

Overview of Simulation of Video-Camera Effects for Robotic Systems in R3-COP

Michal Kučíš, Pavel Zemčík, Oliver Zendel, Wolfgang Herzner

Brno University of Technology
{ikucis, zemcik}@fit.vutbr.cz

AIT Austrian Institute of Technology GmbH
{oliver.zendel, wolfgang.herzner}@ait.ac.at

Abstract. This paper outlines the proposed process of video-camera simulation used in R3-COP for generating realistic test data for robotic perception. The outlined process enables lens effects simulation, exposure adaptation, and camera motion. This paper, besides description of important stages of generating video output, also presents the results of the simulation software and illustrative figures produced by the developed within the project.

Keywords. simulation of video-camera, r3-cop, lens effects, exposure adaptation

1 Introduction

One of the goals of the R3-COP project is the assessment of computer visual (CV) components that are used in robotic systems as inputs for manipulation planning, navigation, and general scene understanding. The developed approach allows the automatic generation of test images for validation by simulating cameras in virtual environments. This is done by creating models of all needed objects and simulating the interaction of light with the object surfaces to finally create a perspective projection of the scene at the virtual camera position.

In computer graphics, this process is called rendering and most rendering solutions simulate an ideal pin-hole camera. Typically they apply no distortions or depth of field effects. The testing of systems with virtual test data should create results similar to the results that the algorithm will deliver in operation. Thus the artificial test data should be realistic (from the algorithms point of view). To increase the realism of generated test data, we developed a dedicated simulation framework for the simulation of essential camera effects.

The effects that are supported by this framework are: lens effects, exposure adaptation, motion blur and sensor noise effects.

The geometry for models is usually defined as a triangular mesh by specifying the individual points (vertices) and the respective triangles (faces) to create a surface

model of the individual object. In addition to the geometry, the texture and material properties of each object are also specified. The material has to describe the surfaces effect on reflected light. There have been many models proposed to describe the material properties, the most used one is the Phong reflection model [7].

In Section 2, we will briefly describe inputs for the simulator. Section 3 describes our approach to simulation of video-camera effects, and this part also outlines the supported effects with description of used algorithm. Section 4 demonstrates achieved results. In Section 5, we draw conclusions.

2 Inputs for Simulation of Camera Effects

A primary goal was to allow the framework to be able to work with a number of rendering engines. The minimum information for all specified effects needed is:

- The camera input image
- Corresponding depth information
- A representation of the surroundings of the virtual camera

To be able to use high dynamic range, the openEXR image format with its floating point representation was chosen to specify the input image using three color channels (RGB). The depth information is needed to calculate the depth of field effect and is also specified using a floating point openEXR image. The surroundings of the virtual camera can be of interest as the inner lens reflections can involve light that is coming from outside of the virtual camera's field of view. These surroundings are specified using a cube map (six images representing six virtual cameras with a 90° field of view oriented at each coordinate axis $\langle -x,+x,-y,+y,-z,+z \rangle$). The cube map itself is again specified using a floating point RGB openEXR image

Altogether, this allows the framework to operate on three images alone which can be easily created by all common rendering engines. This input to the framework represents the least common denominator of the different engines that is meaningful enough to allow the simulation of the specified camera effects.

3 Simulation of Camera Effects

This section contains a general description of the proposed algorithm and a brief outline of the individual effects and their simulation. Main topics of this section are lens features, motion blur, exposure adaptation, and sensor features.

3.1 General description of simulation

The design of the simulator is based on the provided inputs (Section 2) and required modularity of the simulator. The video-camera simulation is designed as a pipeline of effects, where the outputs of the renderer are processed step-by-step by the

effect-blocks. This process is illustrated in Figure 1. The design allows adding effects or rearranging the order of effects to achieve requested output.

