Image Retrieval for Visual Understanding in Dynamic and Sensor Rich Environments

Noah Lesch, Andrew Compton, John Pecarina, Michael Ippolito, Douglas Hodson
Air Force Institute of Technology
United States Air Force
Wright-Patterson Air Force Base, USA
{*noah.lesch*, *andrew.compton.1*, *john.pecarina*, *douglas.hodson*}*@us.af.mil*, *michael.p.ippolito@gmail.com*

***Abstract*— Vision is vital to decision making, as humans naturally trust their eyes to enhance situation awareness. Yet the modern age has overwhelmed humans with massive amounts of visual information, which is problematic in time sensitive and mission critical situations, such as emergency management and disaster response. More efficient search and retrieval systems address some of these issues, which is why many seek to develop and extend Content Based Image Retrieval (CBIR) techniques to support situational awareness in a more autonomous fashion. However, there is currently no adequate system for CBIR to support situational awareness in dynamic and sensor rich environments. This research proposes an extensible framework for CBIR to support a holistic understanding of the environment through the automated search and retrieval of relevant images and the context of their capture. This constitutes assisted CBIR as embodied in the multi-sensor assisted CBIR system (MSACS). We design the MSACS framework and implement the core CBIR system of MSACS using the state of the art Bag of Visual Words paradigm. The system is evaluated using a dataset of GPS tagged images to show favorable precision and recall of spatially related images. Applications for localization and search for Wi-Fi access points demonstrate improved situational awareness using the system. Assisted CBIR could enable vision based understanding of an environment to ease the burdens of information overload and increase human confidence in autonomous systems.**

Keywords—Content Based Image Retrieval; Computer Vision;

I. INTRODUCTION

Humans rely upon visual information and understanding when operating in time sensitive and mission critical situations. However the exponential increase in visual information in the present time has not guaranteed increased effectiveness of the human decision maker. This is most problematic in dynamic environments such as those that characterize disaster response and emergency management where situations change rapidly. Time sensitive and mission critical implications may demand reestablished visual perception of the dynamic environment.

Human instincts for visual understanding could be supported with wide scale distribution of cameras in the environment either deployed by the responding organization or from voluntary contribution by smartphones. Distribution often includes its own issues with information overload and data quality from large sets of contributed images that are poorly labeled and/or indexed. There are many potential applications for automating visual understanding, which include change detection for detecting emergent events [1], object recognition for identifying structures, landmarks or items of interest (guns, cars, fires, etc.) [2], facial recognition to detect persons of interest [3], structure from motion (SFM) [4] to create a 3D image of an area affected by a disaster, simultaneous localization and mapping (SLAM) [5], visual odometry [6] and localization for navigating an environment affected by loss or unavailability of GPS. Many of these applications often require images that bear similarity to or match other images and make use of image features for the computer to automate visual understanding.

By extension, a framework for dynamically building and retrieving useful images from image repositories based on the image content could improve processing efficiency, thus, situational awareness. Content based image retrieval (CBIR) systems return a set of ranked digital images to a user based on a query; where queries come in many forms, such as text and images. The system can be used to parse, query, filter and catalog large amounts of images to automatically feed computer vision applications, bridging the gap between human and computer understanding of dynamic environments.

A CBIR system that is assisted by other sensed information could provide further granularity for the application. Hardware miniaturization and advanced processing algorithms have enabled a sensor rich environment. The proliferation of sensors is an everyday facet of modern life, as smartphones and the internet of things find increased utilization. One may benefit from magnetometers, barometers, or Wi-Fi sensors as one builds an operational view of a dynamic environment but correlating image and sensor data can be difficult if not linked at the point of capture. This information could assist in understanding the context of an image and may improve image search. However, no system integrates the rich sensor environment with indexing, search and retrieval of the image repositories.

The opportunity to exploit images from a dynamic, sensor rich environment calls for *assisted CBIR*, which integrates search and retrieval mechanisms for image content with other environmental context. We introduce a framework for assisted CBIR with the multi-sensor assisted CBIR system (MSACS). This framework is supported by a CBIR system which utilizes the Bag of Visual Words (BoVW) approach for image search and retrieval. We utilize Google Map's street view to automate the building of an image repository to be stored in the open source database Cassandra. The database enables updates for a self-building world model and queries to improve situational awareness. A localization application demonstrates the ability to leverage GPS information in the event of outages and Wi-Fi Access Point lookups.

