

*Foreword by Ken Smith*

# DIGITAL DRAWING for LANDSCAPE ARCHITECTURE

Contemporary Techniques and Tools  
for Digital Representation in Site Design



BRADLEY CANTRELL • WES MICHAELS



# **Digital Drawing for Landscape Architecture:** Contemporary Techniques and Tools for Digital Representation in Site Design

Bradley Cantrell & Wes Michaels



John Wiley & Sons, Inc.

This book is printed on acid-free paper. ∞

Copyright © 2010 by John Wiley & Sons, Inc. All rights reserved

Published by John Wiley & Sons, Inc., Hoboken, New Jersey

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at [www.wiley.com/go/permissions](http://www.wiley.com/go/permissions).

**Limit of Liability/Disclaimer of Warranty:** While the publisher and the author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor the author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information about our other products and services, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books. For more information about Wiley products, visit our web site at [www.wiley.com](http://www.wiley.com).

**Library of Congress Cataloging-in-Publication Data:**

Digital drawing for landscape architecture : contemporary techniques and tools for digital representation in site design / Bradley Cantrell and Wes Michaels.

p. cm.

Includes index.

ISBN 978-0-470-40397-6 (pbk. : alk. paper) 1. Landscape architecture—Computer-aided design. 2. Landscape design—Data processing. I. Michaels, Wes. II. Title. III. Title: Contemporary techniques and tools for digital representation in site design.

SB475.9.D37C36 2010

712.0285—dc22

2009049253

Printed in the United States of America

ISBN: 9780470403976

10 9 8 7 6 5 4 3 2 1

# Contents

Foreword . . . . .	viii
Preface . . . . .	ix
Acknowledgments . . . . .	xi

## Part 1: Concepts

<b>Chapter 1: Introduction/Overview . . . . .</b>	<b>2</b>
Computing Basics. . . . .	5
Hardware . . . . .	6
Software . . . . .	11
Workspace . . . . .	15
<b>Chapter 2: Analog and Digital Rendering Comparisons. . . . .</b>	<b>17</b>
Efficiency and Editability . . . . .	17
Commonalities and Parallels . . . . .	19
Hybrid Techniques . . . . .	20
<b>Chapter 3: Basic Overview of Digital Concepts . . . . .</b>	<b>23</b>
Raster-Based Programs . . . . .	23
Resolution in Raster Images . . . . .	24
Upsampling and Downsampling . . . . .	26
Vector Images . . . . .	27
Using Raster Images in Vector-Based Programs . . . . .	28
Color . . . . .	28
<b>Chapter 4: Digital Drawings in the Design Process . . . . .</b>	<b>32</b>
Applications for Specific Tasks . . . . .	32
Moving between Analog and Digital Techniques . . . . .	34

## Part 2: Base Imagery

<b>Chapter 5: Setting up the Document . . . . .</b>	<b>38</b>
Drawings at Multiple Sizes . . . . .	38
How Drawings Move through the Digital Workflow . . . . .	39
Setting the Image Size. . . . .	39

<b>Chapter 6: Base Imagery and Scaling</b>	<b>42</b>
Aerial Photography	42
Obtaining the Aerial Photograph	44
Tiling Aerial Photographs in Photoshop	44
Manual Method	44
Tiling Photographs with Photomerge	47
Scaling the Aerial Photograph	49
Calculator Method	49
Scale by Reference Method	51
Pixel Conversion Method	53
Adjusting the Hue, Saturation, and Lightness of Base Imagery	53
Using CAD Linework as a Base	57
Exporting the CAD Linework as a PDF	57
<b>Chapter 7: Hand-Drawn Linework</b>	<b>61</b>
Sketches in CAD	61
Using Magic Wand and Color Range Selections	62
Using the Color Range Selection Tool	65
<b>Chapter 8: Source Imagery/Entourage</b>	<b>68</b>
Selections	68
Manual Methods	69
Partial Selections	69
<b>Part 3: Design Diagrams</b>	
<b>Chapter 9: Introduction to Diagrams</b>	<b>86</b>
Passive Diagramming	86
Active Diagramming	87
Communication	89
Abstraction	89
Distilling and Culling	90
Diagram Types	90
<b>Chapter 10: Setting up an Illustrator Drawing</b>	<b>91</b>
Document Size/Color Mode	91
Based Programs for Design Diagrams	91
Importing an Aerial Photo into Illustrator	92
Link versus Embed	92

<b>Chapter 11: Linework in Illustrator.</b>	<b>95</b>
Shape Tools.	95
Pen Tool	96
Editing Tools	98
Appearance of Lines and Shapes	100
Stroke Weight and Dashed Lines	101
Transparency	102
Appearance Palette.	102
<b>Chapter 12: Custom Linework</b>	<b>105</b>
Creating a Pattern Brush from Shapes.	105
Altering the Pattern Brush	111
Updating the Pattern Brush with New Shapes.	112
<b>Chapter 13: Symbols</b>	<b>113</b>
Creating Symbols from Custom Artwork	113
Updating/Replacing Symbols	115
Managing Symbols	118
Creating Clipping Masks for Image Symbols.	118
<b>Chapter 14: Text, Leaders, and Page Layout.</b>	<b>120</b>
Text Tools.	120
Point Text.	120
Paragraph Text	120
Differences between Point Text and Paragraph Text.	121
Formatting Text.	122
Custom Type Tools	124
Creating Text with a Clipping Mask	125
Leaders.	127
Effects versus Filters	130
Layout	131
<b>Chapter 15: Exploded Axonometric Diagrams.</b>	<b>134</b>
Creating an Exploded Axonometric Diagram	134

## Part 4: Plan/Section Renderings

<b>Chapter 16: Introduction to Renderings.</b>	<b>146</b>
Design Process	147
Issues in Digital Media	147
Illustrative Components.	148

