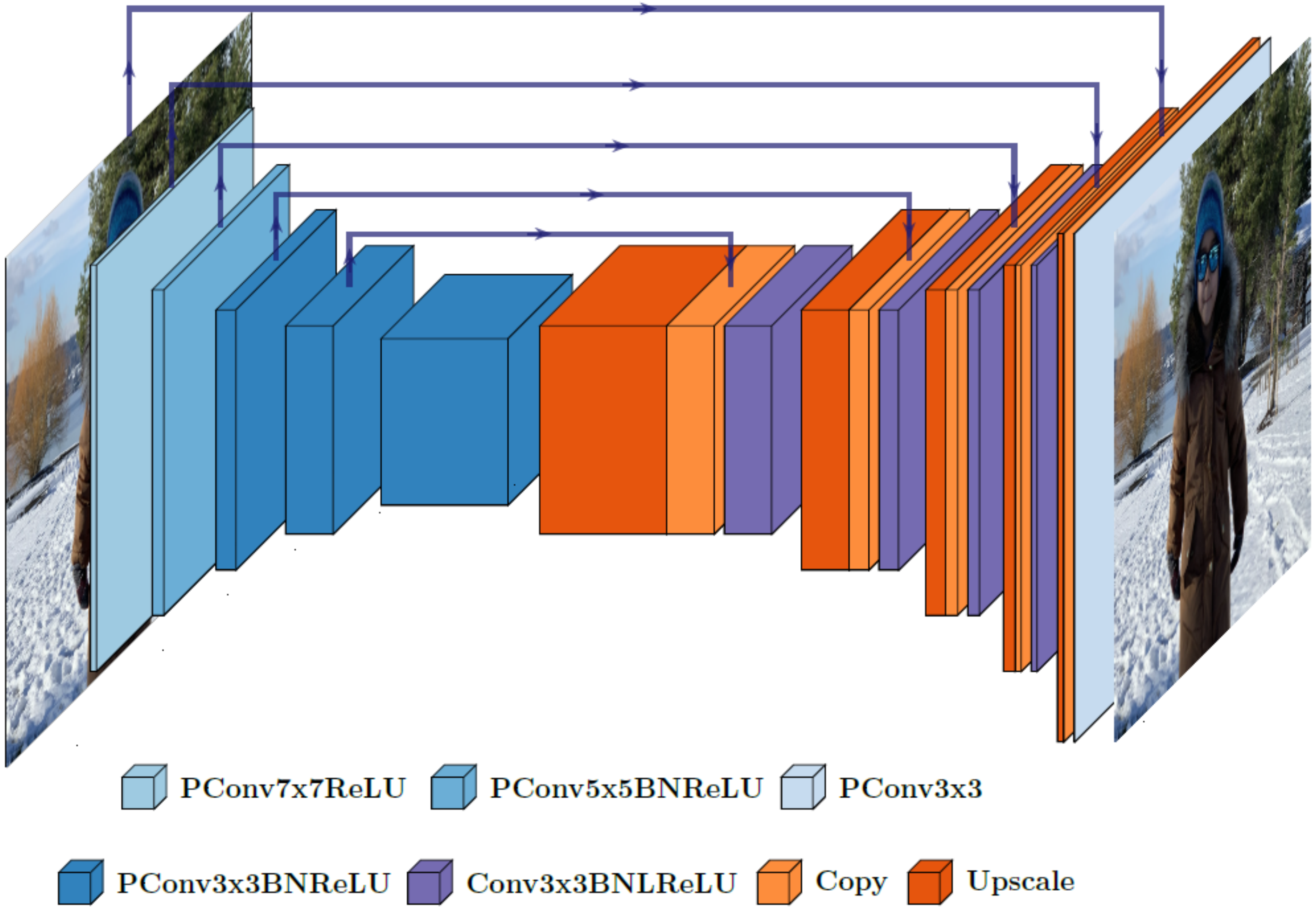
- Depth Estimation
- Layer Generation
- Color Inpainting
- Meshing