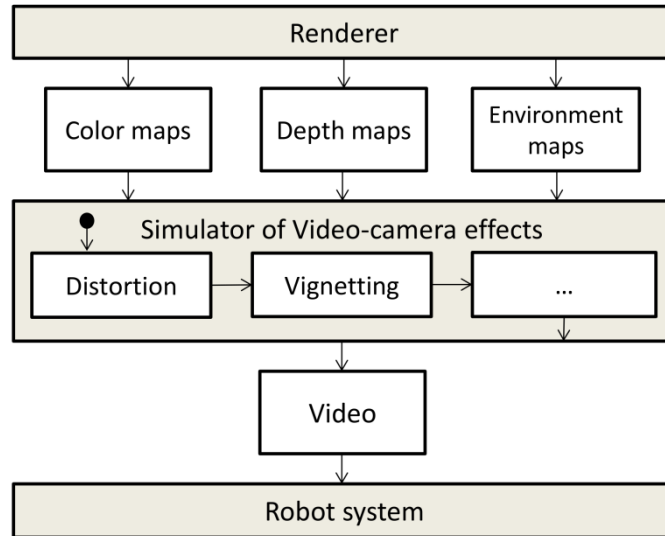


Fig. 1. Scheme of Simulator of Video-camera effects

3.2 Distortion and chromatic aberration

The camera image is a 2D-projection of the real world. In real world, the projection is affected by deviation in which straight lines in a scene do not remain straight in the image. This effect is called geometrical distortion and the most commonly encountered distortion is radially symmetric one, due to the symmetry of camera lens. Chromatic aberration is a type of distortion which is caused by different refractive indices for different light wavelengths. The aberration is mainly observed on edges between a bright and a dark zone where color contours display.

In this implementation, an algorithm based on [1] was chosen. The implementation is straightforward. In the first step, a remapping function is constructed that describes the displacement of output pixels in the input image. In the next steps, the input image is resampled using Lanczos filter. The filter used in this approach is far better from the point of view of robot system than other interpolation filters as this filter is good at preserving the frequency spectrum of the input image. The simulation of chromatic aberration is performed in a similar way. Every color channel is remapped by the different remapping functions.

3.3 Vignetting

In optics, vignetting means reduction of brightness or saturation at the image periphery compared to the center. Vignetting occurs when light located off-axis is

blocked by external objects, such as a lens filter or a lens hood. Vignetting falloff is a natural property of optical lenses and simulation of this phenomenon is straightforward. Vignetting falloff defines a mask that the input image is multiplied pixel by pixel.

3.4 Depth of field effect

A typical 3D scene renderer uses a pinhole camera model to acquire an image. The image is considered to be in perfect focus. An image in the real world application, however, is formed in an optical system where the light from points at different distances in the scene converges at different depths behind the lens. The depth is not necessarily that of the sensor depth. Therefore, the point in the real world appears spread over a region in the image. This region is called circle of confusion and it causes a blurring in the image. The size of the blur is influenced by the distance between the light and the camera. This behavior is called depth of field effect.

The effect is intensively studied and many methods for its simulation exist today [2]. Due to specific requirements, two methods were chosen for implementation. The first one is a fast method based on Reverse-Mapped Z-Buffer Depth of Field [3]. This method is fast but output image suffer by artifacts. The second method is based on principles of diffusion [4]. This method is slower, but artifacts in the output image are reduced. The method can optionally be chosen based on the preferences of the simulator user.

3.5 Exposure adaptation

Exposure influences the amount of light allowed into the optical system during the acquisition process of a single video frame. Therefore the exposure affects brightness of the frame. It is an important task of a camera to control the brightness because the image brightness affects how much information can be obtained from the image/video. The adaptation is usually not instantaneous and reaching optimal exposure can take some time.



Fig. 2. Same image with different exposure

In our simulation of the exposure adaptation, we compute the optimal exposure for each frame. The optimal exposure is defined by the brightness of the frame. The user specifies the initial exposure. During the simulation, this exposure is modified for

each frame so that the error between actual exposure and the optimal exposure is minimized. The computed exposure is used to adapt the brightness of the input HDR image. The adaptation is performed via a PID controller (proportional-integral-derivative controller) known from automation theory [6].

3.6 Motion blur

A simple 3D scene renderer creates a perfect still image. In real world application, the image is acquired over a short period of time. Changes in the scene during the acquisition process are recorded in the image and moving objects are blurred along the path of their relative motion. This effect is called motion blur. Motion blur can also be caused by the camera's motion itself.

