Visual Search of an Image Collection

Nathan Strickland

Master of Engineering
Electronic Engineering with Computer Systems

December 3, 2019

Abstract

The purpose of this report is to explore different visual search techniques using a dataset of 591 images (MSVC-v2) [1]. The performance of a variety of visual search algorithms and distance measures will be analysed by devising experiments using MATLAB [2].

Contents

1 Description of Visual Search Techniques **1**
- 1.1 Image Descriptors . . . 1
  - 1.1.1 Global Colour Histogram . . . 1
  - 1.1.2 Spatial Grid: Colour . . . 1
  - 1.1.3 Spatial Grid: Edge Orientation Histogram (EOH) . . . 2
  - 1.1.4 Spatial Grid: Colour and EOH Combination . . . 2
- 1.2 Principal Component Analysis (PCA) . . . 3
- 1.3 Distance Measures . . . 4
  - 1.3.1 L_1 Norm . . . 4
  - 1.3.2 L_2 Norm . . . 4
  - 1.3.3 Mahalanobis Distance . . . 4

2 Experimental Results **5**
- 2.1 Evaluation Methodology . . . 5
- 2.2 Experimental Results . . . 7
  - 2.2.1 Global Colour Histogram . . . 7
  - 2.2.2 Spatial Grid: Colour . . . 9
  - 2.2.3 Spatial Grid: Edge Orientation Histogram (EOH) . . . 11
  - 2.2.4 Spatial Grid: Colour and EOH . . . 14
  - 2.2.5 Distance Measures . . . 15
  - 2.2.6 Principle Component Analysis (PCA) . . . 16

3 Conclusions **16**

References **17**

1 Description of Visual Search Techniques

1.1 Image Descriptors

1.1.1 Global Colour Histogram

The Global Colour Histogram (GCH) is an image descriptor that represents the average colour distribution of an image. This descriptor does not take into account edges or location of colour, solely it's representation in terms of colour.

To compute the GCH the colour space of each image (for a RGB image the colour space would be 3 dimensional) is quantised into different levels based on the pixel's value, this is then counted and binned into a histogram. The resultant histogram is then normalised and this histogram then represents the image as its descriptor.

Mathematically, in order to compute the GCH several steps are taken to get the pixel values into a workable format.

$$r' = \left\lfloor \frac{r' * q}{256} \right\rfloor \qquad g' = \left\lfloor \frac{g' * q}{256} \right\rfloor \qquad b' = \left\lfloor \frac{b' * q}{256} \right\rfloor \tag{1}$$

$$bin = r' * (q^2) + g' * (q) + b' \tag{2}$$

Equation 1 is used to quantise each colour into q number of regions (q is known as the quantisation level number). This now places q into a region between 0 and $q - 1$ in preparation for binning into the histogram. However, since this is an RGB image the features are in three dimensional space, in order to be used in a histogram it needs to be one dimensional. To make this conversion equation 2 is used. The resultant value can easily be represented as a histogram in the feature space.

The GCH satisfies the need for a compact descriptor but is not discriminative due to spatial information being neglected. As the GCH only takes into account overall colour distribution, images where colours are concentrated in different regions would still be identified as the same image, providing poor search results.

1.1.2 Spatial Grid: Colour

As noted prior, the downside of the GCH was its inability to take into account the spatial features of an image. To overcome this, the image can be split into grids and calculating the average colour of each grid instead of the image as whole. This provides the spatial information required to distinguish between separate areas of concentrated colour. This results in this descriptor being more discriminative than the GCH.

The process is simple, the image is split into grids and then the mean RGB value of each grid is calculated. The image descriptor is then the concatenation of the average colour of each grid.

This method provides a more discriminative image descriptor at the expense of a less compact descriptor. However, this method still does not yield great results - it could be further improved by calculating a local colour histogram at each grid cell. Ultimately this method would provide better performance but would produce a very large image descriptor and would require more computation.

1.1.3 Spatial Grid: Edge Orientation Histogram (EOH)

The Spatial Grid: Edge Orientation Histogram(EOH) uses a similar process as the Spatial Grid: Colour technique in terms of splitting the image into grids. But instead of calculating the mean RGB value of each grid, edge information is calculated and used to compute an edge orientation histogram. In images, edge information is identified by looking for high frequency information.

