The Visual Computer (2021) 37:2931–2949
https://doi.org/10.1007/s00371-021-02200-8

SURVEY

Image and video processing on mobile devices: a survey

Chamin Morikawa[1] · Michihiro Kobayashi[1] · Masaki Satoh[1] · Yasuhiro Kuroda[1] · Teppei Inomata[1] · Hitoshi Matsuo[1] · Takeshi Miura[1] · Masaki Hilaga[1]

Accepted: 5 June 2021 / Published online: 21 June 2021

Abstract
Image processing and computer vision on mobile devices have a wide range of applications such as digital image enhancement and augmented reality. While images acquired by cameras on mobile devices can be processed with generic image processing algorithms, there are numerous constraints and external issues that call for customized algorithms for such devices. In this paper, we survey mobile image processing and computer vision applications while highlighting these constraints and explaining how the algorithms have been modified/adapted to meet accuracy and performance demands. We hope that this paper will be a useful resource for researchers who intend to apply image processing and computer vision algorithms to real-world scenarios and applications that involve mobile devices.

Keywords Mobile devices · Computer vision · Image processing

1 Introduction

In general terms, a *mobile device* is an electronic device that can easily be moved from one place to another. Early mobile devices had limited functionality and performance when compared to their non-mobile counterparts. An example is the portable wireless handset (*walkie-talkie*). With the miniaturization of computing hardware and advances in battery technology, however, this gap has been closing. The category of mobile devices has now expanded to include a variety of gadgets such as laptops, mobile phones, game consoles, smartphones, tablets, and smart watches.

Digital cameras were a later addition to most mobile devices. The first consumer laptop [103] was made in 1982, and the first webcam [44] made it to the market only in 1994. Even then, it was not a built-in-device but a peripheral. The first mobile device with a built-in camera was the mobile phone. In 2000, Samsung and Sharp released mobile phones with built-in rear cameras [20]. Interestingly, it took another 6 years before a laptop with a built-in camera was made. Convenience of carrying and use, and the ability to quickly share photographs, resulted in quick early adoption of camera phones [24]. By the time smartphones arrived, built-in cameras on a mobile phone were an expected feature.

The early cameras on mobile devices were much more limited in functionality and image quality than analog cameras, or even digital cameras. The common form factors of mobile devices [39] required the lenses and image sensors to be small, resulting in a relatively small amount of light available for capturing a photograph. With ordinary digital cameras, this can be compensated by increasing the exposure time. However, mobile devices are mostly held in hand during use. The camera is very likely to shake during a long exposure, resulting in blurry photographs. As a consequence of these limitations, the early mobile phone cameras were

✉ Chamin Morikawa
c-morikawa@morphoinc.com
http://www.morphoinc.com

Michihiro Kobayashi
m-kobayashi@morphoinc.com

Masaki Satoh
m-satoh@morphoinc.com

Yasuhiro Kuroda
y-kuroda@morphoinc.com

Teppei Inomata
t-inomata@morphoinc.com

Hitoshi Matsuo
h-matsuo@morphoinc.com

Takeshi Miura
tmiura@morphoinc.com

Masaki Hilaga
m-hilaga@morphoinc.com

[1] Morpho Inc, Chiyoda-ku, Tokyo 101-0065, Japan

mostly used in situations where it is difficult to use a larger camera.

However, even the early mobile devices had one advantage over other digital cameras. Camera phones had more computing power when compared to low-end digital cameras. The presence of a digital camera and computing power on the same device provided an opportunity for processing the photographs before the user got to see them. Digital image processing techniques had been used in several application areas as early as in 1960s [65] and were already an established field of research by the time the first camera phones arrived. Therefore, it did not take much time before computer vision and image processing algorithms started getting deployed in mobile devices.

Images acquired using mobile devices can—in theory—be processed with generic image processing algorithms. However, there are numerous constraints and external issues that make such algorithms less effective when directly applied. Most mobile devices still have less computing power when compared to computers that are specifically designed for image and video processing. They are also battery-powered, and the limited energy contained in the battery has to be conserved for the primary function of the device (in case of a mobile phone, connectivity, and communication). Smartphones in particular are bringing in additional challenges to mobile image processing. Market competition has resulted in smartphones with cameras that have very high resolutions, having a large number of pixels to process calls for efficient algorithms. Another important, non-technical factor is the need to meet the user expectations. Smartphone users expect high-quality photographs and videos from their devices, and they also want the processed content to be available very quickly.

In this paper, we survey image processing and computer vision algorithms that are used on mobile devices, explaining how the algorithms have been modified/adapted to work under the constraints and requirements mentioned above. As industry researchers whose primary focus is mobile image processing, we believe that we have a good grasp of research advances, device limitations, and user needs, all of which are key factors in selecting the right algorithm for a given task. Focusing the survey on mobile devices will help researchers, who intend to apply image processing and computer vision algorithms to mobile applications, to identify the critical issues and select algorithms that are more effective for the intended usage scenarios.

While drones and autonomous vehicles are not mobile devices in the above context, the technology surveyed in this paper can also be deployed in these two device categories. This is because they share several characteristics and limitations that mobile devices do. Examples are limited processing power, use of battery power, and unstable photography due to movement.

The rest of this paper is organized as follows: Sect. 2 describes the evolution of mobile devices to their current state, in terms of hardware, features, and software libraries. Section 3 surveys in detail the algorithms for still image processing, while Sect. 4 covers video processing techniques. Section 5 introduces image processing techniques that make use of multiple cameras, or cameras supported by other multidimensional sensors, that have an overlapping view of the same scene. Section 6 is a brief but broad survey of image analysis, with particular emphasis on machine learning-based techniques. After a quick look at recent trends (Sect. 7), we conclude the paper in Sect. 8.

2 Mobile devices as image processors

As mentioned in Sect. 1, the cameras in mobile devices lack larger lenses and sensors, requiring image processing techniques to improve image quality. The present-day mobile devices contain both hardware and software that support faster and more efficient image processing. This section is a brief survey of such hardware and software.

2.1 Processing units

According to the von Neumann architecture [79] for digital computing devices, the central processing unit (CPU) of a computer handles all computational tasks. However, by the time camera phones were developed, this architecture had already evolved to distribute computing tasks in multiple ways. Multiple cores within the CPU operate in parallel to speed up computations. Graphics processing units (GPUs) handle the display of content at high resolution and frame rates. Digital signal processors (DSPs) are used where extensive signal processing is required. These hardware technologies were already available when the first smartphones were introduced to the market. While von Neumann architecture is based on serial data processing, parallel processing of image data was an active research topic by the time commercial digital cameras became available [19].

Due to the large number of pixels in an image that has visually pleasing resolution, image processing algorithms are computationally intensive. The presence of multiple CPU cores can allow the device to make sure that the CPU is not fully occupied with image processing, allowing the device to perform other required tasks. This is especially useful since image processing is not the primary function of many consumer mobile devices; for instance, the primary function of a smartphone is to keep the user connected to a mobile communication network. Surveys by Georgescu [30] and Singh et al. [93] demonstrate how mobile CPUs became more powerful and feature-rich over time.

GPUs were originally designed to relieve the CPU from the burden of rendering high-quality graphics on display devices. They consist of a large number of smaller computing units, so that graphics operations on pixels could be carried out in parallel. Almost all present-day mobile devices have GPUs in them. However, it was soon observed that the computing power of a GPU was not fully utilized at all times, depending on what has been displayed on the device. This resulted in general-purpose computing on graphics processing units (GPGPU), allowing computing tasks that are normally carried out on a CPU to be performed on the GPU [57]. Given that images contain a large number of pixels, and most image processing techniques require a repetition of operations on some or all of these pixels, a GPU is a good candidate for handling image processing tasks. Consequently, it is quite common to use the GPU of a present-day mobile device for image processing tasks. Software libraries such as OpenCL can be used for using GPUs for general purpose computing tasks.

Deep neural networks (DNNs) are an AI technology that is very effective in a variety of image processing and analysis tasks. DNNs benefit from the ability to process image pixels in parallel, and therefore the parallel architecture of GPUs. On mobile devices, the already-present GPU could be used for deep neural inference (processing or analyzing data using DNNs) when the workload of rendering graphics is low. Hardware manufacturers have also designed *AI accelerators* that are dedicated for implementing DNNs. Some mobile device makers such as Apple (*Bionic AI Processor*), Google (*PixelCore*), and Qualcomm (*Hexagon 780*) have added hardware AI accelerators to their devices [5,84].