In this approach, just the camera motion is simulated due to the static nature of most objects in the R3-COP use cases. Movement of the camera is defined as a change of camera position and change of camera viewport. First a velocity map is created which defines the velocity of each pixel in the camera image. The velocity defines a relative motion within the viewport. The pixels along the motion direction are accumulated and an average value is stored at the corresponding output pixel. This technique is based on [5].

3.7 Inner reflection in camera lens

Inner reflections in camera lenses present an important problem in robotics especially in applications where robots are exposed to direct sunlight or strong light sources, such as outdoor applications but also some special indoor applications. Inner reflections in lenses are sources of variously shaped artifacts in the image, such as "light rays", "light circles and ellipses", etc. An example of the artifacts is shown in Figure 3. Such artifacts can adversely affect the robot performance especially because they can generate false objects inputs and also mask some existing structures in the image. Several experiments were performed with inner reflection, and while data was collected and some preliminary results are available, the problem seems to be quite complicated with no general solution available.

In current version of the simulator, the inner reflections are simulated with a simplified process. The input environment map is convoluted with the lens flare artifact. The result is cut, resampled, and added to the input image. The research of the inner lens reflection artifacts is still in progress. In the final state, the simulation will most probably be performed still in a simplified form but it will be accompanied with statistical approaches where parts of the artifacts will be generated randomly.



Fig. 3. Demonstration of inner lens reflection artifacts acquired by camera

3.8 Sensor features

The camera sensor itself can also affect the output image. Currently, there are two types of sensors frequently in use: CCD and CMOS. Problems, such as noise and influence of color filter array are common, but each of the sensors has specific problems. Images acquired by CCD sensors can suffer from vertical smearing on bright light (called “bleeding”). On the other hand, images acquired by CMOS sensors can suffer by skew or partial exposure, where part of the image is more exposed than the rest of the image. The acquisition process can also suffer by imperfections in the construction of the sensor.

In the current version of the simulator, simple noise models are implemented. The sensors features and imperfections will be the aim of the further research.

4 Results

The main topic of simulation of camera effects is to produce simulated photorealistic image/video output. This section discusses what photorealism level of lens effects is achieved in the simulator and if the photorealism can be improved in the future. Due to our limitation of no renderer modification and types of inputs, only specific image-based methods are considered in the following section. In Figure 4 the results are presented.

Effects	Result	Robot system ¹	Status
Geometry distortion	Photorealistic	Sufficient	Enclosed
Vignetting	Photorealistic	Sufficient	Enclosed
Depth of field effect	Semi-realistic	Sufficient	Enclosed

¹ The result is considered the point of view of robot systems testing.

Motion blur of camera	Semi-realistic	Sufficient	Enclosed
Motion blur of objects	Not implemented		Enclosed
Exposure automation	Simulated	Sufficient	Enclosed
Inner lens reflections	Simulated	Insufficient	In research
Noise	Simulated	Sufficient	In research

