Bench report : CPUvsGPUsm20V2 (28-03-2012-16-40)

Wednesday 28th March, 2012

Contents

1 Introduction

For a bivariate random sample $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$ drawn from a density f, the kernel density estimate is defined by

$$\mathcal{F}_{\boldsymbol{X}}(\boldsymbol{x}; \mathbb{H}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{K}_{\mathbb{H}}(\boldsymbol{x} - \boldsymbol{X}_i)$$

where $\boldsymbol{x} = (x_1, x_2)^t$ and $\boldsymbol{X}_i = (X_{i,1}, X_{i,2})^t$, $\forall i \in [\![1, N]\!]$.

Here $\mathcal{K}_{\mathbb{H}}(\boldsymbol{x}) = |\mathbb{H}|^{-1/2} \mathcal{K}(\mathbb{H}^{-1/2}\boldsymbol{x})$ and $\mathcal{K}$ is the kernel which is a symmetric probability density function, $\mathbb{H}$ is the bandwidth matrix which is symmetric and positive-definite. We take $\mathcal{K}(\boldsymbol{x}) = \frac{1}{2\pi} \exp(-\frac{1}{2}\langle \boldsymbol{x}, \boldsymbol{x} \rangle)$ the standard normal density.

The choice of bandwidth matrix $\mathbb{H}$ is crucial in determining the performance of $\mathcal{F}_{\boldsymbol{X}}$. We use botev function `kde2d` to obtain *optimal* diagonal bandwidth matrix given by

$$\mathbb{H}^{[\text{botev}]} = \begin{pmatrix} h_1^{[\text{botev}]} & 0 \\ 0 & h_2^{[\text{botev}]} \end{pmatrix}$$

where $h_1^{[\text{botev}]} > 0$ and $h_2^{[\text{botev}]} > 0$

Let $\boldsymbol{x}_1 = (x_{1,1}, \ldots, x_{1,n_1})^t \in \mathbb{R}^{n_1}$ and $\boldsymbol{x}_2 = (x_{2,1}, \ldots, x_{2,n_2})^t \in \mathbb{R}^{n_2}$. We note $\boldsymbol{Z} = (Z_1, \ldots, Z_N)^t$ and $\mathbb{F} = \begin{pmatrix} f_{1,1} & \cdots & f_{1,n_2} \\ \vdots & \ddots & \vdots \\ f_{n_1,1} & \cdots & f_{n_1,n_2} \end{pmatrix}$ defined by

$$Z_j = \mathcal{F}_{\boldsymbol{X}}(\boldsymbol{X}_j; \mathbb{H}^{[\text{botev}]}), \ \forall j \in [\![1, N]\!] \tag{1}$$

and

$$f_{i,j} = \mathcal{F}_{\boldsymbol{X}}((x_{1,i}, x_{2,j})^t; \mathbb{H}^{[\text{botev}]}), \ \forall i \in [\![1, n_1]\!], \ \forall j \in [\![1, n_2]\!] \tag{2}$$

1.1 Z computation with `kde2D` function

We want to compare several versions of `kde2D` function implementation in Matlab.

`kde2D_OptV0` : F. Cuvelier code without Toolbox.

`kde2D_OptV1` : N. Belaribi code using `mvnpdf` function from Matlab Statistics Toolbox.

`kde2D_GpuV0` : F. Cuvelier code using GPU under Matlab (version 0)

`kde2D_GpuV1` : F. Cuvelier code using GPU under Matlab (version 1)

1.2 $\mathbb{F}$ computation with `ksdensity2D` function

We want to compare several versions of `ksdensity2D` function implementation in Matlab :

`ksdensity2D_OptV0` : F. Cuvelier code without Toolbox.

`ksdensity2D_OptV1` : N. Belaribi code using `mvnpdf` function from Matlab Statistics Toolbox.

`ksdensity2D_GpuV0` : F. Cuvelier code using GPU under Matlab (version 0)

`ksdensity2D_GpuV1` : F. Cuvelier code using GPU under Matlab (version 1)

1.3 Bivariate distribution

Let $a_1 < b_1$ and $a_2 < b_2$. We note $\mathcal{U}([a_1, b_1] \times [a_2, b_2])$ the bivariate uniform distribution over $[a_1, b_1] \times [a_2, b_2]$.

