

# Adaptive Dither Voting for Robust Spatial Verification

Xiaomeng Wu and Kunio Kashino  
Nippon Telegraph and Telephone Corporation  
{wu.xiaomeng, kashino.kunio}@lab.ntt.co.jp

## Abstract

Hough voting in a geometric transformation space allows us to realize spatial verification, but remains sensitive to feature detection errors because of the inflexible quantization of single feature correspondences. To handle this problem, we propose a new method, called adaptive dither voting, for robust spatial verification. For each correspondence, instead of hard-mapping it to a single transformation, the method augments its description by using multiple dithered transformations that are deterministically generated by the other correspondences. The method reduces the probability of losing correspondences during transformation quantization, and provides high robustness as regards mismatches by imposing three geometric constraints on the dithering process. We also propose exploiting the non-uniformity of a Hough histogram as the spatial similarity to handle multiple matching surfaces. Extensive experiments conducted on four datasets show the superiority of our method. The method outperforms its state-of-the-art counterparts in both accuracy and scalability, especially when it comes to the retrieval of small, rotated objects.

## 1. Introduction

Local feature-based image encoding has been shown to be successful in particular object retrieval. However, local features do not offer sufficient discriminative power and so their direct matching leads to massive mismatches. Of the methods used to handle this problem, Hough voting (HV) has received considerable attention because of its better balance between accuracy and scalability [3, 8]. Here, consistent feature correspondences are found in a geometric transformation space via a Hough transform. Despite its success, HV remains sensitive to feature detection errors generating noise during transformation estimation. Since a correspondence is hard-mapped to a single transformation, confident correspondences (Fig. 1a) are never identified if they are affected by noise and fall into disjunct bins (Fig. 1b).

To address noise sensitivity, we first consider an unadaptable solution, called dither voting (DV), where an observed

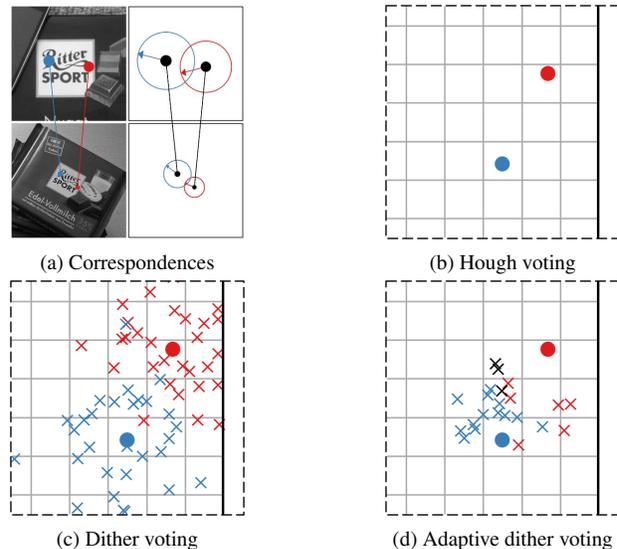


Figure 1: Comparison of ADV with HV and DV. The correspondences in (a) are voted as filled circles in a 4D transformation space. Only one 2D projection is depicted for normalized translation  $(x, y)$ . We show a close-up of the  $5 \times 5$  bins where the correspondences fall. Crosses represent dithered votes that are randomly sampled according to a Gaussian distribution (c) or deterministically obtained for each correspondence concerned (d). Common dithered votes are represented in black.

transformation is polled to a Hough space as a probability density distribution rather than a single vote (Fig. 1c). The distribution can be Gaussian if the noise is assumed to be normally distributed with a zero mean. Provided that the Gaussian can be sampled by a number of random transformations, called dithered votes, HV is converted into polling the dithered votes to multiple bins in the transformation space. However, straightforward DV is highly sensitive to mismatches because a Gaussian distribution is assumed to have the same dispersion for all tentative correspondences.

In this study, we propose a novel adaptive dither voting (ADV) method for robust spatial verification. For the distribution of true transformations, instead of assuming it to be Gaussian, we sample it by using the other correspondences that satisfy certain geometric constraints responding to the observed correspondence. Dithered votes can thus be

deterministically obtained and are expected to be located in closer proximity to the true transformation (Fig. 1d). The aforementioned constraints provide dithering with a greater advantage as regards geometrically-correlated correspondences, and suppress the augmentation of their isolated counterparts that tend to be mismatches. In addition, we propose exploiting the non-uniformity of a Hough histogram as the spatial similarity, which favors correspondences converging in the transformation space. The similarity is measured simultaneously, rather than consecutively, with the voting process, making ADV much faster than the state of the art. In summary, our contributions include:

- A novel adaptive dither voting method that is the first *deterministic* method handling both quantization error and mismatch in a *simultaneous* manner. It significantly outperforms the bag-of-visual-words (BOVW) model and current methods with spatial verification.
- A novel entropy-based similarity measure that provides great flexibility for handling multiple matching surfaces. It is realized simultaneously with voting and so provides much higher efficiency than standard solutions.
- Informative and thorough experiments performed on four datasets. All comparisons that would be expected are available, and those with related researches show consistent performance benefits using the proposed method.

## 2. Related Research

In this section, we review the literature on spatial verification. Other topics, e.g. soft assignment [15], Hamming embedding [7, 8], database augmentation [1] and query expansion [5, 6], concern the field of image retrieval but are not related to our topic. Hence, they are excluded from the discussion. Spatial verification can be categorized as prior spatial context methods or posterior methods: the former explores the spatial configuration of features before matching; the latter rejects mismatches online.