Table 1. Status of the simulated lens effects

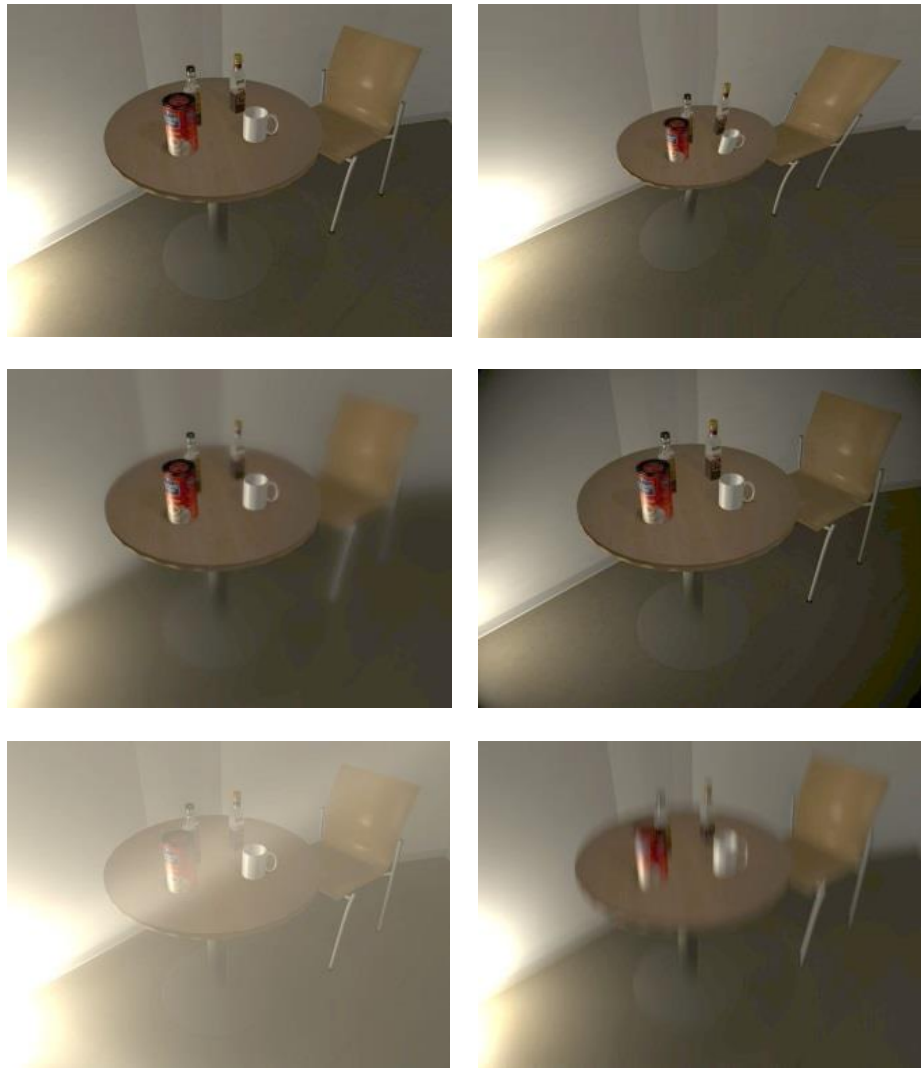


Fig. 4. Video-camera effects

Left, top to bottom: no effect, depth of field effect, inner lens reflection.

Right, top to bottom: geometric distortion, vignetting, motion blur

5 Conclusion

In this paper, we presented a framework for simulation of camera effects for robotics, as researched in the R3-COP project. Our implementation allows simulating of key lens effects such as geometry distortion, chromatic aberration, vignetting, inner reflections and depth of field effect. The implementation also simulates motion blur, exposure automation, and sensor features.

Our research shows that the effects of geometry distortion and vignetting need and can be simulated as photorealistic; motion blur and depth of field effect are simulated with tolerable deviation from reality while the exposure adaptation is simulated realistically but has not yet been fully evaluated and compared to the real world video camera. The presented framework provides modularity and easy modification of the individual effects caused by the separate application of each effect.

Inner lens reflections are still under research. The current simulation of the reflections is acceptably realistic. Future work will also include more sensor features; currently, noise effects are simulated.

6 Acknowledgements

This work was supported by the Artemis JU project R3-COP grant no. 100233 and the TAČR grant V3C TE01010415.

7 References

1. Zemčík, P.; Příbyl, B.; Herout, A.; et al.: “Accelerated Image Resampling for Geometry Correction“. *Journal of Real-Time Image Processing*, 6, 2011, ISSN 1861-8200, pp. 9.
2. Barsky, B. A.; Kosloff, T. J.: “Algorithms for rendering depth of field effects in computer graphics“. In *Proceedings of the 12th WSEAS international conference on Computers, Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS)*, 2008, ISBN 978-960-6766-85-5, pp. 999–1010.
3. Demers, J.: “Depth of Field: A Survey of Techniques“. In *GPU Gems*, Addison-Wesley, 2004, pp. 375–390.
4. Lefohn, A.; Owens, J.: “Interactive depth of field using simulated diffusion“. University of California, Davis, 2006.
5. Rosado, G.: “Motion Blur as a Post-Processing Effect“. In *GPU Gems 3*, Addison-Wesley, 2008, pp. 575–581.
6. Katebi, R.: “Robust multivariable tuning methods“. Springer-Verlag, 2012, ISBN 978-1447124245, pp. 255–280
7. Phong, B. T.: “Illumination for computer generated pictures“. *Communications of ACM* 18 (1975), no. 6, 311–317