To obtain the edges of an image an edge detection filter such as a Sobel filter can be used. The Sobel is defined as shown in Equation 3:

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \frac{\partial f}{\partial y} = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & -1 \end{bmatrix} \tag{3}$$

Once an edge has been located using Equation 3 the edge strength and orientation can be calculated using Equation 4:

$$||\nabla f|| = \sqrt{\frac{\partial f^2}{\partial x} + \frac{\partial f^2}{\partial y}} \qquad \theta = arctan \frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}} \tag{4}$$

Once these values have been calculated weak edges are often excluded by comparing the magnitudes to a threshold magnitude, τ. For edges with a magnitude greater than the threshold, their orientations are then quantised into one of θ evenly spaced bins where $\theta \in [0,\ 2\pi]$. This gives us the EOH for each cell.

Similar to the Spatial Grid: Colour descriptor, the image descriptor here is the concatenation of the EOH in each grid cell.

As the EOH only looks at edges/textures within an image, an image with very few, non distinct edges will cause this image descriptor to perform poorly and not be very discriminative. So despite this descriptors dimensionality, the performance may not always be better for all images. Furthermore, this system is sensitive to affine transformations as it is not scale invariant.

1.1.4 Spatial Grid: Colour and EOH Combination

The Spatial Grid: Colour and EOH Combination descriptor combines the Spatial Grid: Colour and Spatial Grid: EOH descriptors. This is done by simply concatenating the two descriptors. This allows the visual search system to take into account colours and textures of an image, all spatially.

Combining the two descriptors hints that the resultant descriptor would lack compactness but would be more discriminative. This trade-off needs to be considered when designing a visual search system and would be dependent on system requirements. Such system requirements would aid in determining the grid sizes. A large number of grids would reduce the compactness of the descriptor, conversely a fewer number of grids may not cover an image correctly and would reduce the discriminative nature of the descriptor.

1.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical method to project data which is in a high dimensional space to a lower dimension. Projecting data to a lower dimension often allows observations to be visualised in a more concise way and redundant values are often removed. This also has the added benefit of reducing the size of an image descriptor, improving its compactness without sacrificing much performance. However there will inevitably be a negative impact on precision due to information loss.

The process of building an eigenmodel is outlined below:

1. Arrange data such that features are aligned one per column.

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & x_3 & ... & x_n \\ y_1 & y_2 & y_3 & ... & y_n \end{bmatrix} \tag{5}$$

2. Calculate the mean, μ.

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{6}$$

3. Subtract the mean, μ, from all the data points.

$$p = x - \mu \tag{7}$$

4. Calculate the covariance matrix from the previous mean subtracted points.

$$\mathbf{C} = \frac{1}{n}\ p.p^T \tag{8}$$

5. As the covariance matrix $\mathbf{C}$ can be factorised in terms of fits eigenvectors, $\mathbf{U}$, and eigenvalues, $\mathbf{V}$ eigen decomposition can be performed. In MATLAB this can be done with the `eig()` function.

$$[\mathbf{U}, \mathbf{V}] = \texttt{eig}(\mathbf{C}) \qquad or \qquad \mathbf{C} = \mathbf{U}\mathbf{V}\mathbf{U}^T \tag{9}$$

Now, using τ, $\mathbf{U}$ and $\mathbf{V}$ the relevant eigenvalues can be found where the data set is the most expressive. Once these values have been identified the data is then projected into a lower dimensional space by moving to a new frame.

1.3 Distance Measures

Once image descriptors have been plotted in a feature space, a distance measure is required to determine the 'similarity' between two image descriptors. This similarity comes from how close the two image descriptors are to each other, the closer the descriptors, the greater the similarity.

This report contains experiments with three different distnace measures, L_1 norm, L_2 norm and the Mahalanobis distance.

1.3.1 L_1 Norm

The L_1 norm, or Manhattan distance, is the sum of the absolute difference between two points in space. If given two vectors $\mathbf{p}$ and $\mathbf{q}$ the L_1 norm can be expressed as:

$$L_{1pq} = ||\mathbf{p} - \mathbf{q}|| = \sum_{i=1}^{n} |p_i - q_i| \tag{10}$$

1.3.2 L_2 Norm

The L_2 norm, or euclidean distance is the distance between two points in space. If given two vectors $\mathbf{p}$ and $\mathbf{q}$ the L_2 norm can be expressed as:

$$L_{2pq} = ||\mathbf{p} - \mathbf{q}|| = \sqrt{\sum_{i=1}^{n} |p_i - q_i|^2} \tag{11}$$

1.3.3 Mahalanobis Distance

The Mahalanobis distance is the measure of the distance between a point and a distribution of all data. More specifically, the mahalanobis distance is the number of standard deviations away from the mean distribution in each principle component axis. Therefore to obtain this measure of distance, PCA must be used.

