Robust feature matching in long-running poor quality videos

Craig Henderson and Ebroul Izquierdo, *Senior Member, IEEE*

***Abstract*—We describe a methodology that is designed to match key point and region-based features in real-world images, acquired from long-running security cameras with no control over the environment. We detect frame duplication and images from static scenes that have no activity to prevent processing saliently identical images, and describe a novel *blur sensitive feature detection* method, a combinatorial feature descriptor and a distance calculation that efficiently unites texture and colour attributes to discriminate feature correspondence in low quality images. Our methods are tested by performing key point matching on real-world security images such as outdoor CCTV videos that are low quality and acquired in uncontrolled conditions with visual distortions caused by weather, crowded scenes, emergency lighting or the high angle of the camera mounting. We demonstrate an improvement in accuracy of matching key points between images compared with state-of-the-art feature descriptors. We use key point features from Harris Corners, SIFT, SURF, BRISK and FAST as well as MSER and MSCR region detectors to provide a comprehensive analysis of our generic method. We demonstrate feature matching using a 138-dimensional descriptor that improves the matching performance of a state-of-the-art 384-dimension colour descriptor with just 36% of the storage requirements.**

***Index Terms*—Feature extraction, Image color analysis, Pattern matching**

I. Introduction

FEATURE descriptors and inter-image feature matching have been well researched areas in computer vision for many years. Most works assess the performance of descriptor matching using high quality images, for example, in the field of video analysis, popular techniques have used Hollywood movies as a test dataset [1], [2], [3]. However, security cameras work in uncontrolled environments and record constantly without continual adjustment to focus, lighting and position that a feature film is privileged with. As a result, the low resolution images generally have poor color clarity and little discriminative or representative texture definition (Figure 1). Consequently, contemporary methods are not robust to the challenges of low quality images that result from these systems. Figure 2 shows a comparison of image quality from two popular feature films used in the literature and two typical images from street level fixed closed-circuit television (CCTV) cameras in London, UK. Forensic analysis of security camera video sequences is a less well studied field and demands adaptation of contemporary methods to accommodate the image quality differences that exist. The quality of images from each security camera varies considerably, and this inconsistency can cause difficulties in matching features between camera images.

Our intent is to identify and match patterns such as a brand logo, a colored pattern on a scarf or hat, a tattoo or other distinctive marking on a person, object or clothing. Texture alone is not sufficient to find correspondences between frames in security videos, and therefore the effectiveness of popular gradient-based descriptors such as SIFT and SURF is limited. This paper focuses on color images where color can be a significant, albeit often subtle, discriminator in identifying items within an image and we establish a methodology to improve the robustness of matching features by using information from a blurred image and color detail to increase the discriminative properties of feature descriptors.

Fig. 1. Variation in low-quality CCTV images with poor lighting and long range camera views. Images taken from the same camera at different zoom levels. When a subject appears in the distance, the color and texture definition is poor and inconsistent with frames when the subject is closer.

In mainstream content-based image retrieval (CBIR) systems, color can be useful, but is not absolutely discriminative. For example, a bus is a bus regardless of its color, and a query to a CBIR system for a bus would be expected to find buses of all colors. In such a system, color can be treated as an attribute that may or may not be used in the search. A search for a Daffodil in a database of images of flowers may consider color as an important search criteria as all Daffodils are a variant of yellow. In our environment of CCTV image matching, we are interested in finding and tracking a specific instance of an object rather than a category of object, and so color can be used as a discriminating factor in matching correspondences. In a crowd scene with two people with similarly patterned shirts, color can be used to identify the correct shirt, for example.

Our interest is in large-scale processing of long-running videos which demand fast processing of a very large number of features. We are therefore motivated by simple solutions to complex problems with low computation requirements and minimal storage, intentionally avoiding some of the complexities of other methods in the interest of execution speed.

Our contributions are simple and fast algorithms that com-

Fig. 2. Comparing the definition of broadcast quality films (top) and typical CCTV images. Top left, a frame from the 1993 film Ground Hog Day starring Bill Murray and Andie MacDowell; Top right, a frame from the 1998 German science-fiction action thriller Run Lola Run starring Franka Potente. Bottom row, scenes from CCTV footage have far less definition and clarity.

bine to provide memory- and processing-efficient feature matching which we demonstrate to improve on current methods that use Euclidean distance to match intensity- and color-feature descriptors.

- **Preprocessing to eliminate redundant information** A method to robustly eliminate duplicate frames caused by video recapturing or static frames of no activity, resilient to the presence of an on-screen timer, clock or frame counter.
- **Adaptive blur-sensitive feature detection** An adaptive approach to the detection of features that will correspond between two images, guided by the sharpness of the two images.
- **Combinatorial Texture and Color feature matching** A novel technique to combine texture and color features and measure distance between descriptors for robust feature matching.

We show that our methodology improves feature matching of low-dimensional intensity descriptors with an overhead of just 10 integers per features, to compete with much higher dimensional color descriptors. In addition, the method is shown to improve correspondence of established high-dimensional color descriptors.

The rest of the paper is structured as follows. §III provides an overview of related literature on color features descriptors followed by some background information on the challenges faced in our domain of processing poor quality images, §II. The proposed methodology is then described in detail in §IV and evaluated in §V. Finally, §VI assesses the storage efficiency and accuracy trade-off that is important in large-scale systems, and we conclude in §VII.

II. Motivation

Fluctuating lighting conditions caused by fire and emergency vehicle lights are commonplace in video that undergoes forensic analysis. Fast camera pan or zoom, frenzied motion within a frame, or a combination of both can cause significant blurring in frame images which results in a lack of texture. CCTV cameras are often sited very high and cover a long field of view where objects in the distance lack color definition. In videos from these cameras, frame-to-frame tracking of texture or color features alone fail to cope with the variations of blur and lighting where images of the object, person, tattoo, logo, or clothing pattern appears differently in each frame. Low image quality and inter-frame inconsistencies such as these are generally absent from Hollywood movies, but are a very serious barrier to applying state-of-the-art computer vision algorithms to CCTV videos.

Texture alone is therefore not sufficient to find correspondences between frames in security videos, and the effectiveness of matching popular intensity descriptors such as SIFT and SURF is limited. In this paper we establish a method to improve the robustness of matching features by using color information to increase discriminative properties of a texture feature descriptors.

III. Related Work

Popular key point feature detectors SIFT [4], SURF [**?**] are slow to calculate and designed to identify texture structures in gray scale images, defined only by pixel intensity variations. Computationally efficient key point detectors such as BRISK [5] and corner detectors such as Harris Corner Detector [6] and FAST [7] enable use of keypoint tracking in real time systems by reducing the computation overhead [8]. Leutenegger *et al.* [5] observed that extremely efficient key point detectors such as FAST and BRIEF [9] offers a much more suitable alternative for real-time applications. Alternatively, Maximally Stable (MS) regions are discovered using a common *Maximally Stable Extremal Region* (MSER) detection algorithm [10], [11]. MSER is accepted to be a reliably effective and computationally efficient method of detecting feature regions in single channel images. Early work to extend MSER to multi-channel color images was presented in [12] but did not achieve bottom up feature detection as in [13] where the author presents a derivative work specifically for maximally stable color regions, MSCR.

There have been a number of proposals for color descriptors that describe color attributes of an image. These are conveniently small in dimensionality (Table I) and represent the color information around a key point using a color histogram. A detailed description of histogram based color descriptors is provided in [14]. Color alone is not robust for achieving good correspondences between images. There have been many descriptors proposed to use color, such as the biologically inspired SODOSIFT [15] and those that use texture descriptors in various color channel combinations, concatenating the texture from each channel. Many of these are based on the SIFT descriptor resulting in a $128\times3 = 384$ dimension descriptor. HSV-SIFT [16] calculates a SIFT descriptor on each of the three channels in HSV color space and RGB-SIFT [14] is a similar algorithm using RGB channels, with values equal to the *Transformed Color SIFT* method [14]. rgSIFT [14] builds descriptors on the r and g chromacity components of the normalized RGB color model $(r,g,b)^T = \left(\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B}\right)^T$ and C-SIFT [17] uses a normalized opponent color space, dividing

Descriptor	Dimension
Normalized RG Histogram	30
Hue-Histogram	37
Opponent Histogram	45
RGB-Histogram	45
Transformed Color Histogram	45

TABLE I
COLOR HISTOGRAM DESCRIPTORS AND THEIR DIMENSIONALITY. SEE [14] FOR A DETAILS OF THESE DESCRIPTORS AND THEIR PROPERTIES.

the first two channels by the intensity channel O_3, $\frac{O_1}{O_3}$ and $\frac{O_2}{O_3}$, making it invariant with respect to light intensity [14]. OpponentSIFT identifies features in opponent color channels, red-green (RG) and yellow-blue (YB) [18] by computing SIFT descriptors in each of the channels $(O_1, O_2, O_3)^T = \left(\frac{R-G}{\sqrt{2}}, \frac{R+G-2B}{\sqrt{6}}, \frac{R+G+B}{\sqrt{3}}\right)^T$. OpponentSURF uses the same technique with SURF features. The interested reader is referred to [14] for a comprehensive review of color descriptors.

