Order Consistent Change Detection via Fast Statistical Significance Testing*

Maneesh Singh, Vasu Parameswaran and Visvanathan Ramesh
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540, USA
{maneesh.singh, vasu.parameswaran, visvanathan.ramesh}@siemens.com

Abstract

*Robustness to illumination variations is a key requirement for the problem of change detection which in turn is a fundamental building block for many visual surveillance applications. The use of ordinal measures is a powerful way of filtering out illumination dependency in representing appearance, and several such measures have been proposed in the past for change detection. By design, these measures are invariant to unknown monotonic transformations that may be caused due to global illumination changes or automatic camera gain. However, previous work has left theoretical and practical gaps that limit their full potential from being realized. For instance, random noise has not been given a principled treatment. In this paper, we formulate the change detection problem in terms of order consistency and show that in the presence of noise with known statistical properties, significance tests for order consistency yield much better results than the state of the art. Since ordinal measures require a reordering of patches, they are usually expensive in practice (O(n*log n) at best). We improve upon this by connecting the problem to* monotonic regression*, and applying a fast algorithm from the corresponding literature. We also show that good trade offs between speed and accuracy can be made by quantization to achieve accurate and very fast matching algorithms in practice. We demonstrate superior performance on statistical simulations as well as real image sequences.*

1. Introduction

Order relationships in the visual signals we receive are a powerful cue to succinctly and robustly represent visual information. We often describe objects and shades with adjectives like ‘brighter’ or ‘darker’. Sketches, apart from displaying object boundaries, often depict shading information to illustrate brightness changes which are an important cue to discerning the object geometry. Xie et al. [18] point out that under assumptions of locally constant illumination and smooth surface geometry, the ordering between pixel intensities is quasi-invariant to an ambient or point light source intensity change. Ordering is also preserved under automatic gain change, employed by most commercial cameras to extend their dynamic range. Note that pixel orderings are invariant to any monotonic transformation of the data. Due to their robustness, order statistics have been used copiously in vision techniques and algorithms in various forms. Gradient orientations represent ordering information between a subset of neighboring pixels. Feature descriptors based on gradient orientations (e.g., in SIFT [10], GLOH [14]) have been used in tracking [7, 17], object detection and recognition [2, 6, 12, 13]) and image retrieval [9] among other applications. Order statistics have also been used for stereo correspondence [5, 19] and change detection [15, 18].

Various order-based statistics have been proposed previously and we review below only the ones most relevant to our work. Bhat and Nayar [5] use rank-based statistics for measuring similarity between two image patches. Zabih and Woodfill [19] define local rank-based image transforms - *census* and *rank* transforms and compare the transformed patches. These approaches disregard the statistics of order violations. One of the first works to model order statistics was Xie et al. [18] who used noise statistics to transform the image into a confidence image where each pixel is replaced by a probability that it is likely (under a known noise model) to retain its sign with respect to the most different pixel in its neighborhood. Image blocks are then compared using a *symmetric* Bhattacharyya correlation metric. Mittal and Ramesh [15] determine the set of disjoint pixel pairs whose relative orderings have changed and design a measure that is a function of the intensity differences in a ”maximal” set of these pixels. These two approaches, unlike previous others, incorporate intensity into statistical measure of order consistency resulting in better performance. However, since they do not model the geometry of ranks (i.e. order) systematically or follow a rigorous probabilistic approach, they are not *optimal*. This indeed is the motivation for our work.

Our work includes several important contributions: (1) It fills an important gap in the usage of ordinal measures.

*This work has been partially funded by the Office of Naval Research (ONR) grant N00014-05-1-0543.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

In contrast to previous methods, we explicitly model noise under which rank-consistency is tested. Our approach uses a probabilistic generative model under which image patches are generated. Using this model, we pose order-consistency as a hypothesis validation problem using significance testing. We show that this approach consistently produces better results than previous approaches. (2) We show that the proposed significance testing problem is equivalent to the problem of monotonic regression allowing us to apply fast algorithms from the monotonic regression literature. (3) In general, ordinal statistics are expensive to compute as they require sorting the data. We propose a method to trade off sensitivity for speed and demonstrate that in practice, one can get a very fast algorithm with a small drop in performance.

The rest of the paper is organized as follows. In Section 2, we pose the order consistency problem as a hypothesis testing problem under additive noise. Section 3 presents the statistical performance of the proposed model and compares it with previously proposed methods. Section 4 maps the hypothesis testing problem to monotonic regression. In Section 5, we present a method to trade off sensitivity for computational speed. In section 6 we validate the model by doing comprehensive statistical comparison with previous methods and by demonstrating its performance on real data for the task of change detection. Finally, we conclude in section 7.

2. Order Consistency and Significance Testing

In this section, we present the model used for testing order consistency under measurement noise. We first explain the notation used in the paper.

