Fast Local Spatial Verification for Feature-Agnostic Large-Scale Image Retrieval

Joel Brogan, *Member, IEEE*, Aparna Bharati, *Member, IEEE*, Daniel Moreira, *Member, IEEE*, Anderson Rocha, *Senior Member, IEEE*, Kevin W. Bowyer, *Life Fellow, IEEE*, Patrick J. Flynn, *Fellow, IEEE*, and Walter J. Scheirer, *Senior Member, IEEE*

***Abstract*— Images from social media can reflect diverse viewpoints, heated arguments, and expressions of creativity, adding new complexity to retrieval tasks. Researchers working on Content-Based Image Retrieval (CBIR) have traditionally tuned their algorithms to match filtered results with user search intent. However, we are now bombarded with composite images of unknown origin, authenticity, and even meaning. With such uncertainty, users may not have an initial idea of what the search query results should look like. For instance, hidden people, spliced objects, and subtly altered scenes can be difficult for a user to detect initially in a meme image, but may contribute significantly to its composition. It is pertinent to design systems that retrieve images with these nuanced relationships in addition to providing more traditional results, such as duplicates and near-duplicates — and to do so with enough efficiency at large scale. We propose a new approach for spatial verification that aims at modeling object-level regions using image keypoints retrieved from an image index, which is then used to accurately weight small contributing objects within the results, without the need for costly object detection steps. We call this method the Objects in Scene to Objects in Scene (OS2OS) score, and it is optimized for fast matrix operations, which can run quickly on either CPUs or GPUs. It performs comparably to state-of-the-art methods on classic CBIR problems (Oxford 5K, Paris 6K, and Google-Landmarks), and outperforms them in emerging retrieval tasks such as image composite matching in the NIST MFC2018 dataset and meme-style imagery from Reddit.**

***Index Terms*— Image forensics, forgery detection, image retrieval, content retrieval.**

Manuscript received August 4, 2020; revised April 28, 2021 and June 25, 2021; accepted July 2, 2021. Date of publication July 21, 2021; date of current version August 6, 2021. This work was supported in part by Defense Advanced Research Projects Agency (DARPA), in part by the Air Force Research Laboratory (AFRL) under Agreement FA8750-16-2-0173 and Agreement FA8750-20-2-1004, in part by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) through the DéjàVu Project under Grant 2017/12646-3, in part by Coordenação de Aperfeiçoamento de Pessoal de Nível (CAPES) through the DeepEyes Grant, and in part by National Council for Scientific and Technological Development (CNPq) under Grant 304472/2015-8. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Francesco G. B. De Natale. *(Corresponding author: Joel Brogan.)*

Joel Brogan is with the Multimodal Sensor Analytics (MSA) Group, Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA (e-mail: broganjr@ornl.gov).

Aparna Bharati is with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015 USA.

Daniel Moreira, Kevin W. Bowyer, Patrick J. Flynn, and Walter J. Scheirer are with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556 USA.

Anderson Rocha is with the Institute of Computing, University of Campinas, Campinas, Sao Paulo 13083-970, Brazil.

Digital Object Identifier 10.1109/TIP.2021.3097175

I. Introduction

COMPLEX composite images such as Internet memes are pervasive in the pools of content shared by users on social media [5]. Content overlap between images is often nuanced and subtle, perhaps even to the extent of not being immediately noticed. It can be the result of processes both natural (i.e., a company logo present within two otherwise unrelated photographs) or synthetic (i.e., software-manipulated composite imagery). Some instances of these images convey significant cultural value [47], while others represent odious extremist propaganda [11]. These examples are relevant and timely topics of study, and they represent cases where the retrieval of the source content in reference image collections, as well as quantification of their shared content, are both paramount. Research cases that stand to benefit from this type of retrieval capability are already emerging within the field. For example, the identification of near-duplicate and overlapping content shared between artistic imagery has received attention recently. This problem can be addressed using densely-matched learned features that are both discriminative and invariant [45]. But methods like these, which discover shared content between pair-wise sets of images, are computationally intense and slow. Further, the evolution of an image like an Internet meme can be traced by finding all of the related images, including images that contributed small-donor objects (which are prevalent in the example shown in Fig. 1), and referencing associated timestamps [6], [30]. Similarly, one can debunk misleading or forged images being used for disinformation purposes by verifying identified source material. As an answer to the retrieval-based needs these research tasks require, in this paper, we propose a new solution for a general-use image retrieval system designed to accurately represent the nuanced relationships between images that partially share content, in addition to performing classical near-duplicate retrieval. Our solution utilizes a novel combination of spatial verification and scoring methods to model object-level regions and characterize small areas of matching content between images in large datasets.

Considering this problem more generally, one of the major challenges in image retrieval is the *semantic gap* [49], [57],

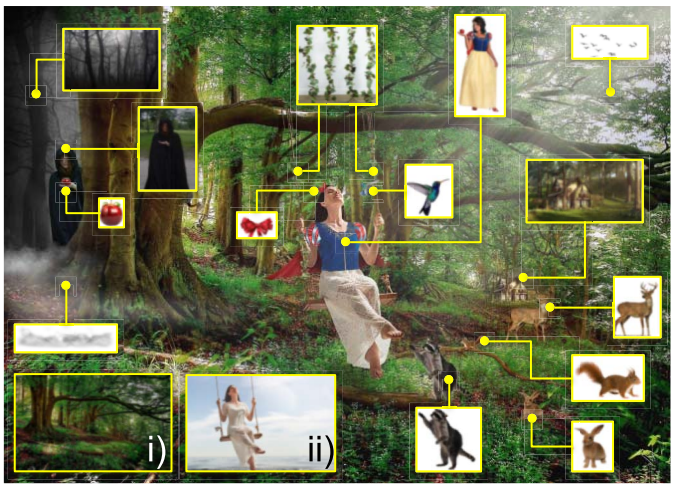

Fig. 1. An example of a meme-style image from the *r/photoshopbattles* subreddit. Internet memes are humorous messages that are spread on social media, often conforming to a set genre with a distinct style. In this paper, we provide spatial verification to find object-level correspondences between images, which assist in retrieving the donor images (highlighted in yellow) that contribute to composites like the one above.

where a user's understanding of the content transcends the low-level representation and subsequent recall capabilities of a content-based search engine. Traditional image retrieval tasks start with the knowledge of what a user is looking for at query time and end with a set of relevant results that match those initial expectations. Thus it follows that this retrieval approach must be reimagined in the age of social media, where expectations can shift rapidly. Such an algorithm must characterize, retrieve, and appropriately score images with only partially matching, possibly transformed, content, such as in Fig. 1. Ideally, a good algorithm should return results that satisfy the user's higher-level intent, not just images with similar low-level features. However, retrieval approaches developed only around this idea may not be sufficient for parsing composite imagery and retrieving relevant results, especially if there are aspects of an image that are not immediately apparent to the user at query time. In addition to the context of image creation and use, an image's extent is also shifted as it can be reused in composite images later on. Finally, given the free-form nature of composite imagery, a practical retrieval approach is expected to operate over a web-scale database to maximize potential matching candidates.

With the above aspects in mind, our goal is to adapt image retrieval to different contexts, such as composite images, especially those containing small spliced objects. For example, in Fig. 1, we see a composite image created from many smaller donor images. This type of image poses a significant challenge for existing image retrieval algorithms because if the host (*i.e.*, background) and donor images are matched globally, the latter group would not be highly scored since the content shared with the composite is very small. Nonetheless, tasks like meme analysis and disinformation debunking require retrieving each meaningful piece of content in an image under scrutiny. Because the image in question is a conglomeration from many sources, we can assume that the goal of a retrieval system should be to return instances of all images contributing to it.

To do so, we propose a new spatial verification method that allows object-level instance scoring of retrieved results without the need for costly object detection steps. We devise a feature-agnostic algorithm that utilizes a geometrically consistent voting measure inspired by the Spatially Constrained Similarity Measure (SCSM) [46] technique, with the major difference being that object regions of interest in the query need not be known ahead of time, in order to make the solution more appropriate to the current reality of unspecified retrieval context. As we show through experiments, the proposed method quickly and accurately localizes and ranks rigid objects contained within the query image to objects contained within a large image database. We call this method the Objects in Scene to Objects in Scene (OS2OS) score, and it is optimized for fast matrix operations on CPUs or GPUs.

Through the rest of this paper, we look at approaches related to the proposed OS2OS score, detail our methodology, and then perform experiments utilizing several relevant image retrieval datasets and feature representations, including both classic handcrafted and contemporary deep features. In summary, the contributions are:

- A new perspective on the problem of image retrieval, which aims to address the deficiencies in existing problem formulations for retrieval in cases of partial overlapping content, including object sharing, complex composite images, and manipulated images.
- A new method called the OS2OS score for spatial verification of matching objects between images, including tiny objects that are important for understanding memes and other emerging Internet media.
- A series of experiments showing the generalized viability of the OS2OS approach on classical retrieval datasets, including the Oxford 5K [38], Paris 6K [40], and Google-Landmarks [33], and specific viability of the algorithm for composite image retrieval using the NIST MFC2018 [31] datasets, as well as meme-style imagery from Reddit [30].
- A new experimental protocol for the Reddit Photoshop Battles dataset [42], preparing it to be used for benchmarking potential solutions to the problem of retrieving the donors of composite and meme-style images.

II. Related Work

Typical CBIR solutions rely on the multi-level representation of the images to reduce the semantic gap between the pixel values and the system user's retrieval intent. In the lowest levels, typical methods use local features (*e.g.*, keypoints) to obtain n-dimensional descriptions of the image content, ranging from handcrafted representations, such as SIFT [27] and SURF [4], to representations learned via neural networks, such as LIFT [26] and DELF [33].

In the subsequent levels, these local features are then used to index and compare, within the n-dimensional space they constitute and through Euclidean distance or a similar method, pairs of image localities (*i.e.*, image patches). State-of-the-art large-scale indexing solutions comprise methods such as Optimized Product Quantization (OPQ) [13] or Generalized Product Quantization (GPQ) [20] for approximate

nearest neighbor (ANN) search, with Inverted File Indices (IVF) [24].

As proposed by Lowe [27], two features and their respective image patches are probably a *match* (*i.e.*, they depict the same object or scene region in different configurations), if one is the nearest neighbor of the other within the feature space. Depending on the nature of the CBIR application, local features may be indexed in ways that use only the feature space and ignore or underutilize the scale, orientation, or (x, y) positions of the features within the images to which they belong. For instance, local feature aggregation methods such as VLAD [1] or Aggregated Selective Match Kernels (ASMK) [52] can provide some implicit spatial proximity of features, but are not able to utilize the aforementioned geometric data for stricter verification. Likewise is the case for bag-of-features [48] and similar approaches [23], [37], which are most useful for tasks such as retrieving globally semantically similar images with significant content overlap (*i.e.*, near duplicates). The global voting strategies [22] they employ are not able to provide satisfactory results for small, localized areas, such as those in Fig. 1. This motivates the use of additional spatial verification steps, such as the ones proposed in this work, to ensure that the local features being matched present a geometric coherence in either their scale, orientation, or (x, y) positions.

Local-feature spatial verification methods can be organized into two categories, which are described below.

Region-Based Image Retrieval (RBIR) and Object-Based Image Retrieval (OBIR). In these retrieval approaches, the image is segmented into a set of regions, either grid-based object segmentation-based [19]. Each region is then individually searched, and results from all regions are fused and re-ranked. Methods such as the one described by Sun *et al.* [50] perform CBIR directly on object detections provided by some detection framework (in their case, Binarized Normed Gradients, or "Bing" [8]). Although detection can be fast, image search is effectively slowed by a factor that scales linearly in the number of objects detected in the query. Alternatively, Reddy Mopuri and Venkatesh Babu [43] and Tolias *et al.* [53] pool object-level features from a CNN to obtain a single compact representation. In Mohedano *et al.* [29], images are partitioned into a rigid grid, with each region being described by a CNN, where resulting activation maps are used aggregated in a Bag of Visual Words approach. The method can suffer when objects of interest straddle region boundaries. Vaccaro *et al.* [54] solves this setback by utilizing a multi-scale pyramid of regions, instead.

Some methods employ adaptive region boundaries, however the success of these region-based algorithms [17] relies on the segmentation algorithms they employ. New methods such as Regional ASMK [51] increase performance by training an application-specific object proposal network, tuned specifically for landmark detections. While these methods can be made better with object classifiers, they are not general and do not incorporate explicit geometric constraints. The absence of this property hinders the balanced scoring of relevant small regions, significant for the OS2OS problem.

Hypothesis-oriented spatial verification. Methods in this category start with a set of spatial transformation hypotheses of one image towards the other (*e.g.*, affine transformation). As initially proposed in the RANSAC [12] algorithm, these hypotheses are iteratively generated for random samples of feature matches and are evaluated according to the overall number of matches that are *inliers* (*i.e.*, matches that comply with the hypothesis). Aiming to make the process more accurate and deterministic, Philbin *et al.* introduced the Fast Spatial Measure (FSM) algorithm [38], which generates one hypothesis for every single feature match. Although very accurate, the major drawback of techniques from this category is the significant runtime they demand, which is a quadratic function of the number of available features (as we show through experiments in Fig. 4).

A more recent take on this approach is the work from Cao *et al.* [7], which employs a deeply-learned global feature vector for initial CBIR retrieval, then re-ranks results using deeply-learned local feature matching and spatial verification. While the method produces competitive results, the algorithm's global feature backbone constrains results to have global similarity, which is not an assumption that can be made with composite images.

Hough transform-based spatial verification. Methods from this category start with Hough transforms [3], [9] and the computation of histograms for their parameters, where each bin quantifies the number of feature matches in agreement. Lowe [27] proposed the adoption of a four-parameter Hough transforms, computing bins concerning the product of (i) x and (ii) y position coordinates, (iii) scale, and (iv) orientation of the features. The largest accumulated histogram bin is then chosen to select a better and potentially reduced set of feature matches, along with an initial transformation hypothesis, before applying RANSAC. Aiming to speed up the overall process, Avrithis and Tolias later introduced the Hough Pyramid Matching (HPM) strategy [2], which employs a hierarchical voting strategy to recursively split pairs of feature matches into bins in a top-down fashion (from coarser to finer correspondences), as a way to evaluate the pairwise affinities of matches without enumerating all pairs. Because Hough features are binned using all four spatial features, hypothesis voting is carried out with respect to global rotation and scale values, and cannot account for localized changes to an object that has been transformed inconsistently from its global background.

Similar to HPM, Li *et al.* also suggested the evaluation of pairs of feature matches to develop their Pairwise Geometric Matching (PGM) strategy [25]. However, they proposed to use coarser (and therefore faster to verify) two-parameter Hough transforms, relying only on the orientation and scale change between the compared images. This indeed guarantees a significant speed-up in the spatial verification process, as we show through experiments reported in Fig. 4.

Another relevant work was proposed by Shen *et al.*, namely, Spatially Constrained Similarity Measure (SCSM) [46]. Although this work also makes use of four-parameter Hough transform quantization to enumerate candidate spatial transformations, it differs in the method employed to select the

best transformation. By demanding the establishment of a bounding box over the features of the query object before performing the retrieval task, the algorithm uses the center of this box to measure the candidate transformations' quality. Therefore, the best transformation is the one that — after its application — best preserves the spatial relations among the feature positions and the box center.

Putting the proposed method in context with prior work. Our method belongs to the latter category of spatial verification techniques and is agnostic to the chosen local features and feature indexing approach. Compared to the literature, the novelty of this work comprises, besides the unique combination of strategies for the task at hand:

- Inspired by SCSM [46], the computation of match-set-wise centroids significantly reduces Hough voting distances when evaluating multiple Hough-based hypotheses, without the need for selecting bounding boxes or object regions of interest in the query ahead of retrieval time. These centroids are calculated in a novel way (see Eq. 2).
- Inspired by PGM [25], the use of a coarse (and thus fast), but still accurate, two-parameter Hough transform quantization, which accumulates Hough votes within bins to generate a list of candidate transformations. Contrary to PGM, our method relies on the x and y position coordinates of the matched features transformed by their respective scale and orientation, instead of solely using scale and orientation. This allows for local scale, rotation, and affine transforms of objects within the scene that may not cohere to the global transformation hypothesis.
- A two-stage match filtering strategy that helps eliminate spurious matches with linear computational cost, based on removing all one-to-many and many-to-one matches that fall within a local transformation hypothesis bin.
- A novel image retrieval score (OS2OS score, see Eq. 11), which is based on two complementary measures of object-level image matching quality (namely, object centrality, and angle coherence), and allows for spatial verification of both global content and localized regions while ranking images.

In the following section, we detail each step of the proposed algorithm and discuss the reasons and advantages of adopting each of the above new aspects.

III. Objects in Scene to Objects in Scene (OS2OS) Score

Our proposed spatial verification method for image retrieval can be explained in six steps.

Step 1 (Local Feature Affinity): Images are described through local feature vectors and their respective geometric data, namely (x, y) location, scale, and orientation angle. Let Q be the set of all features extracted from the query, as depicted in Fig. 2 (i), and D be the set of all features extracted from a target image database, as depicted in Fig. 2 (ii). For a particular query feature $q_i \in Q$, with $i = \{1 \dots |Q|\}$, we compute the K nearest neighbors (KNN) $d_j \in D$ in the feature space only, ignoring the images they come from. As a result, q_i participates in K matches $m_{ij} = (q_i, d_j)$, where $j = \{1 \dots K\}$. A score S for a given match m_{ij} that can express the affinity between q_i and d_j is calculated via the L_2-distance and the rank-adaptive scoring outlined in [22]:

$$S(m_{ij}) = max(0, \|q_i, d_\phi\|_2 - \|q_i, d_j\|_2), \tag{1}$$

where ϕ is a fixed rank position of reference (usually $K/2$).

In practice, features are commonly searched and retrieved using approximate K-Nearest-Neighbor heuristics that circumvent the need to enumerate all pairwise comparisons in a feature index exhaustively. In our work, we utilize Inverted File Indexes (IVF) [24], and Optimized Product Quantization (OPQ) [13] to store and retrieve the K-nearest neighbors concerning a query feature q_i. Section IV-C provides in-depth detail as to specific IVF implementation used within this paper.