In each of these, color information is used in detecting features and extracted descriptors are implicitly discriminative by virtue of their construction. However, the color detail of the image area around the feature is not encoded into the descriptor and is not used to discriminate between similar features. HueSIFT [19] describes a concatenation of a quantized Hue Histogram of 37 dimensions with the SIFT descriptor, concentrating on the effective detection of features without consideration for the descriptor encoding. SIFT was also used as a base for the *bag-of-colors* algorithm in the context of image search in [20]. Our method takes a similar approach in descriptor concatenation, but we do not limit our focus on SIFT descriptors and we describe a robust approach to feature distance calculations.

Color descriptors that use three color channels for feature descriptions typically increase the dimensionality three times, compared with their intensity based counterparts, and the size of each descriptor becomes problematic for efficient computation and storage. Principal Component Analysis (PCA) has been used to reduce the dimensionality in PCA-SIFT [21], but is computationally expensive; given n data points, each represented by p features, the computational complexity is the sum of the covariance matrix computation $O(p^2 n)$ and its eigen-value decomposition $O(p^3)$, hence $O(p^2 n + p^3)$.

Robust feature matching is important for many tasks such as image alignment, stitching, object tracking, and search and retrieval. Typical methods for matching feature descriptors find the closest descriptor to another in n-dimensional space and use a threshold to determine if the descriptors are *close enough* to determine that they match. Lowe [4] increased the robustness of this technique by introducing a *distance ratio* measure that calculates a ratio of the distances of the closest and the second closest to distinguish true and false matches.

SIFT descriptors, which are histograms, were designed for use with Euclidean distance measures for comparison and matching [4]. However, it is well known that using Euclidean distance to compare histograms often yields inferior performance than using χ^2 or Hellinger measures. Arandjelović and Zisserman made this observation and proposed rootSIFT [22], which transforms the SIFT descriptor such that the Euclidean distance between two descriptors is equivalent to using the Hellinger kernel, also known as Bhattacharyya's coefficient. rootSIFT is dubbed *Hellinger distance for SIFT*, and can yield a significantly more accurate result in calculating the distance between two descriptors used in feature descriptor matching.

In object tracking systems, matching features between adjacent frames is crucial, and many systems use key point features. Sun and Liu [23] describe a selective method of tracking in which they calculate a confidence of feature matches by two measures - a mean value of the maximum inner product for every descriptor, and a ratio of reliably matched features to the total number of features - and adapt the tracking algorithm appropriately. Fast key point detectors and corner detectors such as FAST enable use of key point tracking in real time systems by reducing the computation overhead [8], and real-time applications may employ probabilistic methods [24] for data association to discover matching consensus [5]. This further motivates us to apply our method to a variety of feature descriptors beyond the popular SIFT. The contribution of color in matching video frames from multiple views was a motivator for the recent work of Fezza *et al.* [25], using a Histogram Matching algorithm in a group of pictures considering pixels within a square window around each SIFT feature that is proportional to the scale parameter of the SIFT key point.

IV. PROPOSED METHODOLOGY

A. Measuring image blur

Accurate models for calculating the motion blur of an image have been described, from estimating the parameters of a Point Spread Function [26] to using machine learning [27]. Our intent is not to accurately calculate the blur parameters such that the blurred image can be restored to a sharp image, but to *quickly* be able to estimate the degree to which an image, or part of an image, is blurred. We therefore use a straightforward method that is fast to calculate and is shown to give a reasonable estimation of blurriness for our purposes.

We derive an efficient technique from the intuition that a blurred image will contain fewer sharp edges than a non-blurred image. The number of edges in an image can therefore be used as an expression of image blurriness (or, conversely, image sharpness). We use a Canny edge detector [28] with a 3×3 Gaussian kernel, a lower threshold of 175 and an upper threshold of 225. The small Gaussian kernel balances execution time with sensitivity salt-and-pepper noise that can be caused by analog-to-digital converter errors or bit errors in transmission. The threshold values have been chosen empirically to avoid breaking noisy edges (if the lower threshold is too high) and reducing fragmentation if the upper threshold is too low.

The Canny edge detector [28] is used to construct a binary edge map E from image I of size $m \times n$,

$$E_I = \{e_1, e_2, e_3, \ldots e_{m \times n}\}, \quad e_i \in \{0, 1\} \tag{1}$$

The *sharpness* of image I is then determined by a function $S(I)$ that calculates the fraction of non-zero pixels in the edge

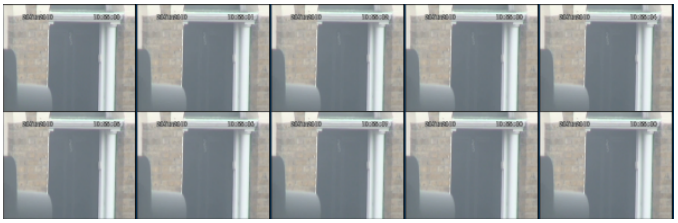

Fig. 3. Ten frames of a video sequence with a counting time stamp but no activity in the scene

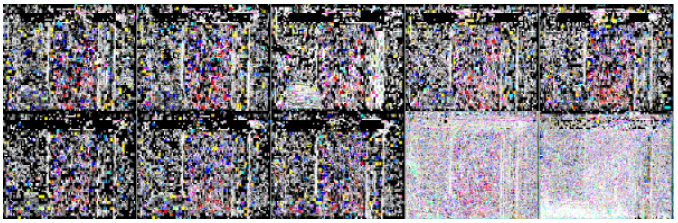

Fig. 4. Difference images of the frames in Figure 3 with their previous frame. The colors represent pixel value differences in the RGB channels

map E, which is the fraction of pixels representing edges in the image I.

$$S(I) = \frac{1}{m \times n} \sum_{x_i \in E_I} x_i \tag{2}$$

We use this measure of image sharpness to eliminate duplicate frames and static scenes (IV-B) and in blur-sensitive feature detection (IV-C).

B. Duplicate frames and static scenes

Frame rate instability can cause non-deterministic duplicate frames to appear in a video sequence. Visually duplicate frames are seldom identical in their pixel values even when recorded by screen-capturing, and so eliminating them from the processing stream requires more effort than simple pixel comparison. A related problem in video surveillance, particularly with fixed cameras, are static scenes; scenes where nothing happens for periods of time, but the camera continues to record and subsequently produces large sequences of video with no activity. A security camera will often produce a time stamp on the captured image, and frames are therefore different in the counting clock (and date roll-over at midnight) but otherwise unchanged (Figure 3).

Processing duplicate images is time-consuming, wasteful of resource and complicates algorithms that are designed to work with movement. It is therefore desirable to eliminate duplicate frames and static scenes from the processing queue. The sparse optical flow is calculated using the multi-scale Lucas-Kanade algorithm [29], with *good features to track* [30] key point features. Feature descriptors are matched in each pair of temporally adjacent frames and if no features change location in the second frame, then the frames are treated as identical. We ignore extreme small or large movements and only consider movements between 1 pixel and 2σ pixels, as 95% of values are within 2 standard deviations of the mean. An exception case is made if one or other of the images is measured as blurred (Eq. 3) and the other isn't. In this case, the feature matching algorithm may fail but given a difference in sharpness measure, the frames are considered not duplicate and not static.

The determination of a *blurred image* is to find a suitable threshold below which $S(I)$ from Eq. 2 must fall to represent an image that is blurred. Our definition of a blurred image is therefore;

$$\text{image } I \text{ is blurred} = \begin{cases} true & \text{if } S(I) \leqslant \varphi \\ false & \text{otherwise} \end{cases} \tag{3}$$

In our experiments, we use an empirical value $\varphi := \frac{1}{32}$, so if edges are present in 3.125% of the image or less, then the image is deemed to be blurred. This value is sensitive to the content of the scene. In our use case of CCTV footage, scenes are generally quite *busy*, in built-up areas containing a lot of activity (people, vehicles, buildings, etc.), and this value has worked well. The duplicate frame elimination is an optimization to avoid processing frames where the result will be the same as a previous frame. In this situation, there is some flexibility of the robustness of the algorithm where false positives will only cause a frame to be processed where perhaps it is need not have been.

The method is robust to time stamp counters within the frame because movement between frames is only identified if a matching feature is in a different position in the two frames. In the case of counting digits of a time stamp, features surrounding a 6, for example, are not matched in the next frame to the features surrounding the 7 that replaces the 6, and therefore no movement is detected. This technique therefore detects movement between frames, not simply differences between the frames.

Figure 3 shows an example of frames from a video sequence recorded with a static camera, with a time stamp counter in the corner. Each of these frames were determined to be duplicates by the above algorithm, despite the substantial pixel value differences as shown in Figure 4.

C. Blur sensitive feature detection

Image blur is a very significant hindrance to matching features between frames in low quality images. This observation leads us to adapt feature detection to maximise correspondence accuracy in a technique we call *blur sensitive feature detection*. The method is designed to optimise a local region within an image with respect to its *blurriness* and that of the adjacent frame image to which correspondence is to be established. Applying a localised blur of the area before detecting features can help to find more similar features to the corresponding image, if the amount of image blur can be more closely matched.

A relationship map M_s is established to correspond the properties of a 2D Gaussian kernel G_k of size k with the sharpness measure of the query image I_Q after a convolution with G_k. The map holds sharpness values for the query image after convolution with 2D Gaussian filters of kernel size in the set Γ. Let

$$\Gamma \equiv \{p \in \mathbb{N} | p = 2q - 1 \wedge q \in \mathbb{N}\} \tag{4}$$

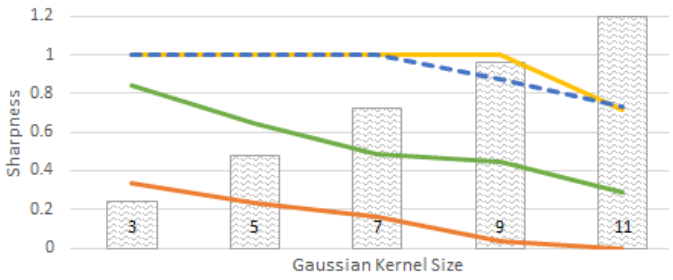

Fig. 5. Relationship between the size of a Gaussian kernel (x-axis) used to artificially blur example query images (*curves*) and the sharpness of the resulting image (y-axis). The kernel size steadily increases while the decline in sharpness (increase in image blur) varies with different query image regions. The relationship map M_s therefore needs to calculated for each query region used for correspondence matching.

The sharpness is calculated for each kernel size and stored in an associative map $k \rightarrow S(I)$

$$M_s(k) = S(G_k * I_Q) \quad \forall k \in \Gamma \wedge k \leq \alpha \in \mathbb{N} \tag{5}$$

where

$M_s(\cdot)$ represents an entry in an associative map
$S(\cdot)$ image sharpness, from Eq. 2
G_k Gaussian filter of kernel size k
$*$ represents 2D convolution
α an upper bound on the Guassian kernel size

Figure 5 shows some examples of the relationships between the size of a Gaussian kernel used to artificially blur example query images and the sharpness of the resulting image in M_s. This demonstrates the variance in the correlation between the steepness in the decline in sharpness (increase in image blur) with steadily increasing kernel sizes for different query image regions, and therefore the need to calculate M_s for each query region used for correspondence matching.

A *sharpness adjustment* S_a is calculated as the difference between the sharpness of the original (unconvolved) query image region I_Q and the target image I_T to which correspondence is to be established.

$$S_a = S(I_Q) - S(I_T) \tag{6}$$

The value of S_a is used to find the corresponding Gaussian kernel size k in M_s which, when convolved with I_Q will produce an image I'_Q with sharpness that will most closely match $S(I_T)$

$$m = \arg\max_k \{S_a - S(G_k * I_Q)\} \quad m \geq 0 \tag{7}$$

$$I'_Q = G_m * I_Q \tag{8}$$

Features are detected in, and extracted from I'_Q and I_T and correspondences are found between these feature sets.

Matching performance is considerably improved by aligning sharpness of the images before feature detection. However, blurring an image reduces texture structure, which reduces the effectiveness of feature detectors, especially corner-based detectors such as FAST and BRISK. If no features are found in I'_Q, we repeat the process with I_Q as the entire query image, not bounded to the query region of interest. We do this with the understanding that the bounded region of interest contains little texture so retrying with greater kernel sizes would offer only minor improvements, whereas the sharpness of the query image as a whole provides more information with respect to blur induced by camera movement. In the unlikely event that no features are found in the revised I'_Q, we fall back features found in I_Q. In all of our experiments, this fall back position is never required as the unbounded I_Q image always produces a usable feature set.

D. Combinatorial Texture and Color feature matching

We create a new combinatorial feature descriptor representing local features with color information from the surrounding region. First, any local feature detector is used to find feature locations and both a key point and a region are defined for each. In the case of a key point detector such as SIFT, a circular region is created with its center at the key point co-ordinates. For region based feature detectors such as *Maximally Stable Extremal Regions* (MSER), the region is approximated using an ellipse fitting algorithm through the region boundary points and a key point is defined at the center of the ellipse.

With the resulting set of key point locations and region definitions, we extract a texture descriptor at each key point. The texture descriptor is a standard feature descriptor that will be extended by our method to improve its discriminative capability in color images. We then build a local histogram color model of each region to create an extension descriptor. Using the region shape as a mask over the color image, pixels falling within the shape are quantized into a local color histogram representing the region. This histogram is transformed into a feature descriptor using the histogram bins as the feature dimension. Finally, the text descriptor and color descriptor are concatenated into a composite descriptor.

The RGB color space is known to be a poor representation for color segmentation as there is no straightforward correlation between the RGB channel values and the intensity of a particular color that lends itself to simple thresholding. We therefore transform the RGB image to the HSV color space for our algorithm. The Hue (H) channel determines the color, the Saturation (S) is the intensity of the color and the brightness or luminance (V) can be used to find non-color white, gray and black.

Allocating a pixel value to its closest histogram bin is done by calculating a partial distance in HSV color space. For color entries in the histogram, the distance is determined by the Euclidean distance of the Hue and Saturation components, $d = \sqrt{H_i^2 + S_i^2}$. Distance to the additional three non-color entries in the histogram, white, black and gray, are calculated using the Euclidean distance of the Saturation and Value (luminance) components, $d = \sqrt{S_i^2 + V_i^2}$. Measuring color distances in the HSV color space in this way maintains robustness to affine illumination changes in the image.

1) Texture descriptor distance: Two features are considered equal if they are close to each other in their high-dimensional feature space. Popular feature descriptors are designed as

Euclidean space vectors such that $\vec{u}$ and $\vec{v}$ represent features

$$\begin{aligned}\vec{u} &= (u_1, u_2, \ldots, u_n)\\ \vec{v} &= (v_1, v_2, \ldots, v_n)\end{aligned} \tag{9}$$

and the distance between them is given by the length of the line segment $\overline{\mathbf{uv}}$ connecting them, i.e. the Euclidean norm.

$$\|\overline{\mathbf{uv}}\|_2 = \|\vec{u} - \vec{v}\|_2 = \sqrt{\left(\sum_{i=1}^{n}(u_i - v_i)^2\right)} \tag{10}$$

A pre-defined or dynamically calculated threshold value is typically used to determine whether features are *close enough* to be considered a reasonable match.

2) Color histogram distance: Color histogram feature space is also multi-dimensional, but the distance between feature points in most color spaces are more accurately calculated using methods such as χ^2 [31], Bhattacharyya distance [32], the Earth Mover's Distance [33]. For histograms with identical palettes, the Normalized Histogram Intersection (NHI) [34] is a light-weight calculation of similarity, and subtracting from one gives a dissimilarity, which is a distance measure between two histograms

$$H(a,b) = 1 - \frac{\sum_{j=1}^{n}\min(a_j, b_j)}{\sum_{j=1}^{n} a_j} \tag{11}$$

3) Designing a combinatorial descriptor: In designing an algorithm to extend an existing feature descriptor, consideration is made to the potential of falsely matching dissimilar features of similar color or moving vectors in feature space closer together where neither their feature descriptor nor the color are similar. Our goal is to produce a generic extension that can be used with any underlying texture feature descriptor, and so we focus on a method to combine an n_1-dimensional texture feature descriptor with an n_2-dimensional color histogram in such a way as to discriminate similar features of different colors without these pitfalls.

Consider a naïve implementation that concatenates an n_2-dimensional color-histogram onto a n_1-dimensional texture descriptor to form an (n_1+n_2)-dimensional feature descriptor, and compares the combined descriptors as single vectors. Ignoring the differences in scale of the two components, extending the dimensionality to accommodate the color information is intuitive, however the method will treat the color histogram as an integral part of the feature, applying Eq. 10 to the vector as a whole and losing the unique properties of the color histogram. Figure 6 shows a correlation between using Euclidean distance and histogram similarity to measure the closeness of features from a typical dataset. The low positive correlation value of 0.49 shows that a Euclidean distance will generally give a reasonable indicative result but is less accurate than the histogram similarity $(1-\text{NHI})$.

4) Distance between descriptors: A feature can be said to correspond to its closest match in a set of candidate features, where the descriptor with the smallest distance is selected, irrespective of the value of the distance or its relationship to its neighbors. Lowe [4] refined this method using a *distance ratio* to determine if the closest match was a *good* match. The distance ratio method finds the closest two features f_c and

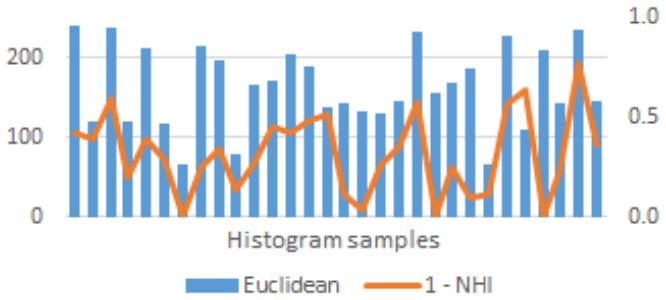

Fig. 6. Correlation of the distance between color histograms using the Euclidean distance (left axis) and a similarity based on Normalized Histogram Intersection (right axis). Correlation ≈ 0.49 indicates a low positive correlation between the two measures.

f_{c+1} and divides the nearest distance by the second closest distance,

$$distance\ ratio = \frac{\|f - f_c\|_2}{\|f - f_{c+1}\|_2} \tag{12}$$

This ratio helps to determine how reliable the match is. If the nearest feature has another feature close to it, then there is a lesser likelihood that the match is correct. Tests in the original paper suggest that 0.8 is a reasonable threshold for this ratio, based on analysis of $40,000$ key points, and that matches with a distance ratio greater than 0.8 should be considered less reliable, thus,

$$match = \begin{cases} true & \text{if } \frac{\|f-f_c\|_2}{\|f-f_{c+1}\|_2} \leqslant 0.8 \\ false & \text{otherwise} \end{cases} \tag{13}$$

We follow this understanding in our method and use the color information of both features to scale the distance between their descriptors. In doing this, a metric of the difference in the color histograms logically moves the features apart, extending the line segment $\overline{\mathbf{uv}}$.

5) Distance Definition: The composite feature descriptor $\mathbf{f}$ is conveniently represented as a single n-dimensional vector, where n the sum of the lengths of the texture $\vec{t}$, and histogram $\mathbf{h}$.

$$\mathbf{f} = (\vec{t}, h) \tag{14}$$

In calculating the distance D between two composite descriptors, we first consider a distance measure between each of the two parts independently, d_1 and d_2, and combine the results. The texture descriptor distance d_1 is a standard calculation of the Euclidean distance between the two vectors and d_2 is the distance between the two colour histogram descriptors, $H(\cdot)$ from Eq. 11.

$$d_1 = \left\|\vec{t_1} - \vec{t_2}\right\|_2 \tag{15}$$

$$d_2 = H(\mathbf{h}_1, \mathbf{h}_2) \tag{16}$$

The individual distance measures d_1 and d_2 are then combined to yield a representative distance between the two composite descriptors. A simple sum $D = d_1 + d_2$ does not account for the difference in scale within each of the descriptors, which itself will be different depending on the choice of texture descriptor. The product $D = d_1 \times d_2$

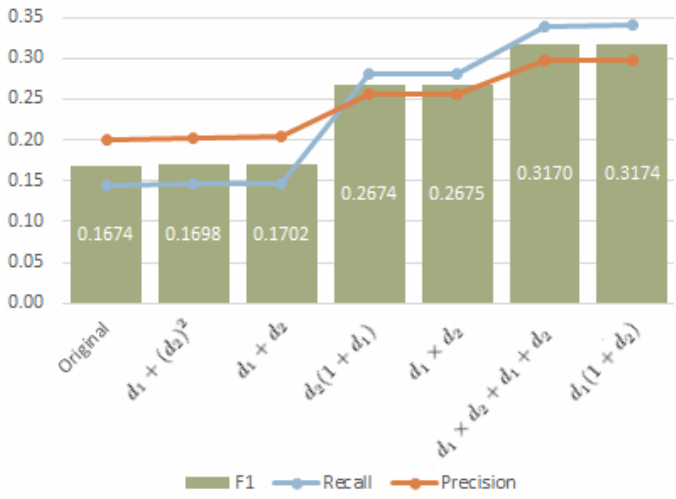

Fig. 7. F_1 score comparing a variety of combination methods in the distance calculation; $D = d_1 + (d_2)^2$, $D = d_1 + d_2$, $D = d_2(1 + d_1)$, $D = d_1 \times d_2$, $D = d_1 \times d_2 + d_1 + d_2$, and the proposed $D = d_1(1 + d_2)$, against the F_1 measure of an unmodified SIFT descriptor

down-scales the texture distance based on the color histogram distance, effectively moving similar texture descriptors closer together. This reduces the discrimination of similar textual descriptors, increasing the number of mismatches and reduces the overall accuracy. We derive a composition applying a constant multiplier to the normalized histogram distance and summing with the texture distance, in general form,

$$D = d_1 + \lambda d_2 \tag{17}$$

The selection of a suitable value for λ has been the subject of many experiments. Any empirically chosen constant value is not robust for the variety of challenging images from surveillance video images, and we therefore look to a dynamic value for λ which represents the conditions within which the feature appears.

Using d_2 as a value for λ reduces the impact of the color histogram distance because d_2 is a normalized value, which when multiplied by itself becomes smaller, and overall less discriminative. However, d_1 is a good candidate. With $\lambda = d_1$, the color distance is used to scale the distance measure of the texture descriptor so that it discriminates between similar descriptors of different colors.

$$\begin{aligned} D &= d_1 + \lambda d_2 \\ \lambda &= d_1 \\ \therefore D &= d_1(1 + d_2) \end{aligned} \tag{18}$$

Figure 7 shows a comparison of performance using a variety of combination methods; $D = d_1 + (d_2)^2$, $D = d_1 + d_2$, $D = d_2(1 + d_1)$, $D = d_1 \times d_2$, $D = d_1 \times d_2 + d_1 + d_2$, and the proposed $D = d_1(1 + d_2)$, against the F_1 measure of an unmodified SIFT descriptor, highlighting the superior performance of the proposed method in our experiment data.

We see from Eq. 18 that with $\lambda = d_1$ we apply the color distance measure as a scalar to the distance between two texture feature descriptors. Increasing the normalized value of d_2 from the range $0 \ldots 1$ into $1 \ldots 2$, thus upscaling the distance of a texture feature by multiplication. The overall distance between two composite descriptors is therefore

$$\begin{aligned} D &= d_1(1 + d_2) \\ &= \left\| \vec{t}_1 - \vec{t}_2 \right\|_2 \times \left(2 - \frac{\sum_{j=1}^{n} \min(a_j, b_j)}{\sum_{j=1}^{n} a_j} \right) \end{aligned} \tag{19}$$

The use of a scalar applied to the texture descriptor distance ensures that attributes of the texture descriptor such as invariance to affine scale and rotation transformations, are preserved. The calculation of the color histogram in Hue and Saturation channels maintains invariance in affine illumination transformations.

Distance Calculation To find the closest descriptor D_c to a given descriptor D_i it is customary to use an algorithm based on Euclidean distance, such as k-Nearest Neighbor. We perform a nearest descriptor calculation in two parts. First, the k-nearest neighbors of the texture descriptor $\vec{t}$ are found using the standard algorithm with $k = 5$, giving $\{\vec{v_1}, \vec{v_2}, \vec{v_3}, \vec{v_4}, \vec{v_5}\}$. For each of the five closest descriptors, we perform the scaling multiplication of Eq. 19 and determine the descriptor with the smallest resulting distance to be the closest, D_c

$$D_c = \arg\min_i \{D_{v_i}\} \tag{20}$$

This is not guaranteed to be optimal, but in our tests increasing k to 10 does not improve the result. This calculation does not produce a worse approximation than the common method to reduce computational complexity in a k-Nearest Neighbor search using Approximate Nearest Neighbor algorithm (ANN), which uses a randomized indexing method making the result non-deterministic, but is widely accepted for most matching tasks.

V. Empirical Evaluation

We evaluate the performance of the proposed descriptor by measuring the accuracy of matching features between pairs of images. The definition of a feature match depends on the matching strategy that is applied [35]. Our intention is to measure the accuracy of our new composite feature descriptor and distance calculation. We therefore compare our results with a nearest neighbour matching algorithm without any threshold filtering, such as Nearest Neighbour Distance Ratio to discard poor matches.