2.2 Cameras and additional sensors

Cameras in mobile devices have improved in terms of sensor resolution; ten Megapixel smartphone cameras are common at the time of writing. Given the constraints due to form factor, however, it has been difficult to improve the optics of mobile cameras. Multiple cameras facing the same direction (see Sect. 5) have been added to smartphones as a solution to this problem.

Mobile devices also contain other sensors that are either essential for their main function (for example, microphone for a mobile phone), or auxiliary (such as gyrosensors and accelerometers on a smart phone). Data from some of these sensors can be used for improving the quality of photographs and enable more accurate analysis of the scene captured in the photograph.

2.3 Software Neural engines

Software libraries (middleware) that cater for faster deep neural inference are another new addition to mobile devices. NVIDIA's CuDNN library is one of the earliest of such libraries, though it was not specifically designed for mobile devices. *ARM Compute Library* [7], Qualcomm's *Snapdragon Neural Processing engine*(SNPE) [83], and Apple's *CoreML* [3] library are some examples that are specifically designed for the platforms of the respective makers. *Tensorflow Lite* [51] and Morpho's *SoftNeuro* [74], on the other hand, provide support for multiple platforms. Tensorflow Lite is open-source and provides a high-level application programming interface (API) for several programming languages, facilitating quick development. SoftNeuro has industry-oriented features such as model encryption and the ability to apply device specific tuning to trained machine learning models, for faster inference.

3 Image processing

This section focuses on still image processing for mobile devices. An image processing algorithm accepts an image as input and outputs a modified version of the input image. A common example for image processing on a smartphone is to improve the visual quality of a photograph taken by its user. The rest of this section highlights the motivation for mobile image processing and describes a few techniques that are widely used.

3.1 Characteristics of mobile image processing compared to DSLR Cameras

One of the most significant differences of hardware specifications between mobile devices and DSLR (digital single lens reflex) cameras is the size of image sensors. DSLRs are equipped with image sensors with sizes such as full frame (36×24 mm), APS-H (28.7×19 mm), APS-C (22.2×14.8 mm), and so on. All of these are more than ten times larger in surface area than those for mobile devices. Pixels in a larger sensor can receive a larger amount of light. This contributes not only to capture bright images with less noise even in dark scenes but also to enlarge the dynamic range, thereby preventing over-exposure and under-exposure.

Ability to attach a variety of lenses is another functionality of DSLRs. Lenses with large aperture facilitate capturing bright images as well as a more impressive *bokeh* effect. On the other hand, lenses with long focal lengths provide optical zoom with high zoom ratio. Such lenses tend to be thick because they consist of systems of multiple lenses. The form factors of mobile devices do not allow for thick lenses, limiting their optical zoom potential. Hence, most cameras for mobile devices need to rely on digital zooming.

From the early days of mobile image processing in feature phones, engineers have taken efforts to improve image quality using software. Improved image quality due to image

processing combined with other advantages (portability, and all-in-one functionality including image editing and network connectivity) has helped mobile devices to successfully establish themselves as alternatives to digital cameras. More and more people take photographs with their smartphones instead of carrying around a digital camera, not only in their daily lives but also for trips and other events.

3.2 Basics of merging multiple images

In order to overcome the limitations of mobile devices with respect to noise, dynamic range and zoom quality, one of the most powerful approaches by software is to capture multiple images in a short interval and merge them to synthesize into a result image [98]. Multi-frame noise reduction (MFNR), high dynamic range (HDR) compression, and super-resolution are techniques to reduce noise, enlarge dynamic range, and increase resolution, respectively.

There are three advantages in smartphone hardware, when compared to DSLRs: (1) faster shutter speed and frame rate due to electronic shutters, (2) large amount of RAM, and (3) powerful processors.

Using electronic shutters enables capture at extreme high speed such as 1/32,000 s, while mechanical shutters generally max out at 1/4000 or 1/8000 s. The faster the shutter speed is, the more robust the image is to motion blur, which is difficult to be removed by post-processing. Because of the absence of moving mechanism in electronic shutters, the frame rate can be increased, shortening the interval between captures. Electronic shutters enable us to shoot with 30 frames per second (fps), while mechanical shutters support around 10 fps. It is highly important for merging images to shorten the interval of captures in order to reduce ghosting artifacts. Therefore, smartphones have more potential in terms of capturing multiple images for post-processing than DSLRs.

Recent image sensors have resolutions of higher than ten megapixels, which sometimes requires more than 100 Megabytes of memory to hold multiple captured images. Smartphones are equipped with several Gigabytes of RAM because they are designed to run rich applications such as web browsers, media players, and games. In addition, such large size of RAM is embedded in SoC (System-on-a-chip), not as an external storage, providing smartphones with sufficient data throughput to store images capture with high frame rates.

Recent smartphones also have powerful processors, which are comparable with those for desktop and laptop computers. As mentioned in Sect. 2, other types of hardware resources are also available for image processing.

In conclusion, smartphones have mechanisms and computational resources to compensate for the disadvantages of the optical system by image processing. In the following subsections, we will deep dive into some algorithms to enhance image quality of smartphones by merging multiple images.

3.3 Blur and image alignment

When we take a picture without using a tripod, the camera may move slightly due to hand jitter during capture even if we take as much care as possible. Such movement causes two kinds of blur [82].

One is called *motion blur*, which results in a streaking of moving objects. Because motion blur occurs when a camera or objects in a scene move during capture, it tends to occur when capturing with long exposure time. Motion blur is hardly removed once caused. Therefore, keeping exposure time short is an important factor to prevent from motion blur.

The other is ghosting artifacts, which is caused by the synthesis of the same object to different positions, while merging objects in multiple images taken with intervals. Ghosting artifacts are caused by both global motion due to hand jitter and local motion due to moving objects in a scene. The former can be reduced by aligning images to be merged to cancel out global motions. The latter should be treated in merging algorithms, but it can be made easier by shortening intervals of continuous shooting.

In the early development of cameras for feature phones, manufacturers considered embedding a motion sensor in the phone. However, it was difficult to install a motion sensor in addition to a camera mechanism in a small housing due to space constraints and was also unprofitable in terms of price. SOFTGYRO® [76], which was developed by Morpho Inc. in 2004, is a software-based estimator for motion between images. The offset between similar image features on adjacent frames of a multi-frame sequence was used to estimate camera motion.

3.4 MFNR—Multi-frame noise reduction

As described above, it is important to take pictures with high shutter speed to prevent motion blur. The drawback of increasing shutter speed is the need to *gain up* pixel values to obtain bright images, which results in noisy images (especially in night scenes). Denoising filters are typical solutions to reduce noise, but they also lose some texture in the scene.

Noise reduction in images is a well-researched topic, and a variety of techniques is available to choose from [35]. Multi-frame noise reduction (MFNR) is a technique to reduce noise by merging multiple images taken within short intervals under the same capture parameter. Assuming that additive independent and identically distributed (i.i.d) noise contaminates an original image, averaging pixel values is the simplest and the most effective approach to reduce noise. As described above, image alignment is mandatory for MFNR to prevent ghosting artifacts. In addition, MFNR needs to detect moving

objects in a scene. When moving objects are detected, they should be either aligned locally before merging or excluded from merging.

MFNR is a common noise reduction scheme on recent smartphones, especially for night scenes. While some smartphone manufacturers research in-house MFNR algorithms, others use MFNR products licensed by software vendors. Morpho's PhotoSolid® [76] is an example MFNR-capable product that has been widely used in smartphones globally.

3.5 High dynamic range (HDR) imaging

Dynamic range of a scene is a relative scale of irradiance in the entirety of the captured scene. On the other hand, dynamic range of an image sensor is a relative scale of a signal that one pixel element of the sensor can detect. If the dynamic range of an image sensor is narrower than that of a scene, high irradiance exceeds the capacity of the pixel element to white out (over-exposure), and/or low irradiance falls below the threshold of the pixel element to detect a signal to black out (under-exposure).

Because over- and under-exposed pixel values cannot be recovered from a single capture, one of the most effective ways to store the information in high dynamic range scenes is to take multiple captures with different amount of exposure [16]. If parameters of an image acquisition pipeline can be obtained, we can estimate the true radiance values in the scene. Because estimated radiance values at the same position of a scene are proportional to the exposure levels of the captures, they can be merged to compensate lost information by over- or under-exposure. A good survey on HDR imaging is available in [27]. Recent research has also focused on HDR imaging for mobile devices, due to more challenging requirements [38].