Spatial context methods exploit the co-occurrence and spatial relationship between features inside a given image and embed them in indexing to avoid online verifications. Yang and Newsman [24] showed that the second-order co-occurrence of spatially nearby features offers a better representational power than single features, and proposed abstracting each image as a bag of pairs of visual words. To incorporate richer spatial information, Liu et al. [9] explored both the co-occurrence and the relative positions of nearby features, and embedded this information in an inverted index for fast spatial verification. Wu and Kashino [23] extended this method to handle anisotropic transformations. Toliás et al.’s method [21] serves as an alternative to Liu et al.’s method [9], in which each feature is described by a spatial histogram of the relative positions of all other features. Depending on the size of the visual vocabulary in use, all

of these methods require a huge memory needed for storing redundant indexes online.

Among posterior methods, the most widely used is RANSAC [13, 14], which repeatedly computes an affine transformation, called a hypothesis, from each correspondence. All hypotheses are verified by counting the inlier correspondences that inversely fit the transformation. Jegou et al. [8] used a weak geometric model realized with a 2D HV whereby correspondences are determined as confident correspondences if they agree on a scaling and, independently, a rotation factor. Zhang et al. [25] set up a 2D Hough space spanned by the translations of correspondences, but it does not support scaling or rotation invariance. Shen et al. [18] proposed uniformly sampling a fixed number of hypotheses from a transformation space. All hypotheses are verified in another 2D Hough space spanned by the normalized central coordinates of the common object. Chu et al. [4] and Zhou et al. [26] replaced voting with geometric verification among all correspondence pairs (quadratic time) but ignored the consistency of scaling and/or rotation.

One of the most current methods based on HV is Hough pyramid matching (HPM) [3]. In HPM, an elegant, relaxed histogram pyramid is developed, and correspondences are distributed over a hierarchical partition of the transformation space to handle the noise sensitivity. Although a reasonable balance between flexibility and accuracy can be expected at the finest level of the hierarchy, it is not guaranteed at coarse levels where the constraints are much less discriminating in terms of mismatches. HPM is one of the methods we use for comparison in our experiments.

It has been pointed out that in theory, posterior methods suffer from a longer search time than prior methods because of the added online verifications [9, 23]. However, no experimental evidence has been shown to bear out this conclusion. We share our knowledge by comparing current prior and posterior methods and by designing a computationally-cheap similarity measure for fast spatial verification.

## 3. Robust Spatial Verification

### 3.1. Problem Formulation

An image is represented by a set  $P$  of local features, and for each  $p \in P$  we have its visual word  $u(p)$ , position  $\mathbf{t}(p)$ , scale  $\sigma(p)$  and orientation  $R(p)$ . The geometries of  $p$  can be given by Hessian affine feature detectors [11, 13] and  $u(p)$  by vector quantization [14] in a SIFT [1, 10] feature space.  $p$  can be given by a  $3 \times 3$  transformation matrix  $F(p)$  mapping a unit circle heading a reference orientation to  $p$ :

$$F(p) = \begin{bmatrix} M(p) & \mathbf{t}(p) \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (1)$$

Here,  $M(p) = \sigma(p)R(p)$  and  $\mathbf{t}(p)$  represent linear transformation and translation, respectively. If  $\sigma(p)$  is given by a

scalar,  $F(p)$  specifies a similarity transformation.  $R(p)$  is an orthogonal  $2 \times 2$  matrix represented by an angle  $\theta(p)$ .

Given two images  $P$  and  $Q$ , the correspondence  $c = (p, q)$  is a pair of features  $p \in P$  and  $q \in Q$  such that  $u(p) = u(q)$ . A transformation from  $q$  to  $p$  is given by:

$$F(c) = F(p)F(q)^{-1} = \begin{bmatrix} M(c) & \mathbf{t}(c) \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2)$$

where  $M(c) = \sigma(c)R(c)$  and  $\mathbf{t}(c) = \mathbf{t}(p) - M(c)\mathbf{t}(q)$ . Equation 2 can be extended to handle out-of-plane transformation with an anisotropic  $M(c)$  estimated from Hessian.  $\sigma(c) = \sigma(p)/\sigma(q)$  and  $R(c) = R(p)R(q)^{-1}$  denote scaling and rotation, respectively. Equation 2 can also be rewritten as a 4D transformation vector, as in Eq. 3, in which  $\theta(c) = \theta(p) - \theta(q)$  and  $[x(c) \ y(c)]^T = \mathbf{t}(c)$ .

$$F(c) = \langle \theta(c), \sigma(c), x(c), y(c) \rangle \quad (3)$$

Given  $P$  and  $Q$  that are related as regards a common object, all parts of the object are expected to obey the same transformation. Given a correspondence set  $C = \{c\} \subseteq P \times Q$ , there is one or more subset  $\hat{C} \subseteq C$  of correspondences that dominate in terms of  $F(c)$ . Spatial verification involves identifying such a subset and giving more advantage to the similarity measure for  $\hat{C}$  with a larger cardinality.

### 3.2. Hough Voting

In HV, a transformation space  $\mathcal{F} = [0, 1]^4$  is spanned by the four parameters presented in Eq. 3. A partition  $\mathcal{B}$  of  $\mathcal{F}$  into  $n^4$  bins is constructed, where  $n$  is the number of bins per parameter. All  $c \in C$  are distributed into  $\mathcal{B}$  according to  $F(c)$ .  $\hat{C}$  can be determined by bins  $b \in \mathcal{B}$  into which more than one vote falls. More strictly,

**Definition 1** Given a correspondence set  $C = \{c\}$  and an arbitrary quantization function  $\beta$ , a subset  $\hat{C} \subseteq C$  is a confident correspondence set if and only if  $|\hat{C}| \geq 2$  and  $\forall c_i, c_j \in \hat{C}, \beta(F(c_i)) = \beta(F(c_j))$ .

HV guarantees sufficient recall if the feature shapes are accurately given and if  $F(c)$  can be flexibly quantized. However, these requirements are often violated in practice.