2.1. Notation

We use capital letters (e.g., $X = (x_1, x_2, \cdots, x_n)$) to denote vectors or (*vectorized versions*) of image patches. Vectorization is arbitrary (for example, row-major) but fixed for all patches. Let π_X be the rank vector associated with vector X. If $X \in \mathcal{R}^n$, then the rank vector defines a bijective mapping, $\pi_X : \mathcal{Z}_n^+ \rightarrow \mathcal{Z}_n^+$ where $\mathcal{Z}_n^+$ is the set of positive integers from 1 to n. If $\pi_X(i) = j$, then $X(i)$ is the j^{th} smallest entry in the vector. π^{-1} is the inverse mapping of π. $\pi_X^{-1}(j)$ denotes the index of the j^{th} smallest entry in X. M is used for a monotonic transformation and $\mathcal{M}$ denotes the set of all monotonic transformations. We define rank set $S(\pi)$ to be the set of all vectors having the rank π equivalently denoted it by $S(X)$ if $\pi_X = \pi$. We clarify with an example: Let $X = (4.2, 11, 2.3) \in \mathcal{R}^3$. Then, $\pi_X = (2, 3, 1)$ and $\pi_X^{-1} = (3, 1, 2)$.

2.2. The Model

Given a template patch P and a target patch Q, we seek to establish if the patch Q is order-consistent with patch P. We assume that Q is generated via the application of an unknown monotonic transformation, $M(\cdot)$ to P and a subsequent transformation the effects of which are represented by the noise process ϵ with a known distribution P_ϵ. For this paper, we assume the noise process to be additive. Thus, our model is given by (1) below.

$$Q = M(P) + \epsilon \tag{1}$$

For several applications, it may be possible to constrain M by a prior on the set of monotonic transformations. For this paper, we assume complete ignorance of M. This implies that M can be any monotonic transformation and we are constrained to work with a representation that is invariant to it. This invariance is expressed by the order relationships between the pixels in P, represented completely by the rank vector π_P. Thus, we define the rank-consistency hypothesis (i.e. the null hypothesis) $\mathcal{H}_0$,

$$\mathcal{H}_0 : Q = Q_0 + \epsilon \quad \text{s.t.} \;\; Q_0 \in S(\pi_P) \tag{2}$$

We define the **significance test** for $\mathcal{H}_0$ as follows: Given a significance level α, we *fail to reject* $\mathcal{H}_0$ if there exists a patch Q_0 such that the likelihood that Q is generated from Q_0 (rank consistent with P) is greater than probability p_α. Otherwise, we reject $\mathcal{H}_0$. The existence of Q_0 can be tested by actually finding a Q_0, rank consistent with P, which maximizes[1] the probability $P(Q|Q_0)$ and testing if $P(Q|Q_0) > p_\alpha$ – this is the approach we take in this paper.

Since the process of testing the hypothesis seeks a feasible point in the rank set, it is helpful to understand the geometry of the rank sets which we describe in Appendix A. Note here that each rank set is convex, open and its boundaries are $(n-1)$ linear hyperplanes defined by equality constraints $x_i = x_j$. The hyperplanes contain the line defined by $x_i = c \;\forall i$ and are perpendicular to the hyperplane, $\mathcal{H}^n \doteq \sum_{i=1}^n x_i = \sum_{i=1}^n i = n * (n+1)/2$. Note that if we embed the permutation polytope S_n[2] in $\mathcal{R}^n$ by mapping the ranking π to the vector represented by π, then each rank set contains the corresponding rank vector.

We illustrate this in Figure 1, with an example for a patch of size 3 pixels and isotropic Gaussian noise ϵ. Since the rank sets are perpendicular to $\mathcal{H}^3$, we just show the projections of the rank sets on $\mathcal{H}^3$. The six possible rankings are shown in square boxes. The associated rank sets are shown using the three bounding hyperplanes ($x_1 = x_2$, $x_1 = x_3$ and $x_2 = x_3$) denoted by dashed blue lines. For the given model patch P with $\pi_P = (3, 2, 1)$, Q_0 is some arbitrary vector in the rank set, $S(\pi_P)$. To test $\mathcal{H}_0$, we just need to find $\hat{Q}$ such that $\hat{Q} = \text{argmax}_{Q_0} P(Q|Q_0)$.

[1] Technicality: Rank sets are open sets and except for the trivial case, the maximization problem is not solvable on open sets. Hence, we solve the problem on the closure of the appropriate rank set.

[2] The permutation polytope S_n is the convex hull of all permutations of vector $(1, 2, \cdots, n)$. Refer to Definition 2.1 on page 6, [11].

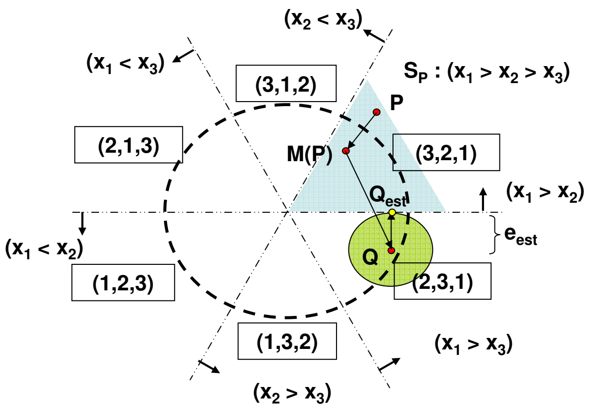

Figure 1. A 3D permutation polytope embedded in $\mathcal{R}^3$ and the associated ranks sets shown projected on the hyperplane $x_1 + x_2 + x_3 = 6$. $P = (p_1, p_2, p_3)$ is a model patch with $p_1 > p_2 > p_3$. Given a target patch Q and isotropic noise, the maximum likelihood estimate is given by Q_{est} ($\equiv \hat{Q}$), the perpendicular projection of Q on the hyperplane $x_1 = x_2$.