Images merged with multiple exposures should be basically stored using a data format with more bits than the number of bits used during capture, in order to preserve both their dynamic range and tone. On the other hand, displays and commonly used image file formats only support images with a lower number of bits (for example, 8 bits per channel). A technique called *tone mapping* is used to map the levels in the merged image to the output. Local tone mapping, which adapts mapping to local appearance, is commonly used to emphasize local contrasts (textures) while limiting the global dynamic range.

Figure 1 shows an example of HDR merge. Three images at the top row are the input images to HDR algorithm, which are sequentially taken with different exposures. In Fig. 1b, while the stained glasses are over-exposed, the other areas are relatively dark. HDR algorithm compensates the textures of the stained glasses from the low exposure input (Fig. 1a) and brightens the other areas by merging the middle exposure (Fig. 1b) and the high exposure (Fig. 1c) images with the optimal weight ratios with respect to the local brightness.

One problem in the above approach is that the parameters of the image acquisition pipeline in smartphones are rarely available to those outside the companies involved in device manufacture. Image processing software vendors therefore use methods that do not depend on such parameters. Morpho HDR™ [76] is one such product that directly merges images taken with different exposures and performs tone mapping to synthesize an output image without using the linearized relation of radiance values.

3.6 Super-resolution

Pixels in an image sensor sum up the number of incoming photons; therefore, taking a photograph with a digital camera amounts to applying a low-pass filter to a scene. Tiny textures (such as characters that measure a few pixels) are observed blurry. This loss of resolution is increasingly noticeable when performing digital zoom. Conventional image resizing algorithms, such as bilinear interpolation, cannot recover the high-frequency information of a scene once lost.

Super-resolution imaging, [99] a technique to increase the resolution of an image, is not just a interpolation of signals. Super-resolution algorithms are classified into two categories: (1) approaches that reconstruct a high resolution image from itself and (2) approaches that register multiple low-resolution images to interpolate sub-pixel information.

Super-resolution from a single image exploits prior information about images of natural scenes such as fractals, which assumes that there are large-scale structures in an image that are similar to the small structures to be super-resolved. Registration-based approaches assume that input images are misaligned to each other due to hand jitter and thus sub-pixel information can be derived. Some cameras have a functionality named *pixel shift*, which captures multiple images while shifting the image sensor. Such active and controlled shift of images enables registration-based super-resolution even if a camera is mounted on a tripod. We have developed Morpho Super-Resolution™, [76] a registration-based super-resolution product that does not require pixel shift. Image alignment at sub-pixel level, using the motion estimator software SOFTGYRO® [76], contributes to the increase of resolution of tiny textures.

Figure 2 shows a comparison between the images upsampled by bilinear interpolation (Fig. 2a) and super-resolution (Fig. 2b). Bilinear interpolation results in blurry edges because it cannot reproduce the high-frequency components by interpolation. Super-resolution, on the other hand, can reproduce the sharp edges and the textures.

Fig. 1 HDR merge. **a** Low exposure input. **b** Middle exposure input. **c** High exposure input. **d** Result of HDR algorithm

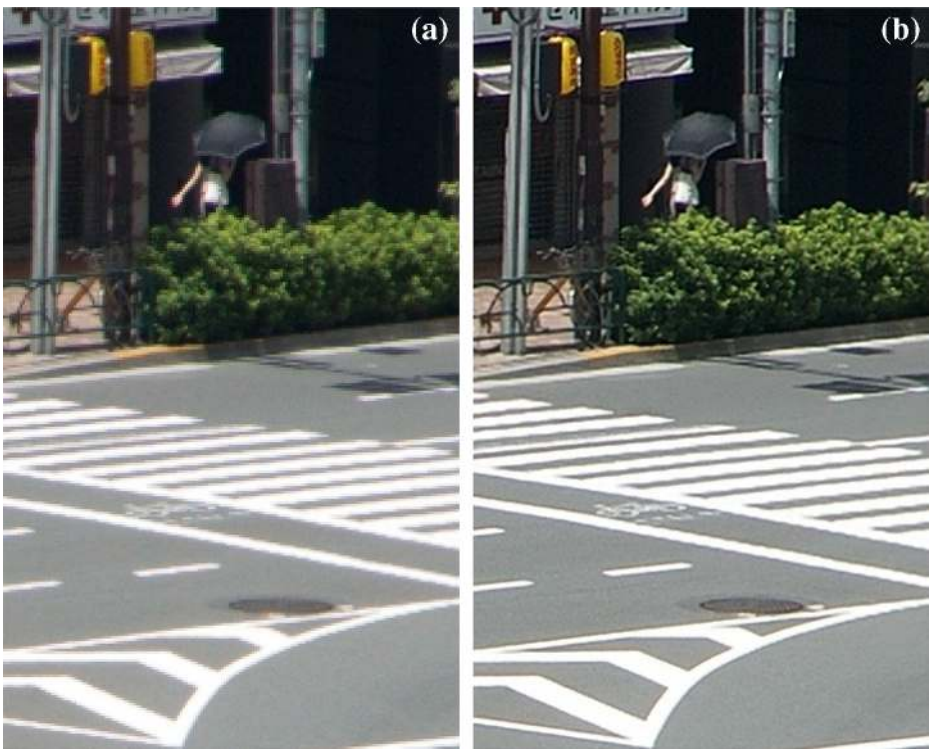

Fig. 2 Comparison between bilinear upsampling and super-resolution. **a** Upsampled image by bilinear interpolation. **b** Upsampled image by super-resolution

3.7 Remarks

The algorithms that we described in this section are based on *Computational Photography* (CP). CP-based techniques try to model optical processes using computations on the captured image data. While this has been the established method for image processing until recently, *machine learning* (ML) methods are increasingly being applied to solve the same problems. Machine learning methods use statistical patterns learned from a collection of *training data*, to process previously unseen data. For instance, an ML-based super-resolution image processor infers finer details from learned patterns while enlarging an image. Machine learning-based methods require a larger amount of computing resources when applied to high-resolution images. Also, their black box nature makes it difficult to explain and correct problems that occur with specific input data. Due to these limitations, CP-based techniques are still more popular for still image processing on mobile devices.

4 Video processing

A video consists of a series of image frames. Applying a single frame image processing algorithm to each frame, we obtain a solution for videos. Then, a question arises: Do we really need special algorithms for videos? The answer is "yes." We will justify this answer with four reasons.

The first reason is an obvious one. Some quantities are defined only for videos. An important example is inter-frame camera shake, which leads to shaky videos. Video stabilization, an algorithm to suppress it, has no counterpart in still image technologies, because they have no inter-frame values. Note that still image stabilization is for suppressing *intra*-frame camera shake.

The second is the difference between spatial and temporal processing. Simply put, the goal of still image algorithms is to create clear and natural images. The point here is that "natural" means natural as a single frame. A natural video consists of natural frames, but a sequence of natural frames does not always form a natural video. There are many "natural as a still image, but not as a video" examples. For instance, suppose that there is an apple in a given video frame. If this apple disappears in the following frame, that looks unnatural as a video, but the frame may still be natural as a single photograph. Because still image algorithms only make use of spatial information, temporal naturalness cannot be guaranteed. Therefore, we need to design algorithms that look after the naturalness of video. Further, if we take temporal information into consideration, results of processing will be better.

The third is processing time. Think about shooting a 60 FPS video. Then, any real-time video processing solution must finish its process for each frame within 1/60th of a second. For the still image case, we do not have this kind of strict processing time limit. For this reason, we should develop much faster algorithms than their still image counterparts.

The fourth and the last reason is power consumption. Think of a fast enough, but power-consuming algorithm. Taking a still image is a one-shot process. A single peak in the power consumption graph might not cause any severe problem to the device, or the user. However, shooting a video is a continuous and relatively long process. A power-consuming algorithm can cause a smartphone to heat up. Because this is not an acceptable situation for many phone or device makers, we have to keep algorithms lite, or less power consuming.

Video stabilization [107], noise reduction, [88] frame rate conversion [81], and motion blur reduction, to name a few, are widely used video processing algorithms. At the time of writing, most video processing algorithms are based on conventional image processing approaches rather than machine learning-based approaches. This is because conventional vision technologies have so far been faster and more stable. However, this situation is changing rapidly. The challenge we are facing at the moment is to find effective approaches that can process video using the current machine learning techniques.

In the following subsections, we will briefly explain two typical video solutions: video stabilization and noise reduction.

4.1 Video stabilization

In this subsection, we abbreviate inter-frame camera shake as camera shake. As stated above, video stabilization is an algorithm only for video. Hence, this is a suitable example to begin with. Stabilization algorithms are classified into either 3D or 2D models. In the 3D algorithm, both scenes and camera positions/orientations are reconstructed in 3D world coordinates. Then, we virtually recapture a video with a smoothed trajectory to obtain a stabilized output. In theory, it is an ideal algorithm. However, for mobile video processing, it is too unstable and heavy.

