The Bunker Cache for Spatio-Value Approximation

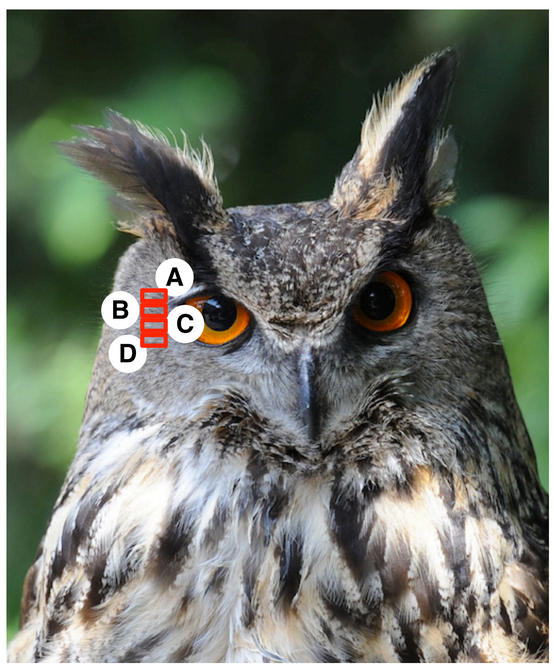
Joshua San Miguel, Jorge Albericio, Natalie Enright Jerger and Aamer Jaleel

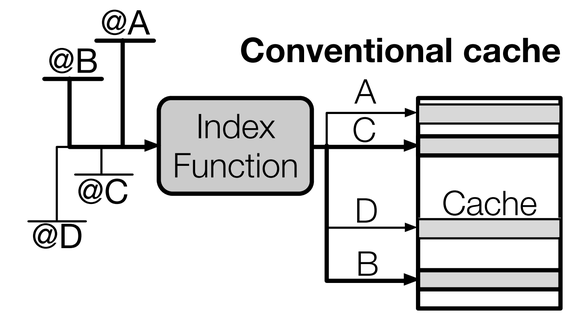
The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
University of Toronto

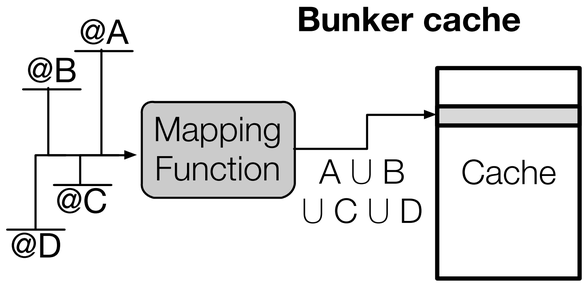
NVIDIA

Bunker Cache Overview

The Bunker Cache exploits **spatio-value similarity**, mapping approximately similar data to the same cache storage location based solely on their memory address. This enables efficiency gains via reductions in 1) last-level cache misses, 2) off-chip memory accesses and 3) cache storage requirements.