Let $\boldsymbol{m} \in \mathbb{R}^2$ and $\mathbb{S} \in \mathcal{M}_{2,2}(\mathbb{R})$ a symmetric positive defined matrix. We note $\mathcal{N}(\boldsymbol{m}, \mathbb{S})$ the bivariate normal distribution with specified mean vector $\boldsymbol{m}$ and covariance matrix $\mathbb{S}$. The density of $\mathcal{N}(\boldsymbol{m}, \mathbb{S})$ is given by

$$f(\boldsymbol{x}; \boldsymbol{m}, \mathbb{S}) = \frac{1}{2\pi \det(\mathbb{S})^{1/2}} \exp(-\frac{1}{2} \langle \mathbb{S}^{-1}(\boldsymbol{x} - \boldsymbol{m}), \boldsymbol{x} - \boldsymbol{m} \rangle)$$

$$\begin{aligned} \mathcal{D}_1 &= 0.50\,\mathcal{U}([-1.00, 1.00] \times [-1.00, 1.00]) + 0.25\,\mathcal{N}\left(\begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}; \begin{bmatrix} 0.30 & 0.00 \\ 0.00 & 0.70 \end{bmatrix}\right) \\ &\quad + 0.25\,\mathcal{N}\left(\begin{bmatrix} 0.50 \\ -0.50 \end{bmatrix}; \begin{bmatrix} 0.10 & 0.00 \\ 0.00 & 0.30 \end{bmatrix}\right) \end{aligned}$$

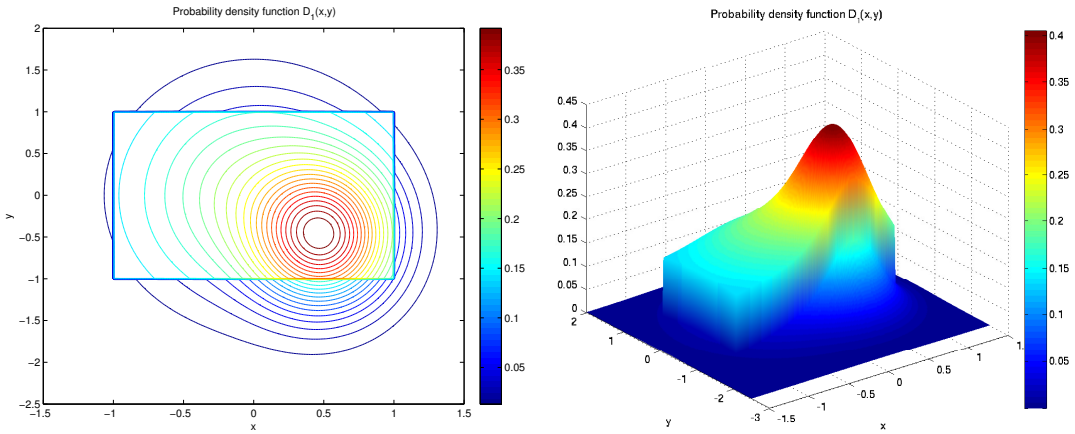

Figure 1: Probability density function $\mathcal{D}_1$

$$\begin{aligned} \mathcal{D}_4 &= 0.33\,\mathcal{N}\left(\begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}; \begin{bmatrix} 0.10 & 0.00 \\ 0.00 & 0.10 \end{bmatrix}\right) + 0.17\,\mathcal{N}\left(\begin{bmatrix} 0.50 \\ 0.50 \end{bmatrix}; \begin{bmatrix} 0.05 & 0.00 \\ 0.00 & 0.05 \end{bmatrix}\right) \\ &\quad + 0.17\,\mathcal{N}\left(\begin{bmatrix} 0.50 \\ -0.50 \end{bmatrix}; \begin{bmatrix} 0.05 & 0.00 \\ 0.00 & 0.05 \end{bmatrix}\right) + 0.17\,\mathcal{N}\left(\begin{bmatrix} -0.50 \\ 0.50 \end{bmatrix}; \begin{bmatrix} 0.05 & 0.00 \\ 0.00 & 0.05 \end{bmatrix}\right) \\ &\quad + 0.17\,\mathcal{N}\left(\begin{bmatrix} -0.50 \\ -0.50 \end{bmatrix}; \begin{bmatrix} 0.05 & 0.00 \\ 0.00 & 0.05 \end{bmatrix}\right) \end{aligned}$$