In the 2D algorithm, there is no reconstruction. We approximate camera movement as 2D homographies between frames and smooth them out using a filter. Suppose that $\mathbf{x}_i \in \mathbb{P}^2$ is coordinates of i-th frame, and H_i is the approximated homography between first and i-th frames:

$$\mathbf{x}_i = \mathrm{H}_i \mathbf{x}_1. \tag{1}$$

Think about a transformation of i-th frame $\mathbf{x}_i \to \mathbf{x}'_i$:

$$\mathbf{x}'_i = \mathrm{H}_i^{-1} \mathbf{x}_i. \tag{2}$$

This transformation is equivalent to the alignment of the i-th frame to the first. Thus, every single frame can be aligned to the first frame. In some sense, this is perfect stabilization, because there is no camera motion left in the output. The problem of this simple algorithm, however, is that it also wipes out intentional camera motion such as panning and tilting. We need to design a filter f to separate camera shake from intentional motion and suppress only the former:

$$\mathbf{x}'_i = f\left(\mathrm{H}_i^{-1}\right)\mathbf{x}_i. \tag{3}$$

This equation shows what 2D video stabilization does in the simplest form. The 2D model has only limited stabilization capability, but it is robust and fast. Hence, in the mobile field, the 2D algorithm is a more realistic choice. Morpho's video stabilization product, MovieSolid® [76], is an example video processing solution based on 2D video stabilization.

One drawback of the 2D model is the smaller angle of view in the stabilized video. The shape of input frames is rectangular, but warping of the 2D model changes this. For many use cases, non-rectangular video is not acceptable. To retain rectangular shape, we have to crop the output, and this results in a smaller angle of view.

Naively thinking, the most important ingredient of 2D video stabilization seems to be filtering of homographies. Of course, it is important, because a better filter results in more smooth output. However, in practice, there is a more critical issue that has to be handled during processing. To avoid artifacts, or unnaturalness, is top priority in video acquisition for consumer devices. To the human eye, a stabilized video with artifacts looks much worse than a video with no stabilization. Here, the main causes of artifacts are lens and rolling shutter distortion. Figures 3 and 4 show examples of them in extreme cases. Because it is impossible to describe them as 2D homographies, thinking of an algorithm handling them properly is essential for 2D video stabilization development.

4.2 Noise reduction

In order to perform noise reduction, we need to manipulate all the pixels in image frames. Hence, this kind of algorithm tends to be slow and power consuming. A technique widely used to mitigate this difficulty is infinite impulse response (IIR) filters. IIR filters utilize not only the input frame but also the output of past frames. The simplest example is as follows:

$$\hat{x}_i = \alpha\hat{x}_{i-1} + (1-\alpha)x_i, \tag{4}$$

where x_i and $\hat{x}_i$ are pixel values of input and output of i-th frame, respectively. The coefficient α is for controlling the strength of the filter. This simple filter is equivalent to a more complicated one:

$$\hat{x}_i = (1-\alpha)\left(x_i + \alpha x_{i-1} + \alpha^2 x_{i-2} + \cdots\right). \tag{5}$$

This is a filter for summing up input values over an infinite number of frames, and α determines the weight of the past frames. It is difficult to implement filter (5) directly. However, using IIR filter (4), an effectively identical filter can be implemented easily. In this way, IIR filters enable us to realize fast and lightweight noise reduction, compared with still image-based techniques.

For temporal filtering, it is important to compare pixels corresponding to the same 3D world coordinates. For example, pixel values $\hat{x}_{i-1}$ and x_i in filter (4) must point to the same 3D coordinates. Hence, image frames must be aligned. There are two types of alignment: global and local. The global alignment is for compensating camera motion and usually approximated by a 2D homography. The local one is for subject motion, such as walking or waving hands. When misalignment occurs, pixels of different 3D points are added up to create output, and the image gets blurred. Local misalignment is particularly problematic, since it leads to ghosting, as shown in Fig. 5. Therefore, an accurate alignment algorithm and a mechanism to suppress blurring artifacts are necessary to develop video noise reduction algorithms.

In Fig. 6, we show a result of Morpho's video noise reduction technology, Morpho Video Denoiser™ [76].

5 Multi-camera image processing

As mentioned in earlier sections, it is difficult to fit a good lens with a wide range of variable focus on most mobile devices. One simple solution to this problem is to add multiple cameras that are facing the same direction. Each camera can have a different lens, providing the device with a good combination of lenses such as tele, normal, and wide. The camera that is most appropriate for the current scene can be used for taking the photograph.

However, in practice, using multiple cameras leads to additional complications. A user should be able to use a single zoom control to zoom in and out, while the camera is automatically selected in a way that is transparent to the user. Additional software support is required for smooth transition between cameras, especially when capturing videos.

The presence of multiple cameras sharing the field of view also brings in the ability to estimate distances to the objects in the scene. By estimating the relative depth between objects, a *depth map* of the scene can be constructed. The most common use for a depth map is applying a *bokeh* effect, something that is difficult to achieve using small lenses in mobile phone

Fig. 3 Lens distortion. Because of the un-homographic nature of lens distortion, a straight object (pole) appears curved

Fig. 4 Rolling shutter is a widely used shutter mechanism for mobile imaging. It induces artifacts, because the image is captured line by line. In this example, the pole appears unnaturally bent

Fig. 5 Ghosting artifacts. Moving subjects, namely the metronomes and the Ferris wheel, are strongly blurred because of local misalignment

Fig. 6 Input and output of a video denoising algorithm. **a** A frame from an input video. **b** The output of the corresponding frame. The IIR filter reduced noises in the frame (see the building wall), and suppressed artifacts (the woman is waving her hand)

cameras. This effect occurs when the depth of field produced by the lens setting of the camera is shallower than the depth range of the scene that is being photographed. The objects near the distance of focus appear sharp, while other objects appear blurry. Most flagship smartphones apply variable amounts of blur to objects depending on their depth inferred from the depth map, to achieve fairly realistic bokeh effects.

Estimating depth using multiple cameras had already been researched well before the invention of mobile phones [10,91]. Camera calibration is a necessary step for enabling depth estimation using computational imaging techniques. Metric calibration Zhang et al. [105] uses a reference calibration object of which the feature dimensions are known with very high precision. Online calibration [43] using information derived by image processing can be used to get rid of errors in initial calibration during manufacture and also cater for errors in degradation.

5.1 Light field cameras

Light field cameras record both the intensity of light in a scene and the direction that the light rays are traveling in space. They are implemented by placing a *microlens array* in front of a conventional image sensor. The arrangement of lenses and the resulting data enable "post-focusing" images, that is focusing images at multiple distances after capturing them. The concept of the light field camera was proposed by Gabriel Lippmann in 1908, but the first digital implementations with post-focusing ability arrived much later [70,80]. Several manufacturers (*Adobe*, *Lytro*, *Pelican Imaging*) have produced consumer-level light field cameras.

Despite their ability to produce photographs with variable focus and depth of field, light field cameras have not been deployed on mobile devices at the time of writing. The key reasons behind this are their low spatial resolution and high cost. If further research and development facilitate overcom-

ing these two weaknesses, they will be a valuable addition to mobile devices.

5.2 Image-like data from other sensors

Several other alternatives to using multiple cameras have also been researched. Structured light had been used for recording depth information since the early days of machine vision [33]. Structured infrared (IR) is a better alternative for use with ordinary photographs, since it is not visible to the human eye. Microsoft Kinect used structured IR to sense depth and motion for gaming applications [73]. Apple's iPhone uses a structured IR grid of approximately 30,000 points for true depth estimation.

Coded light technology is an extension of structured light. With coded light, a rapidly changing series of infra-red patterns are projected onto the scene while recording an image sequence. The resulting image sequence can be used for deriving more accurate depth information than with structured IR. Some models of Intel's *RealSense* depth cameras use coded light for depth estimation [53].

Time of flight (ToF)-based systems have been used in radar and other distance measurement applications for more than 30 years [1]. The basic principle behind these systems is to send a signal from the device, receive its reflection from an object and calculate the distance to the object based on the time between sending and reception. ToF sensors are now used in several smartphone models, for depth estimation and tracking subjects while capturing video.

Light detection and ranging (LiDAR) is a variant of ToF-based systems that uses a laser or a grid of lasers for depth map creation. LiDAR sensors are widely used on automobiles, but their recent adoption to Apple's iPad Pro suggests the possibility of them being used in other mobile devices.