Step 2 (Image-Pairwise Centroid Calculation): Take an image P from the database that shares content with the query. There might be a set of feature matches between them, whose incident locations (x, y) onto the query should give a rough indication of the shared object's location within Q. Therefore, if we calculate the center of these P-wise match locations on the query, we find a point that is *generally* near a potential object of interest, serving as an estimation of its centroid c (see Fig. 2 (iv)). Let M be the set of feature matches m_k shared between P and the query, with $k = \{1 \dots |M|\}$, and let $Q_m \subseteq Q$ be the set that contains only the query features that have a match to P. To obtain c, we use the Euclidean center of the query features $q_k \in Q_m$. Nevertheless, aiming to consider the quality of the matches while computing c, and to deal with spurious matches, we also weight the added features according to their affinity scores $S(m_k)$ (see Eq. 1) associated with the features of P:

$$c = \frac{\sum_{k=1}^{|M|} L(q_k) \times S(m_k)}{\sum_{k=1}^{|M|} S(m_k)}, \tag{2}$$

where $L(q_k)$ is the (x, y) location of the k-th feature $q_k \in Q_m$, and $m_k \in M$ is the respective k-th feature match between the query and P. The motivation for this is that the more similar two matched features are (*i.e.*, the higher their value of $S(.)$), the more they contribute to the position of c. This strategy reduces the Hough voting noise problem described in [44], as shown via the ablation experiments in Sec. V, where we report the decrease in performance due to the absence of centroid computation.

Step 3 (Centroid-Relative Feature Projection): Given the centroid c representing the query feature locations $q_k \in Q_m$, and their respective matched features $p_k \in P$, we can estimate the translation, rotation and scaling transformation from the space of image P towards the query space, for each match $m_k = (q_k, p_k)$, with $k = \{1 \dots |M|\}$:

$$T_k = T_k^R \cdot (c - L(q_k)) \times \frac{\sigma(p_k)}{\sigma(q_k)}, \tag{3}$$

$$T_k^R = \begin{bmatrix} cos(a_k) & -sin(a_k) \\ sin(a_k) & cos(a_k) \end{bmatrix}, \quad a_k = \theta(p_k) - \theta(q_k) \tag{4}$$

where $\theta(.)$ and $\sigma(.)$ respectively provide the angle and the scale associated with the location $L(.)$ of either q_k or p_k

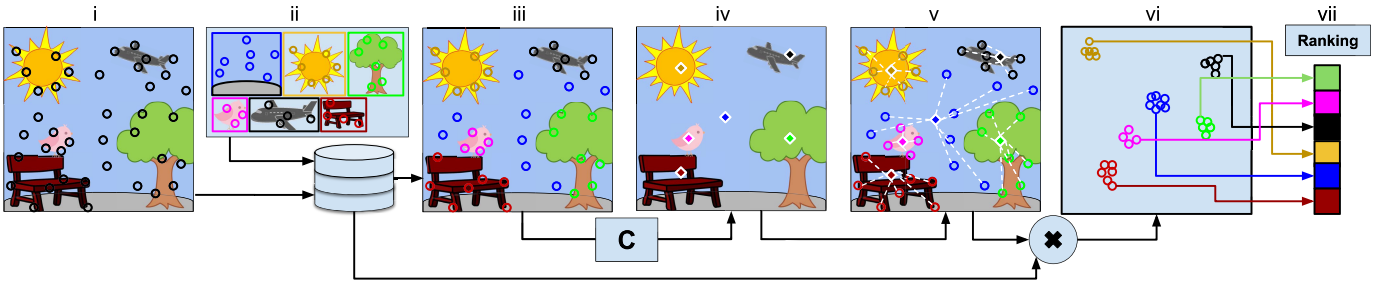

Fig. 2. Steps of the OS2OS method. (i) Local features with associated geometric data (*i.e.*, (x, y) coordinates, scale, and rotation) are extracted from the query. (ii) Local features, along with their associated geometric data, are collected in a database. In this hypothetical example, we associate features for each object with separate images; however, this is not a necessary stipulation for OS2OS. (iii) Query features are assigned to corresponding database matches (represented by feature colors). (iv) For each database image sharing matches with the query, a feature centroid is computed, considering only the matched features. (v) Keypoint geometric transformations are calculated relative to the estimated centroids. (vi) Geometric transformations are applied to the database features clustered in the query (x, y) space. (vii) As each cluster represents a potentially shared object, image ranking scores are calculated on an object-by-object basis.

features. These transformations describe where each query feature expects the object region to be, similar to an "R-table" of the general Hough transform [3], with the added novelty that we only consider the (x, y) feature coordinates. The advantage of doing so is twofold: (i) the Hough space is two-dimensional (given x and y) instead of four-dimensional, making computations faster; (ii) the Hough space maps directly to the query space, making the localization of shared objects straightforward. By applying each transformation to the (x, y) location of its respective matched feature $L(p_k)$, we subsequently build a voting space V_k for each $p_k \in P$:

$$V_k = L(p_k) + T_k. \tag{5}$$

The process of computing the voting space is shown in Fig. 3. Observe, through item (iii), that all p_k features contribute to the centroid, except for the spurious p_5 one.

Step 4 (Density-Based Feature Clustering): We calculate a distinct centroid c for every image P from the database that presents matches with the query. This allows us to transform all matched feature vectors, regardless of the image they come from. After all matched feature locations have been transformed into their respective two-dimensional Hough vote space, as depicted in Fig. 2 (vi), a density-based clustering determines which sets of feature matches are structurally consistent with the query. Instead of using a computationally intensive algorithm such as DBSCAN [10] for clustering millions of points, we apply a linear-time grid-based quantization to each V_k, to find the approximate clusters of highest vote density. Because we assume, for simplicity's sake, cluster morphologies to be roughly circular, we employ a square sliding window to quantize and bin V_k values. The size z of the quantization window varies according to the resolution $w \times h$ of image P:

$$z = \left(\frac{max(w, h)}{b}\right)^{1-\epsilon}, \tag{6}$$

where b is a scaling factor, and $\epsilon \asymp 0$ prevents z from becoming too large. The magnitudes of projection vectors $||T_k||$ are proportional to the resolution of image P. As $||T_k||$ increases, small errors in the scale normalization and rotation transforms (Eq. 4) are amplified, resulting in lower density clustering. Unlike HPM [2], which accounts for this phenomenon by utilizing a hierarchy of window sizes for assigning votes to clusters, we employ a single-window size determined by a sub-linear mapping of the image's resolution (Eq. 6). This is accomplished through ϵ: it constrains the maximum allowable vote cluster area to grow sub-linearly with the image's resolution. The values of b and ϵ are empirically determined via an ablation study, included in Sec V.

Step 5 (Bin-Level Match Filtering): By relying on the value of z computed for an image P, the respective votes V_k are quantized into cluster bins:

$$bin(V_k) = \left\lceil \frac{V_k}{z} \right\rceil \tag{7}$$

We assume that all V_k values sharing the same $bin(.)$ value belong to the same matched object. Let O be the set of transformed features from database image P, which participate in matches between P and the query, and whose respective votes happen to belong to the same bin according to Eq. 6. The meaning of this is that the O features might belong to a unique object shared by the images Q and P. To mitigate matching burstiness, we want to ensure each feature within O, and Q are represented by no more than a single match. To that end, we must remove all sub-optimal 1-to-many and many-to-1 matches between features. For instance, if a feature in O matches to multiple features in Q (1-to-many), we want to remove all but the highest-scoring match, while doing the same for cases of many-to-1 matching. This results in *object-wise match filtering*, in which we perform match filtering locally within each object cluster O, instead of globally across the entire feature set P. To express the affinity of P to Q, we propose to rely on each object $O \in P$ through two novel main scoring mechanisms, each with a particular purpose.

Step 6 (OS2OS Filtering and Scoring): To express the affinity of features within the filtered O to Q, we propose two novel main scoring mechanisms inspired by [55], each with a particular purpose.

The first, called the Centrality Score (CS), measures the centrality of the features and is calculated as the sum of location differences between the elements $o_k \in O$ and their

average (*i.e.*, central) element $\bar{o}$, with $k = \{1 \ldots |O|\}$:

$$CS = \frac{\sum_{k=1}^{|O|} pdf(\|L(o_k) - L(\bar{o})\|_2)}{|O|}, \tag{8}$$

where $L(.)$ is the (x, y) location of the given feature, and $pdf(.)$ is a *probability density function* that inverts the available distance data to a pseudo-probability score (see Fig. 3 (v)):

$$pdf(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}. \tag{9}$$

In addition, CS is normalized by the cardinality of O, as a way to balance clusters of small objects containing few features with respect to large objects that contain many.