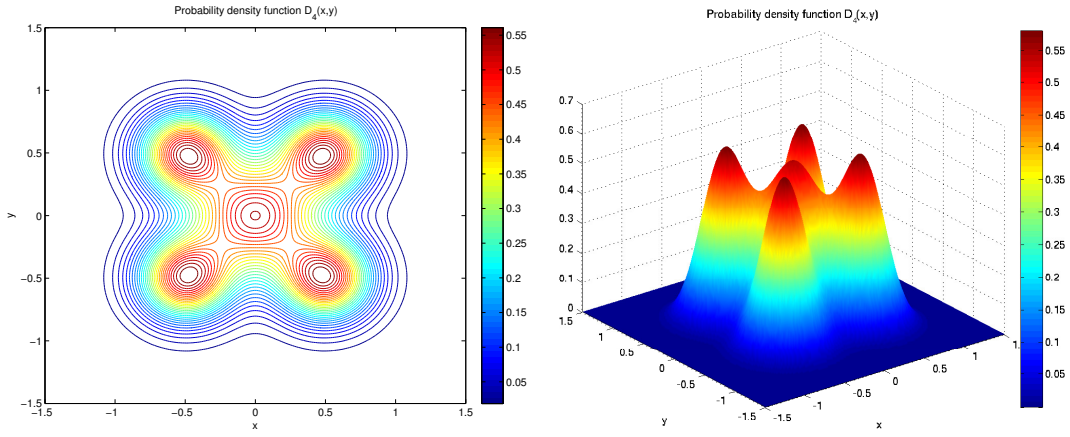

Figure 2: Probability density function $\mathcal{D}_4$

$$\begin{aligned}\mathcal{D}_7 &= 0.36\,\mathcal{N}\left(\begin{bmatrix}-2.00\\2.00\end{bmatrix};\begin{bmatrix}1.00 & 0.00\\0.00 & 1.00\end{bmatrix}\right) + 0.27\,\mathcal{N}\left(\begin{bmatrix}0.00\\0.00\end{bmatrix};\begin{bmatrix}0.80 & -0.72\\-0.72 & 0.80\end{bmatrix}\right)\\ &\quad +0.36\,\mathcal{N}\left(\begin{bmatrix}2.00\\-2.00\end{bmatrix};\begin{bmatrix}1.00 & 0.00\\0.00 & 1.00\end{bmatrix}\right)\end{aligned}$$