Both ToF and LiDAR sensors capture low resolution and sparse depth data due to the limited sensing range and power consumption constraints. This calls for additional image processing where a smooth depth map over all image pixels is required [9,11,77,104].

6 Image analysis

In image analysis, symbolic information regarding the image content is extracted from the image. While such information can be used for a variety of purposes, we herein discuss applications of image analysis on mobile devices, with emphasis on image enhancement.

6.1 Face image analysis

Faces are one of the most common types of content in photographs taken using smartphone cameras. Automatic face detection in digital images has been researched since 1970s, and there is a large amount of published work available [46]. Digital cameras that employ face detection for controlling focus and exposure were commercially available even before smartphones were made. These devices mostly used techniques based on Haar cascade classifiers [100]. Additional features such as facial expression (particularly smile) recognition [61] and age estimation [2] followed, allowing customization of image processing parameters to produce better photographs.

Face landmark detection enabled further localization of different regions of a face, enabling different filtering parameters for skin, facial hair, eyes, lips, etc. A number of techniques, including active shape modeling, Haar cascade classifiers, regression trees and deep neural networks can be used for fast and accurate face landmark detection [56]. Several smartphone makers include face landmark detection software in their devices.

With the advances in machine learning technologies, further detailed face analysis has become possible in smart phones. Face mesh modeling [4,55] and portrait-relighting [34] with good accuracy are now available in high-end smartphones.

6.2 Image classification

In this section, we discuss image classification and object detection. Image classification and object detection are techniques that may focus not only on local features of a specific region of an image but also on global features, and in some cases, context. In recent years, methods using convolutional neural networks (CNN) or transformers have become the mainstream methods to capture both local and global features of an image and utilize them for recognition.

Image classification is a technology that aims to estimate the category to which the image belongs. It has various applications. Some digital cameras and smartphones automatically classify images in albums or change shooting settings based on the result of image classification. Some advanced driver assistance systems (ADASs) recognize road signs in images captured by dash-cams via image classification techniques.

A typical benchmark dataset for image classification is ImageNet [17], where neural-network-based methods continue to have the highest accuracy since AlexNet [60] in 2012. Continuously, researchers proposed new state-of-the-art models with the new network structures following AlexNet, which proposed ReLU and Dropout. GoogLeNet [95] proposed the inception module, ResNet [40] showed the effectiveness of the skip connections, SENet [49] proposed the squeeze and excitation module, and EfficientNet [96] aims to optimize the network depth, width and resolution of the input images. Recently, structures based on

attention, such as Vision Transformer [21], have also been proposed. Besides, normalization of features, learning methods and new data augmentations also contribute to improving the accuracy, for example, Batch Normalization [54], Layer Normalization [8], Dropout [94], DropConnect [101], Cutout [18], Random Erasing [108], mixup [106], and others.

In the context of neural networks, image classification is essential not only for its use but also as a backbone for network architectures used in other tasks. For mobile and edge applications, the efficiency of performance versus computational complexity is important. MobileNet series [47,48,90] are human-designed models with low computational complexity, while NASNet [109] and EfficientNet [96] use the model architecture search techniques to obtain good accuracy versus computational complexity properties.

Also, techniques to reduce the number of parameters of the trained models, such as pruning [36,37], quantization [50, 85], tensor decomposition [58] and distillation [45], make models more suitable for mobile and edge inference. For practical use, it is necessary to adjust the kernel size, the size of intermediate features and other parameters to fit specific hardware.

6.3 Object detection

Neural networks have also become the mainstream technology for object detection tasks in recent years. Object detection, as the basis of image understanding and computer vision, supports image-based applications in various fields such as automatic driving, robot vision and security. Typical benchmark datasets include PASCAL VOC [25,26] and MS COCO [64], while Cityscapes [14] is a well-known dataset for automotive applications.

In the past, handcrafted image features [histogram of gradients (HoG) [15], SIFT [69] and others] and representations such as the deformable part model (DPM) [28] have been successful in the field of object detection. However, in recent years, detection techniques using neural networks have outperformed them in terms of accuracy. Neural network architectures for object detection have been continuously evolving. Conventional R-CNN [31] architecture performs classification on cropped object proposals; Faster R-CNN [89] proposed end-to-end trainable 2-stage architectures, whereas SSD [66] and YOLO [12,86,87] proposed anchor-based one-stage architectures. Cornernet [63] and Centernet [22] estimate object regions by heat map and do not use anchors. DETR [13] uses the transformer structure.

In some applications, there are situations where it is required to produce individual masks of object areas in addition to detecting them as rectangles. Instance segmentation covers such tasks, and Mask R-CNN [42] and SOLOv2 [102] are well-known examples.

For mobile and edge devices, the challenge is to reduce the computational complexity of the neural network architecture. In many cases, one-stage models such as SSD and Centernet are advantageous. However, there are cases where relatively lightweight inference can be made using two-stage methods than one-stage methods. Therefore, when using object detection in edge devices, it is necessary to start by selecting the backbone and the detection architecture depending on the application.

6.4 Single image depth estimation

Using machine learning-based methods, it is possible to analyze a single image and construct a depth map for the corresponding scene. Distance information and patterns learned from the training data facilitate this, despite the complete absence of depth information in the input image. In addition to the applications of its multi-camera counterpart, single image depth estimation provides a more convenient solution for in-vehicle vision systems and drones. Using a single camera eliminates the need for calibration and may require less processing power if an efficient depth estimation algorithm is used.

Many methods for directly estimating depth values using deep neural networks (DNNs) have been proposed [23], [62], and their accuracy has improved. Depth estimation is an ill-posed problem because the depth map for a given input image is not unique. Therefore, in order to obtain a reasonable solution, it is necessary to select appropriate training data for the intended application. For research use, the KITTI Dataset [29] and NYU Depth Dataset v2 [92] are commonly used. Researchers in the industry often create their own datasets, to match the application and the desired accuracy.

In general, it is costly to obtain sufficient depth data. Therefore, in recent years, various methods that are capable of training without ground-truth depth data have emerged. For example, a method that indirectly obtains depth values by estimating disparity map [32] and a self-supervised learning method by reconstructing images from time series images have been proposed [71].

6.5 Image segmentation

The task of assigning a semantic label to each pixel of an image is called semantic segmentation. The result of image segmentation leads to a more detailed understanding of image content. On a mobile device, such information can be used to process each pixel differently according to its semantic label. On a smartphone, accurate image segmentation can be used for fine-level photo-enhancements.

Various methods using CNNs for semantic segmentation have been proposed. For example, FCN [68] in which the entire network is composed of convolution layers, is well

known. Recently, a method that combines multi-scale segmentation structure with an attention mechanism [97] has achieved high accuracy. Another task that is similar to semantic segmentation is to assign different labels to multiple objects of the same class in an image, which is called *Instance Segmentation*. Mask R-CNN [41] which is an extension of Faster R-CNN [89] is well known as a representative method for object detection. Another noteworthy technology is the transformer [67]. This was originally invented in the field of natural language processing, but has recently been applied to a variety of vision tasks too. More recently, the task of adding "Stuff" labels such as "sky" and "road" to Instance Segmentation has been attracting attention. This is called panoptic segmentation [59] and is currently a growing research topic.

Morpho Inc. recently published the results of a joint research project with Denso Corporation that combines depth estimation and semantic segmentation [78] for use in ADAS applications. The depth map and the results of scene segments can be combined to create a 2.5-dimensional scene. Figure 7a, b shows such a scene from two different viewpoints. We expect that such multitask approaches that simultaneously infer different types of information will be actively researched by various companies in future.

6.6 Remarks

With the recent advances in DNN technology, the state-of-the-art accuracy for most image analysis tasks now comes from machine learning-based methods. The presence of GPUs and neural processing hardware in mobile devices has allowed mobile devices to deploy these methods within them. While the large size of some machine learning models makes them less suitable for mobile devices, researchers have identified this as a problem and are working on possible solutions such as creating lightweight models and compressing existing models.

7 Recent trends

Mobile devices are a relatively new category of computing hardware and have been evolving fast. In this section, we will have a quick look at some of the recent trends in mobile devices and mobile image processing.

7.1 Sensing and capture

Ability to take better photographs can make subsequent processing tasks much easier; while mobile devices are not going to get any larger, camera and camera module manufacturers have been working on acquiring images with better quality under the given constraints.