The second mechanism is the Angle coherence Score (AS), which aims to measure the uniformity of angles within the features of O. Consider the feature-wise difference of angles a_k expressed in Eq. 4, and let A be the set of difference of angles a_k computed for each feature of O. Features from keypoints belonging to a single rigid object are expected to present similar values of a_k, while erroneously aggregated unrelated features are expected to present more diverse results. For that reason, we rely on the inverse of the standard deviation of A, $stdv(A)$, to compute the angle uniformity (shifted to avoid division by zero):

$$AS = 1/(1 + stdv(A)). \tag{10}$$

Finally, the OS2OS score is given by CS and AS:

$$OS2OS = CS \times AS \times \log|O|, \tag{11}$$

which lies between 0 and 1 and rewards only clusters in which each point has been transformed similarly. The logarithmic scalar $\log|O|$ penalizes clusters with very low numbers of votes, pushing their score towards zero. The end-to-end method is outlined in Alg. 1 below.

Algorithm 1 OS2OS Matching

Input: A Query image, $query$
Output: A ranked list $Results$ of Images related to the query
Extract feature set Q from query
Match all query features $q_i \in Q$ to K features in database D
Calculate score matrix S using Eq. (1)
initialize Array $Results$;
for *unique image P with features in D* **do**
  Collect features from Q that match within P;
  Call this collection of features Q_m;
  Calculate a weighted centroid c_p of feature coordinates in $q_k \in Q_m$ using Eq. (2);
  for *matched feature pair* (q_k, p_k) *in M* **do**
    Project p_k into voting space v_k relative to q_k and c_p using Eq. (5);
  end
  initialize $Score_P = 0$;
  initialize z using Eq. (6);
  Quantize V_k using Eq. (7) into bin list B;
  for *object O in B* **do**
    if $|O| > 1$ **then**
      Filter out bursty matches from O according to Sec. III, Step 5;
      Calculate CS using Eq. (8);
      Calculate AS using Eq. (10);
      Calculate $OS2OS$ using Eq. (11);
    end
    Sum $Score_P = Score_P + OS2OS$
  end
  Put $Score_P$ in $Results$;
end
Sort $Results$ from highest to lowest value;

A. Time Complexity of the OS2OS Score

In real-world applications, a large number of local features are often extracted from each image. Consider $|Q|$ the number of features extracted from the query. In addition, a large value of K is often employed in the feature-wise retrieval step, to recover a significant number of related images from the database. To be efficient, the scoring algorithm should strive for the smallest time complexity possible as a function of $|Q|$ and K. In Appendix B, we perform an analysis of the time complexity of the proposed OS2OS scoring algorithm, concluding that it is linear w.r.t. the number $|Q|$ of query features and to the values of K:

$$OS2OS \in \mathbf{O}(|Q|K)$$

We analyze Alg. 1 line by line, and also Equations 1, 2, 5, 6, 7, 8, and 10 independently, to combine them and determine the final complexity of the overall OS2OS score from Eq. 11 as a function of $|Q|$ and K.

B. Tensorized Computation of OS2OS Scoring

Alg. 1 is presented as an iterative method, with nested loops to perform computations across every image P within the feature database and every subsequent object O within image P. However, Alg. 1 is optimized for human interpretability, being amenable to improvements in speed performance through tensor computation. In practice, the OS2OS score calculation can be reformulated to perform object-wise binning, clustering, and scoring using tensor computations. To do this, we initially represent the set of retrieved local feature matches as a $|Q| \times K$ match matrix M, with feature pair $m_{i,j}$ corresponding to the i_{th} query feature q_i and the database feature d that matches q_i at rank $j \in \{1, 2, .., K\}$. By looking up the database image that corresponds to each database feature $d \in M$, we can partition M into subsets of retrieved features that correspond to the same database image P. These partitions are used to generate an "extruded" tensor from matrix M, with each channel containing matched features belonging to a unique image P. Different tensors are constructed for each of the individual Hough components: (x, y) coordinates, rotation, and scale. Eqs. 2 through 11 are then extended to perform within this tensor space instead

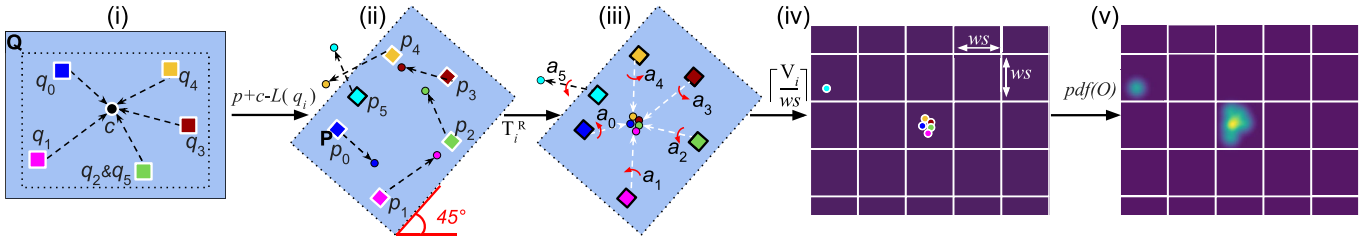

Fig. 3. OS2OS voting space computation. (i) Translation vectors are calculated relative to a computed centroid c in Q. (ii) Translations are applied to matching features in P. (iii) Vectors are rotated by the angle differences a_k to account for object rotation. Notice that the p_5 spurious match votes incorrectly and therefore will not contribute to the score. (iv) Votes are binned using (x, y) coordinates only. (v) A density map is calculated, which conveniently coincides with the query space, providing a visualization of matched object regions.

of iteratively. This approach provides optimal CPU performance and readily lends itself to GPU acceleration through tensor libraries. Because the underlying algorithm is the same, simply with a changed representation, we do not detail this tensorization process. For details, we provide reference code[1] for this method implemented using NumPy, CuPy [35], and PyTorch [36].

IV. OS2OS Score Evaluation

A. Datasets

Oxford 5k and Paris 6k: Oxford 5k [38] and Paris 6k [39] are smaller popular datasets for image retrieval performance evaluation. These two datasets contain hundreds of true positive matches per query, rather than similar-sized datasets that contain only four or five [21], [32].

Google-Landmarks: As a benchmark for DELF features [33], Google released the 2018 Google-Landmarks dataset [15] and subsequent evaluation protocol [14]. This dataset contains $1,098,461$ images with $117,703$ queries in the testing protocol. While ground-truth data for the test set has not yet been released, $1,212,281$ weak labels for the training set are available. The training set contains a total of $14,951$ unique landmarks, with each landmark having an average of 80 instances.

NIST Media Forensics Challenge 2018 (MFC2018): As part of the yearly *Media Forensics Challenge,* run by NIST, the MFC2018 dataset [31] was constructed specifically for the task of finding related manipulated images in a forensics context. This is a large dataset, 3.1TB in size, containing $1,031,080$ images and $3,300$ queries. Many of these queries are composites, with ground-truth results provided in the MFC2018 testing protocol. The dataset also provides ground-truth as to whether a database image contributes a majority of its content (*e.g.*, Fig. 1 (i)), or only a particular object (*e.g.*, Fig. 1 (ii)) to the query.

Reddit: The Reddit dataset [34] introduced by Moreira *et al.* [30] is a meme-style imagery dataset collected from the Reddit Photoshop Battles [42] subreddit. In this dataset, a single user posts an image to be the subject of tampering, while other users utilize a wide variety of photo manipulation techniques to create variations on the original. In many cases, variations are stacked on top of each other, creating evolutionary chains of image manipulation. In the original dataset, Each case provides the original image and all subsequent manipulated versions of the original. This dataset contains $51,245$ images from 185 different Photoshop battles. We generate a query set of one image chosen randomly from each of the 185 cases. The rest of the same-battle thread images are considered as the ground-truth for relevant images to these queries. We will make this new query partition and ground-truth available upon the publication of this paper.

[1]Code available at https://github.com/joelb92/Local_Spatial_Retrieval

TABLE I

Average Computation Time per Feature for a Random Subset of the *Reddit* Dataset. Features for 980 Images Were Calculated, With Their Computation Times Averaged. Each Image was 1MB in Size on Average

Local Feature	Dimensions (#)	Features per image (#)	Features per second
DSURF	64	1000	0.00082
DELF	40	1000	.0621
LIFT	64	500	.178
MobileNet	1280	1	0.01999
Shufflenet	1000	1	.021689

B. Features

For each dataset, we report performance using both hand-crafted SURF [4] and learned DELF [33] features. While other deep image representations have been proposed for image retrieval [16], [18], [41], [56], some deep local feature descriptors such as LIFT [26] are too slow (see Table I) for practical use, as also observed in [33]. Other fast global descriptors such as MobileNet [18] are not applicable to the localization of multiple objects. Our region-wise matching approach requires local descriptors to match coherently within a specific spatial location, which cannot happen with a single global MobileNet descriptor.