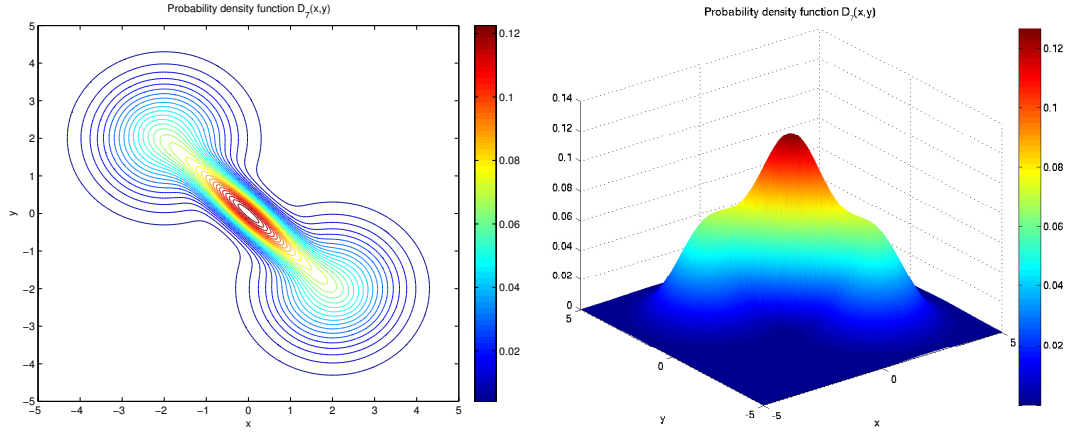

Figure 3: Probability density function $\mathcal{D}_7$

2 Computer : gpuschwarz

Processors : 2 x Intel Xeon E5645(6 cores) at 2,40Ghz .

Graphic Devices : 1x Nvidia GTX 590 (bi GPU)

RAM : 32Go

3 benchs : kde2D

3.1 Probability density : $\mathcal{D}_1$

N	OptV1	OptV0	GpuV0 (sm_20)	GpuV1 (sm_20)
10000	7.699 (s) x 1.00	5.057 (s) x 1.52	0.014 (s) x 566.80	0.016 (s) x 473.84
20000	34.282 (s) x 1.00	16.514 (s) x 2.08	0.026 (s) x 1310.81	0.031 (s) x 1121.63
30000	69.377 (s) x 1.00	34.106 (s) x 2.03	0.046 (s) x 1494.17	0.052 (s) x 1322.25
40000	128.923 (s) x 1.00	44.468 (s) x 2.90	0.075 (s) x 1710.69	0.082 (s) x 1575.11
50000	183.835 (s) x 1.00	77.932 (s) x 2.36	0.117 (s) x 1572.77	0.118 (s) x 1552.49
60000	251.334 (s) x 1.00	120.373 (s) x 2.09	0.156 (s) x 1611.44	0.165 (s) x 1526.53
70000	423.733 (s) x 1.00	144.685 (s) x 2.93	0.211 (s) x 2010.34	0.214 (s) x 1977.69
80000	553.681 (s) x 1.00	174.053 (s) x 3.18	0.268 (s) x 2064.89	0.271 (s) x 2040.90
90000	679.743 (s) x 1.00	278.621 (s) x 2.44	0.332 (s) x 2046.77	0.337 (s) x 2016.19
100000	871.698 (s) x 1.00	328.151 (s) x 2.66	0.423 (s) x 2062.43	0.414 (s) x 2107.89