<b>Chapter 17: Importing PDF Linework</b> . . . . .	<b>150</b>
PDF Linework . . . . .	151
Adjusting the Appearance of Linework . . . . .	160
<b>Chapter 18: Applying Color to a Plan Rendering</b> . . . . .	<b>164</b>
Technique 1: Applying Color with the Paint Bucket Tool . . . . .	164
Technique 2: Applying Color Using Adjustment Layers . . . . .	173
Saving Channels . . . . .	184
<b>Chapter 19: Shading Techniques</b> . . . . .	<b>186</b>
Selecting Fills . . . . .	186
Saving Selections . . . . .	187
Automating the Shading of Edges . . . . .	191
<b>Chapter 20: Creating Textures</b> . . . . .	<b>193</b>
Creating a Texture from an Existing Photograph . . . . .	193
Creating a Seamless Pattern Using the Offset Filter . . . . .	195
Creating the Pattern and Applying It to the Rendering . . . . .	198
Paint Bucket and Pattern Stamp . . . . .	199
Pattern Overlay . . . . .	200
Managing Patterns . . . . .	200
Texturing with Filters . . . . .	201
<b>Chapter 21: Brushes</b> . . . . .	<b>203</b>
Standard Brushes . . . . .	203
Custom Brushes . . . . .	210
<b>Chapter 22: Plan Symbols with Smart Objects</b> . . . . .	<b>213</b>
Creating Smart Objects . . . . .	213
Duplicating and Editing Smart Objects . . . . .	214
Managing Smart Objects . . . . .	215
Smart Filters . . . . .	216
<b>Chapter 23: Managing Large Photoshop Files</b> . . . . .	<b>217</b>
Flattening Layers . . . . .	217
Saving Layer Groups for Flattening . . . . .	218
Printing Issues . . . . .	219
<b>Chapter 24: Creating a Section Elevation</b> . . . . .	<b>220</b>
Methods . . . . .	220



## Part 5: Perspectives

<b>Chapter 25: Perspective Illustration . . . . .</b>	<b>230</b>
Perspective Illustrations, Digital Sketches, and Design Communication . . . . .	232
<b>Chapter 26: Creating a Base for a Perspective Drawing. . . . .</b>	<b>236</b>
Composition . . . . .	237
Virtual Cameras. . . . .	238
Exporting and Rendering . . . . .	244
<b>Chapter 27: Atmospheric Perspective . . . . .</b>	<b>245</b>
Detail. . . . .	246
Color . . . . .	247
Contrast . . . . .	247
Brightness . . . . .	247
(2D) Photoshop Adjustment Layers, Opacity, and Screening. . . . .	249
(2D/3D) Z-Depth . . . . .	254
(3D) Atmosphere/Environment . . . . .	256
Understanding Level of Detail. . . . .	257
<b>Chapter 28: Camera Match 3D Object to Site Photo . . . . .</b>	<b>258</b>
Camera Match with 3ds Max 2009 . . . . .	260
Match Photo with Google SketchUp . . . . .	264
<b>Chapter 29: Create a Photoshop Perspective Collage . . . . .</b>	<b>267</b>
Methods . . . . .	267
<b>Chapter 30: Developing a Perspective Image in Photoshop from a 3D Model . . . . .</b>	<b>281</b>
Base Model . . . . .	281
Adding Site Context . . . . .	282
Textures . . . . .	288
Adding Vegetation . . . . .	290
Adding Scale Figures . . . . .	294
<b>Bibliography. . . . .</b>	<b>297</b>
<b>Image Credits . . . . .</b>	<b>298</b>
<b>Index . . . . .</b>	<b>303</b>

## Foreword

It is easy these days to assume everyone knows how to use new media. One probably imagines that by now new electronic media and image making are embedded in the DNA of everyone who uses a computer, personal device, or other digital tools. But the reality is that media and image making are an art form with learned techniques and protocols. There is always need for excellent tutorials that describe basic techniques and their application and I am pleased that *Digital Drawing for Landscape Architecture* is providing for a new generation of landscape architects training in contemporary digital media and its application as an emerging art form.

I come from the generation that learned manual graphic techniques came of age using digital applications as an extension of these traditional techniques. As a student, I remember learning photography by reading manuals and silk-screening from printmaking books. Skills like collage and montage were acquired much more intuitively, and other techniques such as press-on lettering were learned on the job in an office. It is interesting to me to see how much of the old methods are built into the new digital procedures. One of my favorite Photoshop filters is *pixelate-mezzotint*. From my knowledge of printmaking, this filter makes clear sense to me. Likewise *cut* and *paste* tools are basically collage techniques, and *dry brush* and *cross hatch*, etc., are based on traditional art processes. Expanded electronic techniques go beyond mere digital adaptation of the traditional to create new graphic and design possibilities that were difficult or even impossible to achieve before. Certain kinds of geometric distortions, such as stretching, bending and inversions, are not only transforming the representation of landscape design but also design itself as new forms and spatial relationships are pioneered in digital space. Combining techniques to create non-standard representation forms has emerging potential as well. This is clearly evident in today's contemporary art world where artists are creating new art which incorporates a vast array of new media in innovative ways to challenge our way of seeing and understanding the world. Today, my office uses an array of representation techniques ranging from drawing to physical model building to digital modeling, and all sorts of combinations of digital imaging and animations, all at a range of differing scales. Ultimately, the best design still results from thinking, designing and representing with multiple scales, views and methods.

This book will become a standard manual for students entering the profession and learning their craft, as well a valuable reference for those already in practice who need to keep current with emerging trends. Just as it was impossible to practice twenty-five years ago without knowledge of ozalid printing, leterset, zipatone and rapidograph use, today it is unimaginable to practice in a world without Photoshop, Illustrator, 3DStudioMax, Rhino, SketchUp and CAD.

—Ken Smith

## Preface

*Digital Drawing for Landscape Architecture: Contemporary Techniques and Tools for Digital Representation in Site Design* is the product of many years of professional practice and teaching at the Louisiana State University Robert Reich School of Landscape Architecture. As designers, we attempted to create a book that focused on getting the job done. In this sense, each section tackles the basics of the subject matter and each chapter introduces a short background with an explanation of how to accomplish a phase of the representation process with current digital tools. Our inspiration comes from the books that introduced us to landscape architectural graphics, such as Grant Reid's *Landscape Graphics* and Chip Sullivan's *Drawing the Landscape*. Both books present the reader with techniques that are applicable to a specific topic with just enough background to explain how it fits within the larger profession. Our hope is that *Digital Drawing for Landscape Architecture* will serve as a contemporary, digital version of these books for landscape architecture professionals and students.

We come from a group of academics and professionals who did not take any formal digital media courses. Instead, we were taught analog mechanical drafting and drawing and then applied those skills to our interest in digital media. All of our skills come from exploration through trial and error. We learned that doing it the second or third time was always the most productive. Typically, we would jump into a project and begin to experiment. If we didn't understand a tool, we opened the Help file or just started using it to see what happened. This book outlines techniques, but we encourage you to experiment. There are an infinite number of ways to get to the same solution, and it is important that you find a way that works for you.

*Digital Drawing for Landscape Architecture* is a book about the moment, bridging analog and digital techniques. Digital landscape representation relies heavily on the past, and we attempt to tie past and present together. We are consistently amazed at the work our colleagues and students produce, and our hope is that by putting out defined techniques, individuals will question and evolve these practices. In the long run, landscape representation will eventually begin to leave the conventions of the mechanically drafted orthographic drawing in favor of parametric modeling and geographic information systems. While these systems exist, they currently do not address the needs of site designers as creative design tools.

