PENNSTATE ARL

Optimization of the Earth-Mover's Distance algorithm for a GPU-based real-time multispectral computer vision system

Andrew Shaffer
aps148@arl.psu.edu

High Performance Embedded Computing Workshop
Sept. 22, 2011

The Pennsylvania State University
Applied Research Laboratory

Outline

- Overview
- Color-Based Detection and Tracking
- Metamerism
- Multispectral Video
- Earth Mover's Distance (EMD) Algorithm
- EMD Optimizations
- System Design
- Experimental Setup
- Performance Results
- Future Work
- Questions

Overview

- Problem: Develop a robust real-time detection and tracking system to identify camouflaged objects
- Solution: Exploit multispectral video to identify camouflaged objects using GPU-accelerated algorithm
- Approach: Develop high-performance GPU implementation of Earth Mover Distance algorithm

Multispectral Video Data

NVIDIA C2050 GPU

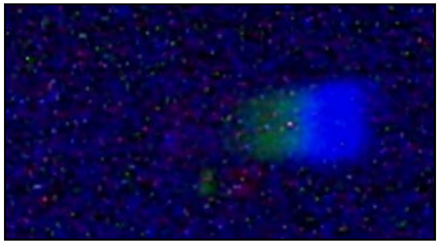
Detect Camouflaged Objects

Color-Based Detection and Tracking

- Compare color distribution histograms from regions of a scene with color distribution histogram of the target
- High correlation between target and scene histograms suggests the location of the target
- Correlation easy to compute for RGB cameras because only 3 color channels to compare

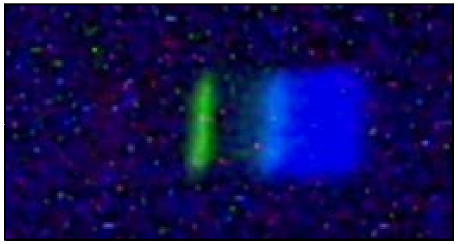
Red highlights indicate potential targets of interest (e.g. faces)

Metamerism



CRT Monitor

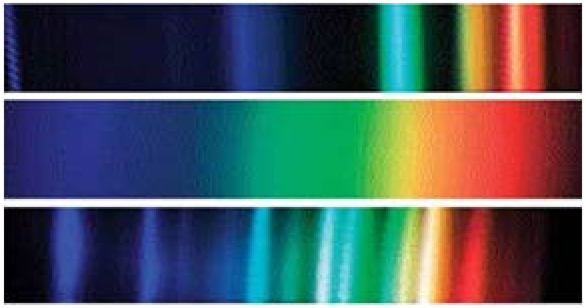
CRT & LCD displaying same RGB value



LCD Monitor

- Objects with different spectral intensity distributions are perceived as having the same RGB values
- Occurs because RGB cameras integrate a wide range of light wavelengths to form each color channel
- Can prevent overlapping objects from being correctly separated during image analysis

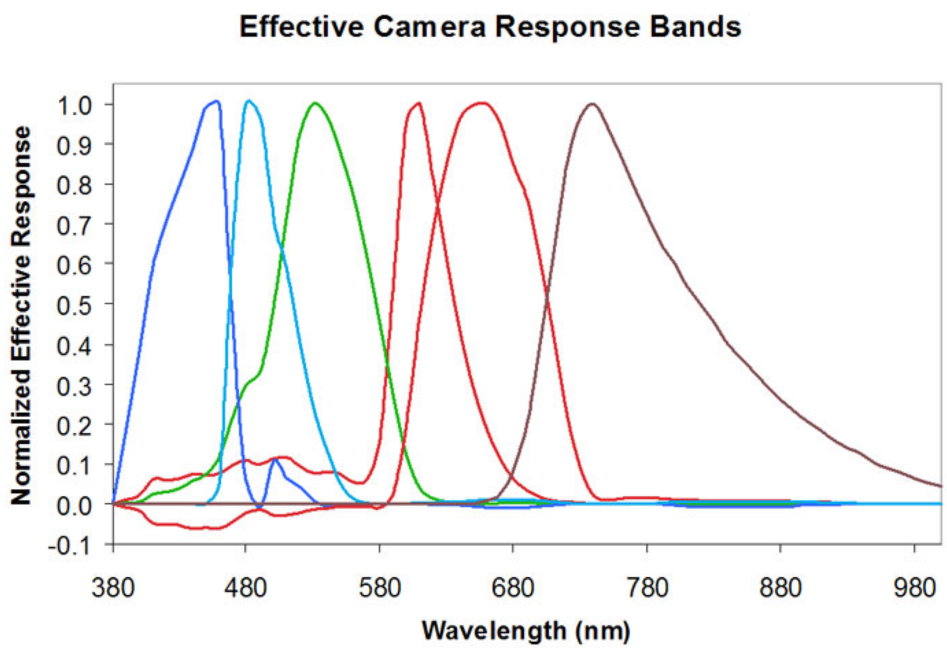
Spectroscopic response of 3 metameric white lights (at right): fluorescent (top), LED (middle), and metal-halide (bottom)



Multispectral Video

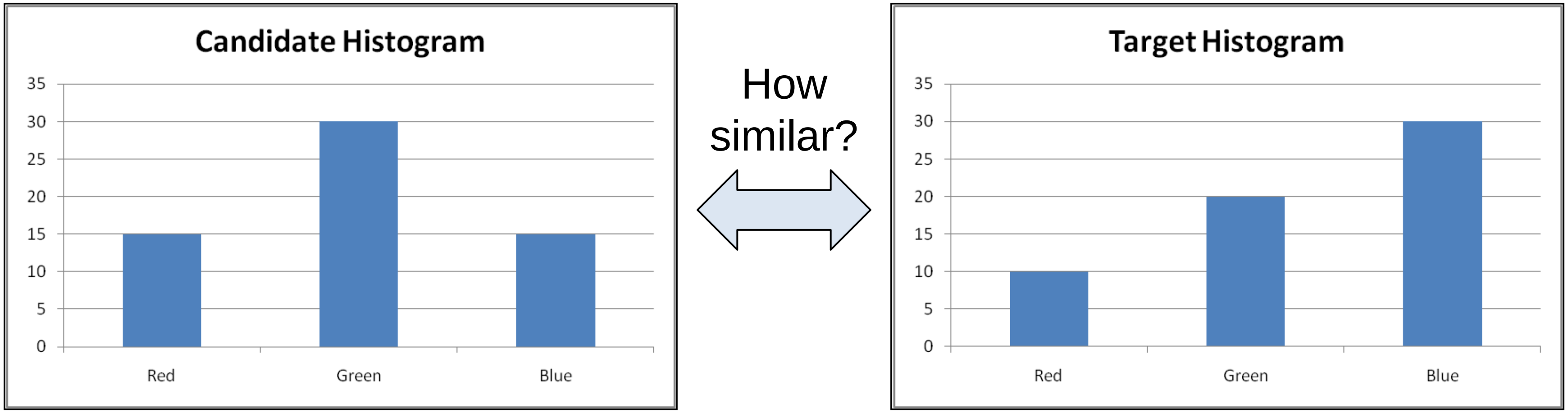
- Multispectral video cameras use more than three color channels to describe scene colors more precisely
- Multispectral video cameras have higher data rates than comparable RGB cameras
- Determining correlation of histograms with more than 3 color channels is more difficult

FluxData Multispectral
Video Camera



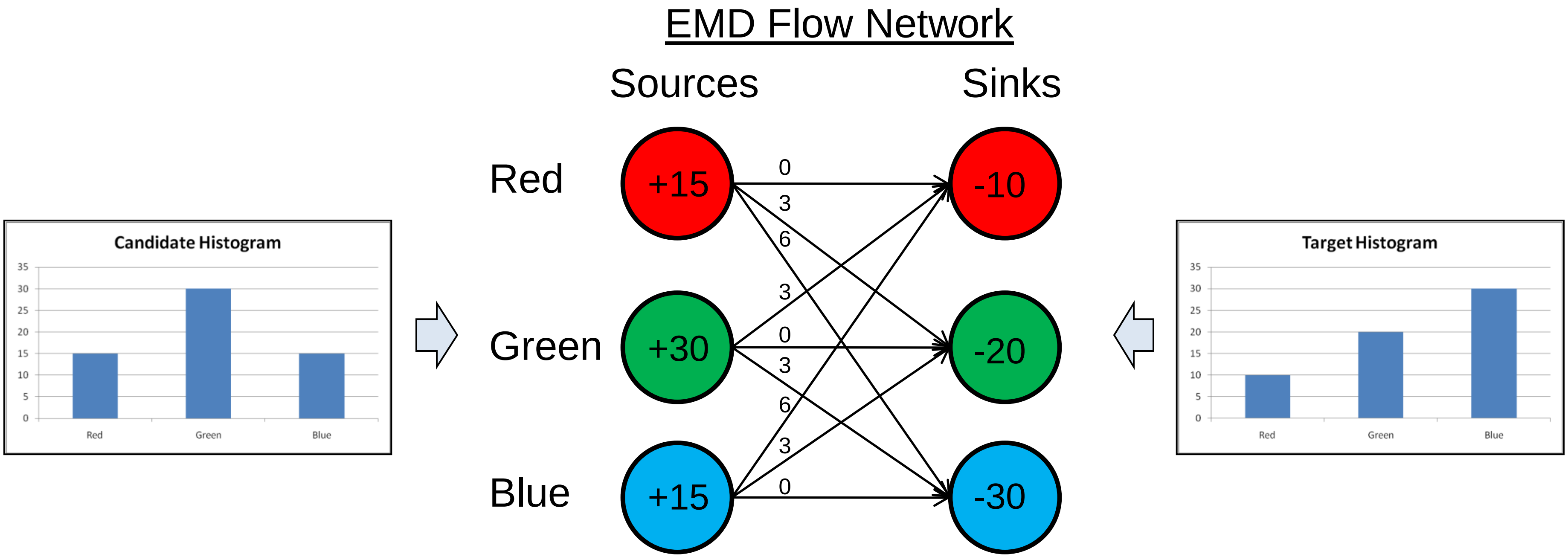
Earth Mover's Distance

- EMD allows calculation of intuitive correlation measure between arbitrary histograms
- Conceptual Description:
  - Each histogram bin corresponds to a pile of "earth"
  - Moving 1 unit of earth from histogram bin *a* to bin *b* costs f(*a*,*b*)
  - EMD finds the minimum cost to convert a candidate histogram into a specified target histogram
  - Lower cost corresponds to higher correlation between the candidate and target histograms



Earth Mover's Distance (cont)

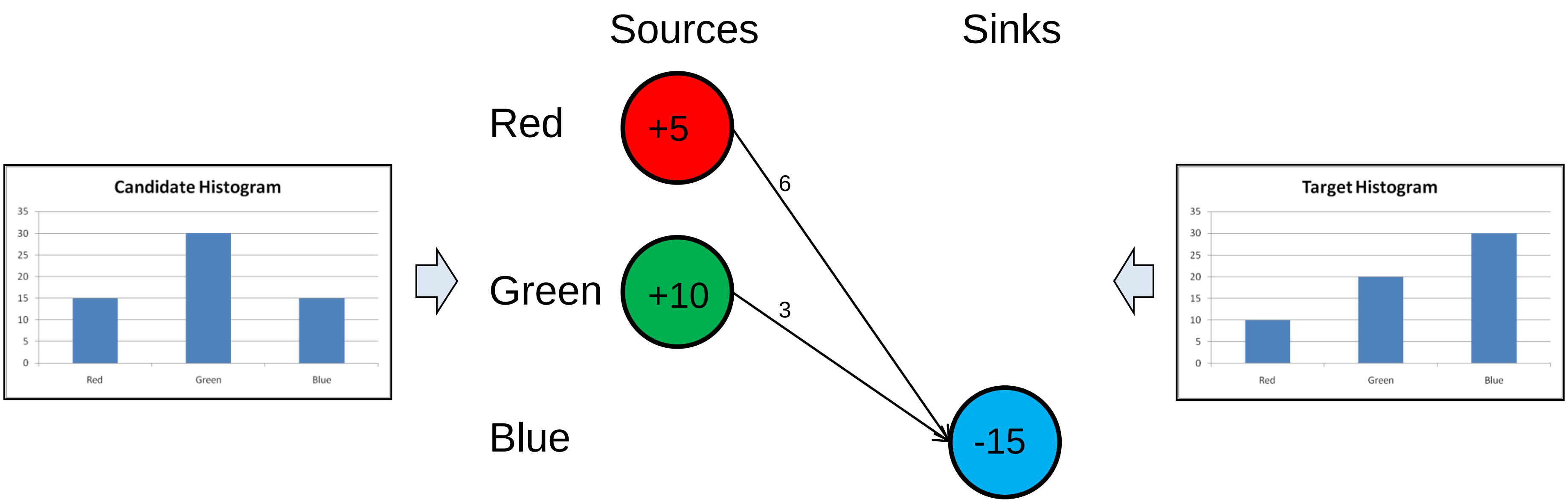
- EMD is an instance of the Transportation Problem
  - Minimum cost flow routing can be solved using linear programming
  - Candidate histogram bins are sources, target histogram bins are sinks, and cost of moving flow along each edge corresponds to distance between histogram bins
  - *…but it is still slow: ~55 hrs to solve 10-bin EMDs for a single 1920x1080 pixel frame using Matlab on Intel Core 2 6600 @ 2.4 GHz*



Algorithmic Optimizations

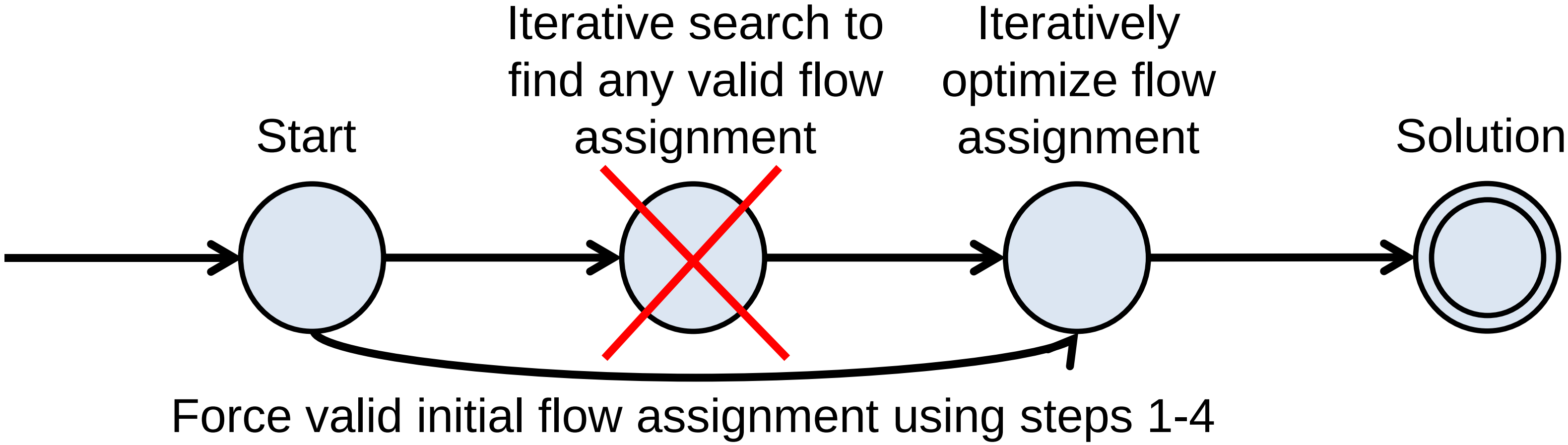
- Create only one source or sink for each histogram bin pair
  - CandBin[n] > TgtBin[n]: Bin n is source generating flow CandBin[n] – TgtBin[n]
  - CandBin[n] < TgtBin[n]: Bin n is sink accepting flow TgtBin[n] – CandBin[n]
- Eliminate histogram bins where the candidate and target histogram bins have exactly the same value
  - CandBin[n] == TgtBin[n]: Bin n does not appear in flow network

EMD Flow Network



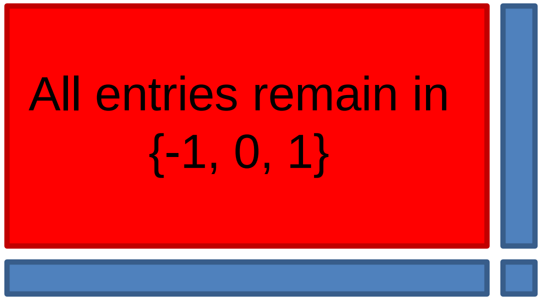
Algorithmic Optimizations (cont)

- Bypass Simplex search to find initial valid flow assignment
  1. Add column of zeros for “unassigned” flow to tableaux with higher cost than any other path in the cost matrix
  2. Add redundant constraint row enforcing that total assigned flow plus “unassigned” flow equals total flow across network
  3. Perform initial pivot with new column and new row as pivot point
  4. Perform row operations to eliminate redundancy in new row
  5. Newly added row and column will not be involved in any subsequent pivots, so *we can apply steps 1-4 to the initial tableaux without materializing the extra row or column*



Storage & Computation Optimizations

- All entries in main body of tableaux always remain in {-1, 0, 1}
  - Eliminates multiplication during pivoting
  - Main tableaux content entries can be stored in bitmap format using only 2 bits per entry
  - All tableaux contents can be stored directly in registers with one row stored by each thread
- Bitmap allows math to be replaced by bitwise logical operations



Bitmap Storage Format		
Value	Pos Bit	Neg Bit
-1	0	1
0	0	0
1	1	0

When using 32-bit variables, the 5 instructions in the example perform the += operation for up to 32 values at a time

(For -=, swap Addend Pos & Addend Neg)

Example: Val += Addend

1. Val Pos. |= Addend Pos.
2. Val Neg. |= Addend Neg.
3. Same = Val Pos. ^ Val Neg.
4. Val Pos &= Same
5. Val Neg &= Same

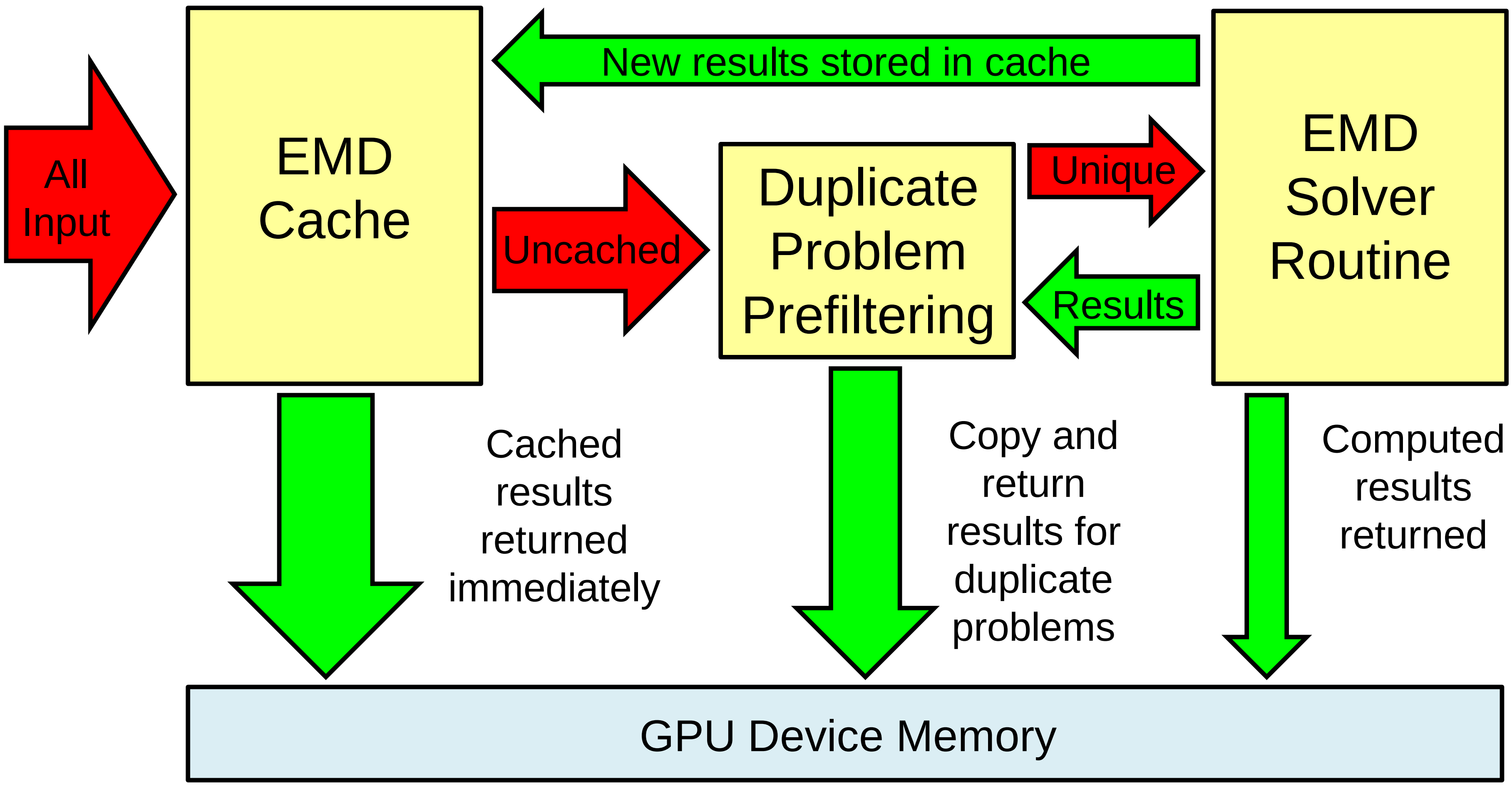
Workload Distribution Optimizations

- For problems with 11 or fewer histogram bins, 16 or fewer threads are needed to store and solve each problem
  - For these problems, each physical warp can be divided into 2 or more “virtual warps” (VW) that operate in parallel
  - The reduced size of these virtual warps allows fewer stages for binary reduction operations (e.g. find min across all threads)
  - Because all threads within a physical warp are implicitly synchronized with each other, there is no need for explicit synchronization calls at any point within the EMD solver routine

Each physical warp contains 32 threads

(Thread color indicates virtual warp assignment; *red* indicates unused threads)

System Design



Red = Input data; *Green* = Results; Arrow size is proportionate to data being moved

System Design (cont)

- EMD Cache exploits temporal locality
  - Pixels that appear in one frame are likely to reappear in subsequent frames due to object persistence
  - Uses difference histogram as key to allow multiple targets
  - Uses large GPU device memory to store many EMD values
- Duplicate Prefiltering exploits spatial locality
  - Any given pixel is likely to be surrounded by other pixels with similar histograms due to objects having regions of similar colors
  - Implemented as open-addressed hash table protected by atomically manipulated lock variables for high performance

Exploitation of locality minimizes workload required to process each frame and maximizes utilization of GPU resources

Experimental Setup

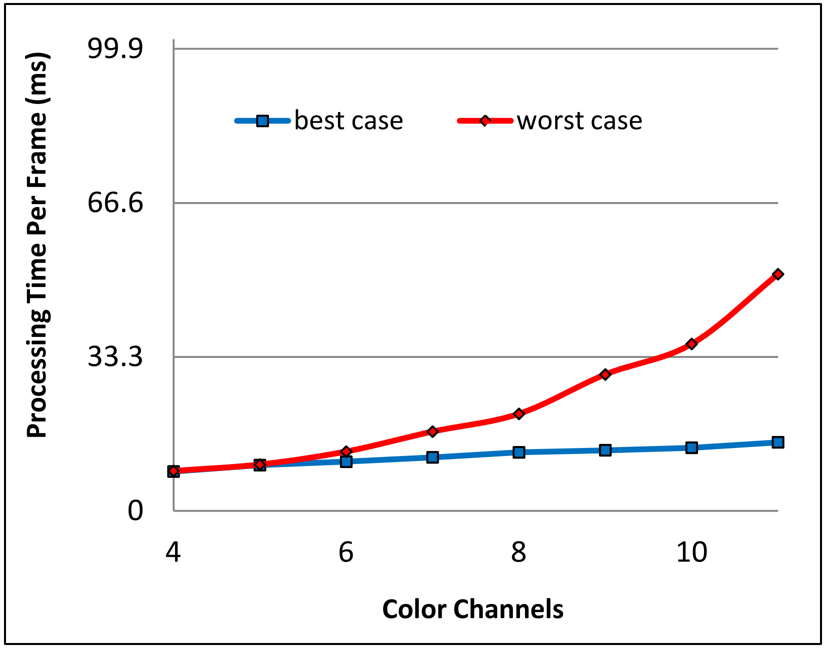
- Generate simulated multispectral input and target images
  - Input and target images generated by computing intensity histograms for windowed regions of grayscale images
- Process simulated multispectral input image twice using NVIDIA C2050 GPU:
  - Data moved from CPU to GPU and back again for each frame
  - First run gives worst-case performance (i.e. no cache hits)
  - Second run gives best-case performance (i.e. maximal cache hits)
- Exclude one-time setup and cleanup processing from timing to obtain steady-state timing values

Experiment designed to estimate the performance of a real-world multispectral object detection and tracking system

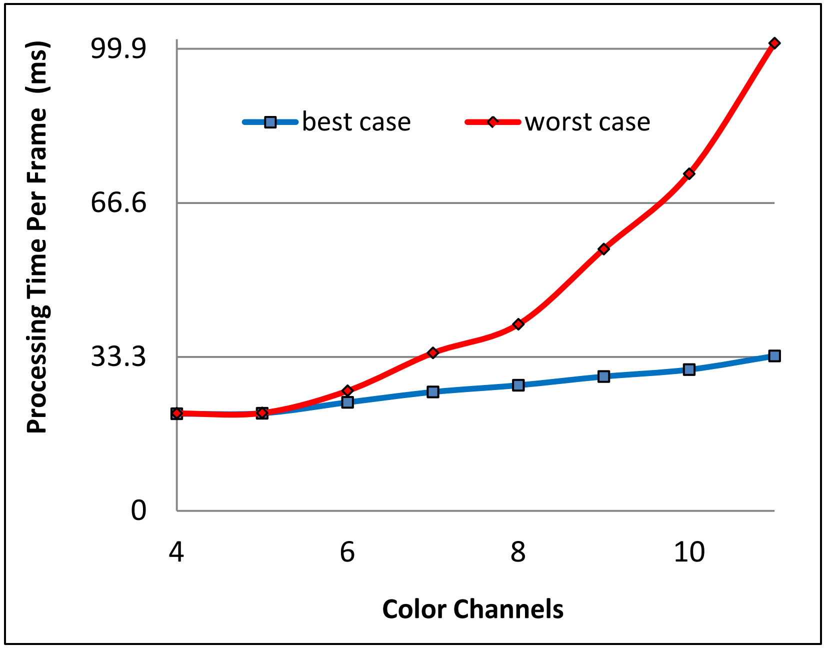
NVIDIA C2050 GPU

Performance Results

Processing Time vs. Number of Color Channels



1280x720 pixel input image



1920x1080 pixel input image

Overall system performance is between 9.9-117.1 frames per second for high definition input data, making the EMD algorithm practical for many real-time computer vision applications

Future Work

- Build fully operational real-time system using multispectral video camera
- Support increased number of histogram bins to compute histogram similarities for hyperspectral video input
- Further improve performance by distributing workload across multiple GPUs
- Support other applications that require fast histogram comparisons

Multispectral Video Camera

NVIDIA C2050 GPU

Detect Camouflaged Objects

Questions?

Andrew Shaffer

aps148@arl.psu.edu