Table 1: kde2D - Probability Density : $\mathcal{D}_1$ - time and speedup

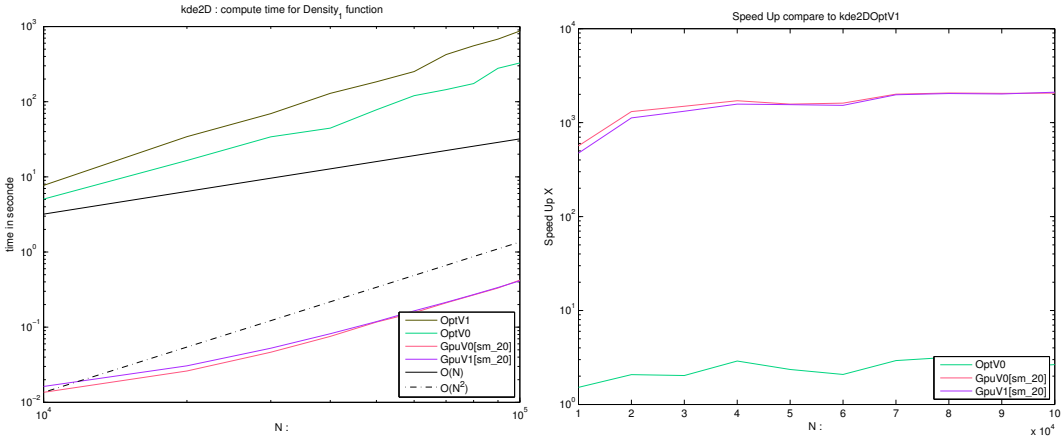

Figure 4: Functions `kde2D` computation times and speedup

3.2 Probability density : $\mathcal{D}_4$

N	OptV1	OptV0	GpuV0 (sm_20)	GpuV1 (sm_20)
10000	10.470 (s) x 1.00	6.187 (s) x 1.69	0.014 (s) x 775.01	0.016 (s) x 645.91
20000	25.690 (s) x 1.00	14.574 (s) x 1.76	0.026 (s) x 987.78	0.031 (s) x 841.86
30000	50.132 (s) x 1.00	31.873 (s) x 1.57	0.046 (s) x 1084.31	0.052 (s) x 956.40
40000	129.011 (s) x 1.00	50.232 (s) x 2.57	0.075 (s) x 1712.06	0.081 (s) x 1585.13
50000	182.639 (s) x 1.00	82.275 (s) x 2.22	0.117 (s) x 1563.49	0.118 (s) x 1550.95
60000	219.862 (s) x 1.00	96.802 (s) x 2.27	0.156 (s) x 1412.91	0.161 (s) x 1362.45
70000	305.749 (s) x 1.00	134.730 (s) x 2.27	0.211 (s) x 1448.84	0.213 (s) x 1436.70
80000	398.089 (s) x 1.00	170.879 (s) x 2.33	0.268 (s) x 1484.46	0.271 (s) x 1468.82
90000	481.935 (s) x 1.00	250.316 (s) x 1.93	0.332 (s) x 1449.69	0.337 (s) x 1430.04
100000	588.218 (s) x 1.00	261.708 (s) x 2.25	0.426 (s) x 1382.24	0.412 (s) x 1427.22

Table 2: kde2D - Probability Density : $\mathcal{D}_4$ - time and speedup

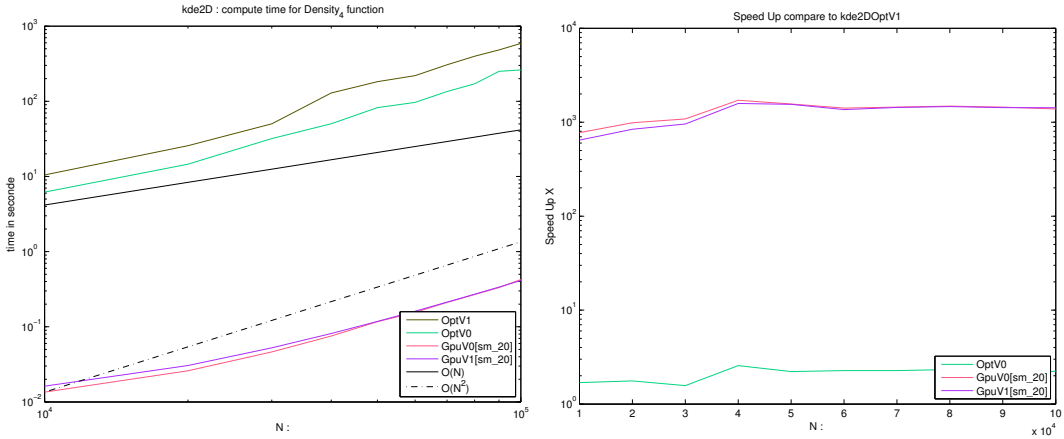

Figure 5: Functions `kde2D` computation times and speedup

3.3 Probability density : $\mathcal{D}_7$

N	OptV1	OptV0	GpuV0 (sm_20)	GpuV1 (sm_20)
10000	6.741 (s) x 1.00	6.203 (s) x 1.09	0.014 (s) x 498.11	0.016 (s) x 415.80
20000	22.427 (s) x 1.00	17.881 (s) x 1.25	0.026 (s) x 861.69	0.031 (s) x 727.66
30000	46.140 (s) x 1.00	40.261 (s) x 1.15	0.046 (s) x 1000.00	0.052 (s) x 880.82
40000	112.250 (s) x 1.00	61.971 (s) x 1.81	0.075 (s) x 1489.09	0.081 (s) x 1379.79
50000	157.776 (s) x 1.00	86.974 (s) x 1.81	0.118 (s) x 1338.29	0.118 (s) x 1339.34
60000	217.545 (s) x 1.00	114.800 (s) x 1.89	0.156 (s) x 1397.21	0.162 (s) x 1341.19
70000	304.128 (s) x 1.00	127.100 (s) x 2.39	0.211 (s) x 1444.26	0.214 (s) x 1422.86
80000	382.841 (s) x 1.00	171.186 (s) x 2.24	0.268 (s) x 1427.02	0.274 (s) x 1399.43
90000	463.140 (s) x 1.00	241.666 (s) x 1.92	0.333 (s) x 1392.68	0.335 (s) x 1381.90
100000	582.468 (s) x 1.00	284.486 (s) x 2.05	0.421 (s) x 1383.89	0.416 (s) x 1399.35