This paper is organized as follows: Section 2 discusses the background and related works of CBIR and CBIR frameworks. Section 3 discusses the MSACS system framework requirements and system overview to implement assisted CBIR. Section 4 describes the implementation and evaluation of our working system, and finally, in Section 5, we present conclusions and discuss future work.

II. RELATED WORK

CBIR has been hypothesized for situational awareness scenarios to aid, for example, emergency services or forensic inquires. In [7-8] the authors suggest moving away from a text-based image retrieval system for hospitals and police stations because the text annotations, or metadata, are subjective in nature. Instead they propose using CBIR to quantize color and texture data to link together mugshots with visual evidence, or by proposing a patient's diagnosis based on past diagnoses using x-rays, scans, etc. Similarly to [7-8], in [9] the author proposes a CBIR system for matching visual evidence with past cases to determine patterns and matches. This system also uses relevance feedback which applies user inputs to assist in narrowing down CBIR results. In [10] the authors use color and texture features to create a CBIR system to detect wildfires within images. These results are annotated with social media tags from Flickr and used for information dissemination during the Colorado wildfires of 2012. The results showed that CBIR along with social media serve as an alternative source of information dissemination and possible emergency services mobilization. This credits the concept of employing CBIR for dynamic situations such as disaster response and emergency management, but none of the prior works have aimed to incorporate a self-building image repository and capture environmental context from sensors.

CBIR searches a digital library (e.g., a database) of images using the visual content extracted from a query image. This visual content can be color, texture, shape, or any salient data able to be extracted from an image. To extract data from an image requires the use of open-source or proprietary algorithms. Parsing an image with a feature extraction algorithm will quantize the visual content within the image. Once this visual information is quantized it can be stored in a database along with any other relevant information associated with the image such as file name, location, etc. for future search and compare operations. In our work, we use the state of the art Bag of Visual Words paradigm [11] since it abstracts important features for situational awareness applications seen in scene reconstruction, navigation and object recognition.

Several CBIR frameworks propose to extract and quantize image features, store quantized data in a database, and return similarity results based on a user query to the system. In [12] the authors create a framework in which the CBIR system relies on multiple types of image features, which were referred to as multimodal features. The CBIR system extracts color, texture, and shape features from an image. Multimodal features are used in CBIR systems because [13] demonstrated that CBIR systems which use only one type of image feature do not perform as well as those which use multiple features. Therefore our system is designed to make use of multiple feature types to improve its image retrieval capability.

There are frameworks that perform CBIR based on hybrid features and fused features. In [12] the authors combine color, texture, and shape features and create a feature vector. In [14] the authors create a framework for color image retrieval by combining low level features of color, texture, and shape information in an image. In [15] the authors use multiple shape features as the means of image retrieval. The first, second, third, and fourth shape moments are extracted which creates a global shape descriptor. They also implement probabilistic classification clustering to classify global shape descriptor as a known class. Once the class is known, a comparison of the query to images of that class is conducted. The authors of [16] implement a multimodal CBIR framework using speeded up robust features (SURF), contour and edge detection, and text search. Thus, their CBIR framework makes use of not only image content, but also textual data. Our framework could make use of image features along with other data as in [16]; however our extraneous data is collected from sensors which at the time of image capture. This sensor data will be used in our modular framework to improve search retrieval and accuracy.

CBIR is used to support applications as in [17], where the authors use latitude and longitude data to automatically annotate new images captured and uploaded to Flickr. Thus, these new images within a similar geographical location will be tagged with labels from other tagged images that currently exist in the Flickr database. Our system replicates this work to demonstrate its usefulness and adds Wi-Fi access point lookups through the Wigle API [18].

With these areas of study in the domains and application of image retrieval, we pursue a framework and implementation that extends towards assisted CBIR to improve situational awareness.

III. MSACS FRAMEWORK

Assisted CBIR will extend the ability of autonomous entities to understand the environment by linking the content of images with sensed inputs. In this section, we expose the design process for the Multi Sensor Assisted CBIR System (MSACS) by developing requirements, describing the system and its key components.

A. Operating Concept

The concept of multi-sensor assisted CBIR is the integration of sensed information in the environment at the moment of image capture within the indexing, search and retrieval process. This concept can be distributed in its operation, allowing for populate/update operations to be distinct from query/response ones. In this way, known information from the populate/update stage can be linked to real-time queries. The results from the queries can be used to propel applications that gain situational awareness of the environment through image repositories.