Dual pixel auto-focus (DPAF) is one approach for improving images and video captured by mobile devices. On DPAF image sensors, each pixel has two photodiodes that can operate either separately or together. Each diode has a separate lens over it. When light goes through the lenses and hits the diodes, the processor analyzes each diode's signal for focus, and once focus is achieved, the signals are combined to record the image. DPAF can greatly enhance the quality of video, since it facilitates quick and accurate focus without the need for adjusting the main camera lens for focus. It also makes following and keeping focus on moving subjects much easier and more accurate. Originally developed as DSLR cameras the main target, dual pixel AF sensors are now available in high-end smartphones like Samsung Galaxy S7.

7.2 Semantic filtering

Another trend in smartphone image processing is to filter different parts of an image using different parameters, to produce a photograph that is aesthetically more pleasing than the original. For example, a selfie can be enhanced by smoothing the skin on faces to diffuse wrinkles, freckles, etc., while still keeping the eyes, hair and facial hair sharp. The background can be blurred to create a visually pleasing portrait. Color correction on skin regions, to achieve better skin tones, is also possible. Such processing that depends on the semantics of the scene can be collectively called "semantic filtering." Semantic-based filtering can be performed by segmenting the image to identify regions corresponding to different objects, and automatically selecting the most appropriate filtering technique to enhance each region [75]. Most high-end smartphones apply some sort of semantic filtering techniques to refine photographs [52].

7.3 Recent trends due to the COVID-19 pandemic

The COVID-19 pandemic (ongoing at the time of writing) resulted in the emergence of new market needs that could be fulfilled using image processing techniques. The increase of remote work resulted in extensive use of video conferencing software. This posed several challenges to the users. Exposing the home environment to business meetings became a considerable burden to the user, since the user has to either clear up the camera's field of view or locate in a way that his/her privacy is not offended. The offset between the camera position and the center of the computer display resulted in perceived lack of eye contact.

The industry responded fairly quickly, using existing technology. Background replacement is now available in major video conferencing software such as Zoom and Google Meet. Gaze correction to emulate better eye contact is provided by Apple iPhone models with depth-sensing capability, and

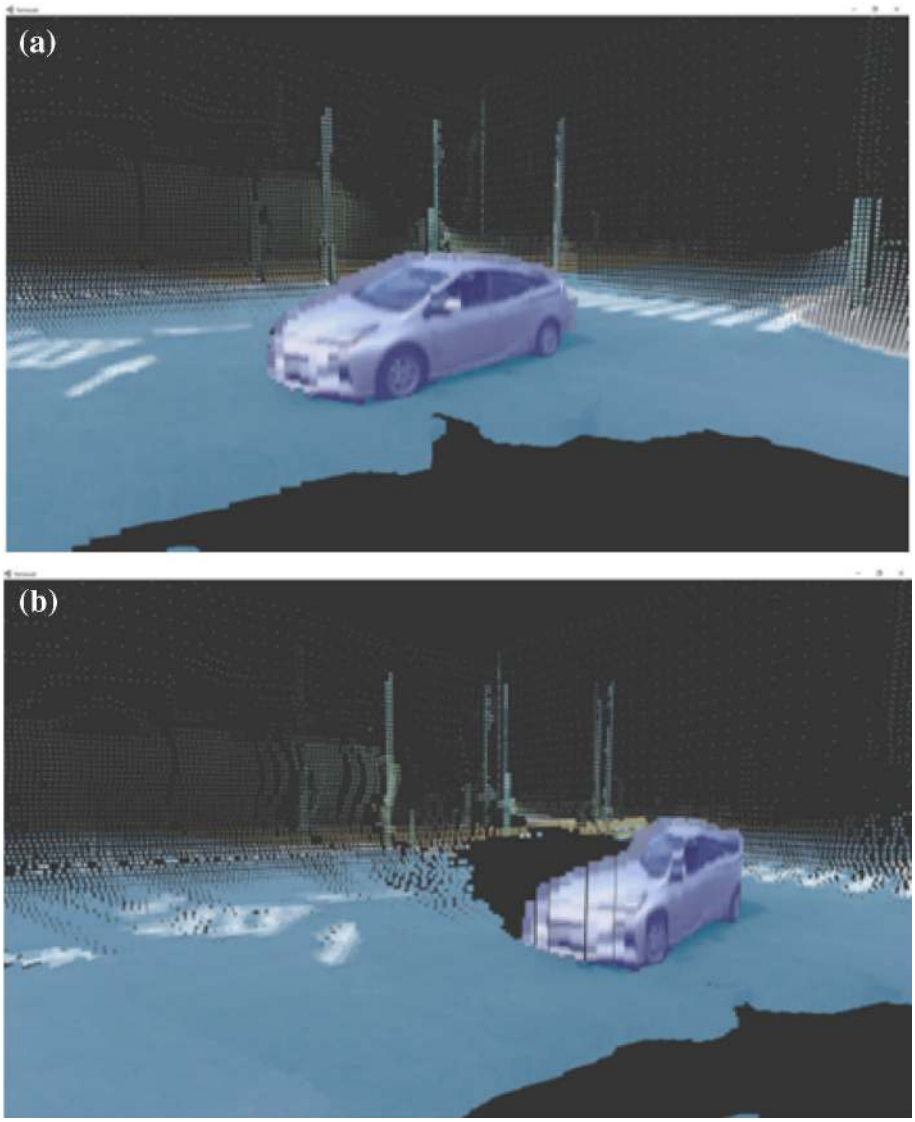

Fig. 7 A 2.5D scene created by combining depth and segmentation information. **a** While the result looks like an image from this viewpoint, it is made of voxels and therefore contains depth information as well. **b** When seen from a different viewpoint, it can be seen that the result contains depth data and occluded regions of the scene are missing in the final result

Microsoft Surface Pro X [6,72]. Zoom also provides semantic filtering functionality to make faces look better.

8 Concluding remarks

With increasing versatility of mobile devices equipped with cameras, the types of image processing and analysis tasks that they carry out have also increased. We presented a somewhat broad survey of such tasks, while highlighting how the constraints and requirements differ from their non-mobile counterparts. Due to the constraints in camera optics and sensors, and also the power consumption, the algorithms used have to be robust, yet efficient. They also require high accuracy and speed and good output quality when it comes to smartphones.

In order to achieve these objectives, researchers and developers in the industry use a combination of algorithms, heuristics, refinements and knowhow. Some of these are not always novel, and some others are unpublished. Therefore, it might be difficult for readers to find a lot of details on image processing pipelines for mobile devices. We hope that this survey provided the reader with some insights on potential challenges in solving mobile computer vision problems and possible approaches to solve them.

Acknowledgements We would like to express our gratitude to the late Dr. Tosiyasu L. Kunii, former Chief Technical Advisor at Morpho Inc., for guiding us in the pursuit of bringing the advances in image processing and computer vision to cameras and mobile phones in the hands of people around the world.

Declarations

Conflict of interest The authors did not receive support from any organization for the submitted work. The authors have no conflicts of interest to declare that are relevant to the content of this article. Product names and organization names included in the publication are included solely for the purpose of surveying.

