**Documentație – P2: CUDA**

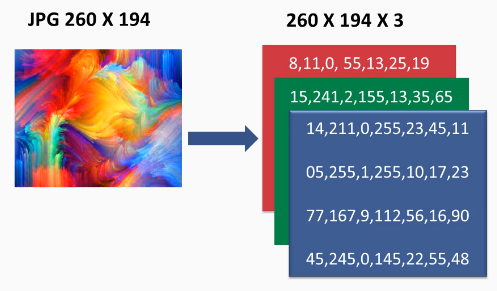
* **Cerinta**

***Filtru Gaussian pe imagini, folosind CUDA ca si tehnologie de paralelizare***

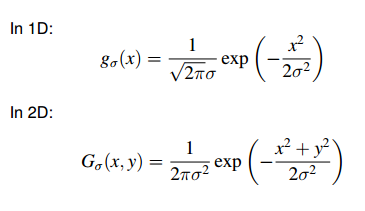
* **Despre CUDA**

*CUDA* este platforma de calcul paralel dezvoltată de *nVidia*, cu scopul de a accelera operaţiile de calcul si reducând sarcina de lucru pentru CPU în anumite aplicaţii, folosind puterea de calcul disponibilă în procesoarele grafice (GPU). De această tehnologie pot beneficia cel mai mult aplicaţiile care au de procesat în mod similar multiple seturi de date, precum: procesare imagini şi video, calcule în domeniul biologiei sau chimiei, simularea dinamicii fluidelor, analiza seismelor şi multe altele.

* **Proiectare:**
* Programul primeste ca si input o imagine, apoi extrage din matricea 3D a acesteia vectorii specifici fiecarui channel de culoare.

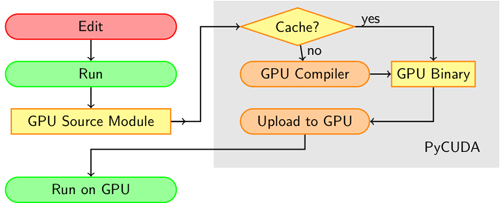


* Se construieste kernelul gaussian, dupa formula de mai jos.

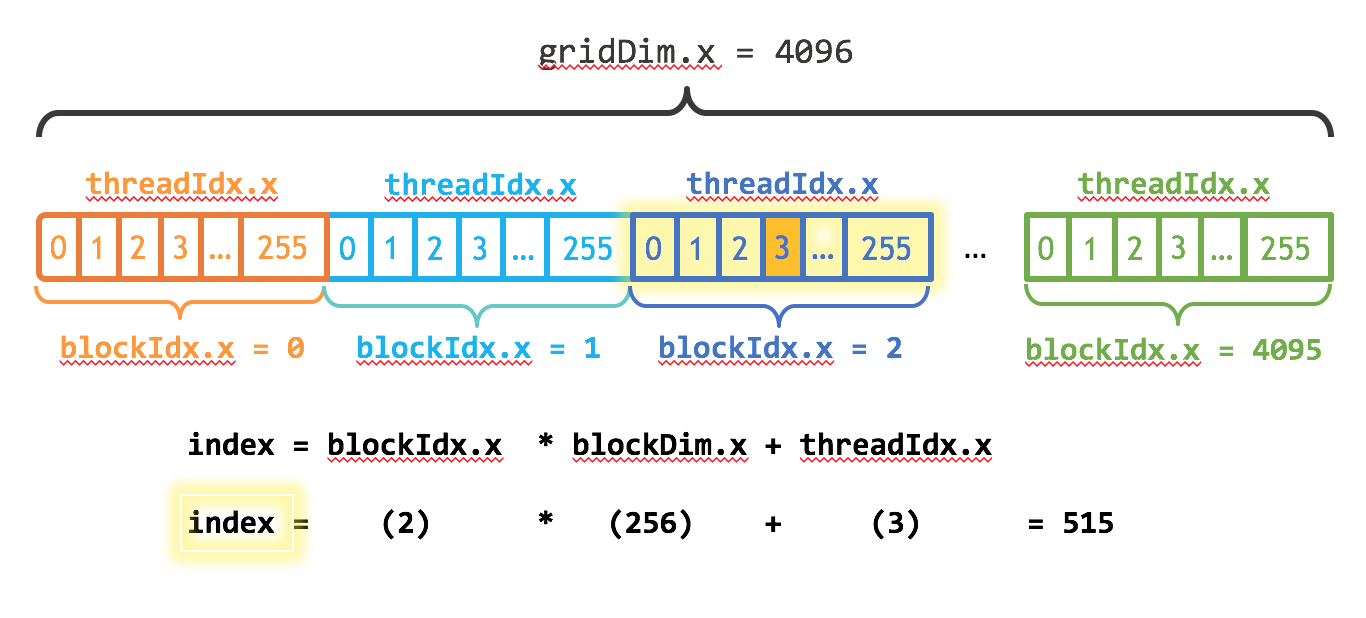
****, unde x,y – pozitiile din matrice ale valorii curent,

σ – deviatia standard

* Se calculeaza gridurile CUDA in raport cu dimensiunile imaginii.
* Cu ajutorul functiei *compiler.SourceModule*, o functie din libraria *PyCUDA*, se compileaza si se incarca in memorie codul CUDA din fisierul ***gaussian\_blur.cu***, care va fi executat pe GPU cand functia va fi apelata. *SourceModule* pastreaza o referinta spre memoria din GPU unde este incarcat codul compilat, iar in acest fel, GPU poate executa codul si CPU poate administra memoria GPU.



* Se apeleaza metoda *applyFilter* din codul CUDA pe cele 3 channel uri de culoare. Kernelul *se executa in paralel de mai multe thread-uri CUDA*, fiecare dintre acestea procesand *cate un pixel* al imaginii. ***threadIdx*** si ***blockIdx*** sunt variabile care calculeaza linia si coloana din imagine ce este procesata de thread. Cu ajutorul lor, kernelul calculeaza valoarea blur a fiecarui pixel.



* Se reconstruieste matricea RGB 3D si se returneaza imaginea finala, adica imaginea data initial asupra careia s-a aplicat filtrul gaussian (blur).
* **Rezultate**

****

***Input*** *(imaginea initiala)*

****

***Output*** *(imaginea cu filtru gaussian)*