If given two vectors $\mathbf{p}$ and $\mathbf{q}$ and its eigenvalues, $\mathbf{v}$, the mahalanobis distance can be expressed as:

$$L_{2pq} = ||\mathbf{p} - \mathbf{q}|| = \sqrt{\sum_{i=1}^{n} \frac{|p_i - q_i|^2}{v_i}} \tag{12}$$

2 Experimental Results

The purpose of the following experiments is to analyse the performance of a range of image descriptors and distance measures. The scripts are implemented using MATLAB and the Microsoft Research (MSVC-v2) dataset. The MSVC-v2 dataset was not designed for visual search and such the image labels are not correct therefore not suitable for visual search. To counteract this, a script, `cvpr_image_category.m`, was used to extract the category from the filename of an image. This information is used when calculated the precision and recall of a query, outlined in section 2.1.

In order to achieve fair results across a range of different images, query images with distinct features were chosen, outlined below in Figure 1.

Figure 1: Query images used in visual search testing.

A summary of the experiment details (algorithm parameters, distance measure and other experimental controls) undertaken can be seen below in Table 1.

Table 1: Details of the experiments undertaken.

Image Descriptor	Script	Parameters	Distance
Global Colour Histogram	`cvpr_extract_GCH.m`	Q $\subset$ {2, 4, 8, 16, 32}	L_2 norm
Spatial Grids: Colour	`cvpr_extract_color.m`	(row x col) $\subset$ {2 x 2, 4 x 4, 8 x 8, 16 x 16, 32 x 32}	L_2 norm
Spatial Grids: EOH	`cvpr_extract_EOH.m`	row x col) $\subset$ {2 x 2, 4 x 4, 8 x 8, 16 x 16, 32 x 32} $\theta \subset$ {2, 4, 8, 16, 32} $\tau \subset$ {0.10, 0.20, 0.30, 0.40, 0.50}	L_2 norm
Spatial Grids: Combined	`cvpr_extract_EOHColor.m`	(row x col) $\subset$ {2 x 2, 4 x 4, 8 x 8, 16 x 16, 32 x 32}	L_2 norm

After the experiments in Table 1 have yielded the best parameters for our query class, experiments comparing L_1 norm and L_2 norm are carried out as well as an exploration of the effects of PCA.

2.1 Evaluation Methodology

The performance of a visual search system comes down to its ability to return relevant results with respect to the query image. Thus, to analyse the performance of the visual search algorithms implemented, two statistics must be calculated, the precision and the recall.

To compute the precision and the recall of a visual search system the relevant images in the dataset (with respect to the query) are marked - this is the ground truth. Once the ground truth has been identified a query image is ran through the system and the precision and recall is calculated using Equation 13 and 15.

The precision of the system is the fraction of retrieved images that are marked as relevant:

$$precision = \frac{|\{relevant\ images\}\ \cap\ \{retrieved\ images\}|}{|\{retrieved\ images\}|} \tag{13}$$

The recall of the system is the fraction of relevant images that are retrieved successfully:

$$recall = \frac{|\{relevant\ images\}\ \cap\ \{retrieved\ images\}|}{|\{relevant\ images\}|} \tag{14}$$

Using the results from Equation 13 and 15 a precision-recall graph can be plotted to visualise the results. An ideal visual search system would have a precision of 1 and a recall of 1. However, this is unlikely to be the case, a more likely result can be seen in Figure 2.

To evaluate the visual search system for an entire query class the average precision (AP) is also calculated:

$$AP = \sum_{n=1}^{M} \frac{P(n)rel(n)}{|\{relevant\ images\}|} \tag{15}$$

where N = [1, M] and M is the size of the dataset, P(n) is the precision of the top 'n' results and rel(n) is 1 if n is relevant otherwise 0.

Finally, to evaluate the performance of the system as a whole across all 6 query classes, the mean average precision (MAP) is calculated:

$$MAP = \sum_{i=1}^{no.\ query\ classes} \frac{AP_i}{no.\ queryclasses} \tag{16}$$