### 3.3. Dither Voting

Each correspondence  $c$  can be voted into multiple bins as a Gaussian  $\mathcal{N}(F(c), \Sigma)$ , similar to related works [10, 18], which is sampled by a finite number of dithered votes given by  $\hat{F}_i(c) = F(c) + v_i$  with  $i = 1, 2, \dots, d$ . Here,  $v_i$  is a random 4D vector drawn from  $\mathcal{N}(0, \Sigma)$  and  $d$  is the number of dithered votes. Let  $B_{DV}(c)$  denote the set of quantized dithered votes (Eq. 4). The confident correspondence set can thus be given by Definition 2.

$$B_{DV}(c) = \left\{ \beta(\hat{F}_i(c)) \mid i = 1, 2, \dots, d \right\} \quad (4)$$

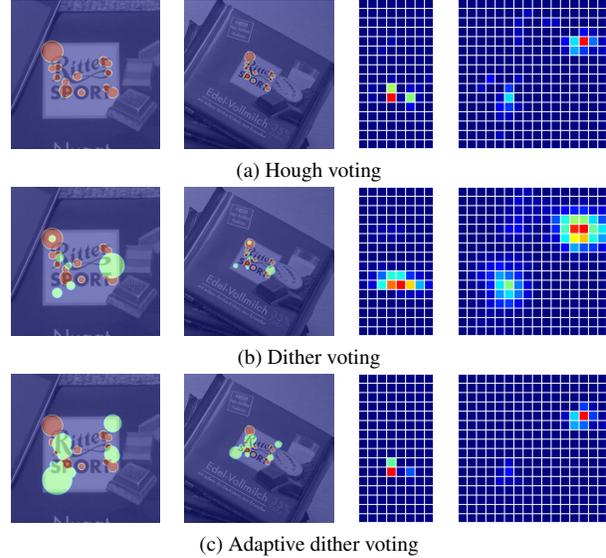


Figure 2: True correspondences and Hough histograms obtained using HV, DV and ADV. True correspondences indicate those corresponding to the histogram maximum. True correspondences found by HV are shown in red, and those newly found via DV or ADV are shown in green. Two 2D histograms are depicted separately for linear transformation  $(\theta, \log \sigma)$  and normalized translation  $(x, y)$ . Red corresponds to the histogram maximum.

Straightforward DV is highly sensitive to mismatches because random sampling gives the same advantage to both true and false correspondences. In Fig. 2b, DV found more confident correspondences that were voted to the bin of the histogram maximum than HV. At the same time, it also augmented the votes for mismatches, as can be seen from the lower left area of the rightmost histogram.

**Definition 2** Given a correspondence set  $C = \{c\}$  and an arbitrary quantization function  $\beta$ , a subset  $\hat{C} \subseteq C$  is a confident correspondence set if and only if  $|\hat{C}| \geq 2$  and  $\forall c_i, c_j \in \hat{C}, B(c_i) \cap B(c_j) \neq \emptyset$ .

### 3.4. Adaptive Dither Voting

We propose deterministically selecting dithered votes instead of randomly sampling them according to a Gaussian distribution. On one hand, more dithered votes are expected to be selected for confident correspondences, while dithering for mismatches has to be minimized. On the other hand, the dithered votes have to be located in closer proximity to the true transformation. When confident correspondences are voted to disjunct bins because of noise (Fig 1b), it can be inferred that the true transformation lies somewhere between the votes. Therefore, the method should avoid selecting dithered votes that lie lateral to both votes (Fig. 1d).

In brief, we look for a set of transformations, called hypotheses, to which the observed correspondence is a geometrical inlier. The hypotheses are selected from the transformations of all tentative correspondences, and are later

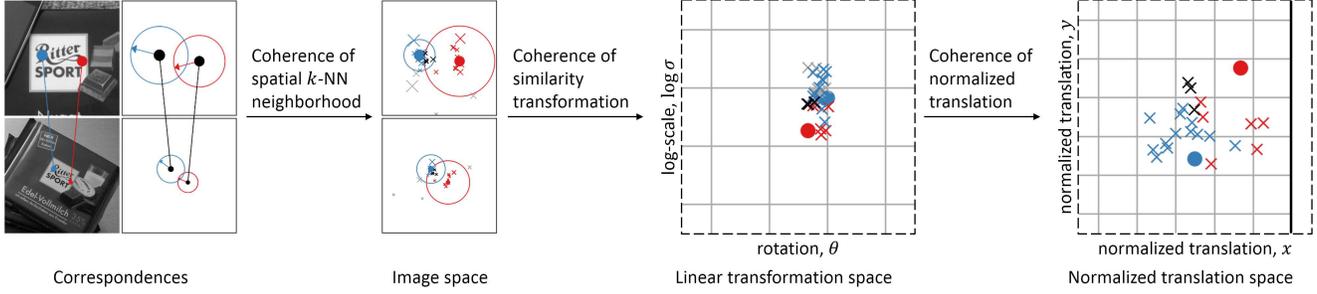


Figure 3: Selection of dithered votes satisfying three geometric constraints in Definition 3. Correspondences are voted as filled circles in a 4D transformation space. Two 2D projections are depicted, separately for linear transformation  $(\theta, \log \sigma)$  and normalized translation  $(x, y)$ . Crosses represent dithered votes. Common dithered votes and rejected dithered votes are represented in black and gray, respectively.

treated as dithered votes. Let  $c$  be the observed correspondence and  $\hat{c} \in C$  a correspondence generating a candidate hypothesis. Let  $r : C^2 \mapsto \{0, 1\}$  be a function mapping an ordered pair  $\langle \hat{c}, c \rangle$  to one if  $c$  is an inlier of  $F(\hat{c})$  and zero otherwise. The set of quantized dithered votes can thus be:

$$B_{\text{ADV}}(c) = \left\{ \beta(F(\hat{c})) \mid \hat{c} \in C, r(\hat{c}, c) = 1 \right\} \quad (5)$$

and the confident correspondence set is given by Definition 2. The hypothesis-inlier relationship is defined by:

**Definition 3** Given two correspondences  $c = (p, q)$  and  $\hat{c} = (\hat{p}, \hat{q})$  and two thresholds  $\epsilon$  and  $\epsilon_t$ ,  $F(\hat{c})$  is defined as a hypothesis of  $c$ , and  $c$  an inlier of  $F(\hat{c})$ , if and only if:

$$\hat{c} \in \mathcal{N}_k(c) \quad (6)$$

$$\|M(c) - M(\hat{c})\|_2 < \epsilon \quad (7)$$

$$\|(\mathbf{t}(p) - M(\hat{c})\mathbf{t}(q)) - \mathbf{t}(\hat{c})\|_2 < \epsilon_t \quad (8)$$

In an image space, correspondences with a larger gap are more likely to be mismatches. This encourages us to employ the neighborhood constraint (Eq. 6).  $\mathcal{N}_k(c)$  represents the spatial  $k$ -nearest neighbors ( $k$ -NNs) of  $c$ . A neighbor of a correspondence  $c_1$  is a correspondence  $c_2$ , both features of which are inside the  $k$ -NNs of the two features of  $c_1$ , respectively. Equation 7 ensures that the observed correspondence has a similar linear transformation to the hypothesis. We decompose Eq. 7 into scaling and rotation constraints:

$$\left| \log(\sigma(c)) - \log(\sigma(\hat{c})) \right| < \epsilon_\sigma \quad (9)$$

$$|\theta(c) - \theta(\hat{c})| < \epsilon_\theta \quad (10)$$

Equation 8 corresponds to the hypothesis-inlier relationship defined in RANSAC [13, 14]. The relationship between related researches and Definition 3 is provided in more detail in the supplementary material.

The relation function  $r$  in Eq. 5 is thus a conjunction of the predicates on Equations 6 to 8. The ADV flowchart is

shown in Fig. 3. The red and blue points are two observed correspondences, projected onto an image, a linear transformation and a normalized translation space. Note how ADV rejects poorly-correlated hypotheses (gray crosses) violating the three constraints in Definition 3.

During the voting process, the quantized dithered votes  $B_{\text{ADV}}(c)$  for each  $c \in C$  are polled to the transformation space, resulting in a 4D histogram. Note that each correspondence itself is also polled to the Hough space like dithered votes. A bin with a larger value is more likely to represent the true transformation. In Fig. 2, we plot the correspondences in the bin of the histogram maximum on the image space. ADV clearly found more confident correspondences than HV (see the supplementary material for more examples). The histograms obtained with ADV are more peaky than their counterparts. Note how the histogram peak resulting from mismatches, i.e. the lower left area of the rightmost histograms for HV and DV, is suppressed with our method. ADV diminishes the proportion of votes from mismatches, in a *relative* way, in inverse proportion to the greatly augmented confident correspondences.

### 3.5. Non-Uniformity of Hough Histogram

In this section, we define a similarity based on the Hough histogram built with ADV. It is observed that confident correspondences often converge in the transformation space while the transformations of mismatches are randomly and uniformly distributed. This motivates us to exploit the non-uniformity of the Hough histogram, which is usually measured by the histogram maximum [8]. The histogram maximum ignores multiple matching surfaces. Instead, we define the non-uniformity via Shannon entropy [17].

Given the notations presented in Section 3.2, a discrete random variable  $F$  with  $|\mathcal{B}|$  possible values  $\{b\}$  and a probability mass function  $P(b)$ , we define entropy  $H$  as

$$H = - \sum_b P(b) \log P(b). \quad (11)$$

---

**Algorithm 1** Generating lookup table.

---

```

1: procedure LUT
2:   for all  $x \in \mathbb{N}_1$  do                                ▷  $\mathbb{N}_1 \triangleq \{1, 2, \dots, 255\}$ 
3:      $\Delta(x) \leftarrow x \log x - (x - 1) \log(x - 1)$ 
4:   return  $\Delta$                                           ▷ lookup table

```

---

Assume that we sampled  $N$  correspondences and  $b$  was seen  $h(b) = N \times P(b)$  times, where  $h(b)$  is the histogram value. The total amount of information that we received is

$$I = N \times H = - \sum_b h(b) \log P(b). \quad (12)$$

$I$  should be maximal if all the outcomes are equally likely such that  $I = N \log N$  when transformations are randomly and uniformly sampled. In contrast,  $I = 0$  when transformations are drawn from a degenerate distribution and represent a perfect match. We define the divergence of the distributions of  $F$  and random samples as:

$$D = N \log N - I = \sum_b h(b) \log h(b). \quad (13)$$

$D$  gives more advantage to a histogram estimated from a larger number of correspondences. It also ensures that isolated transformations, i.e. those in bins with  $h(b) = 1$ , do not contribute to the matching.

## 4. Implementation and Algorithm

In this section, we present our implementation and outline ADC in three algorithms. Given two images, local features are given by Hessian affine feature detectors [11, 13] and described via SIFT [1, 10]. The set of correspondences  $C$  is obtained via a visual vocabulary with approximate  $k$ -means [14]. Given  $C$  as an input, ADV outputs a similarity  $D$ , which is then combined with a non-spatial similarity:

$$S(P, Q) = \begin{cases} D + 1 & \text{if } D \neq 0 \\ S'(P, Q) & \text{else} \end{cases} \quad (14)$$

where  $S(P, Q)$  is the overall similarity and  $S'(P, Q) \in [0, 1]$  is the non-spatial similarity. We adopt  $S'(P, Q)$  as the cosine similarity between TF-IDF histograms [14, 19], but any local feature-based similarity [2, 20] can be used here.