It is always a risk to base any book on specific software, but when techniques are introduced it is almost impossible to be completely software agnostic. It is possible to create amazing work with any software, but we focused on the tools we use everyday: Adobe Photoshop, Adobe Illustrator, 3ds Max, SketchUp, and Vectorworks or AutoCAD. These are not the only tools, but they are the ones we have evolved with over time and, therefore, feel the most comfortable using. While software does change, it has

essentially been very consistent for the past 10 years. Features are added and refined, but the process has not been considerably altered through time. It is very easy to constantly chase the newest tools, but it is typically more productive to evolve our own processes with or in spite of the tools. You will find that most of the techniques discussed in this book will work in software versions that date back 5 to 10 years.

*Digital Drawing for Landscape Architecture* presents examples and techniques for each of the traditional design drawings: diagram, plan, section/elevation, and perspective. These drawings are the basis for all of our representation endeavors; and while we encourage experimentation in how these drawings evolve, it is important to recognize the need for measured drawings when working in digital media. The techniques also focus on speed and efficiency, which translates to getting a job done quickly, with the fewest mouse clicks, and being able to edit the drawing when necessary. You will find that almost every technique allows representation to be an iterative process, creating elements that we assume will be changed or modified. There is very little certainty within the design process and, therefore, it is essential that drawings remain flexible.

This book assumes that readers are versed in basic representation concepts and computing principles. The book spends a small amount of time discussing how computing affects the representation process and the basics of each piece of software, but it is not intended as an introduction to any particular piece of software. There are many great books that catalogue and explain each feature of the software. The software's Help file is a great resource to help you understand every tool and its effect. If you don't understand a concept in the Help file, use a search engine and find out more information on the Web. We are no longer working alone, and someone else may have already figured out or encountered many of the issues you will run into with the software.

This book is intended to highlight examples, explain techniques, and provide context for how we use digital media as designers. Feel free to start at the beginning or jump around to areas of interest; either method is suitable to take advantage of the information. We hope you will take away something new and contribute back to others with new and interesting techniques.

## Acknowledgments

There are many people to acknowledge in the creation of this book. My wife, Susan, has been extremely supportive and her expertise as a graphic designer has contributed to my knowledge of digital media. Without my friend, colleague, and co-author Wes Michaels, this book would have never come to life. I also need to give a special thanks to my friend and the Director of the LSU Robert Reich School of Landscape Architecture, Elizabeth Mossop. My graduate assistants Patrick Michaels and Natalie Yates have made huge contributions that have made this book possible.

I must go back many years and acknowledge the genesis of this work which started during my time in graduate school as an intern for Geller DeVellis. Many of the techniques and approaches to digital media were developed alongside my colleagues Scott Carman and Chuck Lounsberry. I consistently learned new methods thanks in large part to the genius of these two individuals. Working as a team, we were able to push our boundaries and accomplish some very innovative work at the time. Joseph Geller and Bob Corning also gave me opportunities and freedoms that contributed directly to our success.

I also need to acknowledge David Fletcher and Wilson Martin who helped to develop an approach to teaching digital landscape representation that focused on tangible techniques embedded in historical precedent. This was a departure when we first taught the landscape representation module at Harvard but is the focus of this book.

Teaching for the past five years at the Louisiana State University Robert Reich School of Landscape Architecture, I have seen phenomenal changes in the way students are conceptualizing and representing their projects. This book is full of their amazing work, and I feel extremely proud to say that I had a part in teaching them. The students consistently amaze me with their hard work and the design work they produce. While I can't name everyone, I would like to specifically acknowledge the following digital (and analog) media gurus: Chris Africh, Natalie Yates, Christopher Hall, Patrick Michaels, Jamin Pablo, Ying Lou, Xiaoyang Zhao, Joaquin Martinez, Paul Toenjes, Andrea Galinski, Louise Cheetham, Megan Colwart, Zhujun Wang, and of course the dream team: Bob Bass, Chris Barnes, and Connors Ladner. There are many more doing amazing work that I am sure I am forgetting.

Much appreciation goes to our editor Margaret Cummins, who has worked with us through this process.

Brad Cantrell

Many of the ideas found in this book have their origins in a course I first taught several years ago at LSU, and continue to teach to this day. That course, and this book, evolved through many hours of conversation with the co-author of this book, Brad Cantrell. The techniques were refined through many hours of experimentation with Brad while working on design competitions and research grants at the university. Similar to Brad's experience, the ideas and techniques I have contributed to this project are based on my experience in professional offices working on projects with deadlines. In my office today, we use the digital workflow concepts and techniques described in this book. I would like to thank the incredibly talented Jessica Pfeffer for all of her hard work on the projects from my office. Jessica's hand is seen throughout, and I give credit to her for the beautiful work on so many of the images in the book.

Many of the same people that Brad has acknowledged above are some of the same people that I have worked with at LSU. I won't rename each one, but will say a special thanks to Elizabeth Mossop for her support as Director of the school and our editor Margaret Cummins for her patience and good humor over the many long months this book was being put together.

Finally, I would like to thank Anne for her support while I spent evenings and weekends writing this book. Without her encouragement and understanding, this book would have never made it through to the end.