The assisted CBIR operating concept is illustrated in Fig. 1. As shown, cameras and sensors in the bottom half of the figure may publish images annotated with sensed information within a populate/update use case. As frequent updates are essential to minimizing consistency issues between world model and reality; the top half of the figure illustrates use cases that support this objective through queries/responses from autonomous entities and users. The concept also supports image queries to receive similar images as returned values. Non-image queries such as Wi-Fi, magnetometer, and other modules can also retrieve images of relevance. These are built upon a distributed network and database to support scalability for world model building and querying. The proposed method of naming images would concatenate a unique node name with a time stamp created during image capture.

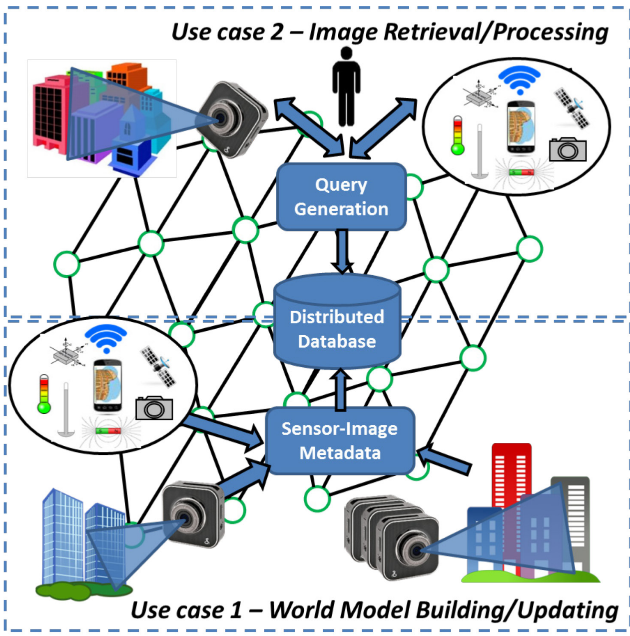

Fig. 1. Assisted CBIR operating concept

B. MSACS System Requirements

Because the framework needs to ingest, parse, and maintain large amounts of text data related to image and sensor features from several sources, this drives four requirements. First, the data obtained from the sensor modules must be associated with an image and incorporated in search and retrieval. Thus at a minimum an image is required to perform the most basic form of CBIR. With this, the core functionality of image retrieval is necessary and we aim to support simple or complex processing algorithms to do this, such as state of the art for CBIR, the BoVW paradigm.

Second, the framework has to be extensible. The extensible sensor modules (implemented in Python), define simple interfaces that includes a data reading function and a database populate function. Configuration files will be used to specify proper relationships between processing modules at the time of operation.

Third, data should be stored in well-established formats such as XML or JSON. It was decided that JSON be used, as it is more lightweight than XML; any existing XML data is converted to JSON as it is parsed. A scalable, distributed database will allow this data to propagate efficiently if used in a networked environment. A database function within the module file instructs the framework to create new columns in corresponding to the type of information collected by that module.

Fourth, we need to expose raw and processed data to support applications. In this use case, a user queries the MSACS application programming interface (API) with data from a camera, Wi-Fi sensor, and magnetometer. For example, a user may perform self-localization in an unfamiliar environment. The localization application can access data through the API. The following summarizes our requirements for MSACS.

1. The framework shall support image-based CBIR using simple or complex processing, such as BoVW, and incorporate these additional inputs to improve search and retrieval. (Assisted CBIR).
2. The framework shall be extensible, supporting multiple sensor inputs and processing of these inputs. (Extensible)
3. The framework shall use a standardized data format, and be supported by a scalable, common database across platforms. (Scalable and Portable)
4. The framework shall support post processing to enable image annotation and application interfaces. (API)

These requirements serve as the basic design concepts for our prototype of MSACS. They address the basic high level design decisions in our system, detailed next.

C. MSACS System Overview

The core of the MSACS is displayed in Figs. 2 and 3. As shown in Fig. 2, MSACS incorporates data obtained from additional sensors. As data is captured it is stored in file-based folder structures enabling it to be assigned a time stamp (t_0, t_1, …, t_n) for first-in first-out processing operations. The types of captured data processed depend on the modules currently loaded and incorporated into the framework.

