CUDA Acceleration of Color Histogram Matching

Antonio S. Montemayor, Raúl Cabido, Juan José Pantrigo
Universidad Rey Juan Carlos, SPAIN

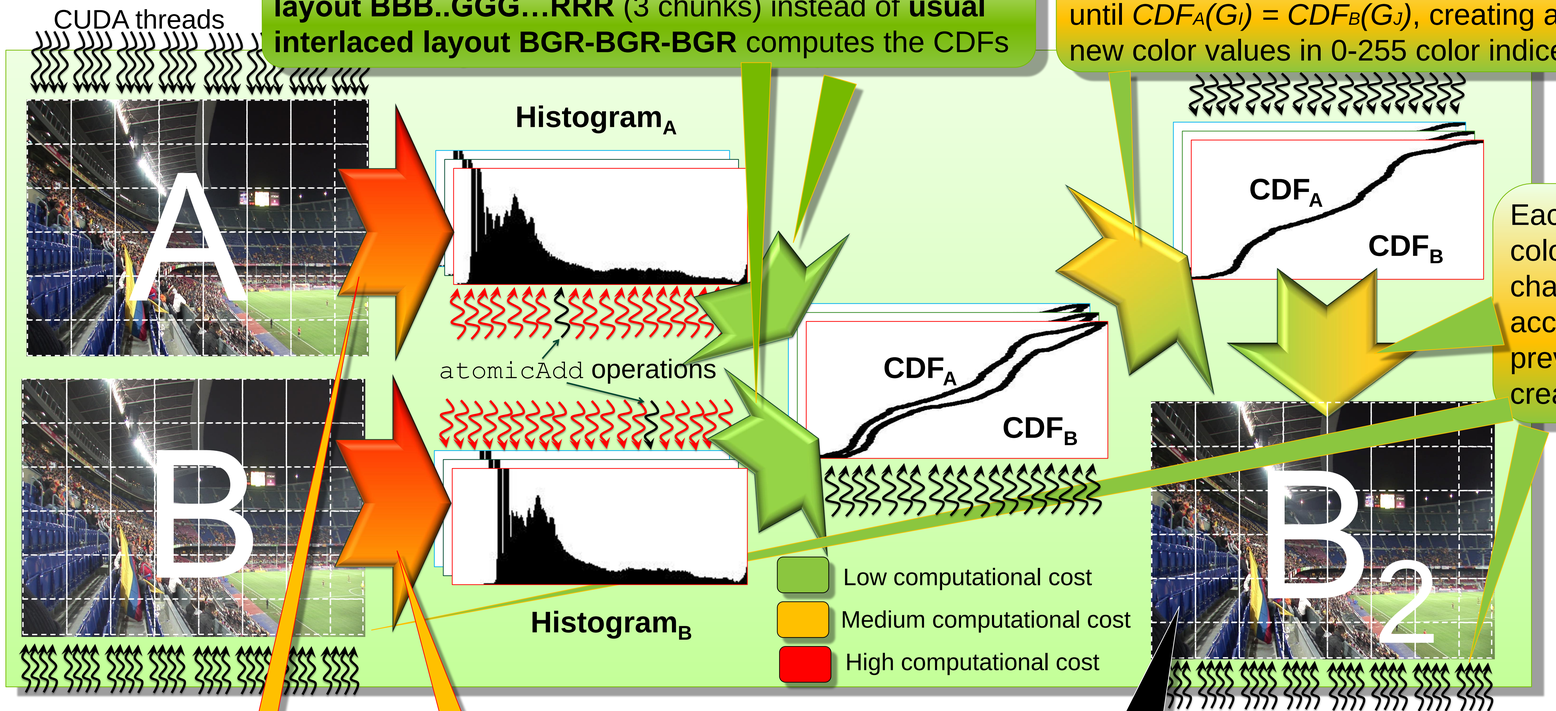
Xavi Rodríguez, Sergi Sàgas
Mediapro Research, SPAIN

A common approach to histogram matching is done by means of the cumulative distribution functions (CDFs). First, we calculate their normalized histograms (h_A and h_B) and their respective cumulative histogram distribution functions (CDF_A and CDF_B). Then, a matching between the CDFs are performed. Given the reference CDF, CDF_A, for each gray level G_I we find the corresponding gray level G_J in which $CDF_A(G_I) = CDF_B(G_J)$, if I and J correspond to different values then a replacement of a gray level in the image I_B is performed in order to match their CDFs. Our approach considers the ideas of [Rolland et al. 2000] with a Nvidia 3D broadcast solution system using professional HD cameras.