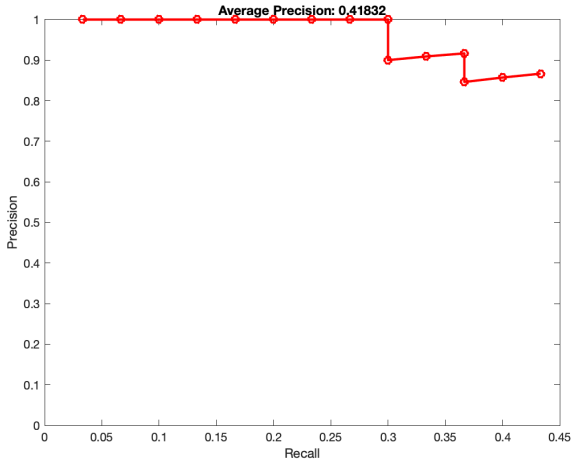


Figure 2: Example PR curve for a Spatial Grid: Colour and EOH system

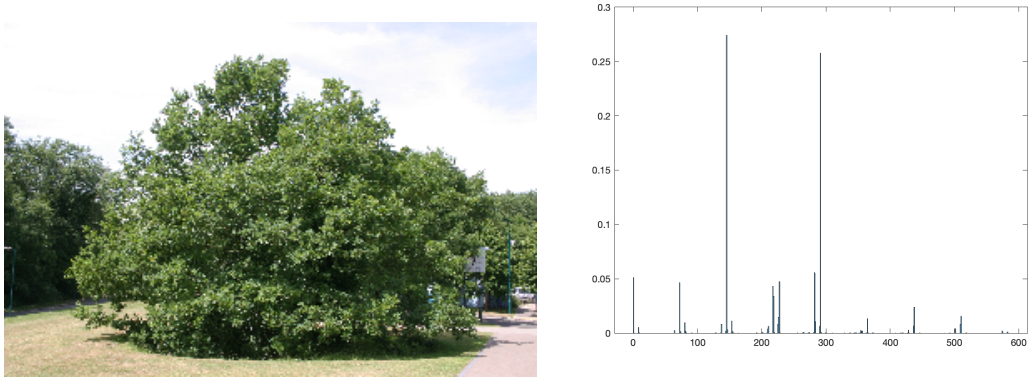
2.2 Experimental Results

2.2.1 Global Colour Histogram

Methodology

To evaluate the performance of the global colour histogram descriptor (GCH), the MATLAB script `cvpr_extract_gch.m` was used. The function takes in a normalised coloured image `img` and a quantisation level, `Q`. The function then calculates the average red, green and blue values and bins the quantised pixel values into a histogram. This histogram is then normalised. The function then returns the histogram as the image descriptor, `F`.

Figure 3 shows an example of the global colour histogram extracted from a test image.



(a) Query image (b) Global colour histogram of image

Figure 3: Example global colour histogram generated from a test image.

To evaluate the performance of the GCH, `Q`, was varied to analyse the effect of changing quantisation level on the average precision of the visual search. The following values of `Q` were used, `Q` $\subset \{2, 4, 8, 16, 32\}$.

Results

Table 2: Average precision of global colour histogram visual search for quantisation levels, Q $\subset \{2, 4, 8, 16, 32\}$.

	Average Precision				
Query Class	Q = 2	Q = 4	Q = 8	Q = 16	Q = 32
Trees	0.156	0.162	**0.198**	0.181	0.170
Flowers	**0.169**	0.120	0.0969	0.102	0.0949
Cows	**0.0555**	0.0332	0.0332	0.0332	0.0332
Face	0.0521	0.0940	**0.105**	0.101	0.106
Planes	0.0926	**0.0940**	0.0832	0.0667	0.0556
Signs	0.0332	**0.0443**	0.0332	0.0332	0.0332
MAP	**0.0931**	0.0913	0.0916	0.0862	0.0822

From Table 2 it is seen that Q = 2 offers the best average precision across all 6 query classes. Although, the Q = 4 and Q = 8 are close. It can be seen that as Q increases the average precision begins to decrease, suggesting sacrificing compactness in a feature does not always yield better results. At high

quantisation levels the pixel values are divided in such a way that the image descriptor cannot identify features correctly, thus reducing the average precision.

Conclusions

Regardless of the values of Q, the global colour histogram does not provide an accurate set of results in visual search. Relying solely on the average colour of an image is not discriminative enough as completely different images can have similar average colours. For example, the sky and ocean can be similar colours but are not the same.

2.2.2 Spatial Grid: Colour

Methodology

To evaluate the performance of the colour spatial grid descriptor the script `cvpr_extract_color.m` was used. The function has three input parameters; `img` - the normalised image to calculate the descriptor for, `grid_rows, grid_cols` - the dimensions of the grid. The function splits the image into grids respective of the previous parameters. The function then calculates the average red, green and blue pixel values in each grid cell. The final image descriptor is the concatenation of the average colour in each grid cell.