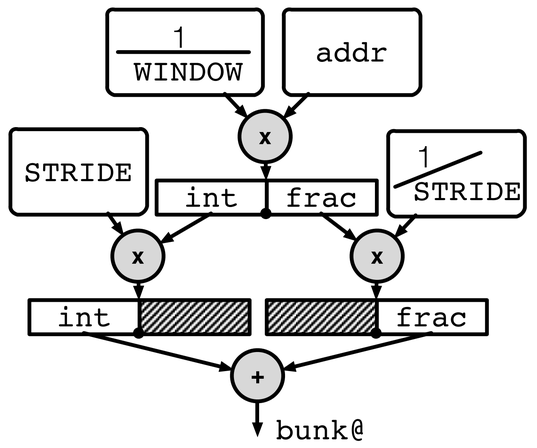


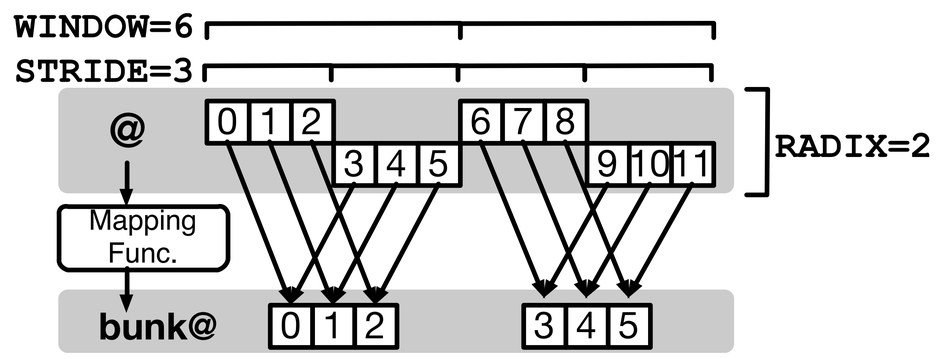
The Bunker Cache requires only modest changes to cache indexing hardware, integrating easily into commodity systems.

Bunker Cache Mapping Function

The Bunker Cache mapping function associates addresses of similar blocks to the same `bunkaddress`. In the example figure below, addresses `0` to `11` are mapped to `bunkaddresses` using a stride of 3 and radix of 2.

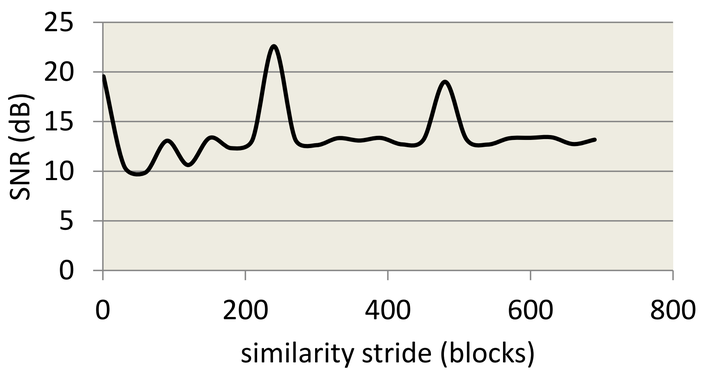
```
int bunk@(int addr, int STRIDE, float RADIX):
    WINDOW = (int) (RADIX * STRIDE);
    bunk@ = addr / WINDOW;
    bunk@ *= STRIDE;
    bunk@ += (addr % WINDOW) % STRIDE;
```



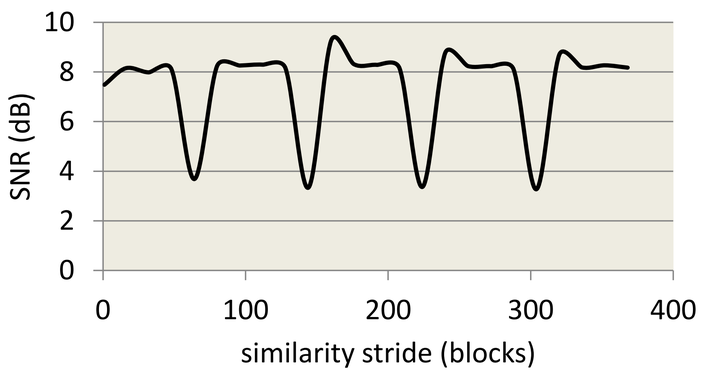


Spatio-Value Similarity

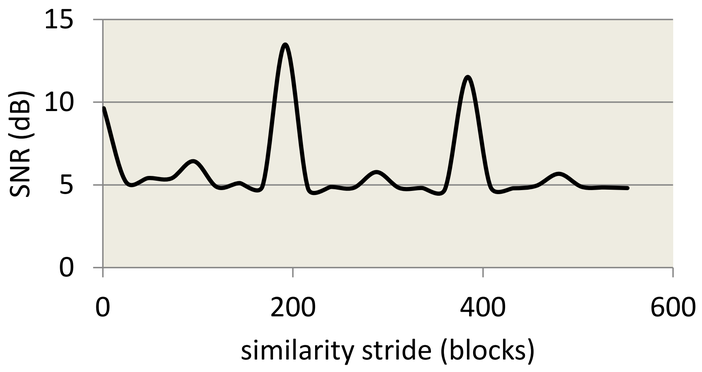
Data elements that are similar in value exhibit spatial regularity in memory. This is inherent to 1) the data values of real-world applications, and 2) the way we store data structures in memory. The figures below show output quality with varying similarity strides, demonstrating significant spatio-value similarity in our applications.