2.3. Isotropic Noise

Now consider the special case of isotropic noise. More general cases, where different pixels can have different noise, will be touched upon later (see Section 4). If the noise, ϵ, is isotropic, the level sets of the noise density function, $f_\epsilon(\cdot)$, are spheres. Further, if the profile of the noise density function is non-increasing, $\hat{Q}$ can be obtained by finding the minimum L_2 distance between $S(\pi_P)$ and the vector Q. $\hat{Q}$ is shown in Figure 1 as Q_{est} for our 3-pixel example and can be estimated by using the Projection Onto Convex Sets (POCS) algorithm [16]. Examples of such distributions include the Gaussian and the Student's t-distribution. Note also that the rank sets can be represented using linear inequalities. Thus, the problem of minimizing the quadratic function represented by the squared-L_2 norm subject to these linear constraints is a quadratic program. We present in Figure 2, Matlab code for estimating $\hat{Q}$ using the quadprog$(\cdot)$ function.

```
function[$\hat{Q}$, $\hat{\epsilon}$] = rankConsistentEstimate($Q$,$P$)
$n$ = length($P$);
[Psorted, indx] = sort($P$,'ascend');
$Q$ = $Q$(indx); $H$ = eye($n$); $f = -Q$;
$A$ = zeros($n$-1,$n$);
for $i$=1:$n$-1 $A$(i,[$i$:$i$+1]) = [1 -1]; end
$b$ = zeros(n-1,1);
[$\hat{Q}$ fval exitflag,output,lambda] = quadprog($H$,$f$,$A$,$b$);
[indxsorted,indx2] = sort(indx);
$\hat{Q}$ = $\hat{Q}$(indx2);
$\hat{\epsilon}$ = Q - $\hat{Q}$
```

Figure 2. Matlab code using quadprog$(\cdot)$ for estimating the vector $\hat{Q}$ closest in distance to Q (L_2 norm sense) such that $\hat{Q} \in S(\pi_P)$.

3. Statistical Validation

In this section, we provide statistical validation for the proposed method. To do this, we construct the following problem: Given a model patch P, the goal is to distinguish between two sets of patches using the proposed method. The first set, $\mathcal{Q}$, consisting of patches Q_i, is generated by applying randomly chosen monotonic transformations (to simulate global illumination changes or camera gain) to P and then adding independent realizations of noise from the random process ϵ to the result. For this experiment, ϵ is additive white Gaussian noise. The second set, $\mathcal{Q}'$, consisting of patches Q'_i, is similarly generated from P_{occ}. P_{occ} is generated from P by covering a random part of the patch P with a 'foreground' patch such that there is a minimum separation between the foreground and the background pixels (separation being a function of the noise standard deviation). This process simulates random partial occlusions of P. The hypothesis is that when a foreground partially occludes a background or that when two object classes are being compared, there is a minimum subset of pixels which can be used to distinguish the patches corresponding to the two cases (the distance between the classes). The intra-class variation is simulated here using additive noise and random monotonic transformations of intensity. Thus, the set $\mathcal{Q}$ is generated according to the model in (1) while $\mathcal{Q}'$ is not. The model image used for these experiments is a 50×50 portion of the *cameraman* image shown in Figure 3. The occluding patch is of size 7×7 and the noise standard deviation is $\sigma = 40$. We carried out 1000 runs each with independent realizations from $\mathcal{Q}$ and $\mathcal{Q}'$ according to the process explained above. We use these 2000 runs to plot the ROC curves shown in Figure 4.

We compare our results to two representative order-consistency measures from literature - one proposed by Bhat and Nayar [5] and another by Mittal and Ramesh [15]. We refer to these as BN-κ and MR-κ respectively. BN-κ is representative of a class of measures that only uses order relationships between pixels and pixel intensities do not contribute directly to the measure. In extensive experiments, BN-κ is shown to outperform normalized cross-correlation (NCC), sum of squared distances (SSD) and Zabih and Woodfill's rank transform [19] for image correspondence. MR-κ is a recent state of the art measure that outperforms those that use only order relationships, like those proposed by Bhat and Nayar [5] and Zabih and Woodfill [19]. MR-κ measure is constructed using both order relationships as well as image intensities. However it is close in spirit to BN-κ and can be looked as a recipe to make BN-κ robust to intensity noise. The ROC curves in Figure 4 show clearly that our method easily outperforms both BN-κ as well as MR-κ. In Section 6 we show that our method also outperforms them for real image sequences. The superior performance is due to the formal incorporation of noise statistics into our method and below, we provide insights that help explain this further.

BN-κ essentially counts the number of order violators for every pixel. In other words, for every pixel i, the number of such pixels j are counted for which the sign of $P_i - P_j$ is different from the sign of $Q_i - Q_j$. This number is then summed over all the pixels to get the measure BN-κ. This measure has been quite successfully applied to several problems including stereo correspondence and motion estimation. However, as pointed out in [15] and discussed by us earlier, this method does not take into account the value of image intensities in computing the change measure. Problems arise with this method when the number of violators is large but relatively insignificant (due to noise in almost constant areas) while the number of significant violators is small (due to valid differences in the two patches). In such a case, the former will overwhelm the latter. Indeed, we found that BN-κ gives much better results when such situations are avoided.