Wes Michaels

# Part 1

## Concepts

# Chapter 1

## Introduction/Overview

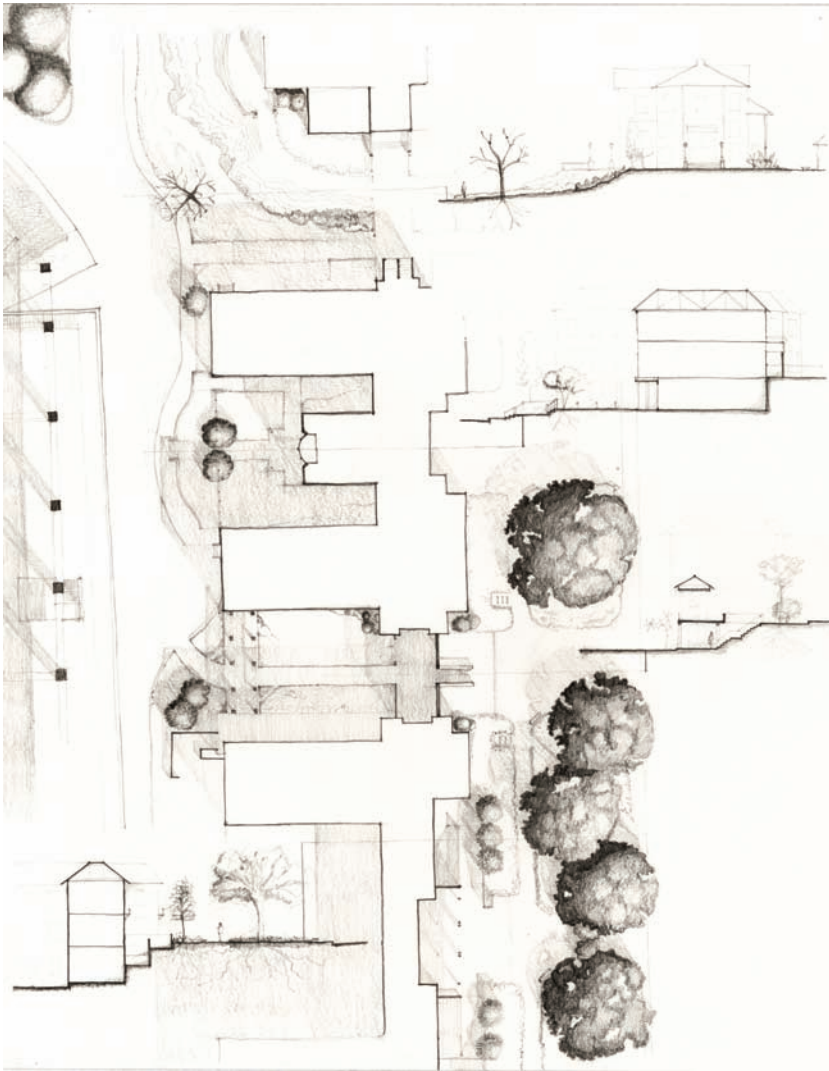
*Digital Drawing for Landscape Architecture: Contemporary Techniques and Tools for Digital Representation in Site Design* provides professionals and students with a clear guide to understanding the digital representation process for a variety of design drawings. Each chapter highlights a specific technique by examining its role in the digital media and landscape representation process through methods available in current software. This provides the reader with tangible tools to explore digital media in the creation of design drawings.

The professions of landscape architecture and urban planning have a strong tradition of representation that has evolved with the professions. During the last hundred years, this has been dominated by analog representation—primarily pencil (graphite), pen (ink), markers (pigment), and watercolor (pigment). The aforementioned analog representation techniques have focused on creating a variety of design drawings such as functional and operational diagrams, orthographic plans, section/elevations, isometrics, and perspective renderings.

The content in this book intends to bridge a fundamental gap between the analog and digital tools used to represent landscape architecture and urban planning projects. The gap has formed in representation methods with the introduction of digital tools that have been adopted despite a generation of designers who are versed in analog methods. *Digital Drawing for Landscape Architecture: Contemporary Techniques and Tools for Digital Representation in Site Design* aims to fill this gap by pulling from the methods of analog representation and applying these concepts to digital media. Examining individual working methods and applying the content of this book to enhance the current design and representation processes are essential to this goal.

A misnomer that many designers intend to embrace when moving to digital representation methods is that the past can be left behind; nothing could be further from the truth. Knowledge of analog representation plays a vital role in understanding the application of digital tools and techniques. Tools such as Adobe Illustrator and Photoshop are born directly from analog processes and tools defined by their physical counterparts. The Paint Bucket tool is used to pour paint into areas, and the PaintBrush tool applies paint to a virtual canvas. This language is intentional and builds on our current knowledge of illustration, avoiding the creation of a new digital tool that has no context in the physical world. It would be confusing and the learning curve would be that much steeper if the Photoshop Paint Brush tool was called the Pixel Application tool and the canvas was called the pixel grid.

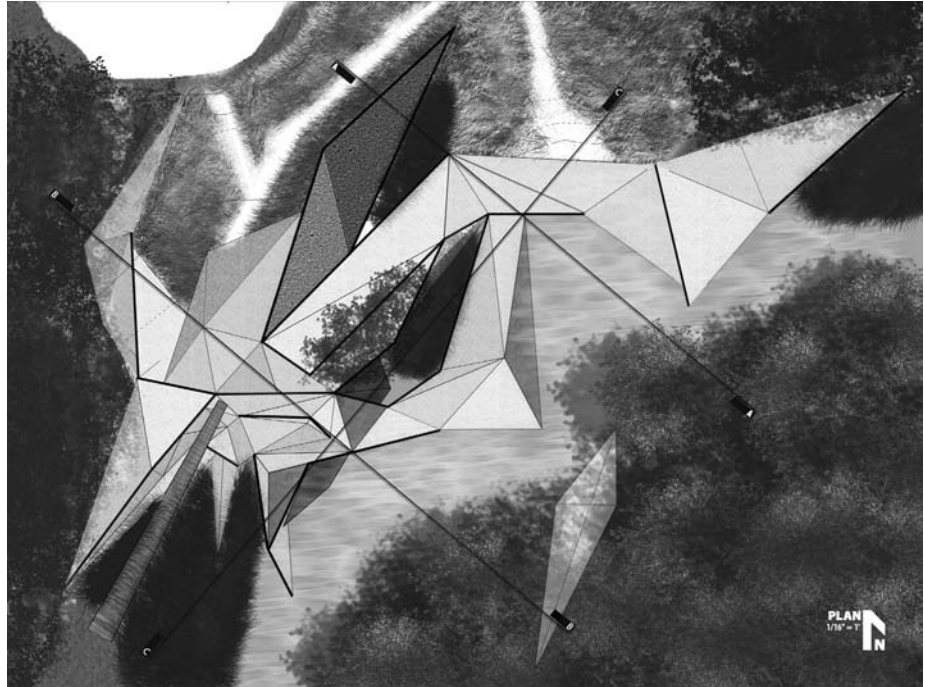




**Figure 1.1.** Graphite, walking plan.

The connections between analog and digital modes go beyond naming conventions into techniques and processes. Current digital rendering processes vary greatly between individuals and firms, as well as across a range of software. It is commonly said that there are an infinite variety of ways to accomplish the same task in image- or vector-editing software. The versatility of most software packages comes from the variety of tools and the options for combining those tools to complete a specific task. This versatility allows the software to be used across a variety of professions from photography to technical illustration. Because of the depth and versatility of the software, the learning curve is typically steep for new users. Similar to using a pencil and pen, there is no way to automatically generate a section, plan, or elevation. Instead, a combination

of tools and methods come together through a proven process to generate the desired results. Digital media provides efficiencies in some areas but does not provide a short-cut to learning the fundamentals of drawing and illustration.



**Figure 1.2.** AutoCAD and Photoshop, Bayou Bienvenue site plan.

Understanding the fundamentals of drawing is essential, but it is not exclusive to either medium. The contemporary design world fully embraces both mediums as valid methods to represent projects and explore design ideas. It is possible to understand the fundamentals of composition, lineweight, texture, color, and/or atmosphere with a pencil or with Photoshop. The physical processes may be different, but conceptually the rules and ideas are similar.