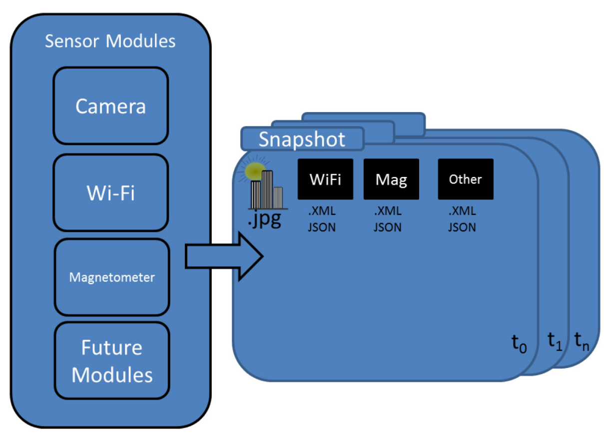

Fig. 2. Proposed assisted CBIR modules

All data is read into the framework as XML or JSON, with the exception of the image capture. Each sensor has a specific framework module which parses sensor-related data to extract and then quantize information for storage in a database. For illustration purposes the data presented in Fig. 3 is shown as XML. After quantization the individual feature vectors are appended to the image as annotations and archived for later retrieval. The database will be available for search and retrieval operations once it contains data. This data supports a localization application tied to our first use case.

Our framework is not limited to a single database. However for the purposes of this prototype only a single database is used. Cassandra databases have the capability to shard and exchange information depending on their configuration, so we select this as our database due to the potential scaling needs.

To the right of Fig. 3, applications can access the MSACS API to make use of the assisted CBIR data. For example, a localization application can utilize the known latitude and longitude information in the database and provide an inference of the latitude and longitude for the query image. This is accomplished by averaging the latitude and longitude of the images in the return set. The cardinality of the return set is user defined in a framework configuration file. The performance of the localization application is affected by the return set size as shown in the evaluation section. The result of the localization application is a pair of latitude and longitude coordinates returned to the user.

D. *Assisted CBIR Description*

Assisted CBIR extends the BoVW approach to support MSACS, as shown in Fig. 3. The module consists of a model generation stage and a query stage. The BoVW generation stage extracts features, performs clustering, generates a codebook, and then populates a database against which query images can be compared. The query stage performs an exhaustive search and comparison of query images within the database.

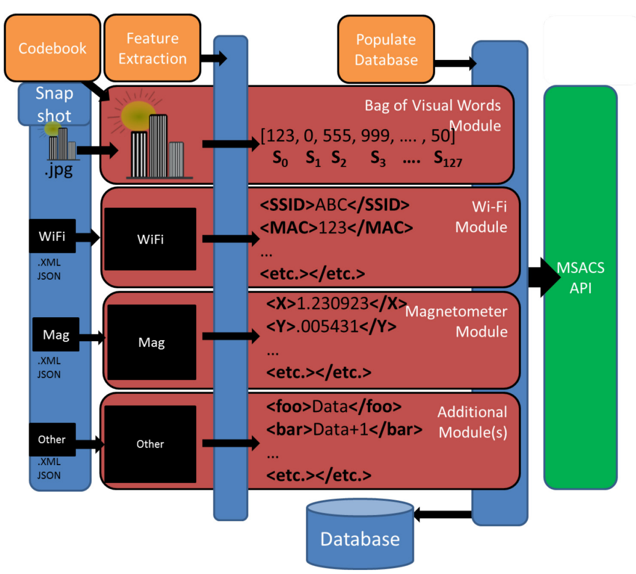

Fig. 3. MSACS framework

BoVW [11] is an adaptation of Bag of Words (BoW) information retrieval paradigm generally associated with text documents. In BoW a histogram of word frequencies in a document is computed. This BoW histogram can then be used for information retrieval problems. An inverted index can be created from the BoW histogram to give greater search weight to terms that occur less frequently in a document. This improves a document's searchability within a database.

In BoVW a similar approach is taken. In BoVW style CBIR, SIFT features [19] are extracted from images. For even greater codebook discrimination, search engines often utilize the RootSIFT [20] feature to describe the image features in the corpus. A codebook is generated by quantizing image features by using a clustering algorithm over all detected features. The clustering algorithm, typically k-means, determines k-unique cluster centers [21]. To calculate the k cluster centers, an iterative process initializes with k random centers. In the iterations, feature vectors are matched to the closest cluster center and new cluster centers are chosen as the mean of all assigned features. These k cluster centers become the codebook which is later used to describe query images. The k-value for k-means clustering is user-defined. The codebook is maintained separately from the database. Codebooks can be precomputed for different environments. A codebook from one environment will not provide the same performance as a codebook intended for another environment.

