

Index

A

Affine transform 302

AIX 153

Algorithms

 bilinear interpolation 300

 Gaussian kernel 227

 high-pass kernel 226

 Laplacian kernel 221

 low-pass kernel 215

 morphology 289

 nearest neighbor interpolation 298

 RGB to HSI 295

Sobel

 angle 292

 kernel 291

 magnitude 292

 thumbnail images 11, 227

 unsharp masking 304

apAlloc 31, 40

 assignment operator 42

clone() method 44

 definition 40

gNull() method 41

 memory access 43

sNull_ pointer 41

 template parameters 41

apAllocator_ 31, 36

 allocation 38

 constructor 36

 deallocation 39

 definition 36

explicit keyword usage 38

 memory alignment 37

 reference counting 37

apAllocatorBase_ 31

 allocation 32

 assignment operator 36

 conversion operators 33

 copy constructor 35

 deallocation 33

 definition 31

 memory alignment 34

 reference counting 34

apBString 175

apClampedTmpl 194

apDelegateInfo 230

apException base class 144

apFunction_s1d1 206

 template parameters 207

apFunction_s1s2d1 212

apImage 198

 caching 274

 image operators 201

storage operations 201
 template parameters 198
apImageIOBase 230
apImageIODelegateList 232
apImageStorage 185
apImageStorageBase 179
apImageStorageLocker 188
apIPP 240
apIPPFilter 242
apJPEG 234
apLimitInfo 193
apPixelIterator 185
apPoint 175
apRect 176
apRectImageStorage 180
 alignment options 181
 locking 181, 182
 pixel storage 180
apResourceMgr 166
apRGBTpl 191
apRowIterator 184
apTIFF 236
apUnitTestFunction 253
apUnitTestObject 254
 ASCII character encoding 164
 Aspect ratio 297
 Assertions 150
 NDEBUG symbol 150
 rules of use 151
 Assignment operator, safe 14
 Atomic operation 127

B
bad_alloc() 139
 Bandwidth 127
 Big endian 170
 Bilinear interpolation 299
 versus Nearest Neighbor 300
 Binary string class 88
 definition 89
 stream operators 175

Bitwise copy 44
 Boost Graph library 287
 Boundary conditions 216
 Byte-order mark (BOM) 170

C

Caching 274
 adding objects 276
 deleting objects 276
 images 280
 Call operator 243
 Casting
 const_cast<> 284
 C-Style 35
 dynamic_cast<> 152, 157
 reinterpret_cast<> 35
 static_cast<> 35
 Catching exceptions 137
 Character encoding 164
 Chinese language support 163
 Clamping values 29, 192
 Class design
 handle class idiom 52
 notations in diagrams 30
 ordering of functions 81
 Code quality 154
 Coding guidelines 71
 comments 77
 copyright information 79
 function ordering within classes 81
 handling restrictions 81
 header file issues 79
 indentation 76
 naming conventions 74
 Color images 5
 conversions 295
 HSI 5
 RGB 5
 Color wheel 5
 Command parser 110
 Comments 77

- Commercial software 2
 - Compiler issues 152
 - compatibility 81
 - errors 155
 - GNU gcc 153
 - Microsoft Visual C++ 154, 156
 - quality level 154
 - SGI Irix 154
 - Sun's native C++ 153
 - versus run-time 152
 - warnings 152
 - const** 283
 - returning a reference 285
 - const_cast<>** 284
 - Constructor 136
 - Conventions
 - coding guidelines 71
 - notations in class diagrams 30
 - used in this book *xiv*
 - Conversions
 - explicit** keyword usage 38, 281
 - operators 33
 - RGB to HSI 295
 - Convolution 214
 - convolve()** 220
 - kernel 214, 219, 288
 - Cooperative multithreaded 119
 - Coordinates 174
 - Copy constructor 27
 - Copy on write 271
 - copy()** 14
 - Copying objects 43
 - bitwise copy 44
 - image storage 189
 - Copyright information 79
 - Critical section 130
 - C-style error handling 148
- D**
- Deadlocks 119, 131
 - Debugging 92
 - connecting sinks to streams 100
 - controlling output levels 101
 - displaying data 99
 - flushing data 98
 - levels 103
 - object registry 104
 - sink 96
 - stream 94
- Delegates**
- file 229
 - image 237
- Destination pixels** 288
- Diagram notation** 30
- diff** 73
- Digital photographs** 302
 - algorithm 303
 - unsharp masking 302
- Dilation operator** 290
- Double-byte language support** 163
- Downcast** 157
- dynamic_cast<>** 157