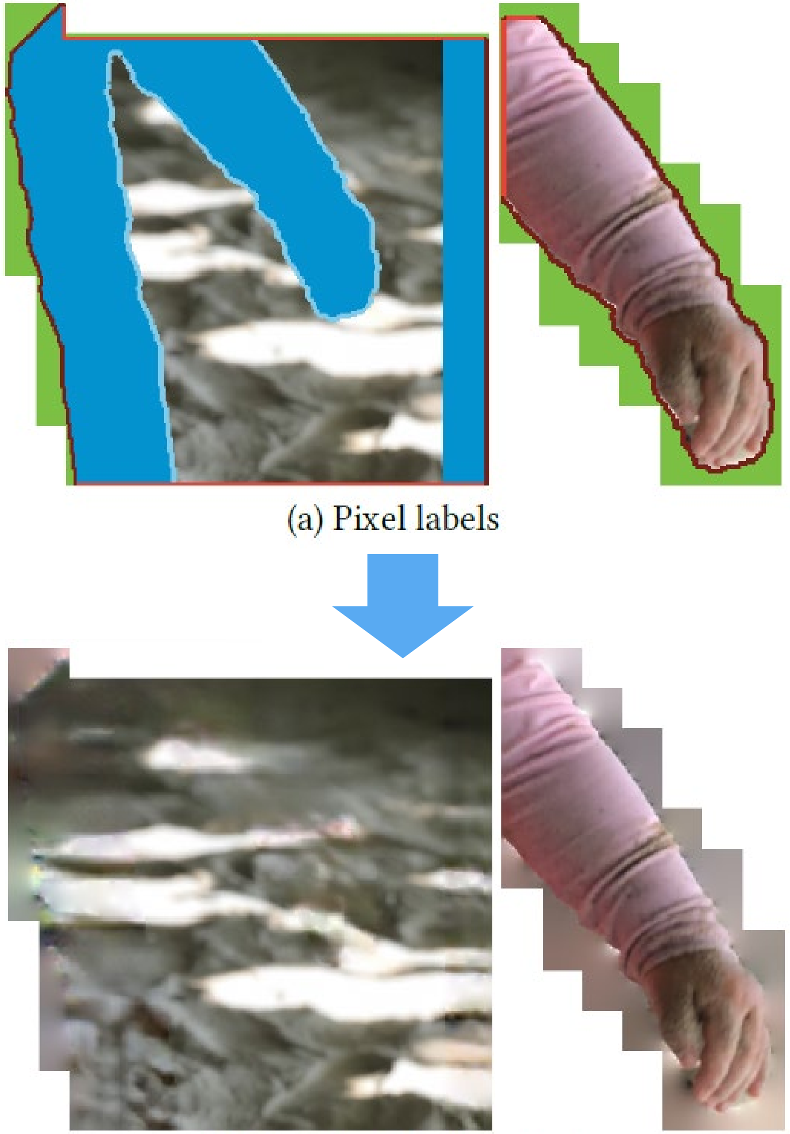


(a) Input (b) Depth estimation (230 ms) (c) Layer generation (94 ms) (d) Color inpainting (540 ms) (e) Meshing (234 ms) (f) Novel view (real-time)

Processing: 1,098ms on a mobile phone (iPhone 11 Pro)