We use seven feature detectors to find initial regions of interest. Five popular intensity based key point detectors; Harris Corners detector (HARRIS), SIFT, SURF, BRISK and FAST, and two region detectors; MSER on gray scale representations and *maximally stable color regions* (MSCR) on color images. For each of these sets of features, we compare feature matching performance of descriptors extracted using SIFT and SURF, with and without our combinatorial descriptor, and later using OpponentSIFT and OpponentSURF 3-channel descriptors, again with and without our combinatorial descriptor.

The key point detectors HARRIS, SIFT and SURF are chosen because of their popularity and widespread adoption in many tasks including object classification and image retrieval [36], and BRISK and FAST for their high performance and

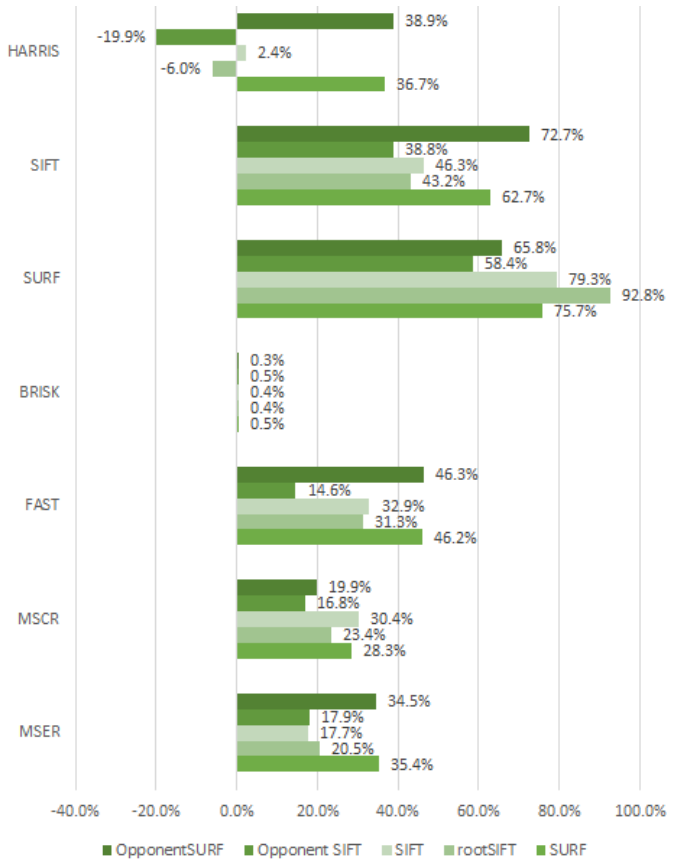

Fig. 8. Matching accuracy *blur sensitive feature detection*. Matching accuracy improvement is subject to the choice of feature detector and performance is broadly consistent across all extractors for each detector. However, Harris Corner features vary considerably for each descriptor type, and decreases matching performance in two cases.

relevance for real-time processing. We are keen to demonstrate the universal improvements of our method and therefore also include region based detectors MSER and MSCR in our comparisons.

A. Blur sensitive feature detection

We evaluate the *blur sensitive feature detection* technique independently using our seven selected feature detectors with state-of-the-art descriptors and Euclidean distance measurements. Our sharpness map contains convolutions with Gaussian kernels up to 11×11, thus $\alpha := 11$ in Eq. 5.

Figure 8 shows the percentage improvements in matching quality achieved by applying the blur sensitive feature detection algorithm to our test database. The matching accuracy improvement is subject to the choice of feature detector, which is expected because the artificial blurring of the image will affect each detector differently. The matching performance is broadly consistent across all extractors for each detector. The exception are Harris Corner features which vary considerably for each descriptor type, and decreases matching performance in two cases; rootSIFT and OpponentSIFT descriptors. BRISK features yielded consistently low improvements, and matching SURF features was generally more improved. With rootSIFT descriptors extracted from SURF key points being improved the most, by 92.8%.

Colour	H	S	V
Red	0°	100%	100%
Brown	15.1°	74.5%	64.7%
Yellow	60°	100%	100%
Green	120°	100%	100%
Blue	240°	100%	100%
Violet	300°	45.4%	93.3%
Pink	349.5°	24.7%	100%
White	0°	0%	100%
Black	0°	0%	0%
Grey	0°	0	60%

TABLE II
COLOUR PALETTE FROM [37] USED IN OUR EXPERIMENTS

B. Combinatorial descriptor assessment

We use a fixed colour histogram for all images. In experiments, the 10-bin palette of Park *et al.* [37] has proven to work well; seven colours and three special considerations for intensities (Table II). This palette has been used for the experiments presented in this paper. The descriptor extension is therefore 10 dimensions in size.

1) Query by example: A rectangular area of an image is specified as a query region containing features that are to be matched in subsequent frames of the video sequence. In our first test the query region represents a distinctive two-color back-pack being worn by a person. This region is matched against 250 video frames, each of which has ground truth information defining the boundaries of the back-pack within it.

Descriptors are created for the query image region and each image under consideration (candidate images) using the method described above. The positions of the features within the candidate image that match with the query region are then assessed relative to the ground truth and determined to be a true or false positive result or a negative result. A true positive result is a feature that matched with the query region (a *query match*) lies within the ground truth region. If a query match falls outside the ground truth then the region is labeled as false negative result. A feature matched between the images from outside the query region that falls within the ground truth region is counted as a false positive result. A match between the images from outside the query region to outside the ground truth region is not used directly within our analysis but are implicitly relative to other metrics.

Results for each feature are tallied for each image, and these are then summed across all of the images in the sequence. The true positive tp, false positive fp and false negative fn totals for the images are then used to calculate the recall and precision measures of performance of each of the four descriptors with and without our extension; $recall = \frac{tp}{tp+tn}$ and $precision = \frac{tp}{tp+fp}$.

In reporting our results we use the F-measure, the weighted harmonic mean of recall and precision, to measure and compare the accuracy of our combinatorial descriptor and distance

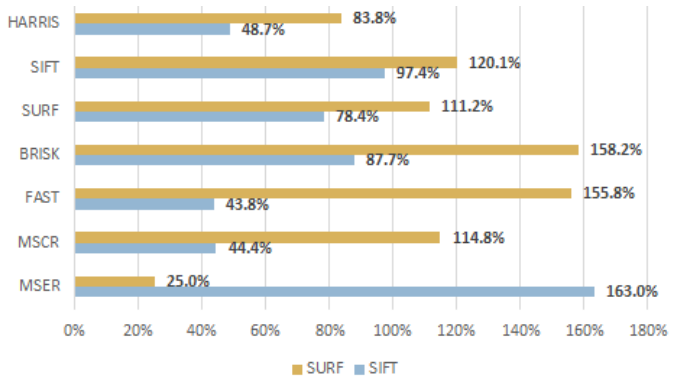

Fig. 9. Improvement of SIFT and SURF intensity descriptors using our combinatorial descriptor and distance measure. Orange bars show percentage improvements of SURF descriptors using our method, and blue bars show improvements in SIFT. The baseline uses standard descriptors with Euclidean distance measures in feature space. The overall average improvement across all of the feature descriptors in this test was 95.2%.

measure with well-known descriptors. We favor neither precision nor recall over the other, and therefore use the F_1 score, defined as

$$F_1 = \frac{(2 \times recall \times precision)}{(recall + precision)} \tag{21}$$

2) Intensity descriptors: It is important to compare the feature matching performance with popular intensity descriptors because these have the smallest dimensionality. In a large-scale processing system, size of descriptors is important for minimizing memory and disk storage and data processing time.

Our experiments compared the matching performance of SIFT and SURF descriptors against our combinatorial descriptor based on SIFT and SURF with our distance measure, for features detected using Harris Corners (HARRIS), SIFT, SURF, BRISK, FAST, MSCR and MSER (Figure 9). Feature matching is determined by the nearest neighbor feature in descriptor space. The greatest improvement was achieved with SIFT descriptors extracted from MSER features where the F_1 measure increased by 163% using our method (from 0.064 to 0.167) compared to a plain SIFT descriptor on the same MSER features.

Overall, the average improvement across all of the feature descriptors in this test was 95.2%.

Figure 10 shows two examples of matching feature descriptors from a region of interest within a query image to a subsequent frame in a surveillance video, using a SURF feature detector. The top images show matches of SURF descriptors extracted from the SURF features within the region of interest in the query image (left), and a blurred frame (right). There is a notable increase in the number of features matched into the bag region in the right hand image. The bottom images show matches from the same query frame to a sharper subsequent frame and demonstrates the reduction in false-positive matches into background features. The less cluttered Figure 11 repeats the second image pairs from Figure 10 using the distance ratio filter (Eq. 13) from [4]. There are new positive matches in both images, matching points within the rucksack that are not matched in the top row. In addition,

Fig. 10. Two examples of matching SURF features on a colored bag from a query frame (left in each pair) to a subsequent video frame. Top, matches to a blurred image perform poorly using Approximate Nearest Neighbor (blue matches) and a significant increase in matches to the target bag using our method (yellow matches). Bottom demonstrates the significantly reduced number of false positive matches to the background using our method (yellow) compared with ANN matching (blue).