E

- Edge
 - angle 221
 - strength 221
- Embedded systems 118
- Erosion operator 218, 290
- Error handling
 - compilers 155
 - memory allocation 139
 - processes 134
- Exception handling 135
 - apException** base class 144
 - avoiding overuse 146
 - catching 137
 - constructors/destructors 136
 - designing a framework 144
 - exception** class 136
 - rules of use 144
 - second-level catch handler 138

- throwing 135
- top-level catch handler 137
- when to use 149
- with locking/unlocking 188

Exception specifications 142

- Microsoft Windows 143
- run-time issues 157

explicit keyword usage 38, 281

explicit template instantiation 30, 283

F

Facet

see Locales

File delegate 229

apDelegateInfo 230

apImageIOBase 230

apImageIODelegateList 232

design 229

JPEG format 233

reading an image 230

TIFF format 236

writing an image 231

Files

saving/restoring strings 169

viewing differences (**diff**) 73

Filtering

see Image processing

FreeBSD 153

Fully preemptive multithreaded 119

Function objects 205

Function templates 29

G

Gaussian filter

image results 227

kernel 227

GB18030 character set 163

gcc compiler 153

geometry of image 297

gettext string handler 171

Global image functions 203

Globalization

see *Internationalization*

GNU gcc compiler 153

gNull() 41, 56, 64

Grayscale image 4

H

Handle class idiom 52

gNull() 56

handle object 55

rep object 56

Header files 79

Heap management

allocation error 140

apHeapMgr object declaration 140

error handler function 141

leaks via threads 126

High-pass filter

image results 226

kernel 226

Histogram 274, 278

HSI color space 5

Hue 5

I

ifstream 148

Image delegate 237

apIPP 240

fully-integrated solution 238

generic filter **apIPPFilter** 242

Intel IPP 237

loosely-coupled solution 239

specific filters via derivation 245

traits classes 240

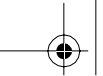
Image framework 1

Image processing 4

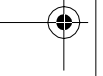
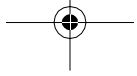
apFunction_s1d1 206

apFunction_s1s2d1 212

bilinear interpolation 299



- boundary conditions 216
- color space conversion 295
- convolution 214
- filtering 205
- Gaussian kernel 227
- high-pass kernel 226
- kernel 214
- Laplacian kernel 221
- Laplacian template specialization 223
- low-pass kernel 214
- mapping pixels 294
- morphology 289
- nearest neighbor interpolation 298
- neighborhood operators 214, 287, 288
- optimizations 264
- single source operations 206
- Sobel filter 291
- thumbnails 227
- transforms 294
- two source operations 212
- unsharp masking 302
- Images
 - 32-bit definition 48
 - 8-bit definition 48
 - aspect ratio 297
 - caching 274, 280
 - color 5
 - coordinates 174
 - derivation -vs- templates 50
 - digital photographs 302
 - filtering 205
 - geometry 297
 - global functions 203
 - grayscale 4
 - HSI 5
 - hue 5
 - intensity 6
 - intersection 205
 - memory alignment 180
 - noise elimination 214
 - null 13, 57
 - null rectangle 176
 - origin 4
 - performance optimizations 264
 - pitch 239
 - pixels 4
 - point 175
 - prototyping 47
 - rectangle 176
 - resizing 297
 - RGB 5, 58
 - definition 59
 - rotation 302
 - run-length encoded 179
 - saturation 6
 - sharpening digital photographs 302
 - sharpening edges 221, 226
 - size 4
 - smoothing edges 227
 - storage 61, 178, 181, 189, 201
 - thumbnail 9, 11, 47, 227
 - windows 180, 183
- Immutable 180
- Indentation 76
- Intel IPP 237
- Intensity 6
- Internationalization 163
 - ASCII 164
 - character encoding 164
 - checklist 164
 - Japanese language support 164
 - legacy character encodings 165
 - locales 171
 - resource manager 165
 - saving/restoring strings from files 169
 - Shift-JIS 164
 - string handling 165
 - Unicode 164
 - UTF-16
 - UTF-32 164
 - UTF-8 164
- Interprocess communication 118
- Intersection
 - images 205