SURF keypoints are detected in a distributed modality (DSURF), as proposed in [30]. The keypoint extraction of DSURF features automatically provides the location, scale, and rotation data needed for computing the OS2OS score. For all datasets, we extract a maximum of $5,000$ 64-dimensional DSURF features per image and their corresponding geometrical data. DELF features are used in a similar way to DSURF. Because the DELF algorithm provides only feature scale information, we use the SURF keypoint angle algorithm [4] as an extension to provide feature angles for the DELF geometric data. We performed experiments using the default parameters and model to produce 40-dimensional local features, and cap

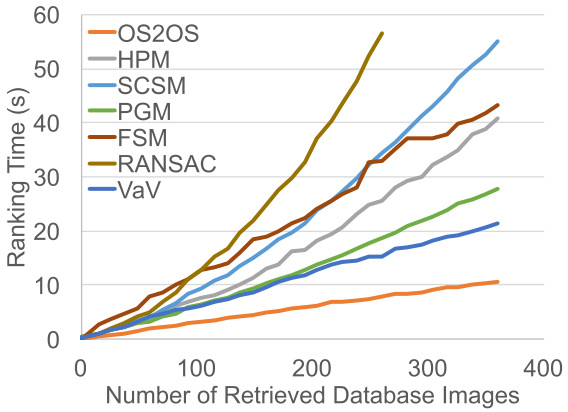

Fig. 4. Spatial verification (ranking) timings for different algorithms. Each algorithm is given an identical subset of K nearest-neighbor features for a query. The lower the ranking time, the better its performance. Results are for the *Google-Landmarks* dataset.

the attention model at a maximum of $1,000$ features per image.

C. Indexing

For experiments on *Oxford 5k* and *Paris 6k* datasets, we keep a consistent indexing backbone among spatial verification methods. To index features, we use Optimized Product Quantization (OPQ) for Approximate Nearest Neighbor (ANN) search, with Inverted File Indices and Asymmetric Distance Computation (IVFADC) [24]. Following the vocabulary hold-out protocol suggested in [52], we utilize a randomized 1% subset of *Google Landmark* images to train the quantization table used for the *Oxford 5k*, *Paris 6k*, *MFC2018*, and *Reddit* datasets. Due to the large volume of images available, a randomized 5% hold-out of the *Google Landmarks* dataset is used to train its tables. These tables are obtained via OPQ matrix computation [13], and used for the IVFADC centroids. OPQ training runs for 25 epochs for 2^{18} centroids using four NVIDIA TITAN Xp GPUs. This process results in a total of four IVFADC tables: two for *Oxford 5k*, *Paris 6k* (trained on both DSURF and DELF features), two for *Reddit* and *MFC2018*, and *Google-Landmarks* datasets (trained for both DSURF and DELF features). In our experiments, *Oxford 5k* and *Paris 6k* utilized the same codebooks for IVF indexing and retrieval, while *Reddit* and *MFC2018*, and *Google-Landmarks* experiments also utilized shared OPQ codebooks. The resulting IVFADC structures are used to index all local image features from their respective datasets. OPQ codebook training took 435.2 seconds when calculating tables for *Oxford 5k* and *Paris 6k*, and took 1,857 seconds when training for Reddit, MFC2018, and Google-Landmarks.

V. Experimental Results

This section presents and analyzes the results for a series of timing experiments, as well as the experiments for each of the five datasets described in Sec. IV.

A. Timing and Complexity

Feature Extraction Timings: Aiming to focus on large-scale retrieval, we performed average timing experiments over a subset of 100 images from the *Reddit* dataset to analyze the tractability of different feature extraction methods on CPUs. Table I shows these results, justifying our selection of DSURF and DELF as candidate features for further experiments. MobileNet stands for the MobileNet-v2 architecture [18], whose features were extracted from its *global_pool* layer. ShuffleNet stands for the ShuffleNet-v2 architecture [28], with features taken from the Fully Connected layer. Although MobileNet and ShuffleNet architectures are considered fast, they produce only a single global feature per image. SURF features are still on the order x20 times faster on average at extracting feature vectors. Additionally, local spatial information is not native to MobileNet or ShuffleNet and is therefore incompatible with the task of spatial verification.

Spatial Verification Timings: We also performed timing experiments against other spatial verification and ranking methods for image retrieval. Here we compared the OS2OS score against HPM [2], PGM [25], FSM [38], SCSM [46], plain RANSAC [12], and VaV [44]. We extracted $5,000$ SURF features from a query image and varied the K retrieved nearest neighbors for each query feature from 0 to 400. For these experiments, we averaged timings across 10 query images from the *Google-Landmarks* dataset. Standard deviations were calculated but were too small to be plotted. While RANSAC is known to provide spatial verification for an arbitrary number of features in quadratic time [25], HPM and PGM are proven to be linear [2], [25]. As can be seen in Fig. 4, OS2OS scoring is faster than HPM, PGM, and VaV, ranking nearly 400 images with 1.8 million features in only 10 seconds. All methods used the same 2.7GHz single-core CPU environment.

TABLE II

Mean Average Precision (mAP) Scores of Different Algorithms for Queries in the *Oxford 5k* and *Paris 6k* Datasets. OS2OS Scoring Provides Competitive or Superior Performance, Without the Need for Bounding Boxes to Pre-Select Regions of Interest

	Oxford 5k		Paris 6k	
	DSURF	DELF	DSURF	DELF
Feature-Only†	66.7	81.5	63.8	83.6
HPM†	72.5	82.2	69.3	83.6
PGM†	75.2	81.9	73.5	83.8
VaV†	77.4	**83.6**	**74.6**	81.2
OS2OS (ours)	**77.9**	83.1	74.1	**86.7**

†Uses groundtruth bounding boxes to pre-select regions of interest.

B. Image Retrieval

Oxford 5k and Paris 6k: The small-scale experiments performed on the *Oxford 5k* and *Paris 6k* datasets are meant to show that the OS2OS scoring algorithm, while designed for object-level spatial verification, is general enough to provide benefits for typical image retrieval. We compare our approach against HPM [2], PGM [25], VaV [44], and plain usage of SURF and DELF features (without spatial verification). Mean Average Precision (mAP) scores are reported in Table II. We find that the OS2OS score significantly improves both DSURF and DELF features, suggesting that the provided spatial constraints work satisfactorily for global geometric verification for instance retrieval. Additionally, we see a much

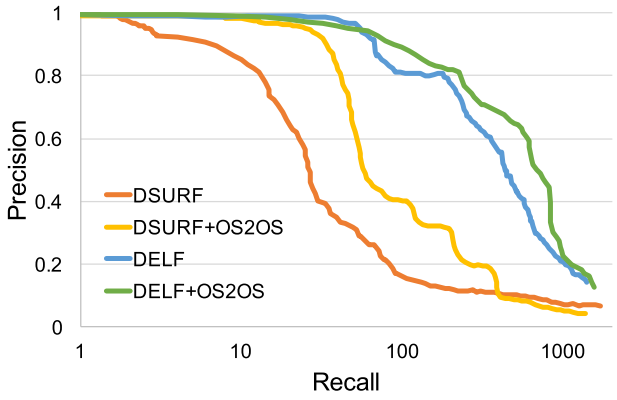

Fig. 5. Precision-recall curves for the *Google-Landmarks* dataset. As can be observed, the usage of spatial verification through the OS2OS score boosts both SURF- and DELF-based image instance retrieval.

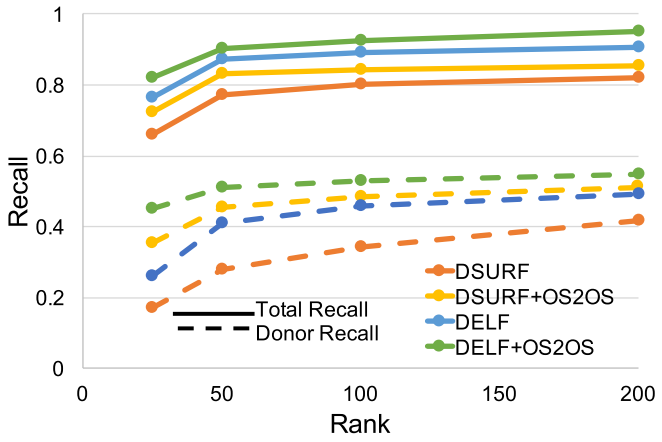

Fig. 6. Recall scores for the *NIST MFC2018* dataset for ranks of 25, 50, 100, and 200 images. Total recall is represented by solid lines, while small-donor-only recall is represented by dashed lines. OS2OS scoring improves retrieval in all scenarios.

larger performance improvement for DSURF, which suggests the OS2OS score helps mitigate erroneous matches from the bursty nature of SURF. Overall, the OS2OS approach is comparable to other approaches in the literature. Of noteworthy significance is the fact that the OS2OS algorithm required no bounding boxes, still performing comparably to other spatial methods. We additionally examined scores from RANSAC [12], SCSM [46], and FSM [38] for *Oxford 5k*, but the results were inferior to the better performing approaches in Table II.

Google-Landmarks: This experiment shows that the OS2OS score generalizes to the instance retrieval task and showcases the algorithm's scalability in the presence of many distractors. Because annotations for the index or test sets have not been released at the time of this writing, we performed our study utilizing the training set, which contains over 1 million images and provides landmark annotations. We selected 1,000 landmarks at random for retrieval, sampling one query per landmark. Finally, we utilized the modified precision and recall measures described in [33] to report results. Fig. 5 shows that DSURF augmented with the OS2OS score improves significantly. The OS2OS score also provides minor improvements to DELF.