Fig. 11. *Good matches* (Eq. 13) are shown for SURF features (top row) and using our method (bottom row).

the number of false positives is visibly reduced, with fewer yellow lines matched to the background in the right-hand images.

3) Color descriptors: We now assess our algorithm using two high-dimensional color descriptors, OpponentSIFT (384-dimensions) and OpponentSURF (192-dimensions), with the same features from the previous section.

The F_1 measure on our test video sequence is improved using color descriptors over using the intensity texture descriptors. This is to be expected as the color information provides a more discriminative comparison. In our test video

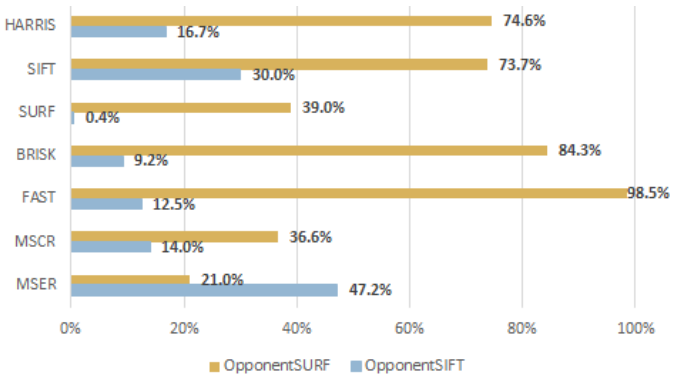

Fig. 12. Improvement of OpponentSIFT and OpponentSURF color descriptors using our combinatorial descriptor and distance measure. Orange bars show percentage improvements of OpponentSURF descriptors using our method, and blue bars show improvements in OpponentSIFT. The baseline uses standard descriptors with Euclidean distance measures in feature space. The overall average improvement across all of the feature descriptors in this test was 95.2%.

Fig. 13. Clarity of color is sensitive to the distance between the object and the camera, and distant objects begin to appear overwhelmingly gray.

Fig. 14. Difference in F_1 accuracy using only the Hue component on near-gray images. The slight reduction (2-4%) in True Positive values (blue) is compensated by a larger reduction (up to 9%) in the False Positive rate (orange). The overall F_1 measure is consistently improved in 92% of the 145 frames (green).

sequence, the best match performance was achieved using the combinatorial OpponentSIFT descriptor with FAST features, achieving an improvement of 12% over the original OpponentSIFT descriptors' accuracy of 0.415. Matching OpponentSURF descriptors of FAST features was improved the most of all color descriptors, by 98.5% but the F_1 score is low, increasing from just 0.149 to 0.296.

OpponentSIFT uses color information in the extraction of the descriptor and can be expected to out-perform those that do not use color information in a dataset in which color is visually distinctive. In their thorough evaluation of color feature descriptors, van de Sande *et al.* conclude that OpponentSIFT is generally a better performing descriptor and is a good choice where there is no prior knowledge of the images or object/scene categories [14]. In our tests, results show that our extension method generally improves matching with this descriptor by up to 47.2% depending on the initial feature detector (Figure 12).

Overall the average improvement across all of the color feature descriptors in this test was 39.8%.

4) Matching with color variations: The representation of color of an object within an image changes with many factors such as illumination, camera, distance, and weather. Our method is invariant to illumination changes by its analysis of Hue and Saturation in the HSV color space and the quantization of color to a fixed palette. The clarity of color is sensitive to the distance between the object and the camera, and distant objects begin to appear overwhelmingly gray (Figure 13).

In video sequences such as these, the color quantization to the fixed palette converges to a spike of gray pixels which subsequently reduces the performance of the color boost algorithm. Instead, such nearly-gray images can be processed using only the Hue channel of the HSV color space to quantize the colors to the fixed histogram and accentuate the dull color. The resulting histogram provides increased color information which is more discriminative than gray.

We have tested combinations of channels for quantizing color to the palette, and found that performance is optimal on a general set of images when the Hue and Saturation channels are used. However, in poor imaging conditions, performance can be improved by using only the Hue channel. The images in Figure 13 are taken from a short video sequence of 486 frames. The improvements that were gained using only the Hue channel to generate correspondences of the green checked shirt (highlighted) are shown in Figure 14. The discriminative nature of the color boosting method is clearly demonstrated with two key performance measures. While *true positive* correspondence of features was reduced by 2-4%, which typically represents only one or two features, the number of False Positive matches reduced by up to 9%. The improvement of the overall F_1-measure, plotted in green against the secondary axis, shows a maximum improvement of 100%, and in only twelve frames the F_1 measure was reduced.

An intelligent implementation can adjust which channels are used for the color quantization based upon real-time analysis of the resulting histogram. If HS channels yield a histogram that is biased to gray, then the quantization calculation should be repeated with only the Hue channel, H.

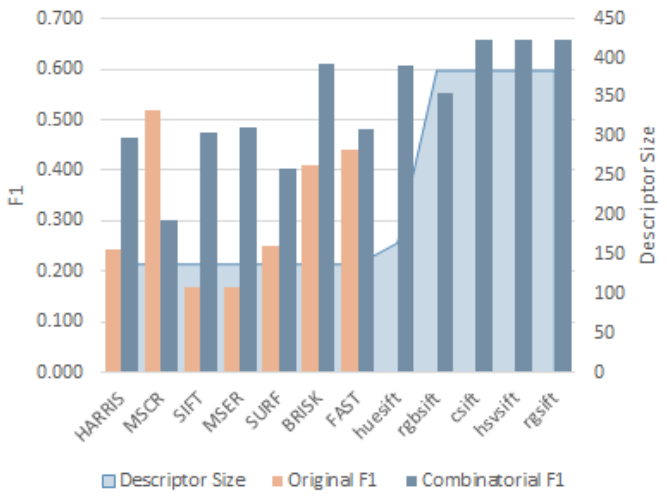

Fig. 15. Comparing the F_1 score of state-of-the-art colour descriptor with our combinatorial method

5) State of the Art Color Descriptors: There have been a number of color descriptors proposed, and we compare state-of-the-art color SIFT descriptors to our extension applied to intensity SIFT descriptors. SIFT descriptors are extracted from the images at feature positions detected by our trial set of detectors; Harris Corners, SIFT, SURF, BRISK, FAST, MSCR and MSER. We apply our combinatorial extension to these SIFT descriptors and measure the F_1 score on our test dataset. A distance ratio filter (Eq. 13) is applied to ensure we are comparing the robust matches in all cases. The F_1 scores are then compared with those achieved on the same dataset using state of the art color SIFT descriptors HUE-SIFT, RGB-SIFT, C-SIFT, HSV-SIFT and RG-SIFT (Figure 15) using the implementation of [14]. HUE-SIFT has dimension $D = 165$ and all the others have $D = 384$. Our combined descriptor is $D = 138$, based on $D = 128$ SIFT with a 10 bin colour extension. The F_1 score of the original descriptor is shown alongside the F_1 score of our combined descriptor. In all cases apart from features detected by MSCR, the combined descriptor shows a large increase in F_1 over the original descriptor.

A combinatorial extension to SIFT descriptors of SIFT features shows the largest improvement, and comes close to matching the accuracy of C-SIFT, which is three times slower to compute and nearly three times the size. SIFT descriptors extracted from BRISK features is a significant result for our application as BRISK is a high performance detector and with our combinatorial extension, we achieve an F_1 score that improves on HUE-SIFT and RGB-SIFT and comes close to the others.

Selection of an appropriate feature detector and descriptor is difficult, and the best performing is not universal across all images or all applications. Our method significantly improves the F_1 measurement of accuracy using fast-to-compute detectors to match or exceed state-of-the-art color descriptors with much lower memory requirements.

C. Feature matching results

The graphs in Figure 16 summarize the results from our test database of 251 images. Each graph shows the F_1 measure of one of our seven selected feature detectors and all four of the feature extractors, comparing the matching performance of four methods of calculating correspondence. The pale blue line shows SIFT features extracted from each of the feature key points or region centers, the orange line shows rootSIFT features, SURF is in gray, and color features of OpponentSIFT and OpponentSURF are in yellow and dark blue respectively. Each of the four methods are represented on the x-axis; the *original* correspondence using Euclidean distance of unmodified feature descriptors is the baseline against which we measure performance improvements. *Blur sensitive* applies the blur sensitive feature detection algorithm using unmodified feature descriptors. *Combinatorial* results are those achieved in using the combinatorial texture and color feature matching descriptor extensions and matching algorithm, and finally *Combinatorial Blur sensitive* are results from the combined methodology described in this paper.

