

補足

簡単なデバッグテクニック

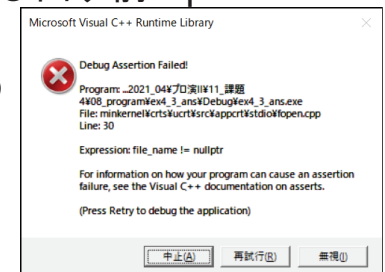
- 怪しい処理行の前にブレークポイントを置き、ステップ実行(1行ずつ実行)
 - どの処理に分岐するか一目でわかる
 - 止まらない場合は、処理がそこまで到達していない
- (全環境共通) 適当なprintf文を大量に設置する
 - コマンドプロンプトに表示される文字から、どこまで処理されたか一目でわかる
 - かなりいいかげんな方法であり、ステップ実行が難しい環境などで有用

より良い例外処理

- ユーザが不適切な方法でプログラムを実行した場合でも安全に終了させるための工夫
 - argcをチェックし、こちらが意図した分だけコマンドライン引数を与えられているか
 - 入力/出力ファイルは正しくopenできているか
 - モードの書式は正しいか
- 不適切な場合、エラーメッセージを出力するとともにmainで1をreturnして終了するとよい

よくある実行時エラー

- エラー文を読むことで解決することが多い
- よくあるケース
 - コマンドライン引数を与えずに実行
 - 動的確保/解放のコードに間違い
 - `ImageMallocImageBuffer`関数を呼び出す前に`pBuffer`に画素値を書き込み/読み込み
 - 画像処理関数において、配列の範囲外にアクセス



画像処理テクニックの例(1)

- 回転, 反転
 - コピー先のインデクスを変える(回転時はサイズ注意)
- モノクロ化
 - 各画素でRGB値を同じ値にする
- 拡大/縮小
 - 拡大: ある画素の値を複数の画素にコピー
 - 縮小: 特定の位置の画素はコピーし, 他は切り捨て
- モザイク化
 - 出力画像の $n \times n$ 画素を入力画像の同範囲の平均値で置き換え

5

バイナリエディタ(1/2)

- 画像ファイルやテキストファイルのバイナリ情報が閲覧できる
- **これを利用すれば減色処理の正否がわかる**
- Visual Studioの場合, ファイル→開く→ファイル... →対象のファイルを選択する→「開く」の下矢印→「ファイルを開くアプリケーションを選択」→「バイナリエディター」を選択

7

画像処理テクニックの例(2)

- 特定の色の物体を検出
 - RGB値がある範囲内の画素値はそのまま, 他は0で置き換える
- ノイズ付加
 - ある確率分布に従う乱数を生成し, 画素値に付加
- ノイズ除去
 - 付加されたノイズによって対処方法が異なる
 - 平均値フィルタやメディアンフィルタなど
- ぼかし処理, 輪郭強調, エッジ検出
 - 講義資料webページの補足説明を参照

6

バイナリエディタ(2/2)

- バイナリデータと位置は16進数表記
- 右側にASCIIコードに従ったテキストも表示

		1バイト×16個の 数値(バイナリ)データ																対応する文字 (対応する文字が なければドット)		
1行に16バイト 分を表示	img01.ppm	#	x																	
	00000000	60	36	0A	23	20	43	72	65	61	74	65	64	20	62	78	20	P6.# Created by		
	00000010	49	72	86	61	6E	56	69	65	77	0A	32	35	36	20	31	39	IrfanView.256 19		
	00000020	32	04	32	35	0A	1A	25	29	09	14	1A	00	0A	0F	0B	22	2,255...%)		
	00000030	1C	23	07	1B	22	00	16	1E	00	19	20	17	32	39	22	42	.#...29"B		
	00000040	4D	27	50	64	23	5A	79	20	64	91	2D	79	AD	39	86	BE	M'PdZy d..y.9..		
	00000050	39	82	B9	34	7B	B1	33	88	BE	33	84	B9	33	7F	B0	38	9..4[.3..3..8		
	00000060	7F	AD	3F	83	B0	3F	83	B2	3B	81	B2	39	81	B3	3B	83	...?...9...		
	00000070	B3	3B	83	B3	3B	83	B3	3C	84	B4	3C	84	B4	3D	85	B5	...?...<...<...		
	00000080	3D	85	B5	3E	86	B6	40	84	B5	40	B4	B5	41	85	B6	41	=>...@..@..A		
00000090	85	B6	42	86	B7	42	86	B7	43	87	B8	43	87	B8	43	87	...B...C...C...			
000000a0	B8	43	87	B8	43	87	B8	44	88	B9	44	88	B9	44	88	B9	...C...C...D...D...			
000000b0	44	88	B9	44	88	B9	45	89	B8	45	89	B8	46	8A	B9	46	D...D...E...E...F			
000000c0	8A	B9	46	8A	B9	46	8A	B9	47	8B	BA	47	8B	BA	3D	8D	...F...G...G...=.			