Virtual Camera Image Processing

Horacio Hernan Moraldo

Google Research

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Moraldo, Horacio Hernan, "Virtual Camera Image Processing", Technical Disclosure Commons, (March 30, 2020)
https://www.tdcommons.org/dpubs_series/3072

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.
Virtual Camera Image Processing

Abstract:

This publication describes techniques for implementing virtual camera image transformations. It can be difficult to properly position cameras to capture certain types of photographs, particularly pictures intended to include the photographer (often referred to as “selfies”). Even with the use of assistive devices, such as “selfie sticks,” these types of photographs often exhibit a characteristic, unnatural look due to suboptimal camera orientation and/or focal length (since the relative distance from the lens and the field of view of that lens affect the look of the subjects within the image). Disclosed in this publication is a virtual camera image processor capable of adjusting, inter alia, camera angle, position, and/or focal length of images during post-processing (e.g., as an after effect) using a machine-learned image generator.

Keywords:

Photography, computational photography, virtual camera, image processing, distortion, machine learning, artificial neural network, generative model, discriminative model, Generative Adversarial Network, selfie.

Background:

One of the most popular uses of cameras, particularly smartphone cameras, is taking “selfie” pictures intended to capture the photographer’s face. Unfortunately, it can be difficult, and often impossible, to properly position and/or orient the camera when taking these types of pictures (e.g., while extending one’s arm as is often required to capture a selfie). As a result, selfies often have an awkward, unnatural appearance (“selfie look”); and, since the relative distance of subjects from the camera lens, and the field of view of that lens affects the look of the subjects, even when assistive devices such as “selfie sticks” are used, the resulting photographs often still
suffer from distortion and poor composition. Correctly positioning the camera to capture selfies that attempt to include multiple people is even more difficult, which can exacerbate these problems. Therefore, it is desirable to provide an image processor capable of implementing virtual camera transforms during post-processing (e.g., manipulate camera angle, distance, focal length, etc.).

**Description:**

This publication describes techniques for adjusting virtual camera parameters of an image, such as camera angle, position, focal length, and/or the like. The disclosed techniques can enable users to capture selfie-type images from comfortable positions, and then correct “selfie look” problems (e.g., distortion and/or poor composition) during post-processing.

As illustrated in FIG. 1, an electronic device 100, such as a smartphone, can include and/or be communicatively coupled to a camera 102. An image processor 110 can operate on the electronic device 100 (e.g., operate on a processor, a graphics processing unit (GPU), and/or other processing resources of the electronic device 100). The image processor 110 is adapted to process input images 104, such as a selfie pictures captured by the camera 102. Alternatively, input images 104 may be retrieved from memory storage of the electronic device 100, accessed through a network (e.g., the Internet), and/or the like. The image processor 110 can include a graphical user interface (interface 112) for manipulating input images 104. The interface 112 can include controls 114 for specifying virtual camera (VC) transforms to perform on an input image 104 (VC parameters 115), and a VC engine 116 to produce an output image (a transformed output image 106) that has been transformed as if captured by a camera having the specified VC parameters 115. The VC parameters 115 may include a set of values representing respective VC settings, such as VC position, VC angle, VC focal length, and so on (e.g., a set of floating-point values).
As illustrated in FIG. 2A, the interface 112 of the image processor 110 can be presented on a display of the electronic device 100 (e.g., on a touch screen or the like). The interface 112 can initially display an input image 104, such as a selfie captured by the camera 102 of the electronic device 100. The controls 114 of the interface 112 can provide for specifying and/or adjusting VC parameters 115; the controls 114 can include but are not limited to, a VC position control 220 for specifying and/or adjusting the position of the VC, a VC angle control 222 for specifying and/or adjusting the angle of the VC, a VC focal length control 224 for specifying and/or adjusting the focal length of the VC, and/or the like.
In some aspects, the interface 112 can suggest recommended VC parameters 115 to achieve a particular look for the transformed output image 106 (e.g., recommend a VC angle to achieve a “professional” look suitable for a resumé or the like). Selecting recommended VC parameters 115 may cause the interface 112 to automatically set the controls 114 to reflect the recommended VC parameters 115 and display the resulting transformed output image 106 (and/or preview thereof). The image processor 110 can automatically implement selected VC transforms (and save a corresponding transformed output image 106) in response to capturing selfie input images 104 using the camera 102.

As further illustrated in FIG. 2B, a preview/save control 226 of the interface 112 can provide for previewing and/or saving transformed output images 106 produced by the VC engine 116; the VC engine 116 can generate a transformed output image 106 from the input image 104 displayed within the interface 112 using VC parameters 115 specified through the controls 114 of the interface 112.
As illustrated in FIG. 3, the VC engine 116 can include a machine-learned (ML) image generator 316. The ML image generator 316 can produce transformed output images 106 in response to input images 104 and associated VC parameters 115. The ML image generator 316 can include an artificial neural network (ANN) having an input layer that receives an input image 104 together with VC parameters 115 (e.g., a set of floating-point values specified through controls 114 of the interface 112), one or more hidden layers, and an output layer at which the resulting transformed output image 106 is output.
In some aspects, the ML image generator 316 can produce preview images in response to user interaction with the controls 114 of the interface 112. The ML image generator 316 can generate lower-resolution preview images for display within the interface 112 (to minimize latency) and produce full resolution transformed output images 106 when required (e.g., in response to a save command or the like).

In some aspects, the ML image generator 316 is trained with a large image corpus, then extended by a corpus of images having three-dimensional (3D) depth information. As illustrated in FIG. 4, the ML image generator 316 can include an ANN (ML depth model 414) trained to estimate 3D depth information 415 for input images 104, and a transform engine 416 to implement classical image deformation and/or translation operations on the input images 104 based on the estimated 3D depth information 415 and VC parameters 115 (e.g., produce transformed output images 106).
In other aspects, the ML image generator 316 can implement image transformations using an end-to-end network (transform network 500), as illustrated in FIG. 5. The transform network 500 can be trained using pre-processed images and post-processed, transformed images (e.g., trained to estimate transformed output images 106 in response to pre-preprocessed input images 104). The transform network 500 can include a multi-task deep neural network trained to implement VC transforms (and/or combinations of VC transforms) corresponding to the controls 114 of the interface 112 (e.g., VC position transforms, VC angle transforms, VC focal length transforms, and so on).
FIG. 5 ML Image Generator Including an End-to-End Transform Network

In some aspects, the transform network 500 includes an encoder 502 trained to produce transform-invariant embeddings of input images 104 (input features 503), a transform task 504 to implement transform(s) corresponding to the VC parameters 115 on the input features 503 (produce transformed features 505), and a generator 506 to reconstruct a transformed output image 106 from the transformed features 505. The generator 506 can be trained using a Generative Adversarial Network (GAN) or the like.

References:


