Name: Harsh Vijay Mamania

CS 5330: Week 4 Homework

Question #1

Given: three vectors v1 = [4, 3], v2 = [7, 7], v3 = [4, 8]

We looked at sum-squared difference (SSD) as a distance metric, which is also called an L-2 distance metric. We also discussed L-1 and L-infinity metrics. For each of the three distance metrics L-1, L-2, and L-infinity, calculate which of v2 or v3 is closer to v1. (Using a calculator is not necessary). How would you describe each metric in terms of what is most important to the distance calculation?

Solution-

L-1 Distance (Manhattan/Taxicab distance):

- Formula: $|\Delta x| + |\Delta y|$
- d(v1, v2) = |3| + |4| = 7
- d(v1, v3) = |0| + |5| = 5
- Winner: **v3** is closer (5 < 7)
- What's important: The sum of absolute differences across all dimensions; treats each dimension equally and additively

L-2 Distance (Euclidean/SSD):

- Formula: $\sqrt{(\Delta x^2 + \Delta y^2)}$ or just compare $\Delta x^2 + \Delta y^2$ (since sqrt is monotonic)
- $d(v1, v2)^2 = 3^2 + 4^2 = 9 + 16 = 25$
- $d(v1, v3)^2 = 0^2 + 5^2 = 0 + 25 = 25$
- Winner: **Tie** (both equal distance)
- What's important: The straight-line distance; penalizes larger differences more heavily due to squaring (so one large difference can dominate over several small ones)

L-infinity Distance (Chebyshev/Max norm):

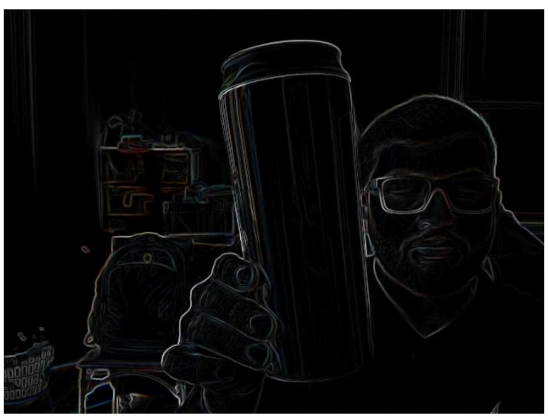
- Formula: $\max(|\Delta x|, |\Delta y|)$
- d(v1, v2) = max(|3|, |4|) = 4
- d(v1, v3) = max(|0|, |5|) = 5
- Winner: **v2** is closer (4 < 5)
- What's important: Only the single largest difference across all dimensions matters; all other dimensions are ignored

---

Question #2

Select two different textures and use your project 1 program to show the gradient magnitude for each texture. Would average energy of gradient magnitude be a useful feature for differentiating these two images?

Solution-



```
Average Gradient Energy: 9.82103
Average Gradient Energy: 9.80626
Average Gradient Energy: 9.80994
Average Gradient Energy: 9.72537
Average Gradient Energy: 9.7261
Average Gradient Energy: 9.71291
Average Gradient Energy: 9.70836
Average Gradient Energy: 9.70904
Average Gradient Energy: 9.70352
Average Gradient Energy: 9.70572
```

```
Average Gradient Energy: 11.471
Average Gradient Energy: 11.4687
Average Gradient Energy: 11.465
Average Gradient Energy: 11.4692
Average Gradient Energy: 11.4786
Average Gradient Energy: 11.4708
Average Gradient Energy: 11.4578
Average Gradient Energy: 11.4069
Average Gradient Energy: 11.4063
Average Gradient Energy: 11.4124
```

Changes made to code / formula used to calculate:

```
if (filter_mode == "magnitude") {
    cv::Mat sobelX_result_16b, sobelY_result_16b;
    sobelX3x3(frame, sobelX_result_16b);
    sobelY3x3(frame, sobelY_result_16b);
    magnitude(sobelX_result_16b, sobelY_result_16b, frame);

    // Added the following 3 lines to get energy (for HW4)
    cv::Scalar mean_val = cv::mean(frame);
    double avg_energy = (mean_val[0] + mean_val[1] + mean_val[2]) / 3.0;
    std::cout << "Average Gradient Energy: " << avg_energy << std::endl;
}
```

Observed Results:

- **Smooth bottle**: Average gradient energy ≈ 9.7-9.8
- **Brick-textured flower pot**: Average gradient energy ≈ 11.1-11.5
- **Difference**: ~15-17% higher for textured surface

Would average gradient energy be useful for differentiating these textures?
Yes, with limitations:

- Clear separation exists between smooth and textured surfaces in this case
- The textured surface consistently shows higher gradient energy, matching intuition
- Simple, fast to compute, and provides quantitative differentiation

However:

- Not rotation or scale invariant
- Sensitive to lighting conditions
- Loses all spatial structure information by averaging
- Different textures could potentially have similar averages
- For robust texture classification, should be combined with other features (gradient variance, directional histograms, spatial frequency content) rather than used alone

Conclusion: Reasonably useful for this binary smooth vs. textured comparison, but insufficient as a standalone feature for general texture discrimination tasks.

Question #3

For each of the following metrics, find or create two example textures. One should represent a high value of the metric, the other should represent a low value of the metric. You can submit photos or drawings for your examples.

- Entropy
- Contrast (pick an offset of your choice)
- Regularity

Solution-

#1 Entropy

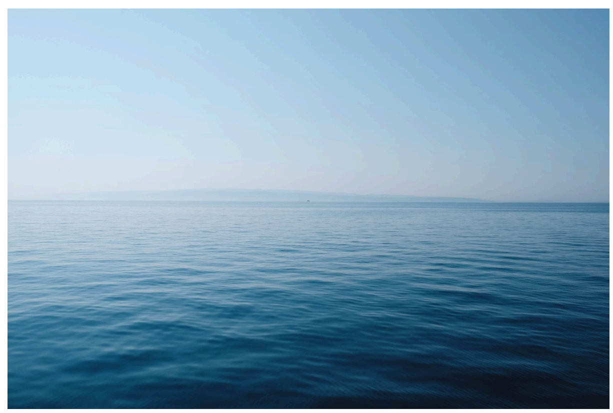



Image 1 (Pebbles/Stones) - HIGH ENTROPY:

- Random colors, sizes, shapes, and positions of stones
- Impossible to predict the next pixel's value
- High information content - lots of variation and unpredictability
- Many different intensity values distributed randomly

Image 2 (Ocean/Sky) - LOW ENTROPY:

- Smooth gradient from blue water to light sky
- Very predictable pixel values - neighboring pixels are nearly identical
- Low information content - mostly uniform with gentle gradation
- Few distinct intensity levels, smooth transitions

#2 Contrast

Image 1 (Zebra) - HIGH CONTRAST:

- Sharp black-to-white transitions in the stripes
- Dramatic intensity differences between adjacent pixels
- At any small offset (e.g., 1 pixel), we'd get large intensity jumps
- Clear boundaries with maximum contrast

Image 2 (Horse) - LOW CONTRAST:

- Smooth, uniform tan/beige coat
- Gentle intensity variations across the body
- At any small offset, neighboring pixels have very similar values
- Soft transitions, no sharp edges on the main subject

#2 Regularity

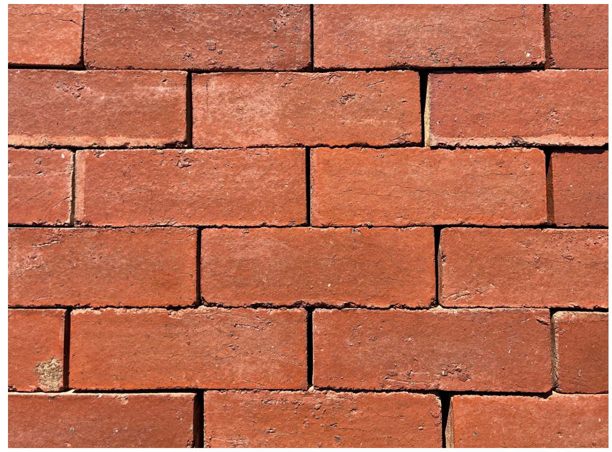


Image 1 (Pile of Bricks) - LOW REGULARITY:

- Random orientations and positions
- No repeating spatial pattern
- Chaotic, aperiodic arrangement
- Cannot find a consistent repeating unit that tiles the texture
- Unpredictable structure

Image 2 (Brick Wall) - HIGH REGULARITY:

- Highly periodic, repeating pattern
- Consistent horizontal rows with staggered offset (running bond pattern)
- Predictable structure - each brick's position follows the pattern
- Clear repeating unit that tiles across the entire surface
- Regular spacing and alignment

Question #4

When using Law's texture filters, why do you think it is helpful to divide the responses by the Gaussian filter (L5 x L5)?

Solution-

Purpose: Normalization for brightness-invariant texture analysis

- Dividing by (L5 × L5) removes the effect of local brightness/intensity variations from the texture response
- Makes texture measurements independent of lighting conditions or overall image brightness
- Allows comparison of textures across regions with different illumination levels

What the division accomplishes:

- (L5 × L5) captures the local average intensity (DC component/mean brightness) in each neighborhood
- Dividing by this value normalizes each filter response relative to the local brightness
- Converts absolute intensity measurements into relative texture patterns
- Two regions with identical texture patterns but different brightness levels will now produce similar normalized responses

Benefits:

- *Lighting invariance:* Same texture appears similar regardless of whether it's in shadow or bright light
- *Fair comparison:* Can compare textures across the entire image without brightness bias
- *Enhanced discrimination:* Focuses on texture structure rather than intensity values
- *Robust features:* More reliable for texture classification and segmentation tasks

Similar to how we'd normalize data by dividing by the mean, it removes scale differences and focuses on the pattern/structure itself rather than absolute magnitudes.

Question #5

What are the two primary uses of K-means clustering? Given an example of each.

Solution-

#1 - "Discovery"

- Discover inherent structure and patterns in unlabeled image data
- Find natural clusters without prior knowledge of categories
- Exploratory data analysis to understand data distribution

How:
- Apply K-means to image features (color, texture, shape descriptors, etc.)
- Algorithm reveals natural groupings based on similarity
- Helps understand what distinct groups exist in the dataset

#2 - "Quantization"

- Find a small set of prototypes/exemplars that efficiently represent the full variation in the data
- Reduce high-dimensional data to a compact codebook of representative vectors
- Data compression through representative sampling

How:
- K-means finds K cluster centers that best represent all data points
- These centers become the "codebook" - a dictionary of representative patterns
- Any data point can be approximated by its nearest codebook entry

---

Question #6

Varma and Zisserman use K-means clustering on what data (there are two possible answers here) as part of their texture classification method?

Solution-

Two possible data types clustered:

1. Filter bank response vectors

- At each pixel location, apply a collection of filters (e.g., Gaussian derivatives, edge detectors, etc.)
- The outputs from all filters at that pixel form a multi-dimensional feature vector
- K-means clusters these feature response vectors across all pixels in the image
- Cluster centers represent common filter response patterns

2. Raw pixel intensity neighborhoods

- Extract a small NxN window (patch) of raw intensity values centered at each pixel
- Flatten this patch into a vector (e.g., 5×5 window = 25-dimensional vector)
- K-means clusters these intensity pattern vectors across all pixel locations
- Cluster centers represent common local intensity patterns

After clustering:

- The K cluster centers become the "texton dictionary" - a vocabulary of representative texture elements
- Each pixel is then labeled with its nearest cluster ID (which texton it belongs to)
- This creates a texton map where pixels are classified by their local texture pattern
- Textures can then be characterized by their distribution of textons

---

Question #7

What was the inspiration for creating Gabor Filters?

Solution-

The human visual system and primary visual cortex.