The upward left-to-right trend in each of the graphs demonstrates the improvement in matching performance that is achieved with each of our method's components, and the combined methodology. The consistent closeness of the orange and yellow lines in the *Combinatorial Blur sensitive* result is particularly striking. The performance of our method using rootSIFT descriptors ($128 + 10$ dimensions) closely matches the performance of the much larger OpponentSIFT $384 + 10$ dimension descriptor and significantly outperforms state-of-the-art feature matching using the OpponentSIFT 128D descriptors with the Euclidean distance measure.

VI. Storage efficiency vs. matching performance

The choice of feature detector to use in the initial step of the processing pipeline significantly affects the ability to match features across images. The variability of matching accuracy is observable in the results presented in this paper. Systems attempting to match features across a high volume of images are becoming increasingly common, and a key consideration for such systems is the storage efficiency of the descriptors used and the trade-off between storage and accuracy.

The accuracy of feature matching using contemporary techniques generally increases in line with the size of the descriptor that is determined by a choice of feature extractor. Figure 17 demonstrates this where the tops of the bars the represent the peak performance on each descriptor, and the yellow bars of established descriptors are generally higher moving left-to-right. The green bars show the F_1 matching accuracy using our method, where there is a peak in matching accuracy at dimensionality $D = 138$ where our method using SIFT and rootSIFT descriptors outperforms all other state-of-the-art descriptor matching using Euclidean distance measures. The saving in storage using our *Combinatorial rootSIFT* over the performance-comparable *Combinatorial OpponentSIFT* is $394 - 138 = 256$ bytes per descriptor.

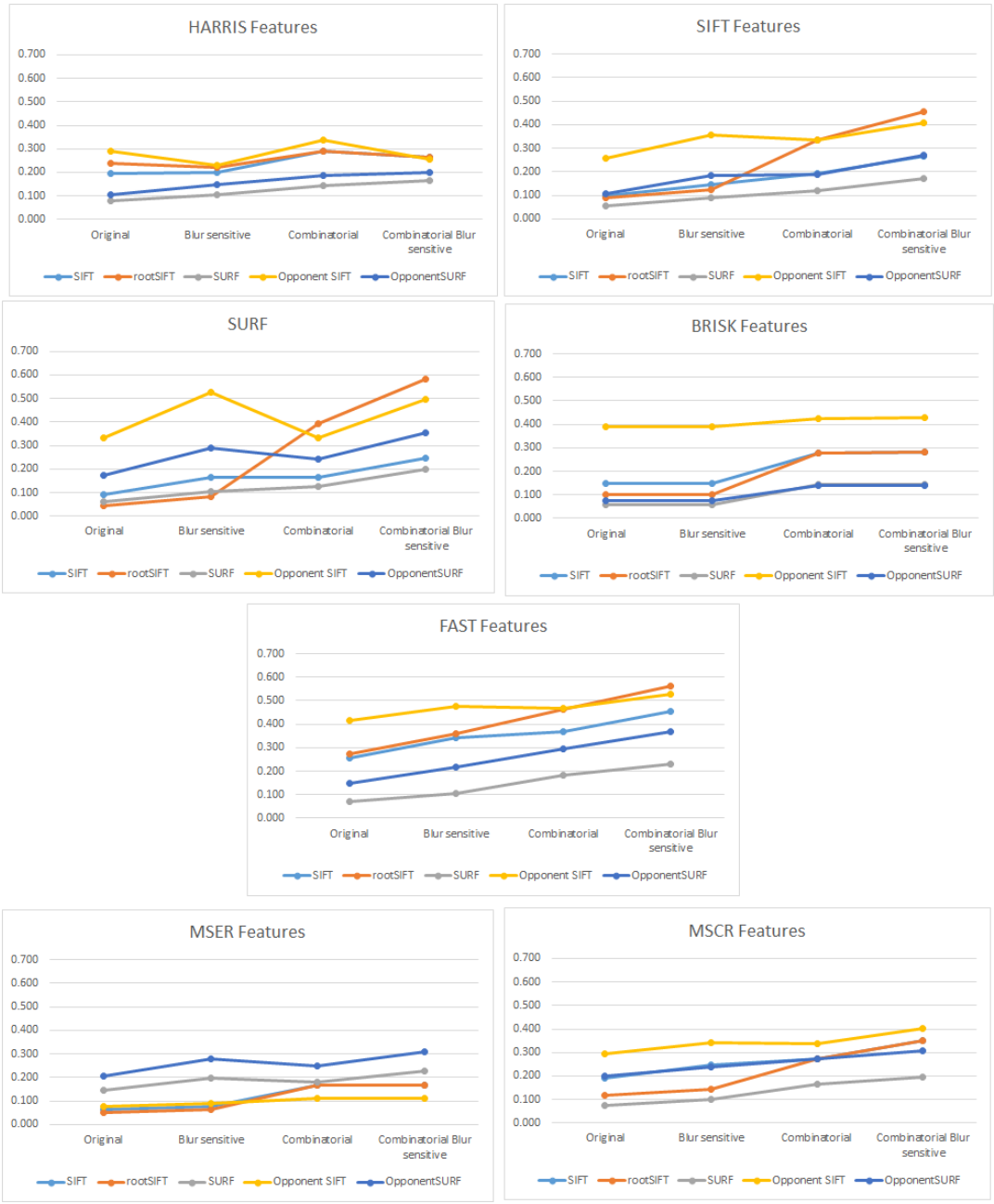

Fig. 16. Summary of the results of feature matching with each component of our method, and the combined methodology (right-most). Each graph shows results from a different feature detector, and compares results with each of four intensity and color descriptors using four methods; *original* – using Euclidean distance of unmodified feature descriptors is the baseline against which we measure performance improvements, *Blur sensitive* – blur sensitive feature detection algorithm using unmodified feature descriptors, *Combinatorial* – combinatorial texture and color feature matching descriptor extensions and matching algorithm, *Combinatorial Blur sensitive* – the combined methodology described in this paper.

VII. Conclusion

We have introduced a methodology for improved discriminative feature correspondence in low-quality images, with an emphasis on storage optimization and execution performance. Our efficient and generic extension for feature descriptors improve the performance of feature matching and the blur sensitive feature detection method further enhances feature matching performance. We have shown the flexibility of the approach by applying it to five common key point descriptors and two popular region descriptors and we have compared the performance of all of them in matching features between images varying in quality and appearance. Our experiments have demonstrated that the introduction of color information to the feature descriptors, a unique feature distance measure and compensating for inter-image blur differences can improve the matching accuracy over the original descriptors in most combinations that were tested, even where the color detail is visually subtle in poor quality images.

Our method provides flexibility as it can be used with any feature descriptor extracted from any key point or region detector. Further, evaluation of the method in our problem domain

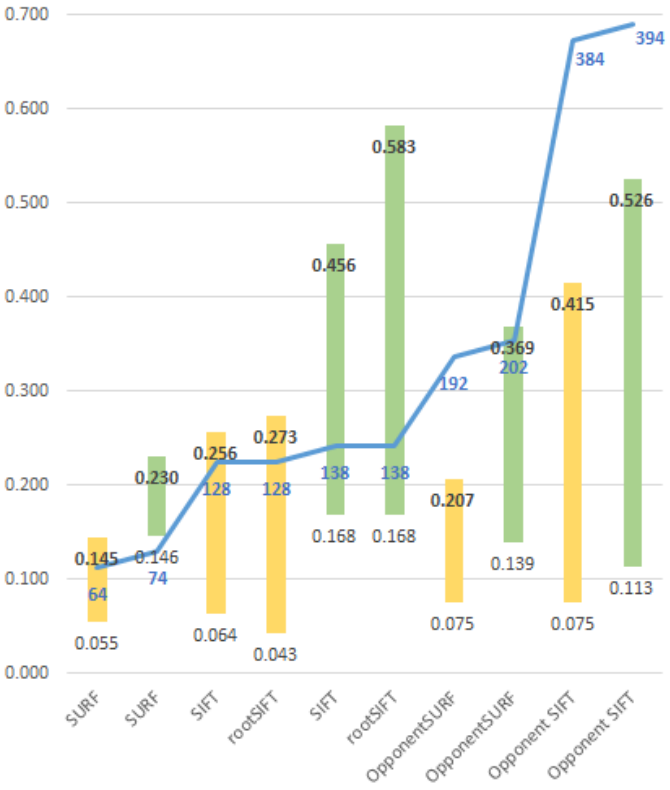

Fig. 17. The correlation between descriptor size and matching accuracy. Yellow bars show measures for established descriptors and Green bars are accuracy measures using our method. Using our method with SIFT and rootSIFT 138D combinatorial descriptors out-perform descriptors of almost 3 times the size.

of frame-to-frame feature tracking in low quality videos has demonstrated that smaller descriptors that are computationally lighter can be used to out-perform larger and more expensive feature descriptors. Our experiments have demonstrated an accuracy in matching features that out-performs all state-of-the-art methods using descriptors of less than 36% of size of the nearest performing color descriptor.