MR-κ extends the basic idea of counting the number of violators in two ways: (a) Instead of computing the number of violator pixel pairs, they compute a statistic over these. In the simplest case, the statistic is the absolute difference between the pixel pair (in other words, the standard deviation of the data corresponding to the pixel pair). (b) Secondly, the measure is computed as a sum over a *maximal* set of pixel pairs with relative ordering changes and not over all such pixel pairs. In essence, this approach finds a set so that the statistics being summed up for pixel pairs remains uncorrelated. This in practice gives a very good measure.

Finally, our approach finds a projection onto rank-set boundaries which obey the constraints: $x_i = x_j = \cdots = x_k$ and computes the l_2 distance to the projected vector. The l_2 distance can be seen as summation of sample variances over disjoint sets (bearing some relationship to MR-κ). It is indeed the correct (and sufficient) statistic to use (given a Gaussian noise model). This advantage is clearly brought forth in the ROC curves.

4. Monotonic Regression

In this section, we show that the problem defined in Section 2 can be mapped to a monotonic regression problem. This connection enables the usage of fast and efficient algorithms from the monotonic regression literature, to solve our problem of rank consistency. In monotonic regression, given an input sequence, the goal is to find the monotonically increasing or non-decreasing sequence that minimizes the cost of deviating from the input sequence (i.e. the 'closest' monotonic sequence to the input sequence, given a cost function). Let $\tilde{Q} \doteq Q(\pi_p^{-1}(i))$, i.e. the sequence obtained by applying the permutation π_P^{-1} to Q (recall that $\pi_P^{-1}(i)$ denotes the index of the ith smallest entry in P). The original hypothesis $\mathcal{H}_0$ (equation 2) can now be mapped to the

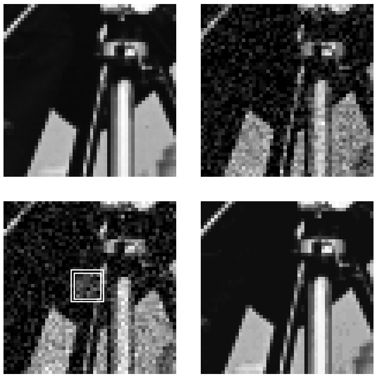

Figure 3. Example for testing performance under AWGN $N(0, \sigma^2 \cdot I)$. (a) Top-left: Model patch; (b) Top-right: Model patch with AWGN, some $Q \in \mathcal{Q}$; (c) Bottom-left: Partially-occluded model patch with AWGN some $Q \in \mathcal{Q}'$. Occluded region shown with a white square; (d) Bottom-right: Estimate, $\hat{Q}$, from (c) that is rank-consistent with the model patch in (a).

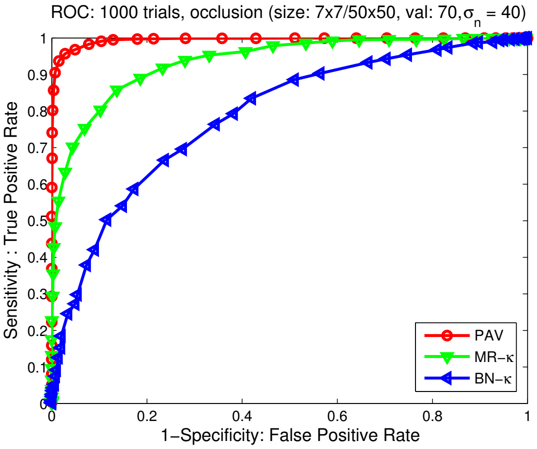

Figure 4. Performance under AWGN $N(0, \sigma^2 \cdot I)$. ROC curves show PAV, RM-κ and BN-κ. Example images shown in Figure 3.

following equivalent problem:

$$\mathcal{H}_0 : \tilde{Q} = \tilde{Q}_0 + \tilde{\epsilon} \quad \text{s.t.} \;\; \tilde{Q}_0(i) \leq \tilde{Q}_0(i+1) \; \forall i \in \mathcal{Z}_{n-1}^+ \quad (3)$$

Thus, the null hypothesis in (2) is replaced by the equivalent hypothesis that $\tilde{Q}$ is generated by some monotonically increasing sequence $\tilde{Q}_0$ and additive noise $\tilde{\epsilon}$. $\tilde{\epsilon}$ is similarly the permutation of ϵ induced by π_p^{-1}. This is precisely the hypothesis used for the monotonic regression problem. We consider below special cases for which efficient algorithms are available in the monotonic regression literature.

4.1. Independent Noise, Convex Cost

If the noise variables are independent, then the log-likelihood factors into a sum of individual likelihoods, i.e., $\log P_\epsilon(\tilde{Q}|\tilde{Q}_0) = \sum_i \log p_{\epsilon_i}(\tilde{Q}(i)|\tilde{Q}_0(i))$. Let us define the cost function, $C_i(\tilde{Q}(i), \tilde{Q}_0(i)) \doteq -\log p_{\epsilon_i}(\tilde{Q}(i)|\tilde{Q}_0(i))$.