Conceptually, each designer must embrace digital media as a tool with analytic, performative, and representational possibilities. Many designers view the computer as a rival that must be conquered in order to accomplish each task. It is important to reverse that role. In order to do this, the designer should have a general understanding of how a computer and operating system function. This environment of hardware and software is where most processes occur; therefore, taking the time to become familiar with your surroundings is very useful. Typically, this is a low priority for designers; we are not computer engineers and, therefore, we often overlook or even overcomplicate basic hardware and software functions.



**Figure 1.3.** Illustrator and Photoshop, ambient space section-elevation.

## Computing Basics

Understanding the basic components of computing and how they affect the design and representation process is necessary. The relationships between hardware and software and the operating system and applications are important to understand in order to efficiently utilize the tools. Understanding this relationship demystifies computing processes that are not readily apparent to the end user. Typically, the hardware, operating system, and applications attempt to hide as much of the computing processes as possible from the end user, but there are times when it is necessary to know enough in order to troubleshoot simple problems.

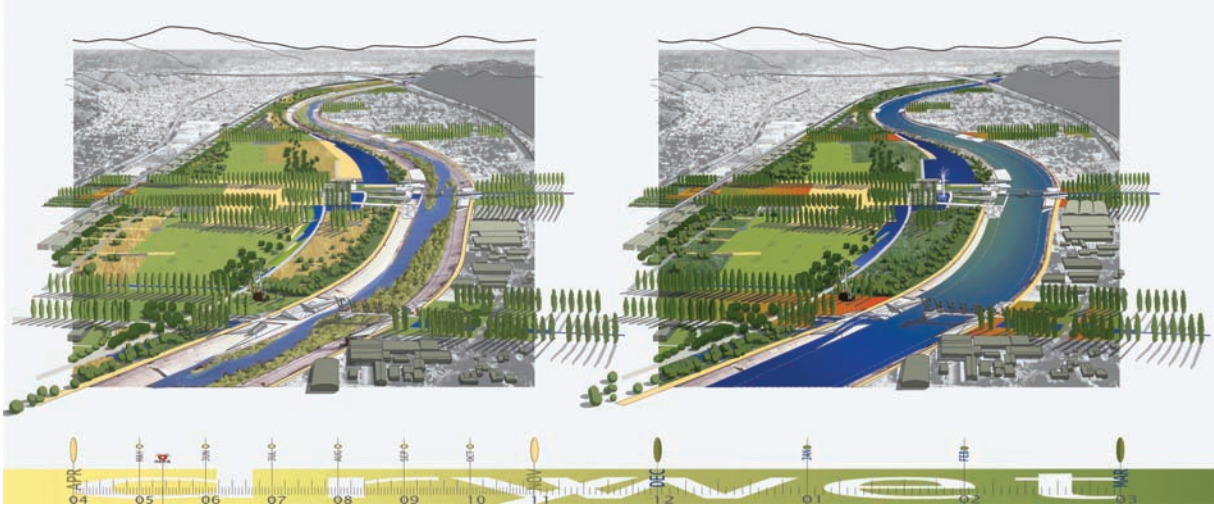


Figure 1.4. Adobe Illustrator, Taylor Yards seasonal-change aerial perspectives.

## Hardware

Laptop and desktop computers typically have a motherboard, a processor, memory, a hard drive, a network connection (wired and/or wireless), and a graphics card. These components provide the necessary functions to do most of the tasks we consider necessary in contemporary computing. From this base, we can begin to add components such as a keyboard, mouse, and monitor in order to have a fully functioning machine. There is little difference between a desktop and laptop computer other than the fact that the keyboard, mouse, and monitor are integrated on a laptop. Because laptop components are usually much smaller and are custom built for each brand, upgrading them can be much more difficult than upgrading desktops. For example, processors are usually not upgradeable components in laptops because they may be soldered directly to the motherboard.

### Motherboard

The main component of computers that we rarely discuss is the motherboard. The motherboard is the framework for the entire machine, but unless we are building the computer ourselves, it is not an extremely important consideration. The motherboard provides connections for the processor, memory, storage, graphics cards, and expansion slots, which would contain the graphics card and other components. Many motherboards will integrate graphics, network, and sound functionality directly into the motherboard. Integrated solutions often provide less functionality and performance; however, in the case of network and sound, this typically is not a problem for designers who rely on the computer to create visual products. At the time of this writing, integrated graphics cards should be avoided; they perform poorly when used for

illustration and three-dimensional modeling, and they are not upgradeable when new graphics cards are introduced.

## Processor

The processor or CPU (central processing unit) can be thought of as the brain of any computer. The CPU is attached to the motherboard in the CPU socket and is dependent on the *chipset* (technologies that constitute the motherboard). Because of the complexity of CPUs, there are very few manufacturers in the commercial computing market. Nearly all computer manufacturers (Dell, HP, Apple, etc.) use either Intel or AMD processors that use the x86 instruction set. Two important factors need to be addressed regarding the CPU: 32-bit and 64-bit addressing and processing speed. Most CPUs at this time are moving toward 64-bit addressing, which is very important for visual artists because it allows the computer to more efficiently execute instructions and address larger amounts of memory. In the past, 32-bit CPUs could only address less than 4GB of memory; however, with the introduction of 64-bit processors, it is theoretically possible to address 16.8 million terabytes of memory. In order to utilize a 64-bit processor, it must have an operating system and software that is coded to take advantage of the 64-bit instruction set.

The second important concept to understand is the speed of processor, which is typically measured in gigahertz. If everything else is equal with a computer, the faster the speed of the processor, the more instructions it can complete in a given time (milliseconds). Many factors, such as memory and hard drive speed, can change this general rule. However, it can generally be assumed that the faster the processor, the faster the computer will complete operations. This affects us directly in the representation process as we apply procedures and effects that require the computer to do large calculations. An example would be the application of a filter in Photoshop; when the filter is applied, the computer must calculate the effect on the image, and a faster processor will typically take less time to accomplish this task.

It is also possible to use multiple processors in some desktop/workstation configurations in order to provide more processing power. In most instances, dual processors provide more efficiency in multitasking rather than doubling the processing power. In essence, this allows a filter to be calculated while switching to another program to accomplish a secondary task without as much of an overall slowdown in the computer. Most of the current processors also use multiple cores, dual cores, and quad cores, which can be thought of as multiple processors embedded within a single processor. This provides greater multitasking possibilities, and software is currently being written with multiple cores in mind in order to take advantage of these efficiencies.