References

1. Adams, M.D.: Coaxial range measurement current trends for mobile robotic applications (2002)
2. Angulu, R., Tapamo, J.R., Adewumi, A.: Age estimation via face images: a survey. EURASIP J. Image Video Process. (2018). https://doi.org/10.1186/s13640-018-0278-6
3. Apple Inc.: Core ML - Apple Developer. https://developer.apple.com/documentation/coreml (2016). Accessed 19 Apr 2021
4. Apple Inc.: Tracking and visualizing faces. https://developer.apple.com/documentation/arkit/content_anchors/tracking_and_visualizing_faces (2019). Accessed 20 Apr 2021
5. Apple Inc.: Apple unveils all-new iPad Air with A14 Bionic, Apple's most advanced chip. https://www.apple.com/newsroom/2020/09/apple-unveils-all-new-ipad-air-with-a14-bionic-apples-most-advanced-chip/ (2020). Accessed 16 Apr 2021
6. Apple Insider: Hands on with Apple's FaceTime Attention Correction feature in iOS 13. https://appleinsider.com/articles/19/07/03/hands-on-with-apples-facetime-attention-correction-feature-in-ios-13 (2019). Accessed 21 Apr 2021
7. ARM Inc.: Arm Compute Library. https://developer.arm.com/ip-products/processors/machine-learning/compute-library (2017). Accessed 19 Apr 2021
8. Ba, L.J., Kiros, J.R., Hinton, G.E.: Layer normalization. CoRR arXiv:1607.06450 (2016)
9. Bartczak, B., Koch, R.: Dense depth maps from low resolution time-of-flight depth and high resolution color views. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) Advances in Visual Computing, pp. 228–239. Springer, Berlin (2009)
10. Bebeselea-Sterp, E., Brad, R., Brad, R.: A comparative study of stereovision algorithms. Int. J. Adv. Comput. Sci. Appl. (2017). https://doi.org/10.14569/IJACSA.2017.081144
11. Bleiweiss, A., Werman, M.: Robust head pose estimation by fusing time-of-flight depth and color. In: 2010 IEEE International Workshop on Multimedia Signal Processing, MMSP 2010, Saint Malo, France, October 4–6, 2010, pp. 116–121. IEEE (2010). https://doi.org/10.1109/MMSP.2010.5662004
12. Bochkovskiy, A., Wang, C., Liao, H.M.: Yolov4: Optimal speed and accuracy of object detection. CoRR. arXiv:2004.10934 (2020)
13. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229. Springer, Berlin (2020)
14. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 886–893 (2005). https://doi.org/10.1109/CVPR.2005.177
16. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: Owen, G.S., Whitted, T., Mones-Hattal, B. (eds.) Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, Los Angeles, CA, USA, August 3–8, 1997, pp. 369–378. ACM (1997). https://doi.org/10.1145/258734.258884
17. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.-F.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
18. Devries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. CoRR. arXiv:1708.04552 (2017)
19. Dew, P., Fuchs, H., Kunii, T., Wozny, M.: Parallel processing for computer vision and display (panel session). In: Proc. ACM Siggraph Computer Graphics, vol. 22, p. 346 (1988). https://doi.org/10.1145/54852.378545
20. DigitalTrends: A complete history of the camera phone. https://www.digitaltrends.com/mobile/camera-phone-history/ (2013). Accessed 16 Apr 2021
21. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: transformers for image recognition at scale (2020)
22. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6568–6577 (2019). https://doi.org/10.1109/ICCV.2019.00667
23. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network (2014)
24. Endo, K.: Personal, portable, pedestrian: mobile phones in Japanese life edited by Mizuko Ito, Daisuke Okabe and Misa Matsuda. Int. J. Jpn. Sociol. **16**(1), 115–116 (2007). https://doi.org/10.1111/j.1475-6781.2007.00103.x
25. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
26. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2015)
27. Fairchild, M.: The HDR photographic survey. In: Color Imaging Conference (2007)
28. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008). https://doi.org/10.1109/CVPR.2008.4587597
29. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013). https://doi.org/10.1177/0278364913491297
30. Georgescu, M.D.: Evolution of mobile processors. In: 2003 IEEE Pacific Rim Conference on Communications Computers and Signal Processing (PACRIM 2003) (Cat. No.03CH37490), vol. 2, pp. 638–641 (2003). https://doi.org/10.1109/PACRIM.2003.1235862
31. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014). https://doi.org/10.1109/CVPR.2014.81
32. Godard, C., Aodha, O.M., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017, pp. 6602–6611. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.699

33. Gonzalez, R., Woods, R.: Digital Image Processing. Pearson (2018). https://books.google.co.jp/books?id=0F05vgAACAAJ
34. Google Inc.: Portrait Light: Enhancing Portrait Lighting with Machine Learning . https://ai.googleblog.com/2020/12/portrait-light-enhancing-portrait.html (2020). Accessed 20 Apr 2021
35. Goyal, B., Dogra, A., Agrawal, S., Sohi, B., Sharma, A.: Image denoising review: from classical to state-of-the-art approaches. Inf. Fusion **55**, 220–244 (2020)
36. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. In: Proc. ICLR 2016 (2016). http://arxiv.org/abs/1510.00149
37. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS'15, vol. 1, pp. 1135–1143. MIT Press, Cambridge (2015)
38. Hasinoff, S.W., Sharlet, D., Geiss, R., Adams, A., Barron, J.T., Kainz, F., Chen, J., Levoy, M.: Burst photography for high dynamic range and low-light imaging on mobile cameras. ACM Trans. Graph. (2016). https://doi.org/10.1145/2980179.2980254
39. Haskell, B.: Portable Electronics Product Design and Development. Electronic engineering, McGraw-Hill Education, McGraw-Hill professional engineering (2004)
40. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
41. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: 2017 IEEE International Conference on Computer Vision (ICCV), Los Alamitos, CA, USA, pp. 2980–2988. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.322
42. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988 (2017). https://doi.org/10.1109/ICCV.2017.322
43. Hemayed, E.: A survey of camera self-calibration. Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance **2003**, 351–357 (2003). https://doi.org/10.1109/AVSS.2003.1217942
44. Higher Intellect Vintage Wiki: Connectix QuickCam. https://wiki.preterhuman.net/Connectix_QuickCam (2020). Accessed 16 Apr 2021
45. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Deep Learning and Representation Learning Workshop (2015). arXiv:1503.02531
46. Hjelmås, E., Low, B.K.: Face detection: a survey. Comput. Vis. Image Underst. **83**(3), 236–274 (2001)
47. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications. CoRR. arXiv:1704.04861 (2017)
48. Howard, A., Pang, R., Adam, H., Le, Q.V., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., Chu, G., Vasudevan, V., Zhu, Y.: Searching for mobilenetv3. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27–November 2, 2019, pp. 1314–1324. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00140
49. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018). https://doi.org/10.1109/CVPR.2018.00745
50. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: training neural networks with low precision weights and activations. J. Mach. Learn. Res. **18**(187), 1–30 (2018). http://jmlr.org/papers/v18/16-456.html
51. Inc., G.: TensorFlow Lite—ML for Mobile and Edge Devices. https://www.tensorflow.org/lite/ (2018). Accessed 19 Apr 2021
52. Insider Magazine: How popular smartphones make your skin look 'whiter' in selfies. https://www.insider.com/samsung-huawei-smartphone-beauty-filters-whiten-your-skin-2017-8 (2017). Accessed 21 Apr 2021
53. Intel Inc.: Beginner's guide to depth. https://www.intelrealsense.com/beginners-guide-to-depth (2019). Accessed 20 Apr 2021
54. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 448–456. PMLR, Lille (2015)
55. Kartynnik, Y., Ablavatski, A., Grishchenko, I., Grundmann, M.: Real-time facial surface geometry from monocular video on mobile GPUs (2019)
56. Khabarlak, K., Koriashkina, L.: Fast facial landmark detection and applications: a survey (2021)
57. Khairy, M., Wassal, A.G., Zahran, M.: A survey of architectural approaches for improving GPGPU performance, programmability and heterogeneity. J. Parallel Distrib. Comput. **127**, 65–88 (2019). https://doi.org/10.1016/j.jpdc.2018.11.012
58. Kim, Y., Park, E., Yoo, S., Choi, T., Yang, L., Shin, D.: Compression of deep convolutional neural networks for fast and low power mobile applications. In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings (2016). arxiv:1511.06530
59. Kirillov, A., He, K., Girshick, R.B., Rother, C., Dollár, P.: Panoptic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019. pp. 9404–9413. Computer Vision Foundation/IEEE (2019). https://doi.org/10.1109/CVPR.2019.00963
60. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 25. Curran Associates, Inc. (2012). https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
61. Kumari, J., Rajesh, R., Pooja, K.: Facial expression recognition: a survey. Procedia Comput. Sci. **58**, 486–491 (2015). https://doi.org/10.1016/j.procs.2015.08.011. Second International Symposium on Computer Vision and the Internet (VisionNet'15)
62. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks (2016)
63. Law, H., Deng, J.: Cornernet: detecting objects as paired keypoints. Int. J. Comput. Vis. **128**(3), 642–656 (2020). https://doi.org/10.1007/s11263-019-01204-1
64. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision–ECCV 2014, pp. 740–755. Springer, Cham (2014)
65. Lipkin, L.: Picture processing by computer. Azriel rosenfeld. Science **169**(3941), 166–167 (1970). https://doi.org/10.1126/science.169.3941.166
66. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision–ECCV 2016, pp. 21–37. Springer, Cham (2016)
67. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: hierarchical vision transformer using shifted windows (2021)
68. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Com-