Figure 4 shows an example of how an image is split into a 2 x 2 grid. The image descriptor for this image would be the concatenation of the average colour of each of the 4 cells.

Figure 4: Image split into a 2 x 2 grid

For this descriptor we experiment with different grid dimensions, (`grid_rows x grid_cols`) $\subset$ {2 x 2, 4 x 4, 8 x 8, 16 x 16, 32 x 32}. The results can be seen in Table 3 below.

Results

Table 3: Average precision of a spatial grid colour visual search for grid sizes, (row x col) $\subset$ {2 x 2, 4 x 4, 8 x 8, 16 x 16, 32 x 32}.

	Average Precision				
Query Class	2 x 2	4 x 4	8 x 8	16 x 16	32 x 32
Trees	0.227	0.290	0.317	0.319	**0.322**
Flowers	**0.164**	0.163	0.119	0.116	0.0935
Cows	0.0744	**0.123**	0.0623	0.0596	0.0610
Face	**0.0853**	0.0656	0.0385	0.0378	0.0333
Planes	0.322	0.406	**0.413**	0.379	0.319
Signs	**0.0591**	0.0556	0.0333	0.0333	0.0333
MAP	0.155	**0.184**	0.164	0.157	0.144

From Table 3 it is seen that a 4 x 4 grid offers the best results. This is again an example of how having a larger image descriptor (in terms of dimensions) does not necessarily result in more accurate results. In this case a 2 x 2 grid is not discriminative enough and still has the problem seen in the global colour histogram where the location colour is not taken into account. At larger grid sizes, the average precision again beings to deteriorate due to the pixel values being divided in a high density, as a result the image

descriptor cannot recognise unique features correctly. It is also possible that the key features in an image are not found in the same grid, but adjacent girds, which result in some relevant images not being returned by the search.

Conclusions

The coloured spatial grid image descriptor provides a significantly more accurate result than the global colour histogram (49.4%) across all six query classes. The 4 x 4 grid provides the greatest precision across each class, however all grid sizes perform better than the global colour histogram. It is worth noting that this improvement comes at the expense of compactness, the coloured spatial grid descriptor is larger and will take longer to compute, which would be noticeable on large images and large datasets.

2.2.3 Spatial Grid: Edge Orientation Histogram (EOH)

Methodology

The spatial grid edge orientation histogram is implemented using the script `cvpr_extract_EOH.m`. The function takes in 4 arguments; the normalised image (`img`), the grid dimensions (`grid_rows, grid_cols`), the threshold magnitude (τ) and the number of quantisation levels (θ).

The function splits the image into grids using the method seen in the spatial grid colour script. The function first converts the image into a grayscale image and then convolves the grayscale image with the Sobel filters, equation 3, to identify the edges.

Once the edges are identified, the gradient magnitudes and orientations are calculated using Equation 4. To eliminate weak edges, any edges with a magnitude less than the defined threshold (τ) are ignored. The edge orientation is then adjusted to be in the range $[0 - 2\pi]$ in preparation for the histogram binning process.

Now, for each pixel within a grid cell, the edge orientations are binned with respect to the quantisation level, θ. (If $\theta = 8$ there would be 8 histogram bins of equal width). This process is repeated for all grid cells and the final image descriptor is a concatenated feature vector of all the binned values normalised.

As an example of the above Figure 5 shows the result of extracting the edge magnitudes and orientations from an image. These values are then used to create the edge orientation histogram in Figure 6 which is the final image descriptor for the image.

(a) Visualisation of edge magnitudes. (b) Visualisation of edge orientations.

Figure 5: Visualisation of the edge magnitude and orientation extracted from an image.

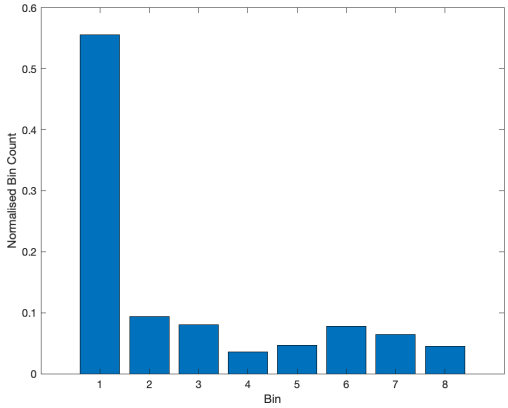


Figure 6: EOH image descriptor for an image with 4 x 4 grids, $\theta = 8$, $\tau = 0.10$.