Each cluster center in our codebook represents one word in our BoVW model. All images in the repository are described by how their extracted features are quantized according to the codebook. In essence, the codebook is used to create a BoVW histogram for each image. This BoVW histogram can then be compared to other BoVW histograms from a dataset or query to determine a similarity or distance metric. This method is faster than matching features across an entire corpus every time a query is submitted.

In the query stage the images in the query set are compared against those in the dataset. The RootSIFT algorithm is applied to these images and a k-bin BoVW histogram based on the codebook is created for each image. Each query image's histogram is then exhaustively compared against the corpus histograms in the database. The user is able to specify the top L results from the image search.

Because the histograms can also be represented as quantized vectors, the distance between them can be calculated as a similarity metric. For the purposes of this CBIR system the cosine distance is calculated between the query image histogram and each image in the corpus. The images in the corpus are ranked by the images that have the largest cosine distance from the query. For two histograms, a query image Q=(Q1,Q2,…,Qk) and a dataset image D=(D1,D2,…,Dk) are compared by the cosine distance which is calculated as:

$$Similarity = \cos\theta = \frac{\vec{Q}\cdot\vec{D}}{|\vec{Q}|\cdot|\vec{D}|} \tag{1}$$

To make this assisted CBIR the additional sensor modules are tied to the framework with a Python interface. When added, the modules enable filtering of results. This filtering provides greater fidelity resulting in multiple levels of search. By having multiple levels, search can be hierarchically arranged, reducing the number results at every level to save indexing. A set of returned images from the image query will be reduced by attributes of the sensor modules, promoting the returns that match sensor attributes and demoting those that do not in the return results.

IV. MSACS Implementation and Evaluation

For our experimental evaluation, the prototype of the MSACS framework was implemented in Python. This prototype made use of several open source libraries such as OpenCV, SciPy, NumPy, GeoPy, Basemap, and the Cassandra database. The dataset used for evaluation was created using the Google Street View API to query and retrieve street-level images for a specified evaluation zone so that the localization application described in the use-case could be demonstrated.

A. Core CBIR Implementation

The core CBIR module within the MSACS framework is responsible for extracting point features from images, calculating a visual vocabulary, and expressing the content of images in the form of a BoVW histogram. The BoVW implementation for this experimental evaluation made use of several open source libraries to complete these tasks, namely OpenCV 3.1.0, NumPy 1.11.0, and SciPy 0.17.1.

In the first stage, each image was converted from RGB to Grayscale. Once converted, SIFT features are identified and descriptors are extracted. These point feature descriptors are then recalculated as RootSIFT descriptors to improve their matching and differentiability. Next, the descriptors are aggregated into a list containing all of the dataset's image descriptors so that centroids can be identified using the *k*-means algorithm. Once centroids have been identified, the centroid vectors form codewords, known collectively as a visual vocabulary or codebook. The codebook is stored by serializing the centroid data to be recalled and used later to codify images in terms of their visual words.

In the second stage, vector quantization and histogram generation, BoVW signatures are calculated for each image in the dataset. This process begins with the quantization of feature descriptors using the codebook produced previously. The result of this process is a list of visual words, each representing a centroid from the visual vocabulary, which summarize the content of each image. The number of occurrences of each of the k visual words found in an image are tallied and used to build an image histogram.

An example histogram for a $k=100$ codebook is depicted in Fig. 4. The figure summarizes 100 codewords on the horizontal axis, labeled by index. The vertical axis depicts the frequency of occurrence of each codeword in the image. The presence of codewords provide this image with a descriptive representation. Once histograms are calculated, they are stored in the Cassandra database and are accessed as needed to facilitate search and comparison operations. When queried with an image, a histogram is computed for the query, the cosine distance is calculated against our corpus, and the dataset images are ranked by the cosine distance to the query.

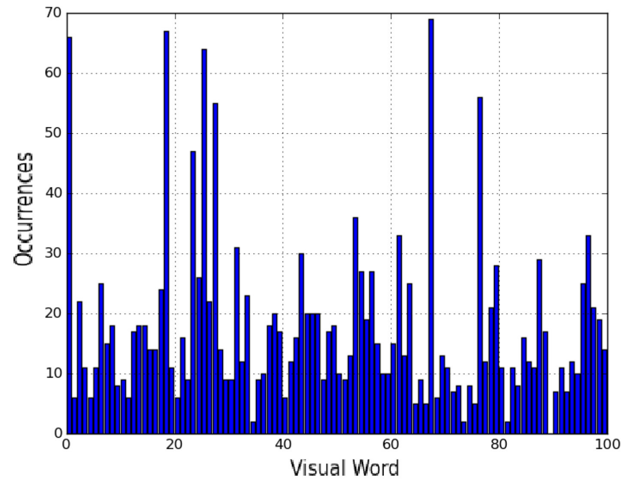

Fig. 4. An example BoVW histogram summary for the content of an image with a vocabulary size of 100 visual words.