Color Inpainting

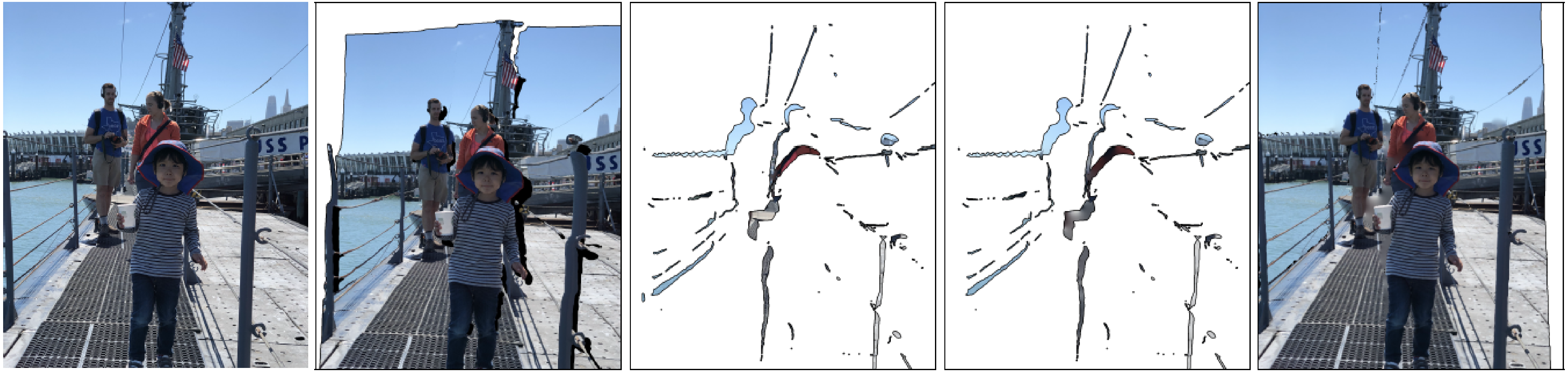




(a) Pixel labels

(b) Inpainted and padded

Color Inpainting



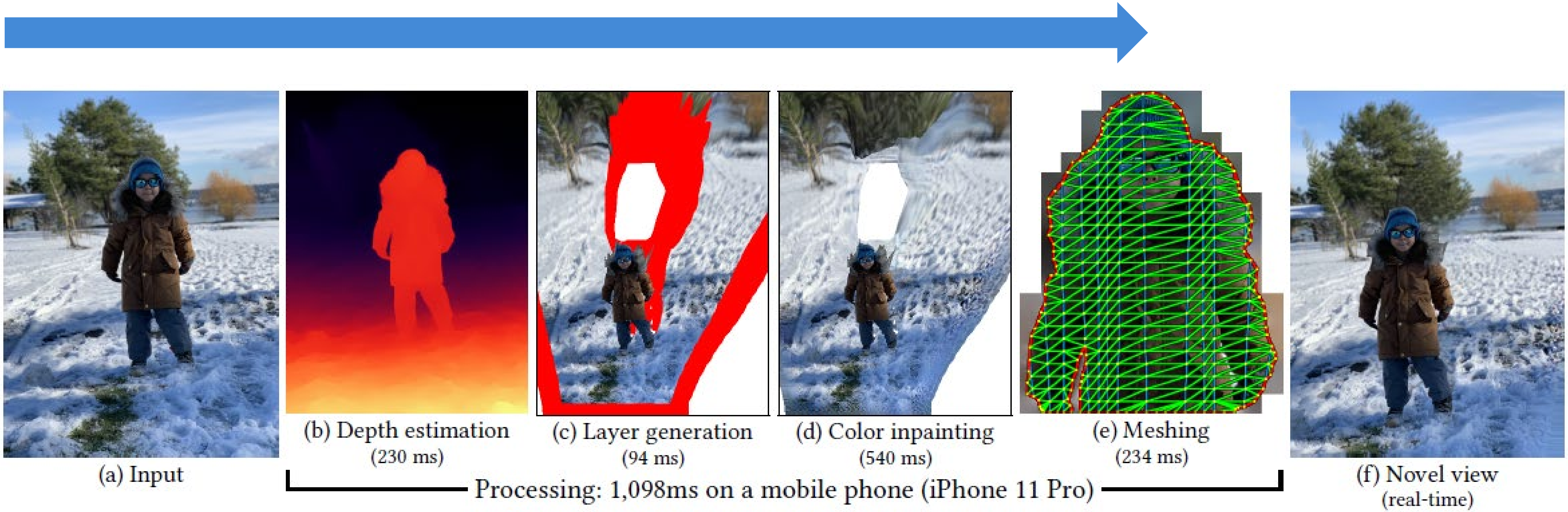
(a) Original view, ground truth | (b) First layer, ground truth | (c) Second layer, ground truth | (d) Second layer, Inpainted | (e) Original view, inpainted