puter Vision and Pattern Recognition (CVPR). IEEE Computer Society, Los Alamitos, CA, USA, pp. 3431–3440 (2015). https://doi.org/10.1109/CVPR.2015.7298965
69. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 91–110, (2004)
70. Lumsdaine, A., Georgiev, T.: The focused plenoptic camera. In: 2009 IEEE International Conference on Computational Photography (ICCP), pp. 1–8 (2009)
71. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5667–5675 (2018). https://doi.org/10.1109/CVPR.2018.00594
72. Microsoft Windows Blog: Make a more personal connection with Eye Contact, now generally available. https://blogs.windows.com/devices/2020/08/20/make-a-more-personal-connection-with-eye-contact-now-generally-available/ (2021). Accessed 21 Apr 2021
73. Milani, S., Calvagno, G.: Correction and interpolation of depth maps from structured light infrared sensors. Signal Process. Image Commun. **41**, 28–39 (2016)
74. Morpho Inc.: Featuring Technology: SoftNeuro. https://www.morphoinc.com/en/featuringtechnology/softneuro (2018). Accessed 19 Apr 2021
75. Morpho Inc.: Featuring Technology: Semantic Filtering. https://www.morphoinc.com/en/featuringtechnology/semanticfiltering (2020). Accessed 21 Apr 2021
76. Morpho Inc.: Morpho: Technology. https://www.morphoinc.com/en/technology#tab-02 (2020). Accessed 18 May 2021
77. Nair, R., Ruhl, K., Lenzen, F., Meister, S., Schäfer, H., Garbe, C.S., Eisemann, M., Magnor, M., Kondermann, D.: A Survey on Time-of-Flight Stereo Fusion, pp. 105–127. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-44964-2_6
78. Narioka, K., Nishimura, H., Itamochi, T., Inomata, T.: Understanding 3d semantic structure around the vehicle with monocular cameras. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 132–137 (2018). https://doi.org/10.1109/IVS.2018.8500397
79. Neumann, J.V.: Introduction to "The First Draft Report on the EDVAC". http://qss.stanford.edu/~godfrey/vonNeumann/vnedvac.pdf (1945). Accessed 16 Apr 2021
80. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. In: Stanford Tech Report CTSR 2005-02 (2005)
81. Park, J., Lee, C., Kim, C.S.: Deep learning approach to video frame rate up-conversion using bilateral motion estimation. In: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1970–1975 (2019). https://doi.org/10.1109/APSIPAASC47483.2019.9023270
82. Poulose, M.: Literature survey on image deblurring techniques. Int. J. Comput. Appl. Technol. Res. **2**, 286–288 (2013)
83. Qualcomm Inc.: Qualcomm Neural Processing SDK. https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk (2016). Accessed 19 Apr 2021
84. Qualcomm Inc.: Mobile AI—On-Device AI—Qualcomm. https://www.qualcomm.com/products/smartphones/mobile-ai (2020). Accessed 16-Apr 2021
85. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: imagenet classification using binary convolutional neural networks. In: European Conference on Computer Vision (2016)
86. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016). https://doi.org/10.1109/CVPR.2016.91
87. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. Preprint at arXiv:1804.02767 (2018)
88. Reeja, S.R., Kavya, N.P.: Noise reduction in video sequences: the state of art and the technique for motion detection. Int. J. Comput. Appl. **58**(8), 31–36 (2012)
89. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017). https://doi.org/10.1109/TPAMI.2016.2577031
90. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018). https://doi.org/10.1109/CVPR.2018.00474
91. Shinagawa, Y., Kunii, T.: Unconstrained automatic image matching using multiresolutional critical-point filters. IEEE Trans. Pattern Anal. Mach. Intell. **20**(9), 994–1010 (1998). https://doi.org/10.1109/34.713364
92. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision–ECCV 2012, pp. 746–760. Springer, Berlin (2012)
93. Singh, M., Kumar, M.: Evolution of processor architecture in mobile phones. Int. J. Comput. Appl. (2014). https://doi.org/10.5120/15564-4339
94. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(56), 1929–1958 (2014). http://jmlr.org/papers/v15/srivastava14a.html
95. Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2015). https://doi.org/10.1109/CVPR.2015.7298594
96. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114. PMLR (2019)
97. Tao, A., Sapra, K., Catanzaro, B.: Hierarchical multi-scale attention for semantic segmentation. arXiv:2005.10821 (2020)
98. Tico, M., Vehvilainen, M.: Robust image fusion for image stabilization. In: 1988 International Conference on Acoustics, Speech, and Signal Processing, 1988. ICASSP-88, vol. 1, pp. I–565 (2007). https://doi.org/10.1109/ICASSP.2007.365970
99. van Ouwerkerk, J.: Image super-resolution survey. Image Vis. Comput. **24**(10), 1039–1052 (2006)
100. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I–I (2001). https://doi.org/10.1109/CVPR.2001.990517
101. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 28, pp. 1058–1066. PMLR, Atlanta (2013)
102. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: Solov2: dynamic, faster and stronger. CoRR. arXiv:2003.10152 (2020)
103. Wikipedia: Dulmont Magnum. https://en.wikipedia.org/wiki/Dulmont_Magnum (2015). Accessed 16 Apr 2021
104. Xu, Y., Zhu, X., Shi, J., Zhang, G., Bao, H., Li, H.: Depth completion from sparse lidar data with depth-normal constraints. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
105. Zhang,Y., Zhou, L., Liu, Z., Shang, Y.: A flexible online camera calibration using line segments. J. Sens. **2016**, Article ID 2802343, 16 pages (2016). https://doi.org/10.1155/2016/2802343

106. Zhang, H., Cisse, M., Dauphin, Y., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: Proc. ICLR 2018 (2018)
107. Zhang, L., Xu, Q.K., Huang, H.: A global approach to fast video stabilization. IEEE Trans. Circuits Syst. Video Technol. **27**(2), 225–235 (2017). https://doi.org/10.1109/TCSVT.2015.2501941
108. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. Proc. AAAI Conf. Artif. Intell. **34**(07), 13001–13008 (2020)
109. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. Preprint at arXiv:1611.01578 (2017)

Chamin Morikawa received the PhD degree in Frontier Informatics from the University of Tokyo, Japan, in 2007, where he also served as an assistant professor in Information Sciences. He is currently a senior research engineer at Morpho Inc. He specializes in multi-camera mobile image processing and face image analysis. Chamin has been a technical program committee member of ACM SIGMAP since 2008 and is also a member of the "Building Trustworthy AI" working group of Mozilla Foundation.

Michihiro Kobayashi received the MS degree in engineering from Yokohama National University, Japan, in 2007, and the PhD degree in Information Science and Technology from the University of Tokyo, Japan, in 2010. He worked at the Institute of Industrial Science, the University of Tokyo, Japan, in 2010, as a project researcher in computer vision. He joined Morpho, Inc. in 2011, where he is currently managing development of the products for mobile devices. His research interests include image enhancement such as noise reduction, HDR synthesis and super-resolution. He is also active on advanced computational photography running on modern mobile platforms.

Masaki Satoh is a senior research engineer at Morpho, inc. His primary research field is video processing for embedded devices, especially smartphones. Since he joined Morpho Inc. in 2011, he has developed a number of commercially proven video algorithms, including video stabilization, video noise reduction, and frame rate conversion. He completed his PhD in the field of Physics and Astronomy at Kyoto University, Japan, in 2011.

Yasuhiro Kuroda received his MS and PhD in Physics from the University of Tokyo, Japan, in 2010 and 2015, respectively. In 2014, he started his career at Morpho Inc. and is currently working as a senior researcher developing image recognition technologies especially related to automobiles, technologies related to efficient and fast training of deep learning, and inference engines on edge devices.

Teppei Inomata received his MS and PhD degrees in Engineering from Keio University, Japan, in 2004 and 2010, respectively. In 2010, he joined Morpho Inc., where he is currently working on research and development of image recognition algorithms using machine learning. Currently, he is mainly interested in semi-supervised learning methods in situations with insufficient training data and data augmentation methods.

Hitoshi Matsuo received the MS degree in Mathematical Informatics from the University of Tokyo in 2017. Since joining Morpho Inc. in 2017, he has been working on research and development of machine learning and image processing. He is particularly interested in optimizations and algorithms that enable real-time image processing on real devices.

Takeshi Miura received his MS and PhD in information science from the Tohoku University in 2002 and 2005, respectively. From 2005, he worked at IT institute, Kanazawa institute of technology as a researcher, specializing in image processing and image recognition. After joining Morpho Inc. in 2008, Takeshi has conducted research and development on several themes including multi-frame noise reduction, HDR imaging, super-resolution, and panorama synthesis. He currently heads Morpho's CTO Office, which is its research and development division.

Masaki Hilaga received his MS and PhD degrees in Information Science from the University of Tokyo in 1999 and 2002, respectively. In 2004, he founded Morpho Inc. to develop and commercialize cutting-edge imaging technologies and has led its R&D as CEO and CTO. He also pioneered the productization of image stabilization and panoramic photography for mobile phones.