In Section 3.5, the computation of Eq. 13 occurs after the construction of the histogram  $\{h(b) | b \in \mathcal{B}\}$ . Hence, voting is in linear time in  $|B_{\text{ADV}}(c)|$  but computing Eq. 13 is in linear time in  $|\mathcal{B}|$  (larger than 30K for a 4D transformation). To reduce the complexity, we propose an acceleration algorithm, which is in linear time in the number of votes (much less than 1K). Let the quotient of the function  $f(h) = h \log h$  be  $\Delta(x) = x \log x - (x - 1) \log(x - 1)$ . We reformulate Eq. 13 as

$$D = \sum_b f(h(b)) = \sum_b \sum_{x=1}^{h(b)} \Delta(x). \quad (15)$$

---

**Algorithm 2** Adaptive dither voting.

---

```

1: procedure ADV( $C, \mathcal{B}$ )
2:    $D \leftarrow 0$                                           ▷ similarity
3:    $\Delta \leftarrow$  LUT                                    ▷ Alg. 1
4:   for all  $b \in \mathcal{B}$  do
5:      $h(b) \leftarrow 0$                                     ▷ histogram
6:      $X(b) \leftarrow \emptyset$                              ▷ common words
7:   for all  $c \in C$  do
8:      $\mathcal{N}_k(c) \leftarrow k$ -NN of  $c$                      ▷ KD-tree [12]
9:   for all  $c \in C$  do                                     ▷ observed correspondence
10:    VOTING( $c, \beta(F(c)), X, h, D, \Delta$ )                 ▷ Alg. 3
11:    for all  $\hat{c} \in \mathcal{N}_k(c)$  do                           ▷ dithering with Eq. 6
12:      if [Eq. 7] = false then continue
13:      if [Eq. 8] = false then continue
14:      VOTING( $c, \beta(F(\hat{c})), X, h, D, \Delta$ )             ▷ Alg. 3
15:  return  $D$ 

```

---



---

**Algorithm 3** Voting with one-to-one constraint.

---

```

1: procedure VOTING( $c, b, X, h, D, \Delta$ )
2:   if  $u(c) \in X(b)$  then return                       ▷ one-to-one constraint
3:    $h(b) \leftarrow h(b) + 1$                              ▷ update Hough histogram
4:    $D \leftarrow D + \Delta(h(b))$                          ▷ update similarity
5:    $X(b) \leftarrow X(b) \cup u(c)$                        ▷ update common words
6:   return  $X, h, D$ 

```

---

In our implementation, all histogram values  $h(b)$  are initialized by zero (Alg. 2 line 5) and later updated each time a transformation  $F$  is voted (Alg. 3 line 3). Let  $h'(b)$  be the temporarily updated histogram value of  $b$ . We have:

$$D = \sum_F \Delta(h'(F)). \quad (16)$$

The complexity of Eq. 16 only depends on the number of  $F$ , i.e.  $\sum_c |B_{\text{ADV}}(c)|$ . The computation of  $\Delta(x)$  can be skipped with a lookup table (Alg. 1) and replacing runtime computation with a much faster indexing operation.

In summary, our method is outlined in three algorithms: Alg. 1 builds the lookup table of  $\Delta(x)$ ; Alg. 2 realizes ADV given a correspondence set  $C$ ; Alg. 3 summarizes the steps for updating the histogram and the similarity  $D$ . Note that a widely-used one-to-one constraint [3] is imposed on voting (Alg. 3 lines 2 and 5) to penalize the visual words appearing in repeating structures, e.g. building facades and foliage.

## 5. Experiments

### 5.1. Dataset

We tested our method against the latest spatial verification methods in a particular object retrieval scenario. We used four datasets: Oxford Buildings (OB) [14], Paris [15], Flickr Logos 32 (FL32) [16] and Flickr 100K (F100K) [14], which are compared in Table 1. The median scale of the object is around 30% of the image for OB and Paris, and 5%

Table 1: Dataset comparison.

Dataset	Category	#Queries	#Images	#Features
OB [14]	Buildings	55	5.1K	13M
Paris [15]	Buildings	55	6.4K	15M
FL32 [16]	Logo	960	4.3K	13M
F100K [14]	Distractor	n/a	100K	217M

Table 2: MAPs obtained with various transformation quantization configurations. *SADB* represents the range of transformation parameters with  $\sigma_m = A$  and  $\delta = B$ .  $k = 15$  for  $k$ -NN.

Method	OB [14]	Paris [15]	FL32 [16]
ADV-WORST	.805	.740	.656
ADV-BEST	<b>.815</b>	<b>.745</b>	<b>.662</b>
ADV (S10D2.0)	<b>.815</b>	.743	.658
ADV (S12D2.2)	.809	<b>.742</b>	.659
ADV (S14D1.6)	.811	.742	<b>.662</b>

for FL32. We used Hessian affine region detectors [11, 13] and Root SIFT [1] for feature detection and description, respectively. For OB and Paris, we conformed to the widely-used configuration [3, 18] that assumes the datasets include no rotated images. For such datasets, we used a modified Hessian affine region detector [13] and switched off rotation for spatial verification. A visual vocabulary with 1M visual words was constructed for each dataset via  $k$ -means [14].

We measured the accuracy using mean average precision (MAP) [22]. We measured the memory use in terms of peak resident set size (PRSS) in increments of bytes per 1K images. The times for feature detection and visual word assignment were excluded from the evaluation since they are independent of the database size. All the times are in increments of msec per query and per 1K images. All the methods were tested in single threads on a 2.93GHz CPU.

## 5.2. Transformation Quantization

For quantization, we deal with the transformation parameters in Eq. 3 separately. Let  $\rho$  be the maximum dimension of the query in pixels. We only keep correspondences with translations  $x(c), y(c) \in [-\delta\rho, \delta\rho]$  [3]. We also filter out correspondences such that  $\sigma \in [1/\sigma_m, \sigma_m]$  [3, 18]. The rotation  $\theta$  is discretized into eight bins and the others into 16 bins. We varied  $\sigma_m \in \{10, \dots, 15\}$  and  $\delta \in \{1.5, 1.6, \dots, 2.5\}$ , in accordance with 66 configurations, each of which is denoted as *SADB* for  $\sigma_m = A$  and  $\delta = B$ . The results for all datasets are shown in Table 2. We can see that the accuracy is somewhat insensitive to the choices of  $\delta$  and  $\sigma_m$ . Among the MAPs obtained using all the configurations, the difference between the highest and the lowest MAPs was less than 1% for all datasets. Hence, we chose  $\{\sigma_m, \delta\} = \{15, 1.6\}$  for all datasets.

## 5.3. Transformation Dithering

In this study, we assume  $\epsilon_\sigma = \epsilon_\theta = \epsilon_t = \epsilon/n$  where  $n$  is the number of bins per parameter and  $\epsilon$  is a scale factor. As described in Section 5.2,  $n = 8$  for  $\theta$  and  $n = 16$  for the other parameters. The relationship between the MAP and  $\epsilon \in [0, 1]$  is shown in Fig. 4. We can see that the  $\epsilon$  leading to the highest MAP for FL32 was much smaller than its counterparts for OB and Paris. That is, the transformation dithering for FL32 required stronger geometric constraints than for the other datasets. This is because the query is in the form of an ROI for OB and Paris. In contrast, there are more mismatches when no ROIs are specified as in FL32. We chose  $\epsilon = .55$  for OB and Paris and  $\epsilon = .25$  for FL32.

In addition to  $\epsilon$ , we also tested the performance with various  $k \in \{5, 10, \dots, 100\}$ . Figure 5 shows the relationship between the MAP and  $k$ .  $k = 0$  equals HV and  $k = \infty$  equals removing Eq. 6 from Definition 3. We can see that the MAP is increasing monotonically for OB and Paris, which demonstrates the effectiveness of ADV, before becoming fairly constant over  $k$ . It may be argued that the neighborhood constraint did not help here. This is true for OB and Paris again because of the ROI: mismatches only come from the inside of the ROI, which do not violate Eq. 6. When no ROIs are specified as in FL32, the neighborhood constraint comes in useful, and so  $k$  first increases and then decreases the MAP (Fig. 5c).

The relationship between the search time and  $k$  is shown in Fig. 6. Given a large  $k$ , searching FL32 was much faster than searching OB and Paris because of the small scale of the object in this dataset. In other words, the adaptively-determined number of dithered votes for FL32 was much smaller than its counterparts for the other datasets. Since the best MAP was achieved when  $k \approx 15$  for all datasets (Fig. 5), we chose  $k = 15$  for all subsequent tests.

## 5.4. Evaluation and Comparison

We compared our method with BOVW, three prior spatial context methods [9, 23, 24] and the posterior method HPM [3]. Other posterior methods such as RANSAC [14] and Jegou et al.’s method [8] were not tested because they were reported to underperform HPM [3]. Table 3 compares the performance obtained with various methods. The highest MAPs were obtained with  $k = 100$  for the  $k$ -NN used in all prior methods. For HPM, the best performance stabilized at five levels. The results of the methods compared in Table 3 are even higher than those, e.g. .789 MAP and 210 msec for HPM (OB), reported in the literature [3, 23]. This demonstrates the propriety of our implementation. The similarity for DV is the same as that for ADV (Section 3.5). All methods were applied on all images.

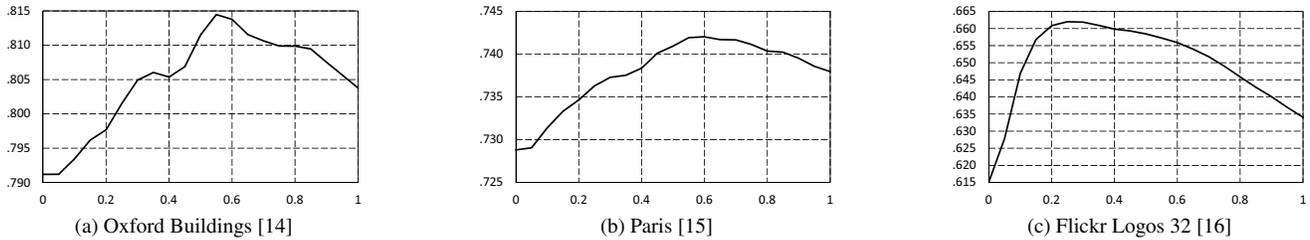


Figure 4: Relationship between MAP ( $y$ -axis) and  $\varepsilon$  ( $x$ -axis).  $k = 15$  for  $k$ -NN.

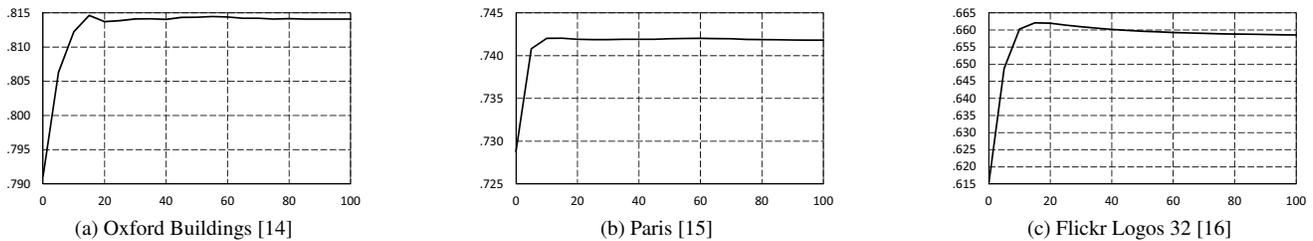


Figure 5: Relationship between MAP ( $y$ -axis) and  $k$  ( $x$ -axis) used in  $k$ -NN.