For this visual search algorithm there are 3 parameters that can be varied, the grid size, the quantisation level and the threshold magnitude. Initially the grid size was changed, (row x col) $\subset$ {2 x 2, 4 x 4, 8 x 8, 16 x 16, 32 x 32}, then using the best result the quantisation level was changed, $\theta \subset \{2, 4, 8, 16, 32\}$. Finally using the previous two results, the threshold magnitude was adjusted, $\tau \subset \{2, 4, 8, 16, 32\}$.

Results

Table 4: Average precision of a spatial grid EOH ($\theta = 4$, $\tau = 0.20$) visual search for grid sizes, (row x col) $\subset$ {2 x 2, 4 x 4, 8 x 8, 16 x 16, 32 x 32}.

	Average Precision				
Query Class	2 x 2	4 x 4	8 x 8	16 x 16	32 x 32
Trees	0.0467	0.123	0.169	**0.199**	0.0911
Flowers	0.0312	0.0312	**0.0354**	0.0312	0.0312
Cows	**0.0972**	0.0500	0.0385	0.0622	0.0733
Face	0.0533	0.0758	**0.0801**	0.0577	0.0333
Planes	0.134	**0.332**	0.275	0.138	0.0792
Signs	**0.0500**	0.0417	0.0333	0.0333	0.0333
MAP	0.0687	**0.109**	0.105	0.0869	0.0569

From Table 4 it is seen that the 4 x 4 grid provides the greatest mean average precision across all 6 query classes. This is due to similar reasons as the spatial grid colour algorithm. The 2 x 2 grid is not discriminative enough and doesn't take into account the spatial information accurately. Also at larger grid sizes the edge orientations and magnitudes are being divided at too high a density.

Table 5: Average precision of a spatial grid EOH (grid = 4 x 4, $\tau = 0.20$) visual search for quantisation levels, $\theta \subset \{2, 4, 8, 16, 32\}$.

	Average Precision				
Query Class	$\theta = 2$	$\theta = 4$	$\theta = 8$	$\theta = 16$	$\theta = 32$
Trees	0.127	0.123	0.144	0.146	**0.147**
Flowers	0.0312	0.0312	0.0357	**0.0427**	0.0365
Cows	0.0400	**0.0500**	0.0467	0.0444	0.0417
Face	0.0778	0.0758	0.113	0.110	**0.0956**
Planes	0.303	0.332	**0.350**	0.345	0.343
Signs	**0.0467**	0.0417	**0.0467**	0.0444	0.0429
MAP	0.104	0.109	**0.123**	0.122	0.118

Now when using a 4 x 4 grid, it can be seen from Table 5 that a quantisation level of $\theta = 8$ provides the most accurate results across our query classes. It is interesting to note that for the chosen query classes the difference in average precision between levels of θ is not significant. At low quantisation levels, the average precision is lower as the image descriptors are not discriminative enough to return relevant results. Likewise, at high quantisation levels the performance does not improve, again proving that image descriptors with more dimensions do not necessarily perform better.

Table 6: Average precision of a spatial grid EOH (grid = 4 x 4, $\theta = 8$) visual search for threshold magnitudes, $\tau \subset \{0.10, 0.20, 0.30, 0.40, 0.50\}$.

	Average Precision				
Query Class	$\tau = 0.10$	$\tau = 0.20$	$\tau = 0.30$	$\tau = 0.40$	$\tau = 0.50$
Trees	**0.350**	0.144	0.0471	0.0870	0.0917
Flowers	0.0382	0.0357	0.0521	**0.0563**	0.0312
Cows	0.0444	0.0467	**0.0887**	0.0555	0.0378
Face	0.0744	**0.113**	0.0722	0.0444	0.0407
Planes	0.319	**0.350**	0.279	0.165	0.151
Signs	0.0333	0.0467	**0.0827**	0.0528	0.0429
MAP	**0.143**	0.123	0.104	0.0768	0.0659

Finally, when using a grid size of 4 x 4 and quantisation level, $\theta = 8$, the threshold magnitude can be adjusted. From Table 6 it is seen that a threshold magnitude of $\tau = 0.10$ provides the most accurate results. Setting the threshold magnitude too high eliminates a lot of the edges in the image causing the resultant image descriptor to be in-discriminative resulting in poor average precision. Likewise, setting the threshold magnitude lower then 0.10 would identify edges in images incorrectly.

Conclusions

Interestingly, in terms of the mean average precision, the edge orientation histogram performed slightly worst than the mean colour grids. However, this is mostly due to the query classes chosen where certain classes performed much worst than others. For example, the edge orientation histogram provided more accurate results for trees, cows and planes (prominent edges) whereas the mean colour grids provided significantly better results for images with lots of colour such as flowers.