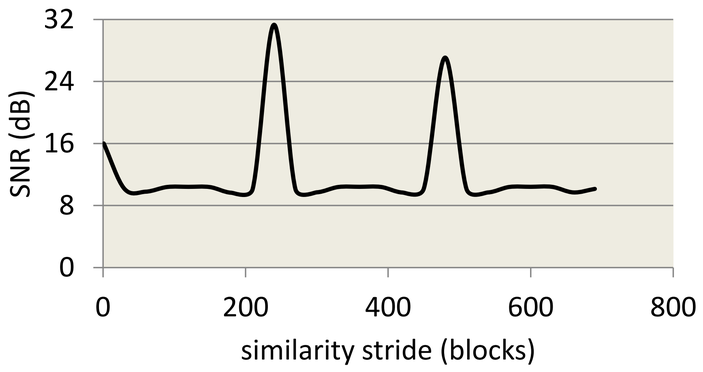
(a) 2dconv



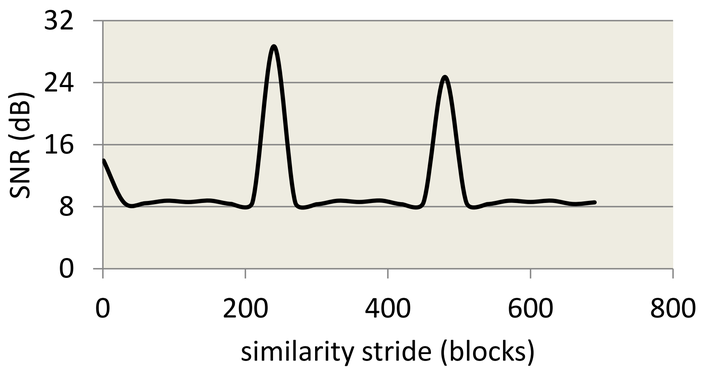
(b) change-detection



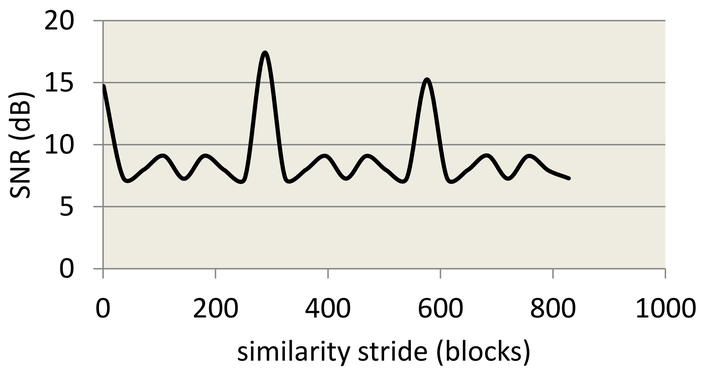
(c) debayer



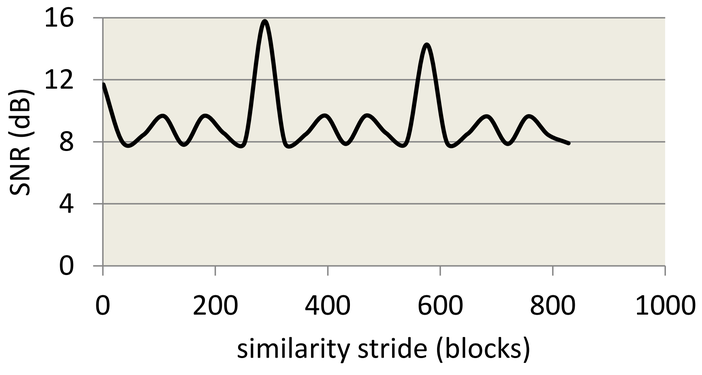
(d) dwt53



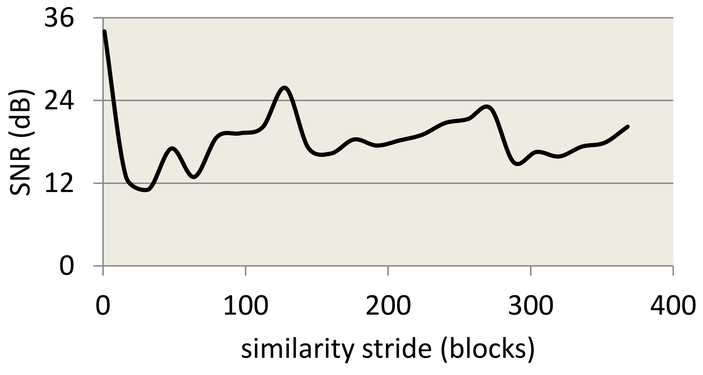