- Gabor filters were designed to replicate the processing functions observed in the primary visual cortex (V1)
- Studies discovered that neurons in the early visual cortex respond to specific orientations, spatial frequencies, and localized image features
- Gabor filters mathematically model these cortical cell responses, mimicking how the human visual system decomposes visual information
- This makes them effective for texture analysis and feature extraction because they align with natural visual perception

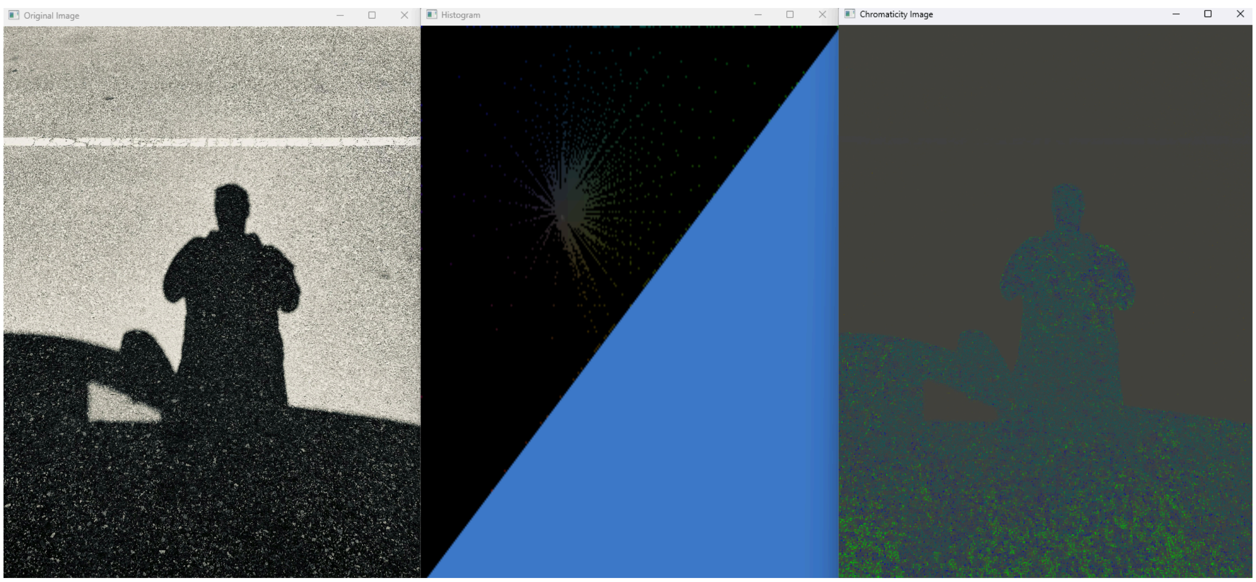
---

Question #8

(If not submitted last week) Take a picture of a cast shadow outside on a sunny day, ideally on a sidewalk, asphalt, or some other surface that is close to grey. Use my 2-D histogram code (or your own code) to make an rg chromaticity histogram of the image. Explain the shape of the histogram.

Add to the code the ability to visualize the rg chromaticity version of the image (this should look like the original image, but with all of the pixels at the same intensity). Once you compute the (r, g) chromaticity for each pixel, you can compute b = 1 - r - g, and then multiply r, g, and b by a constant like 200 to create the color for the transformed image.

Solution-



The histogram displays two distinct chromaticity regions corresponding to the lighting conditions in the image:

- **Bluish cluster (lower-left, darker region)**: Represents the shadowed concrete area. The shadow receives primarily indirect illumination from the sky, which has a blue color temperature. This results in lower r and g values, indicating a relatively higher blue (b) component in the chromaticity.

- **White/yellow region (scattered points extending outward)**: Represents the sunlit concrete surface. Direct sunlight has a warmer, yellowish-white color temperature, resulting in higher r and g chromaticity values compared to the shadowed regions.

Explanation:

The separation between these two clusters demonstrates that shadows on neutral surfaces don't just differ in brightness/intensity - they have fundamentally different chromaticities due to different illumination sources:

- Shadowed areas are illuminated by diffuse skylight (blue)
- Sunlit areas receive direct solar illumination (yellow-white)

Chromaticity Image:
The chromaticity-normalized image shows relatively uniform grey tones because the concrete surface itself is color-neutral. However, subtle differences between shadow and sunlit regions are preserved in the chromaticity representation, with shadows retaining their blue-shifted color properties even after intensity normalization.

The histogram confirms that color information (chromaticity) can be used to distinguish shadows from directly lit surfaces, as they occupy different regions in rg chromaticity space due to the different spectral properties of skylight versus direct sunlight.