rectangles 177

Iterators

apPixelIterator 185

apRowIterator 184

traits 185

J

Japanese language support 164

JPEG file delegate 233

K

Kernel

Convolution 214

Gaussian 227

high-pass 226

Laplacian 221

low-pass 214

neighborhood operations 287

Sobel 291

L

Laplacian filter

image results 222

kernel 221

laplacian3x3() 225

performance 225

Legacy character encodings 165

list class 106

Little endian 170

locale class 171

Locales 171

Localization

see Internationalization

Locking mechanism 128

exception-safe 188

Low-pass filter

average computation 215

image results 214

kernel 215

Luminance

see Intensity

M

Machine vision 5

Macros 112, 256

map class 106, 166, 276

Mapping pixels (transforms) 294

memcpy() 14

Memory allocation

alignment 34, 35

catching errors 139

copy on write 271

heap exhaustion 140

Memory allocation object 22

class hierarchy 31

requirements 22

STL solutions 23

Memory management 21, 271

Microsoft Foundation Classes (MFC) 74

Microsoft Visual C++ 154

Morphology 289

dilation 290

erodeCross() 291

erosion 290

structuring elements 289

Multilingual code

see Internationalization

Multithreaded applications

see Threads

mutable 131, 284

Mutex 130

N

Naming conventions

`_` suffix 75

enumerations 75

general 74

member variables 74

Microsoft Foundation Classes (MFC)

- 74
- static functions 75
- static variables 75
- template type arguments 76
- Tmp1** suffix 61
- NDEBUG** symbol 150
- Nearest neighbor interpolation 298
 - algorithm 298
 - image results 299
 - versus bilinear interpolation 300
- Neighborhood operations
 - destination pixels 288
 - source pixels 288
- Noise elimination 214
- Non-**const**
 - pass by reference 285
- Notations in class diagrams 30
- Null allocation 41
- Null image 13, 57, 64
- Null rectangle 176
- numeric_limits** class 193

O

- Object registry 104
 - base class 114
 - definition 112
 - design 105
 - remote access 104
- operator-** 195
- operator T*** 33
- operator()** 243
- operator+** 214
- operator<<** 175
- operator=** 14, 42
- operator>>** 91
- operator[]** 272
- Operators
 - images 201
 - neighborhood 214
- Optimizations
 - images 264

- performance 259
- return value 22
- Origin
 - image 4
 - kernel 288
- ostream** class 95
- OutputDebugString()** 99
- Overflow 29
- overflow_error** 135

P

- pair** class 175, 287
- Parsers
 - command 110
 - strings 109
 - XML 169
- Pass by reference 286
 - const** 285
 - returning non-**const** 285
 - template instantiation 286
- Passing by **const** reference 286
- Pels
 - see* Pixels
- Performance
 - checklist 261
 - general tuning guidelines 259
 - image optimizations 264
 - improvement techniques 261
 - results for laplacian filtering 225
 - run-time issues 158
 - template specialization 162
 - timing your code 268
 - virtual functions 158
- Persistence 175
- Photographs 302
- Pitch 239
- Pixels 4, 190
 - basic data types 190
 - overflow 54, 192
 - RGB 190
 - run-length encoding 179

- Point 175
 - POSIX 120
 - Processes 117, 133
 - error handling 134
 - threads vs. processes 133
 - Prototyping
 - advantages 45
 - handle class idiom 52, 55
 - handle object 55
 - rep object 56
 - image framework strategy 46
 - image objects 47
 - partitioning image storage 61
 - rules 44
 - templated image objects 52
 - Proxy classes 272
 - pthreads 120
- ## Q
- Quality level of production code 154
- ## R
- RAII 130, 136, 188
 - Rectangle 176
 - intersection 177
 - null 176
 - Reference counting 23
 - for memory allocation 24
 - thread-safe 133
 - References
 - passing by 286
 - passing by **const** 286
 - returning a **const** 285
 - returning a non-**const** 285
 - returning multiple values 286
 - template instantiation 286
 - reinterpret_cast<>** 35
 - Remote access via object registry 104
 - Resizing images 297
 - bilinear interpolation 299, 300
 - computing source pixels 298
 - nearest neighbor interpolation 298
 - Resource Acquisition Is Initialization 130, 136, 188
 - Resource manager 165
 - adding strings design 166
 - apResourceMgr** object 166
 - retrieving strings design 166
 - storing strings design 166
 - usage
 - accessing strings via ID 167
 - accessing strings via names 168
 - Return value optimization 22
 - Reusable code 83, 85
 - RGB images 58
 - color space 5
 - conversion to HSI 295
 - definition 59
 - RGB pixel data type 190
 - ROI 209
 - Rotating images 302
 - six degrees of freedom 302
 - RTTI 232, 242
 - Run-length encoding 179
 - Run-time issues 157
 - dynamic casting 157
 - exception specifications 157
 - performance 158, 162
 - template specialization 161
 - virtual functions 158
 - Run-Time Type Identification 232, 242
- ## S
- Saturation 6
 - Second-level catch handler 138
 - set** class 107
 - set_new_handler()** 139
 - set_unexpected()** 143
 - SGI native Irix C++ compiler 154
 - Shallow copy 44
 - Shared resources 124, 127