Novel view

Method	Quality (LDI) PSNR↑	Quality (reprojected) PSNR↑	SSIM↑	LPIPS↓	Performance FLOPs↓	Model footprint float32↓	Caffe2 Size↓
Farbrausch	**33.852**	**34.126**	0.9829	0.0232	-	**0.37 M**	**1.9 MiB**
Partial Convolution	33.795	34.001	**0.9832**	**0.0224**	-	32.85 M	164.4 MiB
Farbrausch (screen space)	-	32.0211	0.9784	0.0325	**2.56 G**	**0.37 M**	**1.9 MiB**
Partial Convolution (screen space)	-	33.225	0.9807	0.0280	37.97 G	32.85 M	164.4 MiB

Method

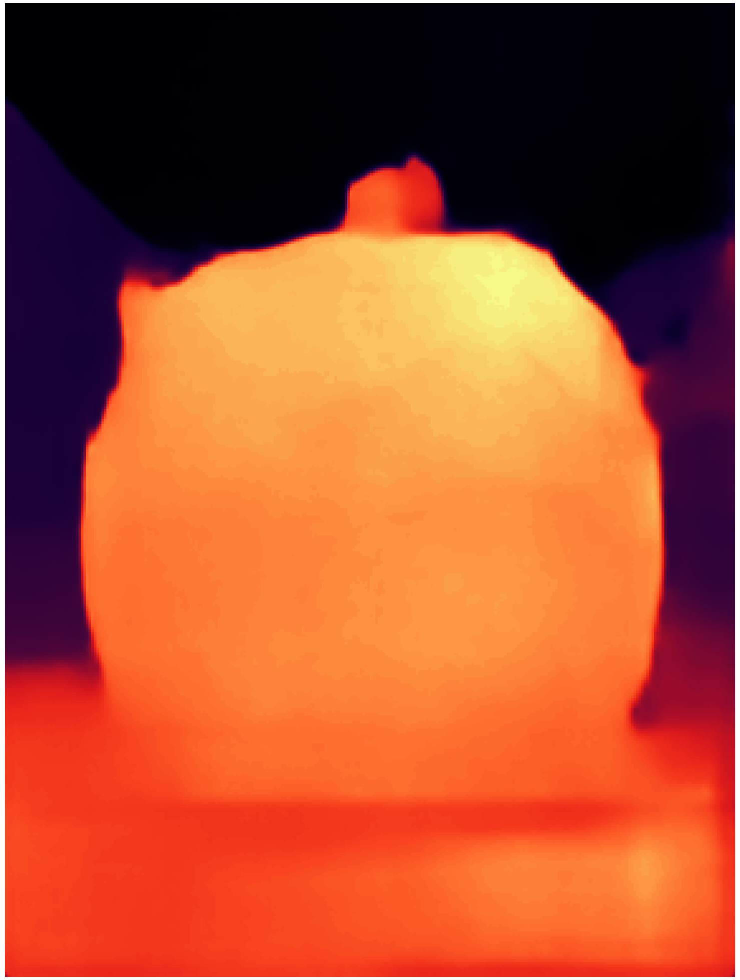
- Depth Estimation
- Layer Generation
- Color Inpainting
- Meshing