Table 3: kde2D - Probability Density : $\mathcal{D}_7$ - time and speedup

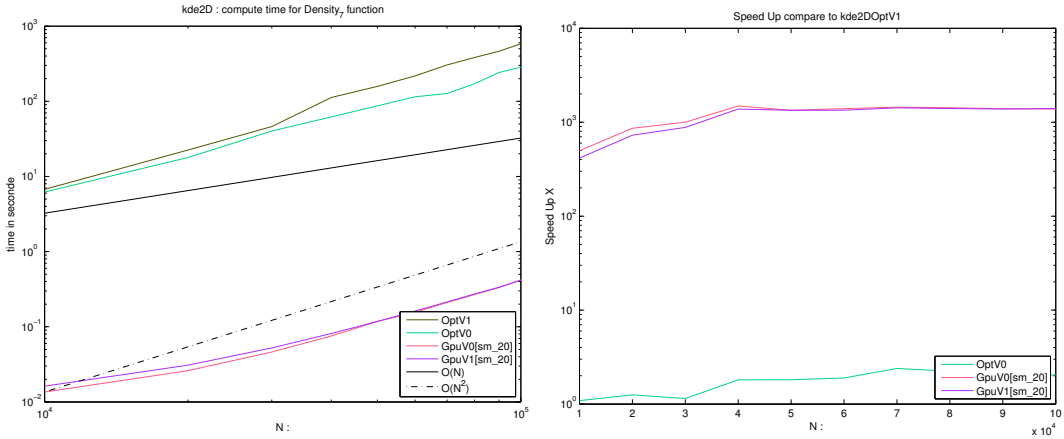

Figure 6: Functions `kde2D` computation times and speedup

4 benchs : ksdensity2D

4.1 Probability density : $\mathcal{D}_1$

N	OptV1	OptV0	GpuV0 (sm_20)	GpuV1 (sm_20)
10000	44.919 (s) x 1.00	23.691 (s) x 1.90	0.743 (s) x 60.42	0.053 (s) x 841.88
20000	94.996 (s) x 1.00	37.137 (s) x 2.56	1.472 (s) x 64.52	0.082 (s) x 1154.91
30000	139.311 (s) x 1.00	55.307 (s) x 2.52	2.201 (s) x 63.30	0.111 (s) x 1255.52
40000	190.900 (s) x 1.00	65.643 (s) x 2.91	2.929 (s) x 65.17	0.140 (s) x 1368.38
50000	235.426 (s) x 1.00	68.924 (s) x 3.42	3.658 (s) x 64.37	0.169 (s) x 1394.52
60000	246.112 (s) x 1.00	68.362 (s) x 3.60	4.386 (s) x 56.11	0.197 (s) x 1247.24
70000	296.531 (s) x 1.00	101.305 (s) x 2.93	5.116 (s) x 57.96	0.226 (s) x 1314.19
80000	368.510 (s) x 1.00	120.110 (s) x 3.07	5.843 (s) x 63.07	0.255 (s) x 1447.94
90000	425.157 (s) x 1.00	115.732 (s) x 3.67	6.575 (s) x 64.67	0.283 (s) x 1502.20
100000	460.035 (s) x 1.00	128.837 (s) x 3.57	7.410 (s) x 62.08	0.314 (s) x 1465.49