Across a larger range of query classes it would be expected that the edge orientation histogram would perform better than the mean colour grids, however as noted this is enitrely dependent on the type of image.

2.2.4 Spatial Grid: Colour and EOH

Methodology

The Spatial Grid: Colour and EOH image descriptor is the concatenation of both the spatial grid colours and spatial grid edge orientation histogram. This image descriptor is implemented in the `cvpr_extract_EOHColor.m` script.

As experiments to find the optimal parameters for the EOH and mean colour have already been undertook, the only variable requiring variation for this experiment is the grid size, (row x col) $\subset$ {2 x 2, 4 x 4, 8 x 8, 16 x 16, 32 x 32}.

Results

Table 7: Average precision of a spatial grid EOH and Colour combination ($\theta = 8$, $\tau = 0.1$) visual search for grid sizes, (row x col) $\subset$ {2 x 2, 4 x 4, 8 x 8, 16 x 16, 32 x 32}.

	Average Precision				
Query Class	2 x 2	4 x 4	8 x 8	16 x 16	32 x 32
Trees	0.0777	0.345	**0.382**	0.358	0.251
Flowers	0.0938	**0.122**	0.101	0.0742	0.0469
Cows	0.0594	0.0703	0.0802	0.0740	**0.113**
Face	**0.0867**	0.0544	0.0400	0.0333	0.0333
Planes	0.292	0.388	**0.418**	0.305	0.168
Signs	0.0417	**0.0467**	0.0429	0.0333	0.0333
MAP	0.109	0.171	**0.177**	0.146	0.108

Table 7 indicates that the 8 x 8 grid size returns the most precise results. Furthermore, from Table 7, certain query classes have a very high precision such as planes at 0.418 and trees at 0.382. When using the image descriptors individually these query classes also perform well so the concatenation of the two further improves the result.

Again however, the descriptor does not perform well for complex images with many edges and colours such as signs and faces, ultimately reducing the mean average precision of the descriptor.

Conclusions

It would be expected that the combined EOH and colour spatial grids would provide more precise results than the individual descriptors, however in this situation this is not the case. In fact, the combined result lies in between the two individual descriptors (0.143 | **0.177** | 0.184 (when tested with the best parameters)). Currently the combined descriptor is concatenated with equal weighting and as noted before, the colour spatial grids perform better for our query images due to the mean colour being more discriminative than the edges, for our chosen images.

In order to achieve more precise results for the combined EOH and colour, one of the image descriptors could be multiplied with a constant to increase its weighting and give a greater emphasis on certain descriptors. In this case, the EOH descriptor needs scaling as currently the colour spatial grid descriptor is overwhelming it.

Finally, it is worth noting that both the concatenation of the image descriptors and the individual use of the spatial grid EOH and colour all provide a greater mean average precision than the global colour histogram.

2.2.5 Distance Measures

Methodology

The prior experiments have been undertaken using L_2 norm as the measure of distance. The purpose of this experiment is to see how changing the distance measure to L_1 norm effects average precision. The distance measure L_1 norm is implemented in the script `cvpr_compare_l1_norm.m`.

In terms of parameters for the image descriptors, we use the parameters that gave the greatest average precision in prior experiments, Q = 2, $\theta = 8$, $\tau = 0.1$, where applicable.

Results

Table 8: Average precision of each visual search algorithm, using best parameters, (Q = 2, $\theta = 8$, $\tau = 0.1$, where applicable) for L_1 norm $\parallel$ L_2 norm distance measures.

	Average Precision			
Query Class	Global Colour	Colour Grid (4x4)	EOH Grid (4x4)	Hybrid Grid (8x8)
Trees	0.161 $\parallel$ 0.156	0.340 $\parallel$ 0.290	0.423 $\parallel$ 0.350	0.457 $\parallel$ 0.382
Flowers	0.142 $\parallel$ 0.169	0.127 $\parallel$ 0.163	0.0474 $\parallel$ 0.0382	0.0865 $\parallel$ 0.101
Cows	0.0639 $\parallel$ 0.0555	0.0755 $\parallel$ 0.123	0.0394 $\parallel$ 0.0444	0.0863 $\parallel$ 0.0802
Face	0.0721 $\parallel$ 0.0521	0.0385 $\parallel$ 0.0656	0.0667 $\parallel$ 0.0744	0.0573 $\parallel$ 0.0400
Planes	0.0851 $\parallel$ 0.0926	0.4037 $\parallel$ 0.406	0.392 $\parallel$ 0.319	0.467 $\parallel$ 0.418
Signs	0.0333 $\parallel$ 0.0332	0.0667 $\parallel$ 0.0556	0.0333 $\parallel$ 0.0333	0.0491 $\parallel$ 0.0429
MAP	0.0929 $\parallel$ **0.0931**	0.175 $\parallel$ **0.184**	**0.167** $\parallel$ 0.143	**0.200** $\parallel$ 0.177