Note that this is a very generic formulation which covers anisotropic and heteroscedastic distributions. In fact, for different pixels, noise distribution may be completely different. The monotonic regression problem admits a unique solution if the cost C_i is convex. For such cases, there exist efficient algorithms to solve the problem. The distributions that this formulation can deal with include the Gaussian, double-exponential, generalized error and Student's t-distribution among others. The most popular among such algorithms is the pool-adjacent-violators (PAV) algorithm [8, 3]. This problem is also called the isotonic regression problem [4] especially if the cost functions are defined using the the l_p norm.

In Figure 5, we give a pseudo-code for the PAV algorithm. The algorithm maintains a set of blocks $B(i)$ representing a set of adjacent pixels (Note that the block represents a vectorized image patch and adjacency implies neighboring pixels in the vector). The first two elements of $B(i)$ represent the pixel indices of the first and last elements of the block. The third element represents the current estimate of $\hat{Q}$ using the pixels represented by $B(i)$. Initially the number of blocks is equal to the length of the data. The while loop works by looking at adjacent blocks. If a pair of adjacent blocks are violators, i.e. the estimates corresponding to the blocks are in a decreasing order ($B(i)\{3\} > B(i+1)\{3\}$), then they are pooled together into a larger block and the estimate is computed over the pooled block. Finally, either all the blocks are in the right order or there is only one block left. In either case, the block wise constant estimate represented by the third element of $B(i)$'s yields the final estimate $\hat{Q}$.

The PAV algorithm is very simple but has a computational complexity $O(n^2)$. This is due to the while loop in Step 5 requiring up to $O(n^2)$ iterations in the worst case.

```
function[$\hat{Q}$, $\hat{\epsilon}$] = PAV($Q$,$P$)
1.  $n$ = length($P$);
2.  [Psorted, indx] = sort($P$,'ascend');
3.  $Q$ = $Q$(indx);
4.  for $i = 1 : n$, $B(i) = \{i, i, Q(i)\}$; end
5.  while there exists j s.t. $(B(j)\{3\} > B(j+1)\{3\})$,
6.        $\hat{q} = \text{argmin}_q \sum_{i=b(j)\{1\}}^{b(j+1)\{2\}} C_i(q(i), q)$;
7.        $R = \{b(j)\{1\}, b(j+1)\{2\}, \hat{q}\}$.
8.        replace $B(j)$ and $B(j+1)$ by $R$.
9.  end while
10.  nb = number of blocks B
11.  for $i = 1 : nb$,
12.        $\hat{Q}(B(i)\{1\}) = \cdots = \hat{Q}(B(i)\{2\}) = B(i)\{3\}$
13.  end
14. $\hat{\epsilon}$ = $Q$ - $\hat{Q}$
15. indxInverse(indx) = 1 : $n$
16. $\hat{Q}$ = $\hat{Q}$(indxInverse); $\hat{\epsilon}$ = $\hat{\epsilon}$(indxInverse)
```

Figure 5. Pseudo-code for the PAV algorithm.

4.2. Noise with Exponential power distribution

When the noise ϵ has independent components that have generalized error or exponential power distribution, the cost is given by $C_i(q, q_0) = \| \frac{q(i)-q_0(i)}{a_i} \|_i^b$. and a_i and b_i are the respective scale and shape parameters. When the shape parameters are all 1 (independent Laplace noise), then the cost reduces to the l_1 norm formulation. In such a scenario, the optimization problem in line 6 of the PAV algorithm in Figure 5 reduces to the estimation of median of the data. Similarly, when shape parameters are all 2 (independent Gaussian noise), the cost becomes quadratic and the corresponding optimization problem in line 6 reduces to estimation of mean of the data. Consequently, additional savings in computation can be made by a simple mean-update[3] strategy whereby whenever two blocks $B(i)$ and $B(i+1)$ are merged, $\hat{q}$ in line 6 is just a weighted average of $B(i)\{3\}$ and $B(i+1)\{3\}$ where the associated block weights are the respective sizes of the blocks.

Now, we carry out the same experiment as in Section 3. However, this time we use independent additive Laplace noise with variance $\sigma^2 = 400$ and using the PAV algorithm in Figure 5 with median calculation. The performance comparison with BN-κ and MR-κ is shown using ROC curves in Figure 6.

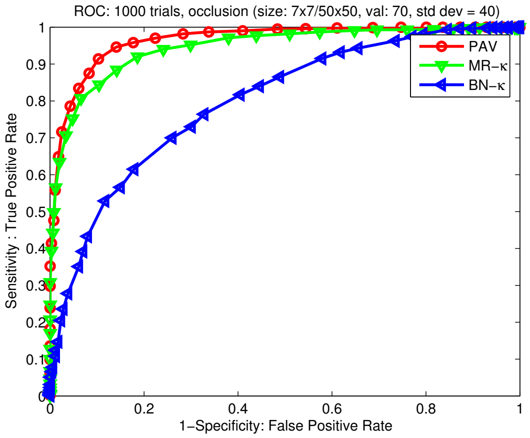

Figure 6. Performance under i.i.d. Laplace noise . ROC curves are plotted for PAV, MR-κ and BN-κ. Example images used for testing are shown in Figure 3.

