Username: MARINet MARINet **Book:** Real-Time Rendering, Fourth Edition. 4th Edition. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

Chapter 2 The Graphics Rendering Pipeline

"A chain is no stronger than its weakest link."

-Anonymous

This chapter presents the core component of real-time graphics, namely the *graphics rendering pipeline*, also known simply as "the pipeline." The main function of the pipeline is to generate, or *render*, a two-dimensional image, given a virtual camera, three-dimensional objects, light sources, and more. The rendering pipeline is thus the underlying tool for real-time rendering. The process of using the pipeline is depicted in Figure 2.1. The locations and shapes of the objects in the image are determined by their geometry, the characteristics of the environment, and the placement of the camera in that environment. The appearance of the objects is affected by material properties, light sources, textures (images applied to surfaces), and shading equations.

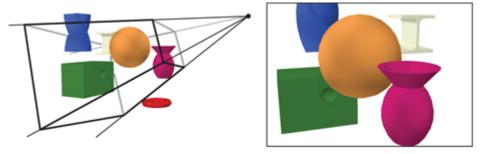


Figure 2.1 In the left image, a virtual camera is located at the tip of the pyramid (where four lines converge). Only the primitives inside the view volume are rendered. For an image that is rendered in perspective (as is the case here), the view volume is a frustum (plural: frusta), i.e., a truncated pyramid with a rectangular base. The right image shows what the camera "sees." Note that the red donut shape in the left image is not in the rendering to the right because it is located outside the view frustum. Also, the twisted blue prism in the left image is clipped against the top plane of the frustum.

We will explain the different stages of the rendering pipeline, with a focus on function rather than implementation. Relevant details for applying these stages will be covered in later chapters.