B. Annotated Image Repository

Dataset collection was automated by using Google's Street View Image API to download both images and their geolocation (latitude and longitude) data. A dataset consisting of 1,080 images was collected from New York's Upper East Side in an area of approximately 50 square city blocks.

The dataset was generated by repeated route planning in the Google Maps API. Source and destination points were entered to obtain a URL by the API. This URL was fed into an online GPS visualizer to obtain an XML-based gpx file. The gpx file contains the source and destination points along with several intermediate GPS points represented by latitude and longitude values. For each latitude and longitude pair in the gpx file four images were collected from the street-view API at random viewing angles; one from 0-90°, one from 90-180°, one from 180-270°, and one from 270-360°. The images are 500x500 pixels in resolution and suffixed with a .jpg extension. The query set was created by taking one random image from 0-360° at each latitude longitude pair in the gpx file. A Cassandra database was used to store information about each image including the image name, latitude, longitude, and field of view (i.e. the degree orientation) from which the image was generated.

Ground truth was computed by comparing each query image against each dataset image to determine if there existed a sufficient number of validated feature matches to establish the existence of relevant matching visual content between the image pairs. For an image to be considered relevant it must have a minimum user-defined number of matches. Applying a constraint of at least 50 matches across our set of query images and dataset images was demonstrated to establish relevance with no false positives for a manual evaluation performed using a subset of the overall data. A ground truth table is constructed of size *n-by-m* where n is the number of query images and m is the number of dataset images. For example if our n^{th} query image and m^{th} dataset image are equal to or greater than the user-defined feature matching limit then position *(n,m)=1* in our ground truth table. Otherwise if the match constraint is not met then position *(n,m)=0*.

C. Image Retrieval Evaluation

The MSACS CBIR prototype was evaluated using precision and recall, the standardized way to evaluate a CBIR system. Precision is a measure of how useful the return set of images are in response to a query, and will generally correlate with a high similarity metric. Recall is a measure of how complete the set of return results are in response to a query [22]. Precision and recall are calculated as follows:

$$\text{precision} = \frac{|\{\text{relevant images}\} \cap \{\text{retrieved images}\}|}{|\{\text{retrieved images}\}|} \quad (2)$$

$$\text{recall} = \frac{|\{\text{relevant images}\} \cap \{\text{retrieved images}\}|}{|\{\text{relevant images}\}|} \quad (3)$$

In this setting, relevant results were calculated from the ground truth table as images that shared at least 50 features with the query image. This ensures the possibility of performing additional processing using computer vision. To be useful for search, most relevant items should appear within a tractable amount of retrieved results. As seen in Fig. 5, 75% of relevant images are returned with 10% of the total corpus size. Tests were conducted with codebook sizes with a *k*-value of 70, 170, 500, and 1000, and it appears that k=170 appears to be the best at ordering the first half of returned images. The results seem to demonstrate that the system overly generalizes at k=70, enabling features which are not instances of one another to be codified into the same visual word. For k=1000, the system seems to perform best when the return set is in the first 10% of the image repository size. This can more clearly be seen on a logarithmic scaled version of precision and recall, shown in Fig. 6.

D. Determining Environmental Information

To enable localization functionality, MSACS can capture images and populate a database with their latitude and longitude coordinates as in use case 1 when the world model is generated. The database could potentially be fed by many cameras and there could be many databases making use of Cassandra's ability to scale. Searching with a query image will lead the system to begin the search and retrieve operation and then execute the localization application per use case 2.