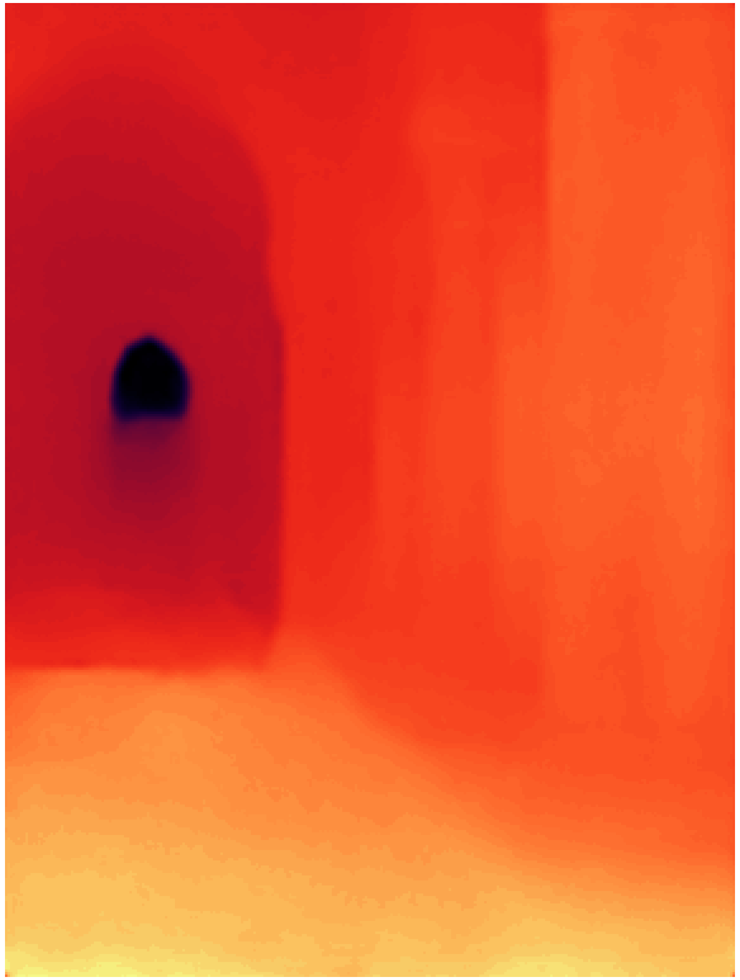
Meshing

(a) Detailed polygon (b) Simplified polygon (c) Interior vertices (d) 2D triangulation (d) Lifted to 3D

Result



Result



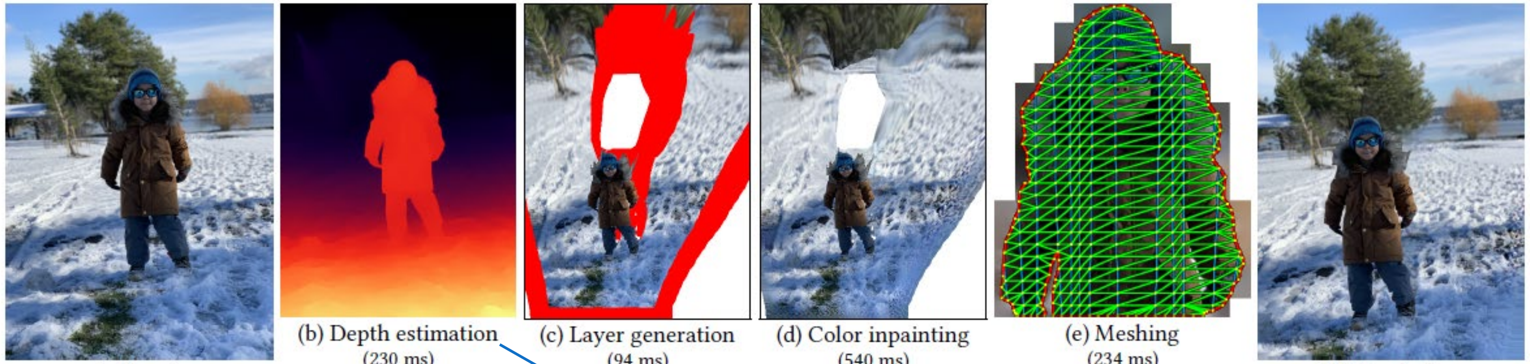
Result

https://facebookresearch.github.io/one_shot_3d_photography/comparison_ken_burns.html