ROLLAND, J. P., VO, V., BLOSS, B., AND ABBEY, C. 2000. Fast algorithms for histogram matching: Application to texture synthesis. *Journal of Electronic Imaging 9(1).*

CUDPP `cudppMultiScan` function **in reordered layout BBB..GGG…RRR** (3 chunks) instead of **usual interlaced layout BGR-BGR-BGR** computes the CDFs

CDFs are **monotonically nondecreasing** functions. Each thread in CDF_A evaluates CDF_B until $CDF_A(G_I) = CDF_B(G_J)$, creating a LUT with new color values in 0-255 color indices.



Each original color in B(x,y) is changed according to the previous LUT, creating B_2(x,y)

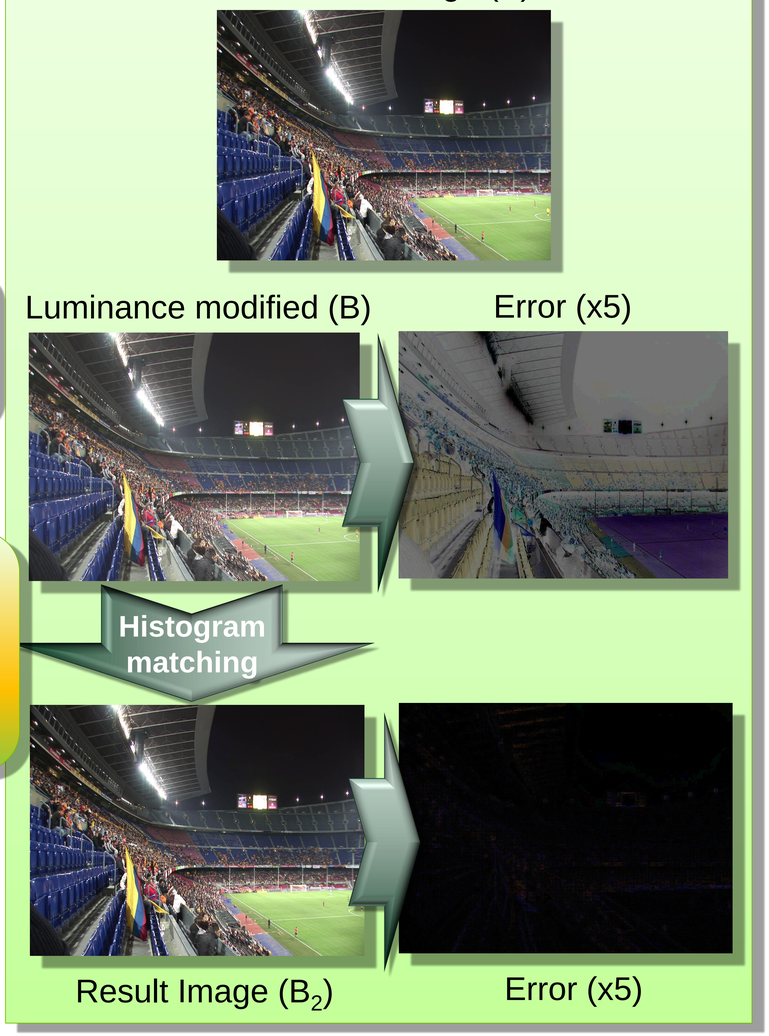


Image histograms are not very prone to parallelization because of the implicit direct access to the distribution container which can lead to race conditions. However, with the recent advances of the CUDA platform we can exploit the **atomic operations in CUDA shared memory** supported by compute capability 1.2 devices.

Atomic operations for 1 color channel histogram

in CUDA **global memory** → 143ms (1440x1080, GTX260)

in CUDA **shared memory** → 7.1ms (1440x1080, GTX260)

Each color channel histogram (256 values) **packed in 1 CUDA block** → 256 threads/block (<512)

Performance	1440x1080	1920x1080 (HD)	3648x2736
CPU (Intel Core2 Duo 3GHz)	33 ms	49 ms	207 ms
GPU (GeForce GTX260)	19 ms	29 ms	104 ms
GPU (GeForce GTX480)	8.9 ms	9.5 ms	52 ms

CONCLUSION

Histogram techniques are not very parallel friendly, however we found very useful the 1.2 compute capability feature of atomic operations on CUDA shared memory to improve about x20 the performance of histogram computation on GPU (compared to global memory usage). Overall, for the **histogram matching problem, we get about x5.2 performance improvement** compared to CPU execution enabling **real time (>30 fps) processing on HD imagery** (1920x1080) for possible 3D content creation and accurate depth estimation.