5. qPAV - Trading off Sensitivity for Speed

While matching algorithms based on order relationships between pixels are more robust than those based only on pixel intensities, they are in general computationally more expensive. In this section, we propose a trade off between sensitivity and computational complexity based on quantizing the model P.

[3] We omit details as this modification is trivially evident.

First we identify the computational complexity of the key steps of the PAV algorithm. (1) Sorting P is $O(n \cdot \log n)$; (2) The while loop in PAV - worst case of $O(m^2)$ iterations where m is the number of bins ($= n$ for PAV); and, (3) The optimization step in line 6 of the algorithm. We note that both steps (1) and (2) can be made faster by quantizing the model P. If the number of quantization levels are $m << n$, then complexity of step (1) reduces to $O(n)$ (by using bucket sort, for example) and the complexity in step (2) is reduced to $O(m^2)$.

We call this new algorithm, which finds the estimate on the quantized model, the qPAV algorithm. The steps of this new algorithm are: (1) Quantize[4] the model P to P_Δ using a sensitivity parameter s. (2) Sort P_Δ and construct $\tilde{Q}$. (3) Initialize the blocks $B(i)$ and follow the PAV algorithm. We present the pseudo-code for qPAV in Figure 7. Let m be the number of quantization levels and Δ_m be the corresponding quantization level. Note that the maximum number of initial blocks is constrained to be the same as number of quantization levels in P_Δ. Thus, the cost of PAV is reduced from $O(n^2)$ to $O(m^2)$. Further, sorting can be done using the bucket sort algorithm.

```
function[$\hat{Q}, \hat{\epsilon}$] = qPAV($Q$,$P$,$s$)
$P_\Delta$ = quantize($P$,$s$);
[$P_\Delta$, indx] = sort($P_\Delta$,'ascend');
$Q$ = $Q$(indx);
$\Delta$ = unique($P_\Delta$);
$m$ = length($\Delta_i$);
for $i = 1 : m$,
    $jmin = \min\{j : P_\Delta(j) = \Delta_i\}$
    $jmax = \max\{j : P_\Delta(j) = \Delta_i\}$
    $\hat{q} = \text{argmin}_q \sum_{j=jmin}^{jmax} C_j(q(j), q)$;
    $B(i) = \{jmin, jmax, \hat{q}\}$;
end
Follow lines 4-16 in PAV pseudo-code.
```

Figure 7. Pseudo-code for qPAV - the quantized PAV algorithm.

For all experiments and results that follow, we use the qPAV algorithm. The input parameter s represents the desired sensitivity of the system and denotes the quantization bin size. If it is set to 1, the quantization step is skipped and qPAV becomes the original PAV algorithm.

We now demonstrate the effects of quantization on both the speed and accuracy of the algorithm on the test example constructed in Section 3 using the L_2 norm as our cost function. In Figure 8 we show the ROC curves when the quantization bin sizes are 1, 25, 50, 100, and 150. The number of quantization levels decrease correspondingly - 215 (all distinct values), 10, 6, 4 and 3. For comparison, we also reproduce the ROC of the MR-κ [15] statistic. Note that we outperform MR-κ even with 3 quantization levels. In Table 1, we show the corresponding computation times for the qPAV algorithm. The code used for the table is in Matlab and uncompiled. Notice the trade off between speed and computational accuracy.

In Table 2, we show results on real data for our C++ compiled code and compare the performance with both BN-κ and MR-κ. These experiments were carried out on the `S3-T7-A` sequence from the PETS 2006 database (refer to Section 6 for details). These results demonstrate that (1) We can methodically trade off sensitivity for speed using the qPAV algorithm. (2) Both on real and simulated data, qPAV outperforms MR-κ both in terms of speed and matching accuracy.

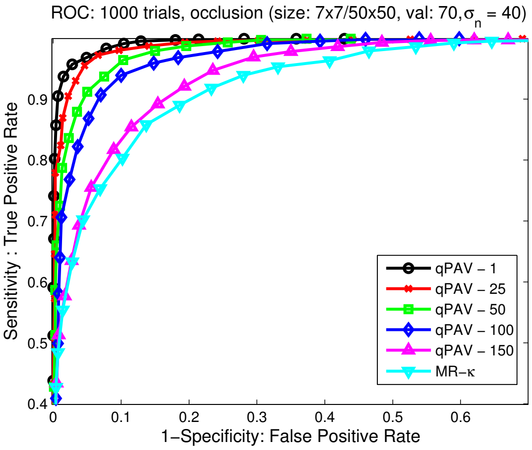

Figure 8. Performance of the qPAV algorithm under AWGN $N(0, \sigma^2 \cdot I)$. ROC curves are plotted for qPAV with different quantization levels. Example images used for testing are shown in Figure 3.

No. of bins	215	10	6	4	3
Time (msec)	240	15.7	11.7	7.8	7.1
Accuracy	0.963	0.953	0.933	0.916	0.869

Table 1. Table of computational performance vs. accuracy for qPAV as a function of different quantizations levels for the Matlab (uncompiled) implementation of qPAV. The time taken is average over 1000 experiments conducted on a patch of size 50×50. Accuracy rates are given at equal error rate, i.e., probability of missed detection and false alarms are equal.