## Memory/RAM

To use another analogy, memory or *RAM* (random access memory) can be thought of as short-term memory. Information that is currently being accessed is stored in the computer's RAM, which allows the information to be accessed very quickly. The major

benefits of RAM are its speed and the fact that data can be retrieved from any location in the memory rather than being retrieved sequentially. This is why the term random access memory is used. RAM is a *volatile* storage medium, which means that when the power is turned off information is lost. If possible, it is desirable to load information into RAM and to complete all of the operations within the computer's RAM without having to offload data to the hard drive. Anytime the computer goes beyond the computer's current RAM, operations slow down as the hard drive is accessed to swap out information.

As mentioned previously, there are certain limitations in computer architecture where 32-bit systems and software can only access memory configurations that are less than 4GB. Currently, most hardware and software is moving toward 64-bit and can access much larger quantities of RAM. A desktop machine should have between 4GB and 8GB of RAM, while laptops typically max out at 4GB—although some workstation replacement laptops are moving to 8GB configurations. Typically, more RAM is better. If the computer, operating system, and software are all 64-bit compatible and if it is affordable, most systems will benefit from up to 8GB of RAM when working with large images or data sets.

## Hard Drives

Another type of memory is the mass storage device, which is usually referred to as the hard drive. A hard drive provides a storage medium that is slower than RAM but capable of storing much larger amounts of information. The hard drive typically stores between 300 gigabytes to 1,000 gigabytes (a terabyte) of data with larger sizes on the horizon. The hard drive is a nonvolatile storage medium; therefore, when the power is turned off, the information is retained. This is why almost everything needed to make a computer function is stored on the hard drive, including the operating system, applications, and all of the user data such as documents, images, videos, and music.

Hard drives operate at different speeds that are measured in revolutions per minute (RPM), which are typically 5,400, 7,200, and 10,000RPM. A faster hard drive will optimize the system's bootup speed, allow applications to load faster, and speed up the opening of files. Currently, 10,000RPM hard drives are prohibitively expensive for very large drives and a common configuration is to use a 10,000RPM drive for the system, applications, and current projects, and another drive as a data archive. In many offices, this second drive is not a concern because most project data is stored on a centralized server; therefore, the bottleneck is often the speed of the network and the number of users accessing the data. The hard drive is often contained within the computer, but it is also possible to use external enclosures that contain a hard drive. This allows data to be more portable and not tied to a specific computer.

## Graphics Cards

Graphics cards provide a secondary processor that specializes in two- and three-dimensional graphics calculations. Current versions of Photoshop take advantage of the

graphics card to display images in both two and three dimensions. True three-dimensional applications depend on the video card to display three-dimensional data on the screen in real time. Graphics cards have three components to be concerned with: the speed of the GPU (graphical processing unit), compatibility with OpenGL or DirectX, and the amount of memory on the card.

Generally speaking, the faster the speed of the GPU, the more graphical operations the card can perform per second. This translates to faster panning and zooming in Photoshop, as well as smoother orbiting, panning, and zooming in Google SketchUp, 3ds Max, and Maya. Upgrading the graphics card is usually possible in most desktop computers; however, this typically is not possible in laptops.

## Monitor

Beyond using these basic components, there are many ways to extend the functions of a computer through peripheral devices that can be categorized as providing an input or output. The most important output peripheral is the computer monitor or display. Displays come in a range of sizes from 13" laptop screens to 30" high-resolution desktop monitors. Most monitors use LCD (liquid crystal display) technology with either fluorescent or LED (light emitting diode) backlighting, with LED backlighting consuming the least power and having more even brightness. There are other technologies for computer displays, but the LCD is by far the most common in both laptops and desktops.

The most important thing to consider for LCD monitors is their native resolution, which is the number of pixels that they can display both horizontally and vertically. *Native resolution* is the resolution at which the display is designed to function. It will typically be between  $1024 \times 768$  and  $2560 \times 1600$  pixels. This resolution is the actual number of pixels that are displayed across the surface of the monitor. Smaller monitors typically display fewer pixels, and larger monitors display more. When a designer is working with high-resolution images, it is useful to have larger monitors in order to see more of the image at its actual size at once. Larger high-resolution screens also make it more convenient to have multiple applications open at once because they will be able to fit on one screen. If the graphics card supports multiple monitors, it is possible in both Windows and OS X to use two or more monitors in order to span the desktop across the screens. This can be very useful and is often an affordable and more versatile solution to having one extremely large monitor.

The second most important factor is the actual size of the monitor. For laptop computers, the monitor is based on the size of the computer. However, most laptops provide either a VGA (analog) or DVI (digital) port in order to connect an external display. The monitor's size is measured diagonally across the surface of the monitor, so a 24" desktop monitor measures two feet from the upper-left corner to the lower-right corner. Most LCD monitors have a 16:10 widescreen *aspect ratio*, which is the ratio of horizontal to vertical pixels. Beyond size and resolution, some monitors provide other input options

that allow video devices to be connected or even act as an input hub for USB and/or FireWire devices.

## Input

Two often overlooked devices are the mouse and the keyboard. They are typically boring, and every computer has them. However, the keyboard and mouse are important and should be comfortable and fully functioning because they are the primary methods of interaction with a computer. Nothing will slow down a designer more quickly than a sticky keyboard or a mouse that does not smoothly track across the screen. Each user will have their own preference when choosing a keyboard or mouse, but generally the keys should provide tactile feedback and the mouse should have enough weight to provide smooth tracking on the screen. In most operating systems, the speed of tracking, double-clicking, and other features can be adjusted in the Control Panel (Windows) or the System Preferences (OS X). These settings need to provide the most comfortable and efficient experience for each user.

Beyond the keyboard and mouse, several other haptic devices expand the ways in which we interact with a computer. Pen tablets provide a decent tool to naturally sketch or draw within most image-editing applications. The main benefit of this type of input device is that it gives the user the ability to draw naturally and have multiple levels of input. This allows the user to press softly for one effect and harder for another effect. In Photoshop, for example, the pen pressure can be mapped to the number of pixels that are applied or even how far away the pixels scatter from the area they are being applied. The main drawback with using a tablet is that it requires some time to become comfortable with the device. Another drawback is that if it is not used every day, it can be hard to jump right back into using it.

Spatial navigation devices are also worth discussing. These haptic input devices provide an alternative two-dimensional (2D) or three-dimensional (3D) axis that allows for navigation across the picture plan in Photoshop or for moving with a SketchUp or 3ds Max scene. The typical configuration is a puck or joystick that can be used in configuration with a keyboard and mouse that can be pushed forward and backward, or pulled upward or pushed down to control movement along each axis. Most devices also provide hotkeys to map frequently used functions.