Connection

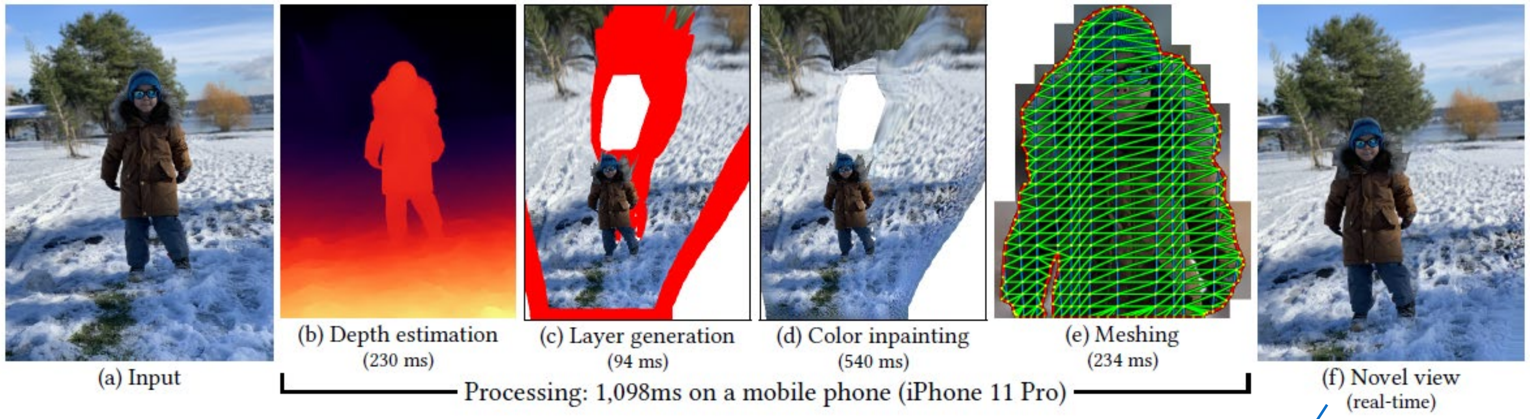
- **Effort:** the capture can been occur in a single shot and not require any special hardware.
- **Accessibility:** creation have been accessible on any mobile device, even devices with regular, single-lens cameras.
- **Speed:** all post-capture processing should at most take a few seconds (on the mobile device) before the 3D photo can be viewed and shared.
- **Compactness:** the final representation have been easy to transmit and display on low-end devices for sharing over the internet.
- **Quality**: rendered novel views should look realistic; in particular, depth discontinuities and disocclusions have been handled gracefully.
- **Intuitive Interaction:** interacting with a 3D photo is in real-time, and the navigation affordances intuitive.

Demo



(a) Input
(b) Depth estimation (230 ms)
(c) Layer generation (94 ms)
(d) Color inpainting (540 ms)
(e) Meshing (234 ms)
(f) Novel view (real-time)
Processing: 1,098ms on a mobile phone (iPhone 11 Pro)

Demo


(a) Input
(b) Depth estimation
(230 ms)
(c) Layer generation
(94 ms)
(d) Color inpainting
(540 ms)
(e) Meshing
(234 ms)
(f) Novel view
(real-time)
Processing: 1,098ms on a mobile phone (iPhone 11 Pro)


BOSTON • DUCK • TOURS
LAND & WATER


BOSTON • DUCK • TOURS
LAND & WATER