6. Application

Following the statistical experiments of section 3, we now apply proposed qPAV algorithm to the application of change detection on real surveillance video sequences from the PETS 2006 database [1].

6.1. Change Detection

We provide results on a representative change detection scenario from a surveillance video from the PETS 2006

[4] In this paper, we used uniform quantization. Better quantization strategies can be used and are a part of future research.

MR-κ	BN-κ	qPAV-1	qPAV-5	qPAV-10	qPAV-25
0.9441	0.8623	0.9873	0.9842	0.9650	0.8871
0.0737	0.0361	0.0268	0.0241	0.0222	0.0217

Table 2. Table of computational performance vs. accuracy for qPAV as a function of different quantizations levels for a C++ implementation of qPAV. For comparison, we also give the time and accuracy numbers for MR-κ and BN-κ. In the first row, the algorithm name is given. Second row shows as performance numbers the area under the ROC curve. The third row displays the corresponding average time taken (msec) per patch of size 12 × 12.

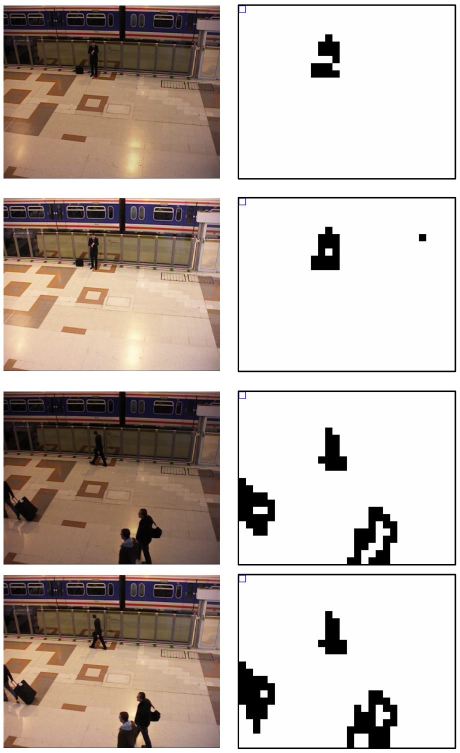

Figure 9. Application of qPAV to change detection: Original images on left column, detections on right. Left column images from top to bottom referred to as (a), (b), (c) and (d). Right column images referred to as (e), (f), (g) and (h).

database [1] which is one of the standard datasets for evaluating surveillance applications. For the reference patch we used the initial image which corresponded to the 'empty scene'. This image (`S3-T7-A.00000.jpeg` from the dataset) is not shown here but is similar to the image in Figure 9 (a) minus the standing person visible in the frame and his bag. We manually labeled several frames in the sequence for creating the ground truth for foreground objects.

For testing our change detection algorithm, each 360 × 288 image frame was divided into 12 × 12 patches for a total of 720 patches per frame. We show two representative frames in Figure 9(a) and (c). We ran the change detection algorithm based on qPAV as well as using MR-κ and BN-κ and constructed the ROC for each case. The designated task was to correctly detect all foreground patches while rejecting the background ones. These ROC curves are plotted in Figure 10. They show clearly that the qPAV algorithm gives 10% better results than MR-κ and 20% better than BN-κ at equal error rates (when the probability of missed detection equals that of false alarms). We also show two change detection results from the qPAV algorithm in Figure 9(e) and (g) for Figures (a) and (c) respectively. For detecting these changes, we estimated the noise standard deviation in the video and set the detection threshold at 95% significance.

Our next experiment involved applying a gamma transformation to these images to simulate automatic gain changes in the camera. Two sample images showing the effects of gamma transformation are presented in Figure 9(b) and (d). Notice the change in intensity and saturation levels. The experimental setup is the same as before and the task is that of change detection in presence of such intensity variations. The corresponding ROC curves are presented in Figure 11. They show that all the algorithms maintain their relative performance vis a vis the earlier experiment (without the gain changes) and that qPAV maintains its better performance. The changes detected by the qPAV for Figure 9(b) and (d) are also shown in (f) and (h) respectively.

Along with the above experiments, we have applied our change detection algorithm on several videos from the PETS 2006 database both with and without simulated gain changes. These videos have been uploaded as additional material with the paper.

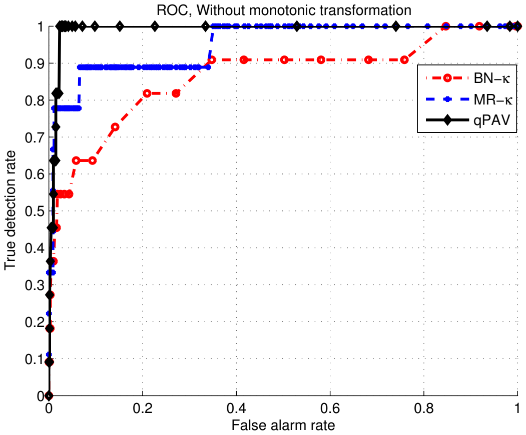

Figure 10. Performance of qPAV, RN-κ and BN-κ on the PETS data. Examples shown in figs. 9(a), (c), (e) and (g).

