Neighbor Search's Bounds

After reading the paper *"Tree-Independent Dual-Tree Algorithms"* in detail, I found some mistakes in the definition of bounds, in the section *"k-Nearest Neighbor Search"*:
I think the recursive definition of the bound B_2 is incorrect. It is defined as:

$$B_2(N_q) = min\{min_{p \in P_q}(D_p[k] + \rho(N_q) + \lambda(N_q)),\ min_{N_c \in C_q}(B_2(N_c) + 2(\lambda(N_q) - \lambda(N_c)))\}$$

It makes sense to use a tighter bound: $\rho(N_q) + \lambda(N_q)$ for points in the node, but the recursive call: $min_{N_c \in C_q}(B_2(N_c) + 2(\lambda(N_q) - \lambda(N_c)))$, doesn't seem to be ok. We are subtracting $2\lambda(N_c)$ but maybe the child's B_2 bound was calculated using the tighter bound: $\rho(N_c) + \lambda(N_c)$, where probably: $\rho(N_c) + \lambda(N_c) < 2\lambda(N_c)$, so we are substracting more than the correct value. So, sometimes, this could result in a smaller value of B_2 than the correct one, and erroneously prune.

From the initial definition:

$$B_2(N_q) = min_{p \in D_q^p} D_p[k] + 2\lambda(N_q)$$

The recursive definition would be:

$$B_2(N_q) = min\{min_{p \in P_q}(D_p[k] + 2\lambda(N_q)),\ min_{N_c \in C_q}(B_2(N_c) + 2(\lambda(N_q) - \lambda(N_c)))\}$$

If we introduce the tighter bound: $\rho(N_q) + \lambda(N_q)$ for points in the node, we not only have to change the base case, but also the recursive call.
I would define it, using an auxiliary function B_{aux}:

$$B_{aux}(N_q) = min\{min_{p \in P_q} D_p[k], min_{N_c \in C_q} B_{aux}(N_c)\}$$

$$B_2(N_q) = min\{min_{p \in P_q}(D_p[k] + \rho(N_q) + \lambda(N_q)),\ B_{aux}(N_q) + 2\lambda(N_q)\}$$

Finally, the total definition of bounds would be:

$$B_1(N_q) = max\{max_{p \in P_q} D_p[k], max_{N_c \in C_q} B_1(N_c)\}$$

$$B_{aux}(N_q) = min\{min_{p \in P_q} D_p[k], min_{N_c \in C_q} B_{aux}(N_c)\}$$

$$B_2(N_q) = min\{min_{p \in P_q}(D_p[k] + \rho(N_q) + \lambda(N_q)),\ B_{aux}(N_q) + 2\lambda(N_q)\}$$

$$B(N_q) = min\{B_1(N_q), B_2(N_q), B(Par(N_q))\}$$

Code of mlpack *neighbor_search*, implements B_2 as mentioned in the paper, so I think we should fix it.