From Table 8 it can be seen that L_1 norm out performs L_2 norm for certain descriptors. L_1 norm performs significantly better on both EOH grid cells and the combined EOH & Colour grid cells (most likely due to the improved standalone performance in EOH). This improvement is seen as L_2 norm finds the shortest distance between two points which is not as robust in high dimensions (will not take into account every dimension), this gives rise to the visual search system finding outliers more often. However, L_1 norm has greater sparsity as it provides multiple solutions, therefore reducing the number of outliers at higher dimensions.

L_2 norm only performs marginally better on the global colour histogram and colour grid descriptors.

Conclusions

From the results the logical output would be labelling L_1 norm as the superior measure of distance for visual search systems. This is due to its scalability and ability to handle image descriptors with a high number of dimensions effectively.

However, there is no one size fits all, L_2 norm can perform better at lower dimensions, the choice of distance measure should depend on the image descriptor being used.

2.2.6 Principle Component Analysis (PCA)

Methodology

The purpose of PCA is to project image descriptors to a lower dimensional space while maintaining similar performance in terms of average precision. The goal is to reduce the size of a descriptor without effecting its ability to return relevant images. PCA is used in conjunction with the Mahalanobis distance to attempt to achieve this.

The script `cvpr_pca.m` implements PCA and the script `cvpr_compare_mahalanobis.m` implements the mahalanobis distance measure.

As in the distance measure experiment, we use the best parameters for each image descriptor identified in prior experiments. The PCA is also set so that it retains 90% of the original image descriptors.

Results

Table 9: Average precision of each visual search algorithm (Q = 2, $\theta = 8$, $\tau = 0.1$, where applicable) using the Mahanalobis distance and PCA ‖ No PCA and L_2 norm distance measure.

	Average Precision			
Category	Global Colour	Colour Grid (4x4)	EOH Grid (4x4)	Hybrid Grid (8x8)
Trees	0.182 ‖ 0.156	0.257 ‖ 0.290	0.213 ‖ 0.350	0.283 ‖ 0.382
Flowers	0.0905 ‖ 0.169	0.0936 ‖ 0.163	0.0437 ‖ 0.0382	0.0742 ‖ 0.101
Cows	0.0556 ‖ 0.0555	0.0675 ‖ 0.123	0.0407 ‖ 0.0444	0.0556 ‖ 0.0802
Face	0.0333 ‖ 0.0521	0.0500 ‖ 0.0656	0.0667 ‖ 0.0744	0.0667 ‖ 0.0400
Planes	0.0811 ‖ 0.0926	0.363 ‖ 0.406	0.297 ‖ 0.319	0.431 ‖ 0.418
Signs	0.0333 ‖ 0.0332	0.0333 ‖ 0.0556	0.0333 ‖ 0.0333	0.0394 ‖ 0.0429
MAP	0.0793 ‖ **0.0931**	0.144 ‖ **0.184**	0.116 ‖ **0.143**	0.158 ‖ **0.177**

As seen in Table 9, the mean average precisions with PCA are approximately 10 - 15% lower than without PCA. This suggests that PCA is successfully reducing the dimensions of the image descriptors and the Mahalanobis distance measure is adjusting for the sparsity in the image descriptors.

The reduction in average precision is expected due to the data loss with PCA.

Conclusions

PCA can be an extremely useful tool for increasing the compactness of very large image descriptors. Such size reduction is often necessary for very large descriptors as without it a visual search could be slow and require a lot of computational power. PCA enables image descriptors to become more compact without sacrificing the discriminative nature of the descriptor.

3 Conclusions

This report has explored only a few basic visual search techniques, only scratching the surface of the field of visual search. From the few image descriptors looked at, a wide range of adjustable parameters were identified, all having an impact on the performance of a descriptor. All these parameters have to be considered when designing a visual search system, this proves how complex a task this can be.

Individual conclusions for each image descriptor can be found in the respective section of the experimental results.

References

[1] Microsoft, MSRC Object Category Image Database v2. United Kingdom: Microsoft., 2005. [2] MATLAB, R2019b. Natick, Massachusetts: The MathWorks Inc., 2019.