



## Prática de Laboratório Extra 01

Prof. Alexandre Zaghetto

[zaghetto@unb.br](mailto:zaghetto@unb.br)

No presente trabalho vamos explorar os princípios do **Full Search Algorithm**, um dos algoritmos de busca utilizados em codificadores de vídeo. Para isso:

- a) Declare duas matrizes **F<sub>0</sub>** e **F<sub>1</sub>** **10x10** iniciadas com os valores que estão no arquivo quadros.txt.

212	200	191	204	233	222	182	197	196	200
196	201	219	239	238	241	218	174	185	196
201	184	224	240	233	232	234	205	175	188
202	214	192	150	131	124	126	169	196	185
208	174	115	101	98	101	91	180	178	194
112	100	116	120	110	125	106	178	212	186
146	75	114	145	126	128	124	102	149	210
198	125	94	134	123	110	125	93	83	131
209	174	117	102	143	152	121	85	87	93
209	201	162	104	91	123	114	96	90	110

**F<sub>0</sub>**

217	216	207	194	216	227	191	203	203	202
192	186	196	221	239	240	213	173	187	193
200	199	193	234	243	241	243	197	174	193
191	186	209	199	156	137	139	177	189	179
214	218	175	111	99	101	91	155	183	191
174	114	110	116	107	113	102	186	199	180
199	109	80	133	139	129	128	118	185	204
209	178	96	109	137	106	112	94	97	176
207	203	146	102	125	143	128	92	83	97
211	207	191	130	93	124	122	89	93	107

**F<sub>1</sub>**

- b) Escreva uma função que a partir de dois vetores **V<sub>1</sub>** e **V<sub>2</sub>** de 4 posições calcula a SAD (*Sum of Absolute Differences*) desses dois vetores. A SAD é definida da seguinte maneira:

$$SAD = \sum_{i=0}^3 |V1[i] - V2[i]|.$$

- c) Defina uma *struct* chamada **movimento**, contendo duas variáveis membro inteiras **i** e **j**.
- d) Declare um vetor **mov** de 25 posições do tipo **struct movimento**.
- e) Considere agora que a matriz **F<sub>1</sub>** é composta por blocos de **2x2**. Para cada um dos 25 blocos 2x2 de **F<sub>1</sub>** procure em **F<sub>0</sub>** o bloco mais parecido, utilizando deslocamentos unitários. O casamento que apresentar a menor SAD indica maior similaridade.



212	200	191	204	233	222	182	197	196	200	217	216	207	194	216	227	191	203	203	202
196	201	219	239	238	241	218	174	185	196	192	186	196	221	239	240	213	173	187	193
201	184	224	240	233	232	234	205	175	188	200	199	193	234	243	241	243	197	174	193
202	214	192	150	131	124	126	169	196	185	191	186	209	199	156	137	139	177	189	179
208	174	115	101	98	101	91	180	178	194	214	218	175	111	99	101	91	155	183	191
112	100	116	120	110	125	106	178	212	186	174	114	110	116	107	113	102	186	199	180
146	75	114	145	126	128	124	102	149	210	199	109	80	133	139	129	128	118	185	204
198	125	94	134	123	110	125	93	83	131	209	178	96	109	137	106	112	94	97	176
209	174	117	102	143	152	121	85	87	93	207	203	146	102	125	143	128	92	83	97
209	201	162	104	91	123	114	96	90	110	211	207	191	130	93	124	122	89	93	107

$F_0$   $F_1$

- f) Construa uma matriz auxiliar **SAD\_AUX** de **9x9** elementos reais que irá armazenar as SADs calculadas para o bloco  $k$  de  $F_1$  que está sendo atualmente procurado em  $F_0$ .
- g) Procure a menor SAD dentro da matriz **SAD\_AUX** e guarde a linha e a coluna referente a menor SAD em **mov[k].i** e **mov[k].j**, respectivamente
- h) Gere uma matriz **RESIDUO 10x10** que irá armazenar a diferença entre cada bloco de  $F_1$  e o bloco de  $F_0$  que resultou na menor SAD. Mostre a matriz **RESIDUO** na tela do computador.