Table 4: ksdensity2D - Probability Density : $\mathcal{D}_1$ - time and speedup

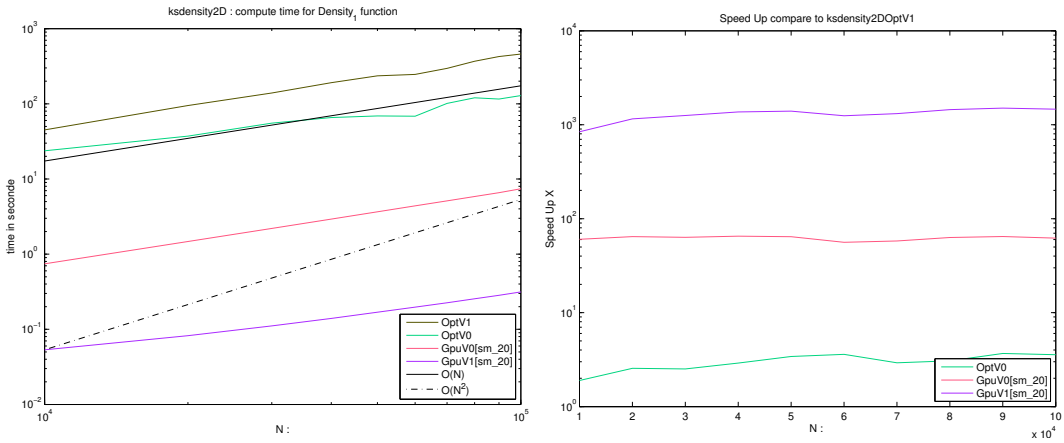

Figure 7: Functions `ksdensity2D` computation times and speedup

4.2 Probability density : $\mathcal{D}_4$

N	OptV1	OptV0	GpuV0 (sm_20)	GpuV1 (sm_20)
10000	15.501 (s) x 1.00	8.766 (s) x 1.77	0.380 (s) x 40.79	0.031 (s) x 501.72
20000	31.530 (s) x 1.00	14.170 (s) x 2.23	0.746 (s) x 42.29	0.044 (s) x 724.09
30000	46.266 (s) x 1.00	20.210 (s) x 2.29	1.111 (s) x 41.65	0.056 (s) x 825.94
40000	64.380 (s) x 1.00	26.229 (s) x 2.45	1.476 (s) x 43.61	0.068 (s) x 942.51
50000	79.215 (s) x 1.00	31.913 (s) x 2.48	1.842 (s) x 43.01	0.081 (s) x 975.82
60000	88.426 (s) x 1.00	34.039 (s) x 2.60	2.207 (s) x 40.06	0.094 (s) x 944.14
70000	108.267 (s) x 1.00	37.523 (s) x 2.89	2.573 (s) x 42.07	0.106 (s) x 1021.68
80000	117.107 (s) x 1.00	44.480 (s) x 2.63	2.938 (s) x 39.85	0.118 (s) x 996.50
90000	133.648 (s) x 1.00	50.412 (s) x 2.65	3.306 (s) x 40.43	0.130 (s) x 1028.99
100000	155.801 (s) x 1.00	51.281 (s) x 3.04	3.725 (s) x 41.83	0.145 (s) x 1071.99