Other input devices include cameras and scanners that can be used to capture a variety of images for use in digital rendering. Digital photography is the subject of an entire book, but it is important to consider some very basic concepts. Digital cameras are the best method to record the environment around us, capturing images of textures, people, plants, materials, etc. It is useful to build a versatile collection of imagery that is well organized and easily accessible for use in renderings. Cameras capture images at resolutions measured in millions of pixels or megapixels. The more megapixels a camera can capture, the higher the resolution of the image and, therefore, the larger the output when printing occurs. Megapixels are only one factor when determining the



quality of the final image; it is also important to consider lighting, stability of the camera, focus, and shutter speed.

Desktop and wide-format scanners are the best method to capture printed media, sketches, and other flat materials. Scanning can capture images as black and white, grayscale, or full color at multiple resolutions, depending on the scanner's ability. Black-and-white scanning captures images without differentiation or shading, and it is the most useful for clip art or pen-and-ink drawings. Grayscale will capture 256 shades of gray and is most often used to capture graphite or charcoal sketches, but it can also be used for line drawings to more accurately reflect changes in lineweight and tone. Full-color scanning should be used when scanning images or materials where color is necessary. The resolution or *dots per inch* (dpi) must be considered when scanning. If the scanned image will be used at the same size in the final output, then the scan can be done at the final output resolution. For example, an 11" × 17" sketch that will be touched up in Photoshop and reprinted at 11" × 17" can be scanned at 200 dpi and then be reprinted at 200 dpi. If the image needs to be larger in the final output, then it is possible to scan the 11" × 17" sheet at 400 dpi and then reprint the sheet at 22" × 34" at 200 dpi. It is possible to always scan at very high resolutions and in full color in order to have an image that is versatile in many situations. The only problem with this method is that it will take longer to scan and will create larger images that take up more hard drive space and are slower to process.

## Software

Software typically describes code or computer programs that perform a specific task within a computer system. Although there are many types of software, designers are typically concerned with specific types of applications for pixel/raster editing, vector editing, three-dimensional modeling, and video/motion graphics editing. Each type of application plays a different role in the representation process but also interacts with and utilizes the hardware in different ways. Beyond applications, it is also important to understand the role of the operating system because it is at the core of any hardware/software relationship.

## Operating System

The operating system handles the intricacies of the interaction between the user and the hardware. Generally, nearly all of the computing devices we use from desktop computers to video game consoles use some type of operating system that we interact with using a graphical user interface (GUI). The two prominent operating systems for design professionals are Microsoft Windows and Apple OS X. For architects and landscape architects, Windows has traditionally been the dominant operating system because Autodesk AutoCAD runs exclusively in Windows. This is slowly changing as compatibility increases. Many offices work in either operating system and exchange information between them seamlessly.

OS X and Windows are different types of operating systems created by the companies Apple and Microsoft, respectively. OS X will only run on Apple hardware (laptops and workstations), but Windows will run on any compatible hardware including Apple hardware. This makes it possible to use Apple hardware to boot into either OS X or Windows when necessary. This method is accomplished by creating two separate partitions on the computer's hard drive and then choosing which system to boot into when restarting the computer. Either OS X or Windows must be chosen while booting up the computer; it is not possible to work in both systems simultaneously.

Another method for running an operating system is *virtualization*, which creates "virtual" hardware on which the operating system then runs. This allows an operating system such as OS X to host or virtualize an operating system such as Windows, which means both can run simultaneously and have access to similar resources. This is an ideal working situation, but it falls short on performance—specifically when using resource-intensive applications such as Photoshop or AutoCAD. Virtualization works best when using applications for word processing or project management, or when accessing the Windows partition in order to do quick edits in CAD.

## Applications

Applications represent a broad range of software created to accomplish specific tasks such as word processing, image editing, or financial management. When considering the representation of design drawings, typically we will use a range of applications to edit photos, create CAD linework, and build virtual models. Excluding applications for programming and word processing, the main types of applications designers will use are image editors (Photoshop, GIMP), vector editors (CAD, Illustrator), three-dimensional modelers (3ds Max, Maya, Blender), and video/motion graphics editors (Final Cut, Premiere, After Effects). Using each application, it is possible to find crossover or even repetition between the functions of one piece of software and another. For example, Photoshop and Illustrator share many of the same vector-editing tools to control pen paths. This crossover makes it easy to attempt to use one piece of software to accomplish everything, but it is important to understand the strengths and weaknesses of each application in order to efficiently use both pieces of software.

## Image Editing

Image-editing software refers to a broad range of applications that are used to manipulate pixels for tasks such as adjusting photographs, editing illustrations, and/or altering image sizes. Pixel-based imagery is also referred to as *raster images*. Image-editing applications typically use three paradigms that are specifically useful for design representation: layers, selections, and brushes. Methods that combine these three types of tools can typically perform all of the tasks necessary to manipulate pixels. Layers are used to organize pixels in order to edit specific pixels separately from other pixels, overlay pixels on top of one another, or apply effects or adjustments to specific layers.

Selections are used to select pixels on layers or multiple layers and can range from simple shapes, such as a square or circle, to complex shapes with multiple selection percentages per pixel. Selections can be made based on shape, the color or value of pixels, vector paths, and/or existing pixels on layers. A selection typically works as a range represented by a range of grays from unselected (0, black) to fully selected (255, white). This creates a selection using 256 values, so that edits or effects can be applied as a ramp or gradient. If an area is selected, it is possible to then edit those pixels. This creates an area in which to apply the edits based on the values in the selection. For example, if an image had a rectangular selection that was fully selected, then filling it with red would create a red rectangle. If the selection were rectangular but went from fully selected on the left to unselected on the right, then filling that rectangle would create a red box that slowly faded away from left to right.

Brushes are the third component that is typical in most image-editing applications, and they are used to apply or erase pixels. Brushes consist of a brush-tip shape and controls for the dynamics of how the tip creates a stroke. This allows brushes to carefully mimic real-world brushes or create all new brushes for specific needs. Brushes can apply a single color, a range of colors, or a pattern; they even transfer pixels from one side of an image to another. All of these tasks, either applying or erasing pixels, are accomplished with a selected brush, giving the artist many options to adjust the desired effect.

## Vector Editing

Vector-editing software refers to the use of points, lines, and shapes in order to represent imagery. To accomplish this, mathematical equations are used to represent the location of points, the direction of lines, and the fill-in to create shapes. There are two main types of vector-editing software for designers: illustration software such as Adobe Illustrator and drafting software such as Autodesk AutoCAD. The main difference between these types of software is that CAD applications focus on precision, and illustration applications focus on effects and appearance. In both types of vector-editing application, the tools and results are slowly becoming more congruent, as AutoCAD provides more tools to adjust style and appearance and Illustrator has third-party applications that increase the range of drafting tools.