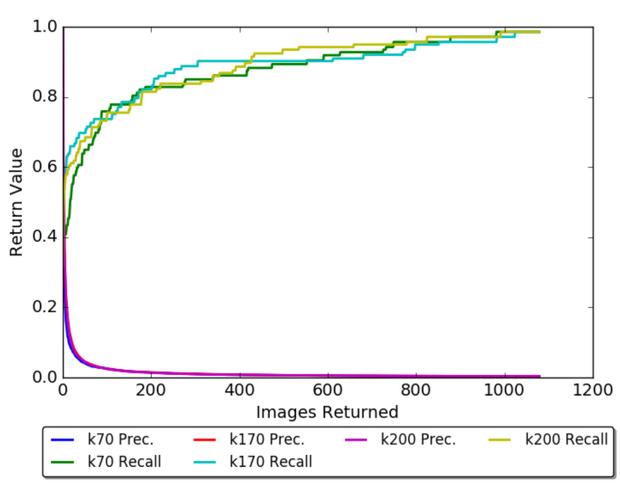

Fig. 5. Precision and recall results for various codebook sizes (*k*)

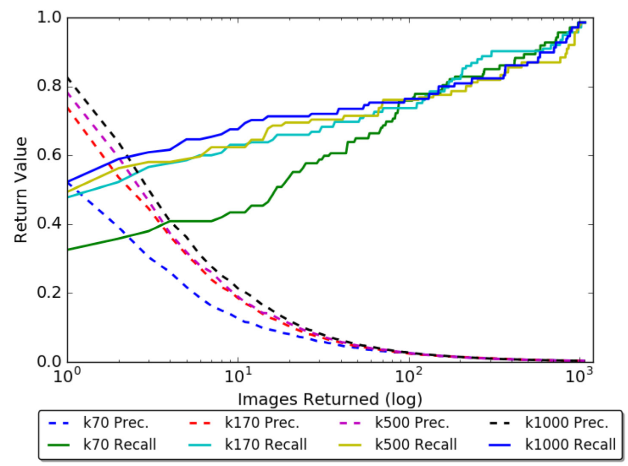

Fig. 6. Precision and recall results, logarthmic scale

A simple approach to show this capability is simply to estimate the locations of the top ranked images for a latitude and longitude estimate for the query image. For each of the top returned dataset images the cosine distances with respect to the query images are calculated. If the cosine distance between the query and any dataset image is greater than .99, then the location of the query image is guessed to be that of the dataset image. If the cosine distance of an image is less than .99 then the dataset image's latitude and longitude are added and averaged with all other dataset image's latitude and longitude values from the return set. The user is then presented with an actual location, guess location and physical distance in meters between the actual and guess locations and the distance between the two was counted as error. The results of the simple localization through coarse averaging among the returned set is shown in Fig. 7. The graph further reinforces that the discriminating power of the larger *k*-values in producing a more accurate location guess.

These localization values are averaged guesses over a path. In an actual localization application, more sophisticated techniques to recognize landmarks and surveyed positions within images would increase the certainty of a guess, but we reserve this for future work.

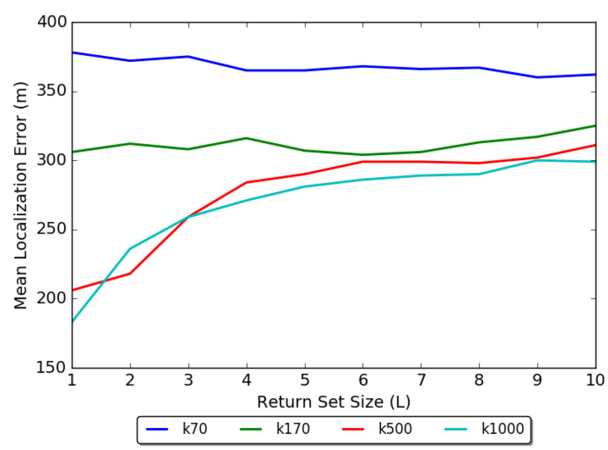

Fig. 7. Coarse localization by averaging location from top returns

In a final demonstration, we capture information about wireless access points from images in the sensed environment. For this, we use location guesses to automate search in the Wigle database [18] from a Python script to show, in Table 1, that access points can be returned from an image search.

TABLE 1. WI-FI ACCESS POINT LOOKUP

Wi-Fi Access Point	Latitude	Longitude	Channel
00:09:5B:XX:XX:XX	40.765308	-73.967735	1
02:22:C2:XX:XX:XX	40.765308	-73.967735	10
20:AA:4B:XX:XX:XX	40.765308	-73.967735	11
20:C9:D0:XX:XX:XX	40.765308	-73.967735	6
58:6D:8F:XX:XX:XX	40.765308	-73.967735	11
68:94:23:XX:XX:XX	40.765308	-73.967735	11
90:84:0D:XX:XX:XX	40.765308	-73.967735	3

The returned results indicate potential to discern a Wi-Fi profile in an environment through image search. It also hints at the reverse, where a set of access points can return a set of images from an environment that matches the wireless profile.