Sharpening image edges 221
Shift-JIS character encoding 164
Singleton object 56, 125, 140
Sink 96
Six degrees of freedom 302
Sobel filter 291
 angle 292
 image results 291
 kernel 291
 magnitude 292
 sobel13x3mag() 293
Software development cycle 3
Solaris 153
Source pixels 288
Specialization 27, 223
Standard library
 bad_alloc() 139
 copy() 14
 exception class 136
 ifstream class 148
 list class 106
 locale class 171
 map class 106, 166, 276
 numeric_limits class 193
 ostream class 95
 overflow_error class 135
 pair class 175, 287
 set class 107
 set_new_handler() 139
 set_unexpected() 143
 string class 34, 282
 terminate() 135
 vector class 106, 254
static_cast<> 35
std::
 see Standard library
Storage
 apImage operations 201
 for images 61, 189
Stream
 binary 89
 debugging 94

locale 171
string class 34, 282
String handling 165
Strings
 binary class 88, 175
 generic parsing routine 109
 gettext 171
 string class 34, 282
Structuring elements 289
Sun's native C++ compiler 153
Synchronization
 deadlocks 131
 mutable 131
 threads 126

T

Tagged data 89
Templates
 class vs. **typename** 25
 compiler compatibility 81
 converting a class to use 24
 copy constructor syntax 27
 default arguments 26
 explicit instantiation 30, 283
 final image class design 198
 function 29
 function template specialization 29
 inline definition 27
 instantiation by reference 286
 non-inline definition 27
 specialization 27, 161, 223
 type argument names 25
terminate() 135
Test application 9
 Image class 10, 12
 Thumbnail class 10
Third-party software
 file delegates 229
 image delegates 237
 interfaces to 229
Threads 117, 118

- cooperative multithreaded 119
- critical section 130
- deadlocks 119, 131
- encapsulating 125
- fully preemptive multithreaded 119
- heap leaks 126
- locking 128
- Microsoft Win32
 - apThread** class 120
 - lock/unlock 131
 - start()** 122
 - stop()** 122
- mutex 130
- POSIX 120
- pthreads 120
- scoping issues 125
- shared resources 124, 127
- switching between 124
- synchronization 126, 127
- thread-safe 126, 133
- UNIX
 - lock/unlock 130
 - start()** 122
- Throwing exceptions 135
- Thumbnail image 9, 11, 47
- Thumbnail method
 - in final design 228
 - in Prototype 1 50
 - in Prototype 2 57
 - in Prototype 3 67
- TIFF file delegate 236
- Timing your code 268
- Top-level catch handler 137
- Traits 185, 240
- Transforms 294
 - affine 302
 - converting between color spaces 295
 - mapping pixels 294
- Tuning performance 259
 - see also* Performance
- typeid()** 113, 189, 232
- typename** 25

U

- Underflow 29
- Unicode 164
- Unit tests 249
 - apUnitTestFunction** base class 253
 - apUnitTestObject** list 254
 - extending framework 258
 - framework design 252
 - framework overview 250
 - macros 256
- UNIX 120
- Unsharp masking 302
 - general algorithm 303
 - image results 303
 - kernel 304
 - unsharpMask()** 305
- UTF-16 character encoding
- UTF-32 character encoding 164
- UTF-8 character encoding 164

V

- vector** class 106, 254
- Viewing changes to files 73
- Virtual functions 158

W

- Warnings 152
- Wide character
 - see* UTF-16
- Windowing 180, 183

X

- XML parser 169