Table 5: ksdensity2D - Probability Density : $\mathcal{D}_4$ - time and speedup

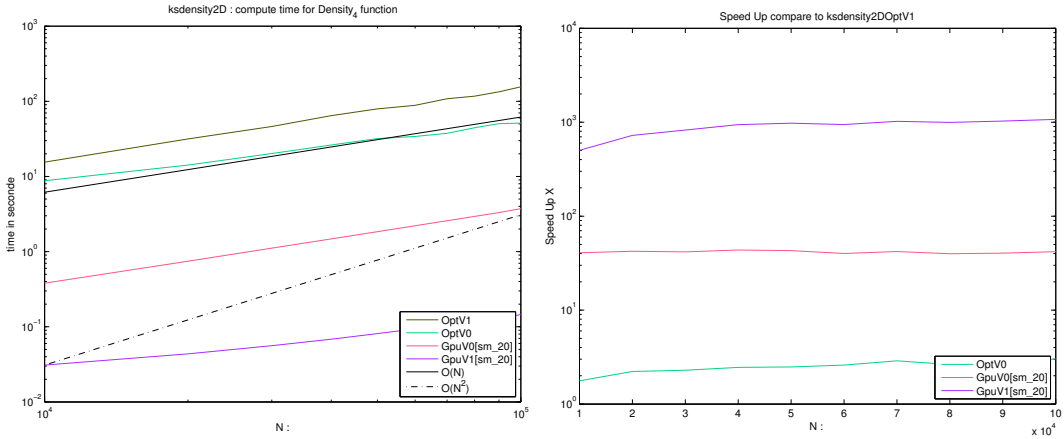

Figure 8: Functions `ksdensity2D` computation times and speedup

4.3 Probability density : $\mathcal{D}_7$

N	OptV1	OptV0	GpuV0 (sm_20)	GpuV1 (sm_20)
10000	48.475 (s) x 1.00	28.207 (s) x 1.72	0.744 (s) x 65.20	0.053 (s) x 908.56
20000	95.942 (s) x 1.00	46.393 (s) x 2.07	1.472 (s) x 65.17	0.082 (s) x 1165.37
30000	140.856 (s) x 1.00	63.929 (s) x 2.20	2.201 (s) x 64.00	0.111 (s) x 1270.52
40000	188.674 (s) x 1.00	68.831 (s) x 2.74	2.929 (s) x 64.41	0.140 (s) x 1350.45
50000	232.740 (s) x 1.00	67.728 (s) x 3.44	3.658 (s) x 63.63	0.169 (s) x 1381.06
60000	281.939 (s) x 1.00	78.715 (s) x 3.58	4.386 (s) x 64.28	0.197 (s) x 1430.63
70000	313.130 (s) x 1.00	84.362 (s) x 3.71	5.116 (s) x 61.20	0.226 (s) x 1388.09
80000	369.719 (s) x 1.00	100.394 (s) x 3.68	5.842 (s) x 63.28	0.254 (s) x 1454.55
90000	419.001 (s) x 1.00	121.777 (s) x 3.44	6.575 (s) x 63.73	0.283 (s) x 1481.25
100000	451.534 (s) x 1.00	125.787 (s) x 3.59	7.411 (s) x 60.93	0.320 (s) x 1411.59

Table 6: ksdensity2D - Probability Density : $\mathcal{D}_7$ - time and speedup

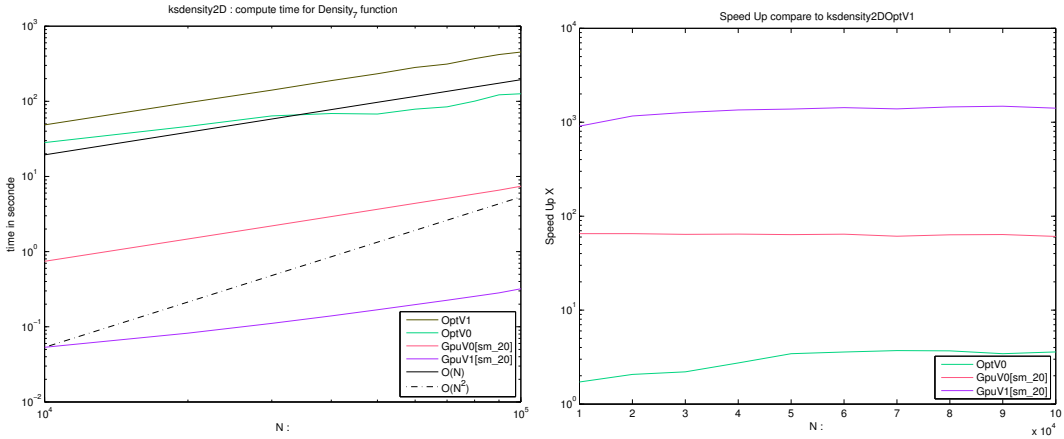

Figure 9: Functions `ksdensity2D` computation times and speedup