NIST MFC2018: Unlike the experiments described thus far, the *MFC2018* dataset comprises of manipulated images. Query images from the dataset's retrieval protocol may or may not contain regions from multiple sources within the image database. We utilized the ground-truth relationship graphs to determine which images donate small objects to their queries. Using this data, we can generate recall curves exclusively for donor image retrieval (namely, donor recall). We report both total recall scores and donor recall scores in Fig. 6. While the OS2OS score shows good boosts in total recall, we see that donor recall is more significantly improved, indicating that the OS2OS score is capable of balancing geometrically coherent matches of image regions from small donors with global matches from backgrounds.

Reddit: The *Reddit* provenance dataset has proven to be a difficult challenge [30]. In Table III, we see a significant increase in retrieval performance (nearly 10% for SURF and nearly 5% for DELF) across the board, with vastly superior results when compared to VaV [44]. The near-baseline performance of VaV suggests that global spatial verification methods are not entirely adequate to solve this problem. Further, Fig. 7 shows qualitative retrieval results using OS2OS scoring, along with the object vote maps from each retrieved match. These results highlight the OS2OS score's ability to localize and appropriately weight small objects from a database image to small objects within the query, without the need for bounding boxes.

TABLE III

RECALL SCORES FOR THE *Reddit* DATASET AT THE TOP-50, 100, AND 200 MOST RELATED RETRIEVED IMAGES. OS2OS SPATIAL VERIFICATION IMPROVES THE RESULTS IN ALL SCENARIOS. THE BOTTOM TWO ROWS DENOTE RESULTS USING OUR OS2OS APPROACH

Method	R@50	R@100	R@200
DSURF	0.317	0.432	0.478
DELF	0.402	0.516	0.551
DSURF + HPM	0.351	0.358	0.437
DSURF + PGM	0.327	0.398	0.442
DSURF + VaV	0.310	0.423	0.479
DSURF + OS2OS	0.424	0.509	0.546
DELF + OS2OS	**0.479**	**0.548**	**0.593**

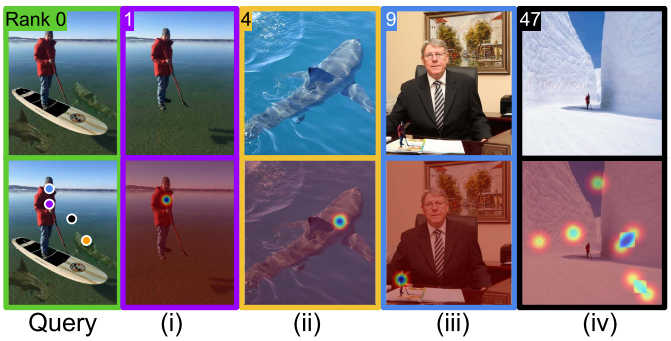

Fig. 7. Set of retrieved images for a query (left) in the *Reddit* dataset. The top row shows retrieved results. Bottom row provides an overlay of the feature vote space $pdf(V_k)$ from Eq. 5. Object centroids for each retrieved image are color-coded and overlayed on the query. Rank 1 (i) is the unmodified version of the query. Rank 4 (ii) is an object donor to the query, while rank 9 (iii) utilizes a scaled version of the man in the query. Rank 47 (iv) is a failure case. See Appendix A for additional examples.

TABLE IV

RESULTS OF THE ABLATION STUDY, ITERATIVELY ADDING IN ALGORITHM COMPONENTS, WITH VOTING WINDOW SIZES (z) OF 5, 10, AND 20 FOR EACH EXPERIMENT. COLUMNS 1-3 (PURE HOUGH) PERFORM SIMPLE SUMMED HOUGH VOTING (WITHOUT FURTHER SCORING) WITHIN THE GIVEN HOUGH BINS

	Pure Hough			+ Centroid (Eq. 2)			+ CS (Eq. 8)			+ AS (Eq. 10)			+ $log\|O\|$ (Eq. 11)		
z (Eq. 6)	5	10	20	5	10	20	5	10	20	5	10	20	5	10	20
R@k=200	76.2	76.3	76.2	77.5	78.1	77.5	80.2	80.9	79.3	81.2	82.1	82.0	83.1	83.2	**83.4**

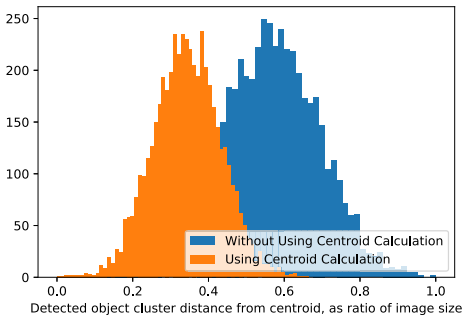

Fig. 8. Comparison of distribution of L2 distances of Hough vote vectors on 5000 images from the *Google-Landmarks* dataset. When not using centroid calculation (blue), we simply set the vote point to be the center of the query image's dimensions. Results show that vote distances are significantly shorter when using weighted centroid voting, which helps to reduce the magnification of noisy voting.

C. Assessment of Individual Algorithm Components

Ablation Study: To verify the efficacy of different OS2OS algorithm components detailed in Sec. III, we provide an ablation study on the different parts of the proposed method using the *MFC2018* dataset. We first verify that centroid utilization does contribute to higher retrieval performance. We then re-introduce the Centrality Score (Eq. 8), Angle Score (Eq. 10), and finally the logarithmic scaling (shown in Eq. 11). These experiments are performed on the *MFC2018* dataset to assess the algorithm's ability to accurately retrieve donor images when contributing only relatively small amounts of content to a composite query. The results of this study in Table IV show that each component provides at least a marginal improvement in retrieval performance, measured as Mean Average Precision (mAP). The results also show the relative stability of results despite changing Hough bin window sizes (z in Eq. 6). This suggests that as long as window sizes divide the voting space by at least 5, we can effectively recover local objects and score them appropriately.

Study of Weighted Centroid Use: This experiment aims to show the benefit of using the weighted centroid calculated from matched keypoint locations as the voting center for given images, as described in Eq. 2. We perform two matching experiments using a random sample of 5000 images from the *Google-Landmarks* dataset, and extract the average vector length of each vote projection (Eq. 3). The results of this analysis in Fig. 8 show that utilizing this weighted average markedly reduces the voting distance in most cases. This, in turn, reduces the distance of each vote vector, subsequently reducing quantization-related voting error.

VI. CONCLUSION

Retrieving images that share small regions in a complex composite scene is a challenge for most image retrieval approaches. In this paper, we proposed an inexpensive scoring technique based on better utilization of pre-trained feature extractors and indexing techniques to yield geometrically consistent localized scores. An advantage of the technique is that it is learning-free, and works as an add-on to existing feature description methods. It provides a way to include spatial verification while performing matching in a large feature space. It is also optimized to be efficiently executed on CPUs without the need for high-end GPUs, but can optionally utilize GPU acceleration when desired.

In our experimental results, we saw that the proposed approach improved recall for the retrieval of images for difficult emerging problems such as Internet meme analysis and image forensics. As with most computer vision algorithms, the proposed approach performs better on datasets with clean matched correspondences between images than with data obtained from the web with different styles of content correspondence.

Although the performance results improve with the proposed scoring, retrieval for specific applications, such as tracking memes on social media, are still in need of improvement. One possible extension of this work could be to utilize the OS2OS score as a loss function for end-to-end feature learning or fine-tuning, which could subsequently help further improve object-level scoring results. Whatever paths future research may take in tackling Objects-in-Scenes-to-Objects-in-Scenes retrieval scenarios, the problem worthy of additional research consideration — especially as such content grows in popularity.

APPENDIX A
ADDITIONAL QUALITATIVE RESULTS

Here we provide more visual results of the OS2OS score discovering objects in datasets. We provide visual representations of query cases from the three most challenging datasets used within the paper, showing success and failure cases for the *MFC2018*, *Google-Landmarks*, and *Reddit* datasets. In each figure, the first row corresponds to the highest voted center of points of interest in the query by the points detected in the corresponding rank retrieved image. The second row shows the heatmap of the probability distribution of votes of matched features to the query for the particular image. The third row contains the original rank retrieved image. The columns correspond to the different ranks. Because we use weighted centroid voting, we can visualize where the algorithm matches objects between the query and retrieved images by overlaying the generated Probability Density Function (PDF) of Hough votes on top of each retrieved image.

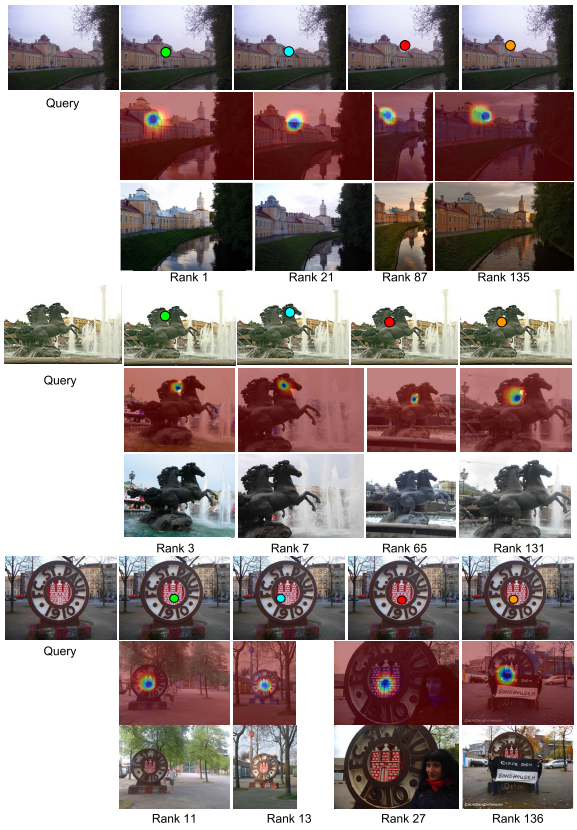

Fig. 9. Results for three queries from the *Google-Landmarks* dataset.

