

Trabalho 2.2

Linguagem de Montagem

Opção 2

Lucas Frank Hollmann

Neste trabalho foram feitos alguns testes envolvendo manipulação de imagens em C e Assembler. O objetivo é comparar o tempo de execução do código padrão em C com os códigos otimizados em Assembler, com a utilização das instruções SSE.

Os testes realizados foram feitos em imagens de vários tamanhos diferentes e consistiram em aumentar o brilho nessas imagens e gerar um ruído aleatório.

Os resultados obtidos podem ser vistos nas tabelas 1 e 2.

Tabela 1 Resultado das manipulações feitas na imagem de 32x32 pixels.

Imagem original	Brilho aumentado em C	Brilho aumentado em Assembler	Ruído gerado em C	Ruído gerado em Assembler
				

Tabela 2 Tempo gasto para executar cada manipulação (tempo em ns)

	Brilho (C)	Brilho (asm)	Ruído (C)	Ruído (asm)
32x32 pixels	55284	29931	356994	293106
640x480 pixels	11203421	6733822	14559204	18419512
1280x960 pixels	12541633	7510352	53161908	51612926
1600x1200 pixels	20241581	11003528	80626813	74586284
2560x1920 pixels	47067304	23338528	205846213	198732801

Com esses resultados é possível notar uma grande melhoria de eficiência entre o aumento de brilho feito em C e o feito em Assembler, pois o algoritmo em Assembler, ao utilizar as instruções SSE, permite somar quatro dados de dois vetores por vez, dessa forma, diminuindo o tempo necessário para essa operação. Já ao gerar um ruído aleatório, não se vê uma melhora tão grande pois é necessário criar um vetor aleatório previamente, para somar

ao vetor original. Uma opção seria manter um ou mais vetores fixos, dessa forma, poupando o tempo de geração do vetor aleatório, porém isso acabaria com a aleatoriedade do ruído.