ACKNOWLEDGMENTS

This work is funded by the European Union's Seventh Framework Programme, specific topic "framework and tools for (semi-) automated exploitation of massive amounts of digital data for forensic purposes", under grant agreement number 607480 (LASIE IP project). The authors also extend their thanks to the Metropolitan Police at Scotland Yard, London, UK, for the supply of and permission to use CCTV images.

REFERENCES

[1] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *Proceedings Ninth IEEE International Conference on Computer Vision*, vol. 2, Oct. 2003, pp. 1470–1477. [Online]. Available: http://doi.org/10.1109/ICCV.2003.1238663

[2] A. Anjulan and N. Canagarajah, "A unified framework for object retrieval and mining," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 1, pp. 63–76, Jan. 2009. [Online]. Available: http://doi.org/10.1109/TCSVT.2008.2005801

[3] S. O'Hara and B. A. Draper, "Introduction to the Bag of Features Paradigm for Image Classification and Retrieval," Jan. 2011. [Online]. Available: http://arxiv.org/abs/1101.3354

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://doi.org/10.1023/B:VISI.0000029664.99615.94

[5] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*. IEEE, Nov. 2011, pp. 2548–2555. [Online]. Available: http://doi.org/10.1109/ICCV.2011.6126542

[6] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in *Procedings of the Alvey Vision Conference 1988*. Alvey Vision Club, 1988, pp. 147–151. [Online]. Available: http://doi.org/10.5244/C.2.23

[7] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Computer VisionECCV 2006*, vol. 3951, pp. 430–443, 2006. [Online]. Available: http://doi.org/10.1007/11744023_34

[8] G. Nebehay and R. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *2014 IEEE Winter Conference on Applications of Computer Vision, WACV 2014*. IEEE, Mar. 2014, pp. 862–869. [Online]. Available: http://doi.org/10.1109/WACV.2014.6836013

[9] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," *Computer VisionECCV 2010*, vol. 6314, no. 4, pp. 778–792, 2010. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-15561-1_56

[10] P.-E. Forssén and D. G. Lowe, "Shape descriptors for maximally stable extremal regions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007, pp. 1–8. [Online]. Available: http://doi.org/10.1109/ICCV.2007.4409025

[11] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *European Conference on Computer Vision (ECCV'08)*, 2008, pp. 183–196. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-88688-4_14 http://www.mendeley.com/catalog/linear-time-maximally-stable-extremal-regions/

[12] P. M. Roth, M. Donoser, and H. Bischof, "Tracking for learning an object representation from unlabeled data," *Proceedings of the Computer Vision Winter Workshop (CVWW)*, pp. 46–51, 2006. [Online]. Available: http://www.icg.tugraz.at/Members/pmroth/pub_pmroth/pmroth06a.pdf

[13] P.-E. Forssén, "Maximally stable colour regions for recognition and matching," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8. [Online]. Available: http://doi.org/10.1109/CVPR.2007.383120

[14] K. Van De Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, Jun. 2010. [Online]. Available: http://doi.org/10.1109/TPAMI.2009.154

[15] J. Zhang, Y. Barhomi, and T. Serre, "A new biologically inspired color image descriptor," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, 2012, vol. 7576 LNCS, no. PART 5, pp. 312–324. [Online]. Available: http://doi.org/10.1007/978-3-642-33715-4_23

[16] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 4, pp. 712–27, Apr. 2008. [Online]. Available: http://doi.org/10.1109/TPAMI.2007.70716

[17] A. E. Abdel-Hakim and A. A. Farag, "CSIFT: A SIFT descriptor with color invariant characteristics," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1978–1983. [Online]. Available: http://doi.org/10.1109/CVPR.2006.95

[18] H. Stokman and T. Gevers, "Selection and Fusion of Color Models for Feature Detection.pdf," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 560–565. [Online]. Available: http://doi.org/10.1109/CVPR.2005.315

[19] J. van de Weijer and C. Schmid, "Coloring Local Feature Extraction," in *9th European Conference on Computer Vision*, vol. 3952, 2006, pp. 334–348. [Online]. Available: http://doi.org/10.1007/11744047_26

[20] C. Wengert, M. Douze, and H. Jégou, "Bag-of-colors for improved image search," in *Proceedings of the 19th ACM international conference on Multimedia - MM '11*. New York, New York, USA: ACM Press, 2011, p. 1437. [Online]. Available: http://doi.org/2072298.2072034

[21] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., vol. 2. IEEE, 2004, pp. 506–513. [Online]. Available: http://doi.org/10.1109/CVPR.2004.1315206

[22] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2012, pp. 2911–2918. [Online]. Available: http://doi.org/10.1109/CVPR.2012.6248018

[23] L. Sun and G. Liu, "Visual Object Tracking Based on Combination of Local Description and Global Representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 4, pp. 408–420, Apr. 2011. [Online]. Available: http://doi.org/10.1109/TCSVT.2010.2087815

[24] M. Chli and A. Davison, "Active matching," *Computer VisionECCV 2008*, vol. 5302, pp. 72–85, 2008. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-88682-2_7

[25] S. A. Fezza, M.-C. Larabi, and K. M. Faraoun, "Feature-based Color Correction of Multi-View Video for Coding and Rendering Enhancement," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2014. [Online]. Available: http://doi.org/10.1109/TCSVT.2014.2309776

[26] R. Dash and B. Majhi, "Motion blur parameters estimation for image restoration," *Optik - International Journal for Light and Electron Optics*, vol. 125, no. 5, pp. 1634–1640, Mar. 2014. [Online]. Available: http://doi.org/10.1016/j.ijleo.2013.09.026

[27] C. Schuler and M. Hirsch, "Learning to Deblur," 2014. [Online]. Available: http://arxiv.org/abs/1406.7444

[28] J. Canny, "A computational approach to edge detection." *IEEE transactions on pattern analysis and machine intelligence*, vol. 8, no. 6, pp. 679–698, Nov. 1986. [Online]. Available: http://doi.org/10.1109/TPAMI.1986.4767851

[29] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674–679.

[30] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*. IEEE Comput. Soc. Press, 1994, pp. 593–600. [Online]. Available: http://doi.org/10.1109/CVPR.1994.323794

[31] H. L. H. Liu and R. Setiono, "Chi2: feature selection and discretization of numeric attributes," in *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, 1995, pp. 388 – 391. [Online]. Available: http://doi.org/10.1109/TAI.1995.479783

[32] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.

[33] Y. Rubner, C. Tomasi, and L. J. Guibas, "Earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000. [Online]. Available: http://doi.org/10.1023/A:1026543900054

[34] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991. [Online]. Available: http://doi.org/10.1007/BF00130487

[35] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005. [Online]. Available: http://doi.org/10.1109/TPAMI.2005.188

[36] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, vol. 2, IEEE. IEEE, 2006, pp. 2169–2178. [Online]. Available: http://doi.org/10.1109/CVPR.2006.68

[37] U. Park, A. Jain, I. Kitahara, K. Kogure, and N. Hagita, "ViSE: Visual Search Engine Using Multiple Networked Cameras," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 1204–1207. [Online]. Available: http://doi.org/10.1109/ICPR.2006.1176

Craig Henderson received a B.Sc. degree in Computing for Real Time Systems from the University of the West of England in Bristol, UK in 1995. From 1995 to 2014 he work in a variety of organisations as a Software Engineer and Engineering Manager.

Since 2014, he is a PhD student in the Multimedia and Computer Vision and Laboratory at the School of Electronic Engineering and Computer Science, Queen Mary, University of London, London, UK. His research interests include computer vision, machine learning and scalable systems.

Ebroul Izquierdo (M'95-SM'03) received the M.Sc., Ph.D., C.Eng., FIET, SMIEEE, MBMVA degrees. For his thesis on the numerical approximation of algebraic-differential equations, he received the Dr. Rerum Naturalium (Ph.D.) degree from Humboldt University, Berlin, Germany.

He is the Head of the Multimedia and Vision Group, School of Electronic Engineering and Computer Science at Queen Mary, University of London, UK. He has published over 500 technical papers including book chapters and holds several patents.