The tools for vector-editing focus on selection, transformation, and stroke/fill manipulation. Selections are typically accomplished in three scales in vector-editing applications: multiple objects, single objects, or subobjects. A single object is typically defined as a series of points, lines, and fills that create an object such as a rectangle (four points, four strokes, and a fill). It is possible then to select a group of rectangles, the rectangle itself, or a single point or line. Once an object or subobject has been selected, it is then possible to transform the element with typical transformations such as move, rotate, or scale. Depending on the application, it may be possible to perform many other types of transformations with a variety of tools. Transformations can typically be applied interactively or by entering values for more precision.

## Three-Dimensional Modeling

Similar to vector-editing applications, three-dimensional modeling applications create wireframe representations of objects using points (vertices), edges, curves, and triangles. The most common type of three-dimensional modeling is polygon modeling, which creates representations of a model through a shell or surface. Other types of modeling include solid modeling, which creates accurate representations of an object's volume and is typically used in medical or engineering simulations. NURBS modeling, or nonuniform rational B-spline, creates surfaces from curves, creating precise freeform models. A fourth type of modeling is called subdivision modeling, which is similar to polygon modeling but uses a series of refinements on the initial mesh in order to create a smooth object. In most applications, each modeling type can be converted from more complex models, NURBS, and subdivisions to simpler polygon models.

Three-dimensional models are viewed in two ways: *real time* (allowing a user to move around the world interactively) and *rendered* (creating an image or animation with preselected lighting, materials, and movements). Real-time viewing typically occurs within the application viewport as the model is created or edited. It is also possible to create real-time models that can be explored in third-party viewers or applications and perform similar to first-person video games. Real-time viewing is ideal, but it is limited by the power of a computer's graphics card. In most cases, the graphics card cannot render the model, materials, and lighting at cinematic quality, which requires drawing 30 to 40 frames every second. A rendered view or animation is created from a three-dimensional scene after the models are built, materials are applied, and the animation is planned. The computer will then calculate the complex interaction between the light and objects with the ability to create extremely complex imagery. The user can choose to render a single image or a series of images in order to create an animation.

## Video Editing and Motion Graphics

Animations and movies require applications specifically suited to sequencing, modifying, and compositing a series of inline images. Two types of applications are specifically suited for this task: video-editing software such as Adobe Premiere or Apple Final Cut Pro, and motion-graphics software such as Adobe After Effects or Apple Motion. Both types of software have specific uses, but there are many overlaps between them. Video-editing software excels at placing clips and sound within a timeline in order to edit sequences and create transitions. There is a huge range of video-editing software from high-end professional packages such as the aforementioned Final Cut Pro to entry-level applications such as Apple iMovie or Windows Movie Maker. Many tasks can be accomplished in the entry-level software, but the output and refining process will be extremely limited.

Motion-graphics software excels at compositing or layering multiple images and movie clips within a timeline. Software such as After Effects can do basic movie-clip

sequencing, but the tools are typically limited compared to the professional video-editing software. Motion-graphics software uses layers and keyframing to animate layers of film, allowing the user to separate areas with masking. Most motion-graphics applications use a three-dimensional environment that makes it possible to build simple geometry with planes that can contain other movies or images. This creates a diorama-like environment that can be used to create film sets, special effects, or even complex animated diagrams.

## Workspace

The space that we create for ourselves when creating drawings is particularly important for designers. This is no different when we are working with digital media. The most important aspects of any workspace are efficiency and comfort. When working with digital media, we need to consider two workspaces: the physical as well as the virtual environment. There is no formula for what a workspace should be because it differs greatly for each individual. Some users prefer a space devoid of distractions, while others relish multiple activities occurring around them. Both types of spaces can provide creative inspiration for different individuals. Because we spend many hours working on drawings, the physical space we occupy must be comfortable for us as individuals. The space should provide room for a computer and all of the peripherals, as well as space for other design explorations such as drawing and modeling.

The computing environment consists of the operating system and application interface, as well as the input devices used to control them. Typically, the input devices will be a keyboard to enter commands and a mouse to interact with elements of the user interface or drawing. The best way to use this combination of devices is to keep one hand on the mouse and the other hand on the keyboard. Maintaining a consistent relationship between the position of the hands and the input devices allows the individual to quickly select hotkeys on the keyboard while maintaining the position of the cursor on the screen. This will allow an individual to look at the keyboard very little and maintain their focus on the screen in order to see feedback from the application.

When working in any application, the user will need to perform many repetitive tasks; therefore, it is important to minimize the amount of effort required to perform each task. If instantiating a command to draw a line requires the mouse to move up to a toolbar in order to select the Line tool, the designer will waste a good portion of their time simply moving the mouse away from the drawing area. If the designer needs to pick up a drafting pencil, draw a line, put down the pencil at the top of the drafting table, and then pick up the same pencil again in order to draw the next line, the extra step of putting down and picking up the pencil will add hours to the drafting time. However, this is what many users do when they use the applications, constantly clicking a button to draw a line.

The easiest way to speed up repetitive tasks is to use hotkeys or key combinations in order to instantiate commands. In an application like AutoCAD, every command can

be entered through the command line. In Photoshop, hotkeys exist for nearly every tool and menu item. It is possible to also create custom hotkeys for most applications, but depending on the working environment, it is advisable to use the defaults as much as possible. Using the defaults makes it much easier to use another computer that may not have the same hotkey customization. Depending on the user, it may be helpful to create a quick reference card in order to quickly see the default hotkeys for the application they are using. In most applications, the hotkeys will also be visible within the menus and as tooltips when the mouse rolls over a button. All designers should make it a priority to learn the hotkeys in order to efficiently use the application they are currently using.

Most applications are used for a range of design purposes. For example, Photoshop can be used to render a plan or adjust photographs. This means that there is a huge range of tools for many different purposes, and the interface can often get cluttered and hard to navigate. It is advisable to only turn on or display the features that are necessary in order to minimize the onscreen clutter. This will also give more space to the drawing area than to tools, palettes, and dialog boxes. It would be silly to put every pencil, marker, and paintbrush on the workdesk, and the same holds true in a virtual workspace. Open and display only the tools necessary to accomplish the job at hand. Depending on the application, it is usually possible to save multiple user interface configurations that can be customized for different tasks.

It is important to understand that specific hardware and software are not necessary to create beautiful digital drawings. Amazing work has been created by the humblest of applications and hardware, while thoughtless, poorly crafted work can just as easily come from the best applications running on a high-end workstation. The goal for designers is to find a combination of hardware and software that functions reliably and comfortably for a specific design environment and user.