(e) histeq



(f) jpeg

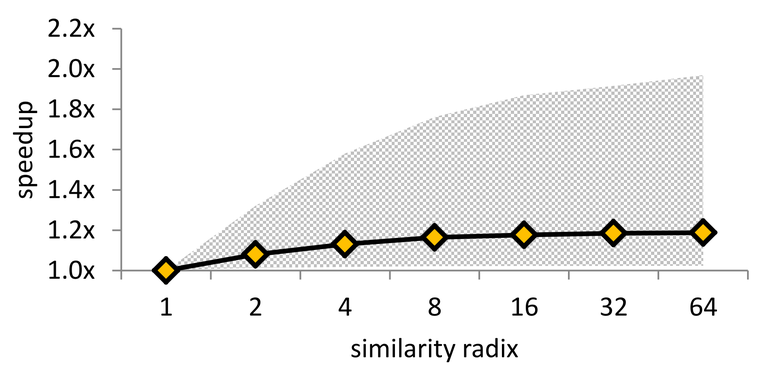


(g) kmeans

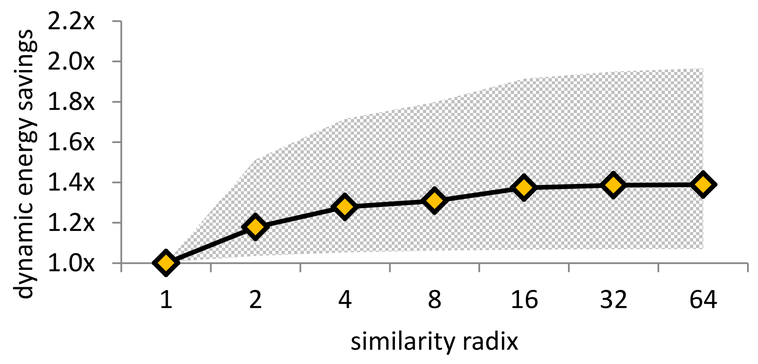


(h) lucas-kanade

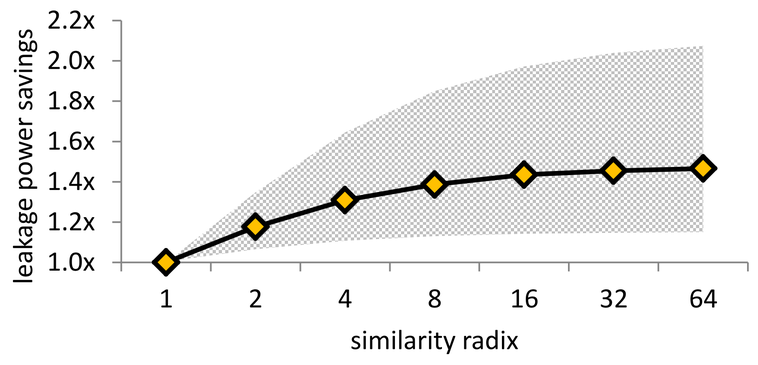
Application Speedup



Dynamic Energy Savings



Leakage Power Savings



Output Quality