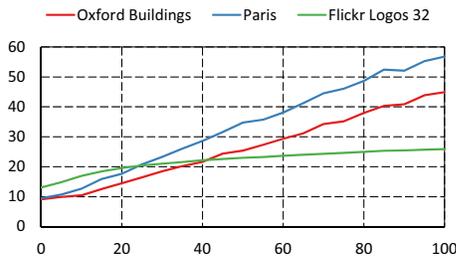


Figure 6: Relationship between search time ( $y$ -axis), in increments of msec per query and per 1K image, and  $k$  ( $x$ -axis) used in  $k$ -NN.

**Retrieval Accuracy** Table 3 shows that our method outperformed all methods except for FL32, where Wu and Kashino’s method [23] obtained a higher MAP. It consistently outperformed HPM, the best baseline we know in this field, by 1.3-4.8%. DV obtained the second highest MAPs for OB and Paris, followed by HPM [3]. For FL32, HPM could not match the other methods. The main reason lies in the small scale of the object and in consequence the high percentage of mismatches. It is difficult for HPM to achieve a good balance between flexibility and accuracy at coarse levels of the pyramid. In contrast, the constraints adaptively incorporated in Definition 3 help ADV handle mismatches.

For each image pair, we measured the number of correspondences found from the histogram maximum. For pairs with more than one such correspondence, the count was averaged over all pairs. From Table 4, we can observe a large drop from positive to negative pairs in both #HV and #ADV, but the drop in #ADV is much greater than that in #HV. The robustness of ADV as regards mismatches is also observed in the two rightmost columns. ADV is more beneficial to the augmentation of correspondences converging in the transformation space, which have a lower chance of being mis-

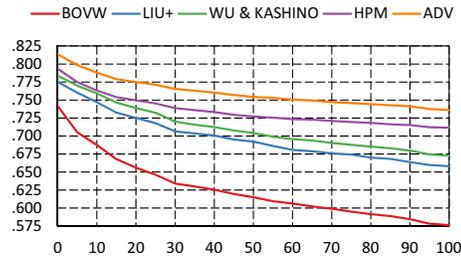


Figure 7: MAPs ( $y$ -axis) versus sizes of distractors ( $x$ -axis).

matches. Therefore, it shows a higher level of #ADV-#HV and #ADV/#HV for positive pairs in Table 4.

We included the F100K distractor dataset [14] in OB for a larger-scale examination (Fig. 7). Because F100K contains no positive images, transformation invariance pales by comparison with the discriminative power in terms of mismatches. As a result, methods imposing stronger spatial constraints enjoy an advantage as regards MAP. The MAP of ADV degrades more smoothly than BOVW and [9, 23], and the degradation is on par with HPM. When 100K images were included, ADV achieved the highest MAP improvement of 16% over BOVW. Table 5 shows the reported MAPs of spatial verification methods on the OB, Paris, and OB+F100K datasets. ADV outperforms all methods on all datasets. Note that Shen et al.’s method [18] shares a common experimental setting with our experiment and is consistently outperformed by ADV.

**Scalability** In Table 3, posterior methods showed much less memory use than prior methods. Posterior methods including ADV are in linear space in the number of features  $|P|$ , while prior methods are in linear space in  $k|P|$  with  $k$  being the parameter of  $k$ -NN. Feature matching, which reduces redundancy, is not available in prior methods, and so

Table 3: Performance comparison. All times are in increments of msec per query and per 1K images.

Method	Oxford Buildings [14]			Paris [15]			Flickr Logos 32 [16]		
	MAP	PRSS	Time	MAP	PRSS	Time	MAP	PRSS	Time
BOVW [14, 19]	.742	36M	.1	.710	32M	.1	.543	36M	.2
Yang and Newsam [24]	.774	15G	72.7	.733	13G	59.6	.634	10G	58.2
Liu et al. [9]	.775	15G	45.8	.731	13G	44.7	.653	11G	59.4
Wu and Kashino [23]	.784	8G	69.4	.735	9G	79.6	<b>.675</b>	7G	93.8
HPM [3]	.794	<b>70M</b>	60.2	.729	<b>66M</b>	67.3	.614	<b>90M</b>	91.6
HV	.786	<b>70M</b>	<b>10.1</b>	.725	<b>66M</b>	<b>12.4</b>	.621	92M	<b>11.6</b>
DV	.800	<b>70M</b>	23.6	.738	<b>66M</b>	17.4	.630	92M	42.2
ADV	<b>.815</b>	<b>70M</b>	12.6	<b>.742</b>	<b>66M</b>	15.8	.662	92M	18.4

Table 4: Statistics pertaining to number of correspondences found from histogram maximum. P/N: positive or negative image pairs. Difference: #ADV-#HV. Ratio: #ADV/#HV.

Dataset	P/N	$ C $	#HV	#ADV	Difference	Ratio
OB	P	99.5	11.9	39.5	27.6	3.6
	N	9.1	1.1	2.1	1.1	2.0
Paris	P	82.4	9.3	32.4	23.1	3.6
	N	10.5	1.1	2.3	1.1	2.0
FL32	P	85.4	15.5	59.2	43.7	3.7
	N	22.8	1.6	3.1	1.5	2.0

Table 5: Reported MAPs of other spatial verification methods.