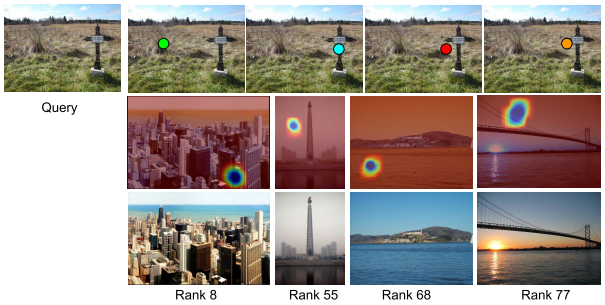

Fig. 10. A failure scenario from the *Google-Landmarks* dataset where the matches do not appear to be coherent and high affinity matches are false positives.

APPENDIX B
COMPLEXITY ANALYSIS OF OS2OS SCORING

The OS2OS algorithm is posed to perform spatial verification between large quantities of image feature pairs using the minimal computational overhead. To understand the time complexity of the OS2OS algorithm as a whole, we subsequently analyze the constituent sub-steps within the OS2OS algorithm, following the numbered lines depicted in Alg. 1.

The complexity of line 3 is not considered in our analysis since it is related to local feature extraction, which is common across many typical CBIR solutions and to which OS2OS is agnostic. Line 4, in turn, is related to the indexing of the extracted features and the retrieval of the K closest features (KNN) for each query's feature. OS2OS is also agnostic to this step (hence, not considered in the time complexity computation), to which we recommend using state-of-the-art

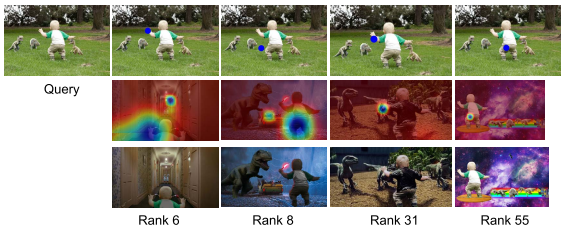

Fig. 11. Example of a query result from the *Reddit* dataset. Incorrect match examples are highlighted in red text. Note the Rank 6 and Rank 8 matches, in which multiple objects from the query are independently donated to the matched result. The heat maps reflect the multiple object matching.

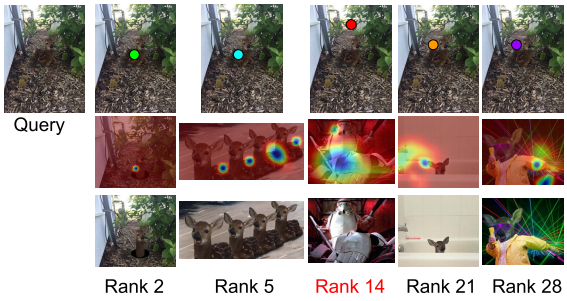

Fig. 12. Example of a query result from the *Reddit* dataset. Incorrect match examples are highlighted in red text. Of particular interest is the Rank 5 match, in which the query donates the same object multiple times to the result. The discovery of these four donations is reflected in the heat map.

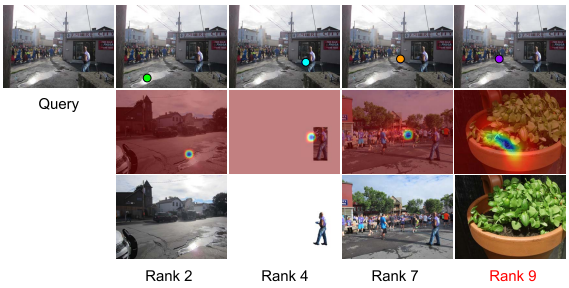

Fig. 13. Example of a query result from the *MFC2018* dataset. Incorrect match examples are highlighted in red text. In this case, the search algorithm finds both the man and the puddle as donated objects within the query.

(OPQ feature compression and IVFADC indexing [24], see Sec. V).

Consequently, the following steps build upon the $|Q|$ local features extracted from the query and the K database features retrieved for each query's feature. Line 5 follows the time complexity of Eq. 1, which is linear to the dimensionality of the local features. Again, since OS2OS is agnostic to these features, we do not consider this dependency on their size to compute the time complexity. However, this line must be executed K times for each one of the $|Q|$ query's features, leading to:

$$Line(5) \in \mathbf{O}(|Q|K).$$

Let us now focus on the lines within the "for" loop related to lines 7-27, which are executed for each database image "touched" by the IVFADC index retrieval. Lines 8 is a selection of elements from Q, hence being linear to $|Q|$. Line 10 follows the time complexity of Eq. 2. In this case, a lookup table is made available to obtain $L(.)$ at constant time. Hence,

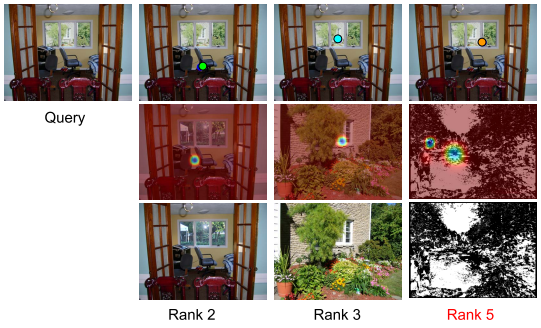

Fig. 14. Example of a query result from the *MFC2018* dataset. Incorrect match examples are highlighted in red text. The Rank 3 match actually appears within the window frame of the query image.

the time complexity depends only on $|M|$ (the number of feature matches between the query and the current database image) and the size of the features, again not considered in our estimation. In the worst time-consuming scenario, $|M|$ is equal to the maximum possible value, $|Q|K$ matches (no more than K matches for each $|Q|$ query features). Therefore:

$$Line(10)_{no\_loop} \in \mathbf{O}(|Q|K),$$

ignoring the lines-7-27 outside loop.

Line 12, in turn, has the complexity of Eq. 5, which alone has all of its members obtained through simple computations (such as sine and cosine) over lookup-table values related to the scale, angle, and position of the local features (hence presenting constant time). Nevertheless, this line is executed $|M|$ times (see the loop on line 11), or in the worst case ($|M|$=$|Q|K$):

$$Line(12)_{no\_loop} \in \mathbf{O}(|Q|K),$$

ignoring the lines-7-27 outside loop. Lines 15 and 16 depend on the complexity of Eq. 6 and Eq. 5, respectively. Both have constant time complexity, by definition.

Next lines refer to the "for" loop defined by lines $17-25$, which are executed $|B|$ times (once for each bin of transformation-agreeing matches, which should refer to a shared "object"). Line 19 is linear to $|O|$, whose value is at most equals to $|Q|K$ (the K retrieved database features for each one of the $|Q|$ features of the query), and executed $|B|$ times. Lines 20, 21, and 22 follow the time complexity of Eq. 8, Eq. 10, and Eq. 11, respectively. All the three depend linearly on the value of $|O|$ (not greater than K, after the filtering of the line 19) and are also executed $|B|$ times. Line 24 has constant time complexity and is repeated $|B|$ times.

Thankfully, the value of $|B|$, which defines the number of iterations within the loop of lines $17-25$, is correlated to the value of $|O|$ inside it. The smaller the value of $|B|$, the larger the possible values of $|O|$, and vice-versa. For instance, for a large $|B|$ (various objects shared among the query and the database images), the K matches for each query's feature will be spread across the different objects, reducing the values of $|O|$ in each iteration, on an average. In the case $|B|$ is small (*e.g.*, equals to two), O will have the chance of being larger and closer to its $|Q|K$ upper bound. In the end, lines $17-25$ will be executed $|Q|K$ times, hence presenting:

$$Line(17-25)_{no\_loop} \in \mathbf{O}(|Q|K).$$

ignoring the lines-7-27 outside loop.

Getting back to the number of repetitions within the more external loop within lines $7-27$, the value of $|D|$ will significantly vary, depending on the queried database. In the scenario where $|D|$ is as large as possible (*i.e.*, equals to $|Q|K$), each one of the $|D| = |Q|K$ database images will share only one local feature with the query (*i.e.*, $|M| = 1$), hence leading to quick executions inside the loop. On the contrary, in the scenario where $|D|$ is close to one database image, $|M|$ will have more possibilities of being closer to its largest value, which is $|Q|K$. The outer loop, however, will be executed nearly once. In the end, the complexity inside the loop is:

$$Line(7-27) \in \mathbf{O}(|Q|K).$$

Combining the time complexity of all the lines from Alg. 1, the dominating (and resulting) complexity is:

$$OS2OS \in \mathbf{O}(|Q|K).$$

ACKNOWLEDGMENT

The authors would like to thank NVIDIA Corporation for the Hardware support provided.