7. Conclusions and Future Work

We have presented in this paper, a formal probabilistic approach for testing order consistency between image patches when the noise probability distribution is available to us. We have demonstrated, both with statistical simu-

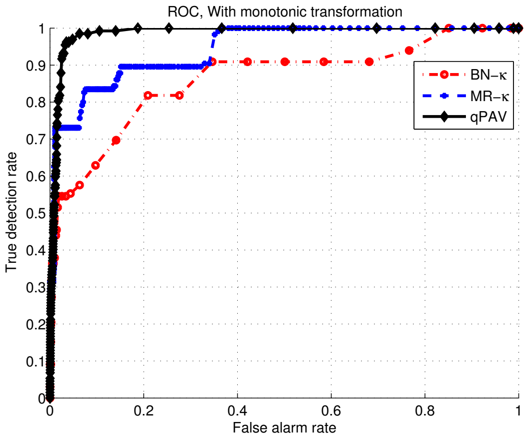

Figure 11. Performance of qPAV, RN-κ and BN-κ on PETS data under arbitrary gamma transformations. Examples shown in 9(b), (d), (f) and (h).

lations of additive random noise as well as with results on real surveillance videos that the proposed algorithm works better (in both speed and accuracy) than the best reported in the literature. We modeled the problem using a generative framework and formulated it as a significance testing problem. This helped us show its equivalence to the monotonic regression problem. Finally, we addressed the problem of large computational costs associated with the order consistency methods. We took one of the most popular monotonic regression algorithms and showed that we could make it very fast by quantizing the model representing the ranks, while incurring only a very small drop in performance. We also demonstrated that our algorithm can be used for fast change detection in video surveillance. There are several directions in which the work can be extended: (1) Integrating appropriate priors (e.g. on the monotonic transformation), (2) Investigating the effect of different quantization schemes beyond uniform quantization used in this work, (3) Using the active set approach proposed by Best and Chakravarty [4] for qPAV implementation, and, (4) Extending it for other problems such as registration, object recognition, etc.

References

[1] PETS 2006 benchmark data - http://pets2006.net/.

[2] B. Alefs. Embedded vehicle detection by boosting. In *Proc. of the IEEE Intell. Trans. Systems Conf.*, 2006.

[3] R. E. Barlow, D. Bartholomew, J. M. Bremner, and H. D. Brunk. *Statistical inference under order restrictions; the theory and application of isotonic regression*. Wiley, New York, 1972.

[4] M. J. Best and N. Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47:425–439, 1990.

[5] D. N. Bhat and S. K. Nayar. Ordinal measures for image correspondence. *IEEE PAMI*, 20(4):415–423, 1998.

[6] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli. On the use of sift features for face authentication. In *IEEE CVPR 06 Workshop*, 2006.

[7] R. D. Changjiang Yang and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *ICCV*, 2005.

[8] W. Hardle. *Applied Nonparametric Regression*. Cambridge University Press, 1990.

[9] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. IEEE CVPR*, 2004.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.

[11] J. I. Marden. *Analyzing and Modeling Rank Data*. Chapman and Hall, 1995.

[12] A. Mian, M. Bennamoun, and R. Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *IEEE PAMI*, accepted, 2007.

[13] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *Proc. ICCV*, 2005.

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE PAMI*, 10:1615–1630, 2005.

[15] A. Mittal and V. Ramesh. An intensity-augmented ordinal measure for visual correspondence. In *Proc. IEEE CVPR*, 2006.

[16] M. I. Sezan and H. Stark. Image restoration by method of convex set projections: Part ii-applications and numerical results. *IEEE Transactions on Medical Imaging MI-l*, 1982.

[17] F. Tang and H. Tao. Object tracking with dynamic feature graph. In *IEEE Int. Workshop on VS-PETS*, 2005.

[18] B. Xie, V. Ramesh, and T. E. Boult. Sudden illumination change detection using order consistency. *Image and Vision Computing*, 22(2):117–125, 2004.

[19] R. Zabih and J. Woodfill. A non-parametric approach to visual correspondence. In *Proc. European Conference on Computer Vision*, pages 82–89, 1994.

Appendix A: Geometry of rank sets

- **(P1) Number**: There are $(n!)$ rank sets which partition $\mathcal{R}^n$.

- **(P2) Convexity**: Each subset $S(\pi_i)$ is convex.

- **(P3) Boundaries** $\delta S(\pi_i)$: The boundaries are linear hyperplanes derived by equality constraints. Each subset $S(\pi_i)$ can be described by exactly $(n-1)$ linear inequalities. Thus, the subset is adequately described by $(n-1)$ hyperplanes with passing through origin. The hyperplanes are described by $xi - xj \geq 0$. The boundaries are all perpendicular to the hyperplane $\mathcal{H}^n \doteq \sum_{i=1}^n x_i = \sum_{i=1}^n i = n * (n+1)/2$.

- **(P4) Intersection** $\bigcap \bar{S}(\pi_i)$: Intersection of the closures of rank sets contain all the constant patches (and nothing else), i.e., $\bigcap_{\pi_i \in \Pi} S(\bar{\pi_i}) = \{X \in \mathcal{R}^n | X = k \cdot 1, k \in \mathcal{R}\}$. Thus, the boundaries represent a sheaf of planes passing through the axis (or pencil) represented by $X = k \cdot 1$.