Method	OB [14]	Paris [15]	OB+F100K [14]
ADV	<b>.815</b>	<b>.742</b>	<b>.736</b>
Perdoch et al. [13]	.789	n/a	.726
Shen et al. [18]	.752	.741	.729
Philbin et al. [14]	.720	n/a	.642
Zhang et al. [25]	.713	n/a	.604

a large  $k \geq 100$  has to be chosen to optimize the accuracy. The indexing thus employs a PRSS more than 100 times larger than posterior methods. In contrast, it is possible for ADV to process 1M images (up to 92GB) in a single thread.

We return to our earlier question: are prior methods truly faster than posterior methods? Table 3 provides a negative response. In practice, ADV outperformed all prior methods in terms of search time. HPM [3] also achieved comparable efficiency, leaving accuracy almost unaffected. The time consumed by prior methods is related to the large search space composed of massive redundant features. Thanks to the constraint in Eq. 6, the determination and voting of dithered transformations for ADV is limited to only spatially neighboring correspondences. ADV is thus in linear time in  $\sum_c |B_{ADV}(c)| \leq \sum_c k = k|C|$ , and so is in linear time in  $C$  for a fixed parameter value  $k$ . HPM suffered from long processing time due to recursive 1-1 constraint verification. The issue becomes significant at coarser levels when the Hough space is divided into larger bins.

Table 6: Search time consumed with and without acceleration. All times are in increments of msec per query and per 1K images. ADV-SLOW: computing Eq. 13 without using Algorithm 3.

Method	OB [14]	Paris [15]	FL32 [16]
ADV-SLOW	27.2	24.6	132.5
ADV	<b>12.6</b>	<b>15.8</b>	<b>18.4</b>

We evaluated the efficiency of the acceleration algorithm proposed in Section 4. The results are shown in Table 6. For comparison, ADV-SLOW is based on a consecutive solution that decouples ADV and Eq. 13. ADV is much faster than ADV-SLOW, especially for FL32 where all transformation parameters are quantized into  $8 \times 16^3 = 32,768$  bins. ADV-SLOW is in linear time in the number of bins, while ADV is in linear time in the number of votes that are usually much fewer than 1K. Our processing time for searching 1K images is 12.6 msec for OB. The commensurate time reported by Perdoch et al. [13] was 238 msec on four cores, and that reported by Shen et al. [18] was 17.6 msec. We can see the high competitiveness of our scalability in the literature.

## 6. Conclusion

We have developed a novel method of spatial verification: Hough transform based on adaptive dither voting. It improves the state-of-the-art performance on four datasets by augmenting correspondence transformations that are lost in the quantization step of previous methods. We also showed experimental evidence related to the open question regarding the efficiency of prior vs. posterior methods. Our method yields a large speed increase compared with both prior methods and current posterior methods, thanks to the acceleration algorithm proposed in Section 4. It indeed searched over 100K distractors in only 0.5 second. In the future, we shall extend ADV to account for the soft assignment [8] of visual words. Because our method can localize the object by accurately estimating the between-image transformation, such information can be used to refine the results, leading to topics of query adaptation [27], query expansion [5, 6] and database augmentation [1].

## References

- [1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012.
- [2] R. Arandjelovic and A. Zisserman. All about VLAD. In *CVPR*, pages 1578–1585, 2013.
- [3] Y. S. Avrithis and G. Toliás. Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval. *International Journal of Computer Vision*, 107(1):1–19, 2014.
- [4] L. Chu, S. Jiang, S. Wang, Y. Zhang, and Q. Huang. Robust spatial consistency graph model for partial duplicate image retrieval. *IEEE Transactions on Multimedia*, 15(8):1982–1996, 2013.
- [5] O. Chum, A. Mikulík, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In *CVPR*, pages 889–896, 2011.
- [6] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, pages 1–8, 2007.
- [7] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317, 2008.
- [8] H. Jegou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336, 2010.
- [9] Z. Liu, H. Li, W. Zhou, and Q. Tian. Embedding spatial context information into inverted file for large-scale image retrieval. In *ACM Multimedia*, pages 199–208, 2012.
- [10] D. G. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [11] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.
- [12] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, pages 331–340, 2009.
- [13] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, pages 9–16, 2009.
- [14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.
- [15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.
- [16] S. Romberg, L. G. Pueyo, R. Lienhart, and R. van Zwol. Scalable logo recognition in real-world images. In *ICMR*, page 25, 2011.
- [17] C. E. Shannon. A mathematical theory of communication. *Mobile Computing and Communications Review*, 5(1):3–55, 2001.
- [18] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Spatially-constrained similarity measure for large-scale object retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(6):1229–1241, 2014.
- [19] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.
- [20] G. Toliás, Y. S. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, pages 1401–1408, 2013.
- [21] G. Toliás, Y. Kalantidis, Y. S. Avrithis, and S. D. Kollias. Towards large-scale geometry indexing by feature selection. *Computer Vision and Image Understanding*, 120:31–45, 2014.
- [22] A. Turpin and F. Scholer. User performance versus precision measures for simple search tasks. In *SIGIR*, pages 11–18, 2006.
- [23] X. Wu and K. Kashino. Image retrieval based on anisotropic scaling and shearing invariant geometric coherence. In *ICPR*, pages 3951–3956, 2014.
- [24] Y. Yang and S. Newsam. Spatial pyramid co-occurrence for image classification. In *ICCV*, pages 1465–1472, 2011.
- [25] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, pages 809–816, 2011.
- [26] W. Zhou, H. Li, Y. Lu, and Q. Tian. SIFT match verification by geometric coding for large-scale partial-duplicate web image search. *TOMCCAP*, 9(1):4, 2013.
- [27] C. Zhu, H. Jegou, and S. Satoh. Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *ICCV*, pages 1705–1712, 2013.