Copyright

REFERENCES

[1] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1578–1585.

[2] Y. Avrithis and G. Tolias, "Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval," *Int. J. Comput. Vis.*, vol. 107, no. 1, pp. 1–19, Mar. 2014.

[3] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[5] N. K. Baym, *Personal Connections in the Digital Age*. Hoboken, NJ, USA: Wiley, 2015.

[6] A. Bharati *et al.*, "Beyond pixels: Image provenance analysis leveraging metadata," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1692–1702.

[7] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 726–743.

[8] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300 fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3286–3293.

[9] R. O. Duda and R. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "Density-based spatial clustering of applications with noise," in *Proc. AAAI Int. Conf. Knowl. Discovery Data Mining (KDD)*, 1996, pp. 1–6.

[11] R. Evans. (2018). *From Memes to Infowars: How 75 Fascist Activists Were, 'Red-Pilled'*. Accessed: Mar. 1, 2019. [Online]. Available: https://bit.ly/2CFglpF

[12] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization for approximate nearest neighbor search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2946–2953.

[14] Google LLC. (2019). *Google Landmark Retrieval Challenge: Given an Image, Can You Find all of the Same Landmarks in a dataset?*. Accessed: Jul. 25, 2020. [Online]. Available: https://bit.ly/2CfTa3a

[15] Google LLC. (2019). *Google-Landmarks Dataset: Label Famous (and Not-so-Famous) Landmarks in Images*. Accessed: Jul. 25, 2020. [Online]. Available: https://bit.ly/34yDwvO

[16] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 241–257.

[17] R. Hinami, Y. Matsui, and S. Satoh, "Region-based image retrieval revisited," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 528–536.

[18] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. Accessed: Jul. 25, 2020. [Online]. Available: http://arxiv.org/abs/1704.04861

[19] W. Huang, Y. Gao, and K. L. Chan, "A review of region-based image retrieval," *J. Signal Process. Syst.*, vol. 59, no. 2, pp. 143–161, 2010.

[20] Y. K. Jang and N. I. Cho, "Generalized product quantization network for semi-supervised image retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3420–3429.

[21] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 304–317.

[22] H. Jégou, M. Douze, and C. Schmid, "Exploiting descriptor distances for precise image search," Ph.D. dissertation, French Inst. Res. Comput. Sci. Automat., Rocquencourt, France, 2011.

[23] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.

[24] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.

[25] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5153–5161.

[26] Z. Liu, S. Wang, and Q. Tian, "Fine-residual VLAD for image retrieval," *Neurocomputing*, vol. 173, pp. 1183–1191, Jan. 2016.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[28] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

[29] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, and X. Giro-i-Nieto, "Bags of local convolutional features for scalable instance search," in *Proc. ACM Int. Conf. Multimedia Retr.*, Jun. 2016, pp. 327–331.

[30] D. Moreira *et al.*, "Image provenance analysis at scale," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 6109–6123, Dec. 2018.

[31] National Institute of Standards and Technology. (2019). *Nimble Challenge 2018 Evaluation*. Accessed: Jul. 25, 2020. [Online]. Available: https://bit.ly/2BEDpSP

[32] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 2161–2168.

[33] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 3456–3465.

[34] Notre Dame Computer Vision Research Laboratory. (2018). *Reddit Photoshop Battles Dataset*. Accessed: Jul. 25, 2020. [Online]. Available: https://bit.ly/2qk2vUJ

[35] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, "CuPy: A NumPy-compatible library for NVIDIA GPU calculations," in *Proc. Workshop Mach. Learn. Syst. (LearningSys)*, 2017, pp. 1–7.

[36] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 1–12.

[37] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[38] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[39] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[40] J. Philbin and A. Zisserman, "Object mining using a matching graph on very large image collections," in *Proc. 6th Indian Conf. Comput. Vis., Graph. Image Process.*, Dec. 2008, pp. 738–745.

[41] F. Radenović, G. Tolias, and O. Chum, "CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 3–20.

[42] Reddit.com. (2020). *Photoshopbattles*. Accessed: Jul. 25, 2020. [Online]. Available: https://bit.ly/2NkcQcv

[43] K. R. Mopuri and R. V. Babu, "Object level deep feature pooling for compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 62–70.

[44] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, "A vote-and-verify strategy for fast spatial verification in image retrieval," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2016, pp. 321–337.

[45] X. Shen, A. A. Efros, and M. Aubry, "Discovering visual patterns in art collections with spatially-consistent feature learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9278–9287.

[46] X. Shen, Z. Lin, J. Brandt, and Y. Wu, "Spatially-constrained similarity measure for large-scale object retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1229–1241, Jun. 2014.

[47] L. Shifman, *Memes in Digital Culture*. Cambridge, MA, USA: MIT Press, 2014.

[48] Sivic and Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, p. 1470.

[49] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[50] S. Sun, W. Zhou, H. Li, and Q. Tian, "Search by detection: Object-level feature for image retrieval," in *Proc. Int. Conf. Internet Multimedia Comput. Service (ICIMCS)*, 2014, pp. 46–49.

[51] M. Teichmann, A. Araujo, M. Zhu, and J. Sim, "Detect-to-retrieve: Efficient regional aggregation for image search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5109–5118.

[52] G. Tolias, Y. Avrithis, and H. Jegou, "To aggregate or not to aggregate: Selective match kernels for image search," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1401–1408.

[53] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–12.

[54] F. Vaccaro, M. Bertini, T. Uricchio, and A. Del Bimbo, "Image retrieval using multi-scale CNN features pooling," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2020, pp. 311–315.

[55] X. Wu and K. Kashino, "Robust spatial matching as ensemble of weak geometric relations," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 1–25.

[56] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 467–483.

[57] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," 2017, *arXiv:1706.06064*. Accessed: Jul. 25, 2020. [Online]. Available: http://arxiv.org/abs/1706.06064

Joel Brogan (Member, IEEE) received the B.S. degree in electrical engineering from the Hope College in 2014, and the M.S. degree in computer science and engineering and the Ph.D. degree in computer science from the Computer Vision Research Laboratory, University of Notre Dame, in 2018 and 2019, respectively. He specializes in machine learning for biometrics and digital media forensics, with the Ph.D. work culminating in a set of algorithms designed to detect and trace the origins of manipulated images circulating online. He currently holds a computer science research professional position at the Multimodal Sensor Analytics (MSA) Group, Oak Ridge National Laboratory.

Aparna Bharati (Member, IEEE) received the B.Tech. degree in computer science and engineering from IIIT Delhi, India, in 2015, with a focus on image analysis and machine intelligence. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of Notre Dame, USA. She is also an Assistant Professor with the Department of Computer Science and Engineering, Lehigh University. Her research interests include media forensics, biometrics, pattern recognition, machine learning, and computer vision.

Daniel Moreira (Member, IEEE) received the B.S. degree from the Federal University of Pará, Brazil, in 2006, the M.S. degree from the Federal University of Pernambuco, Brazil, in 2008, and the Ph.D. degree from the University of Campinas, Brazil, in 2016, all in computer science. After working for five years as a Systems Analyst with Brazilian Federal Data Processing Service, he is currently a Postdoctoral Fellow with the Department of Computer Science and Engineering, University of Notre Dame, USA. His research interests include media forensics, computer vision, and machine learning.

Anderson Rocha (Senior Member, IEEE) has been an Associate Professor with the Institute of Computing, University of Campinas (Unicamp), Brazil, since 2009. His research interests include machine learning and artificial intelligence, reasoning for complex data, and digital forensics. He chaired the IEEE Information Forensics and Security Technical Committee (IFS-TC) for the 2019–2020 term.

Kevin W. Bowyer (Life Fellow, IEEE) received the Ph.D. degree in computer science from Duke University, Durham, NC, USA. He is currently the Schubmehl-Prein Family Professor of computer science and engineering with the University of Notre Dame, Notre Dame, IN, USA, and also serves as the Director for the International Summer Engineering Programs, Notre Dame College of Engineering, Notre Dame. In 2019, he was elected as a fellow of the American Association for the Advancement of Science. He served as the Editor-in-Chief for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He is also serving as the Editor-in-Chief for the IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE.

Patrick J. Flynn (Fellow, IEEE) received the Ph.D. degree in computer science from Michigan State University in 1990. He is currently the Fritz Duda Family Professor and the Chair of the Department of Computer Science and Engineering, University of Notre Dame. He has held faculty positions at Notre Dame, Washington State University, and The Ohio State University. His research interests include computer vision, biometrics, and image processing. He was the past Editor-in-Chief of the IEEE BIOMETRICS COMPENDIUM, the past Associate Editor-in-Chief of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and a past Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.

Walter J. Scheirer (Senior Member, IEEE) received the M.S. degree in computer science from Lehigh University, USA, in 2006, and the Ph.D. degree in engineering from the University of Colorado Boulder, Boulder, CO, USA, in 2009. He is currently an Associate Professor with the Department of Computer Science and Engineering, University of Notre Dame, USA. Prior to that, he was a Postdoctoral Fellow with Harvard University, USA, and a Research Assistant Professor with the University of Colorado. His research interests include computer vision, machine learning, biometrics, and digital humanities.