V. CONCLUSION AND FUTURE WORK

Vision plays an important role in the human decision making process, but modern technology has caused information overload. Making a quick decision can be the difference between mission success and failure in time sensitive and mission critical situations. The MSACS framework, supported by an assisted CBIR system, provides a way to search images quickly and bridge the sensor gap.

Our contribution of the modular framework provided the user with an extensible way to support image retrieval in a dynamic and sensor rich environment. We designed and implemented a framework for an assisted CBIR system, incorporating state of the art core functionality using the Bag of Visual Words model. The evaluation of our prototype showed potential for image retrieval and cross sensor applications. Future studies will expand on the development of user interfaces and mechanisms for assisted CBIR across a wide array of potential applications. In addition, we aim to improve search results and expand image repositories. We will expand the MSACS prototype with multi-sensor inputs and live test experiments using magnetometers, RF receivers and other sensors to iteratively improve assisted CBIR.

ACKNOWLEDGEMENTS

The views expressed in this paper are those of the authors, and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

REFERENCES

[1] R. Radke, S. Andra, O. Al-Kofahi and B. Roysam, "Image change detection algorithms: A systematic survey," IEEE Transactions on Image Processing, vol. 14, no. 3, pp. 294-307, 2005.

[2] J. Deng, W. Dong, R. Socher, L. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 248-255.

[3] M. Jones and P. Viola, "Fast multi-view face detection," Mitsubishi Electric Research Lab, Paper TR-20003-96, 2003, p. 14.

[4] S. Agarwal, et al., "Reconstructing Rome," Computer, vol. 43, no. 6, pp. 40-47, 2005.

[5] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," AAAI Conf. on Artificial Intelligence. Edmonton, Alberta, 2002, pp. 593-598.

[6] D. Scaramuzza and F. Fraundorfer. "Visual odometry [tutorial]," IEEE Robotics & Automation Magazine, vol. 18, no. 4, pp. 80-92, 2011.

[7] J. Singh, J. Kaleka and R. Sharma, "Different Approaches of CBIR Techniques," International Journal of Computing Distributed Systems, vol. 1, no. 2, pp. 76-78, 2012.

[8] M. Mansourvar, et al., "Computer-based system to support intelligent forensic study," Conf. on Computational Intelligence, Modeling and Simulation, 2012, pp. 117-119.

[9] L. Pinjarkar, S. Manisha and K. Mehta, "Comparative evaluation of image retrieval algorithms using relevance feedback and its applications," International Journal of Computer Applications, vol. 48, no. 18, pp. 12-16, 2012.

[10] G. Panteras, et al., "Accuracy of user-contributed image tagging in Flickr: A natural disaster case study," International Conf on Social Media & Society, 2016, London, UK, p. 14.

[11] J. Yang, Y. Jiang, A. Hauptmann and C. Ngo, "Evaluating bag-of-visual-words representations in scene classification," International Workshop on Multimedia Information Retrieval, Augsburg, Bavaria, Germany, 2007, pp. 197-206.

[12] J. Kang and W. Zhang, "A framework for image retrieval with hybrid features," Control and Decision Conf., Taiyuan, China 2012.

[13] X.Y. Wang, Y.J. Yu and H.Y. Yang, "An effective image retrieval scheme using color, texture and shape features," Computer Standards & Interfaces, vol. 33 no. 1, pp. 59-68, 2011.

[14] E. Walia and A. Pal, "Fusion framework for effective color image retrieval," Journal of Visual Communication and Image Representation, vol. 25, no. 6, pp. 1335-1348, 2014.

[15] M. El Alami, "Unsupervised image retrieval framework based on rule based system," Expert Systems with Applications, vol. 38, no. 4, pp. 3539-3549, 2011.

[16] I. Al Kabary, et al., "QUEST: Towards a multi-modal CBIR framework combining query-by-example, query-by-sketch, and text search," IEEE International Symp. on Multimedia, Anaheim, CA, 2013, pp. 433-438.

[17] H. Sergieh, et al., "Geo-based automatic image annotation," ACM International Conf. on Multimedia Retrieval, Hong Kong, 2012. p. 46

[18] Wigle: Wireless Network Mapping, [Online], Available: http://wigle.net. (URL)

[19] D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[20] A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651-666, 2010.

[21] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 2911-2918.

[22] R. Datta, D. Joshi, J. Li, and J.Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," ACM Computing Surveys, vol. 40, no.2, p 5, 2008.