LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-208738

Salient Points for Tracking Moving Objects in Video

C. Kamath, A. Gezahegne, S. Newsam, G.M.Roberts

December 21, 2004

Image and Video Communications and Processing
San Jose, CA, United States
January 17, 2005 through January 20, 2005

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Salient Points for Tracking Moving Objects in Video

Chandrika Kamath, Abel Gezahegne, Shawn Newsam, George M. Roberts

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
7000 East Avenue, Livermore, CA 94550

ABSTRACT

Detection and tracking of moving objects is important in the analysis of video data. One approach is to maintain a background model of the scene and subtract it from each frame to detect the moving objects which can then be tracked using Kalman or particle filters. In this paper, we consider simple techniques based on salient points to identify moving objects which are tracked using motion correspondence. We focus on video with a large field of view, such as a traffic intersection with several buildings nearby. Such scenes can contain several salient points, not all of which move between frames. Using public domain video and two types of salient points, we consider how to make these techniques computationally efficient for detection and tracking. Our early results indicate that salient regions obtained using the Lowe keypoints algorithm and the Scale-Saliency algorithm can be used successfully to track vehicles in moderate resolution video.

Keywords: salient points, detecting moving objects, video tracking

1. INTRODUCTION

Detection and tracking of moving objects is an important task in the analysis of video data for applications such as video surveillance, traffic monitoring and analysis, human detection and tracking, and gesture recognition. A common approach to detecting moving objects is background subtraction, where each frame in the video is compared to a background or reference frame, and pixels that deviate significantly from the background are considered to be moving or foreground objects [1]. This background frame is continuously updated to account for changing illumination, moving objects that come to a stop, stopped objects that start moving again, etc. This updating makes the detection of moving objects computationally expensive, especially if additional logic is required to handle aperture problems, or objects that split up, such as a vehicle going behind a lamp-post.

In this paper, we consider the use of simpler techniques based on salient points to identify and track moving objects in video. Salient points, also referred to as interest points, are extracted from each frame and tracked over time. For video with a relatively large field of view, such as a traffic intersection with several buildings, there can be many salient points in each frame, not all of which move across frames. Many of these can be removed either during tracking or by pre-processing the salient points prior to tracking.

This paper is organized as follows: in Section 2, we provide an overview of the work done in tracking using salient points and describe the salient points used in our work. In Section 3, we describe our tracking methodology based on motion correspondence. Next, in Section 4, using a public domain video sequence, we present the results using salient points in tracking and discuss how we can improve the computational efficiency of the method. Finally, in Section 5, we conclude with a summary and ideas for future work.

2. OVERVIEW OF SALIENT POINTS

Salient points, or interest points, are landmarks in an image which are often intuitively obvious to a human. These include corners of a building, the eyes on a human face, the edges of an object, etc. While salient points such as corners and edges have been used previously in tracking, there has been an increased interest in this approach with the development of new ways of defining salient points in the content-based image retrieval community [2].

Further author information: CK: E-mail: kamath2@llnl.gov

The traditional approach to tracking salient points focuses on edges or corners in an image [3,4]. For example, Barnard and Thompson [5] and Shapiro [6] both use the correlation of small patches around corners to track them across frames. This, however, has the drawback that many of the pixels in a patch around a corner lie in the background and as these pixels change from frame to frame, the method can yield incorrect results. Another approach is to create a small feature vector representing the properties of the image at the corner, for example, by using the image brightness, spatial derivatives or coordinates of a point fixed relative to the corner [7]. It is also possible to use the Kalman filter to track corners across the frames of a video [8].

In this paper, we consider two of the more recent definitions of salient points, namely, the Scale Saliency algorithm of Kadir and Brady [9] and the scale-invariant keypoints from Lowe [10, 11]. We next discuss these in further detail.

2.1. Scale-Saliency regions

Scale Saliency is a method for measuring the saliency of image regions and selecting the optimal scales for their analyses. The Scale-Saliency algorithm was proposed by Kadir and Brady [9] who were motivated by earlier work of Gilles [12] which used salient points to match and register two images. Gilles' definition of saliency was based on the local signal complexity; he used the Shannon entropy of local attributes such as the intensity of the pixels in a neighborhood around the current pixel. Image areas with a flatter distribution of pixel intensities have a higher signal complexity and thus a higher entropy. In contrast, a flat image region has a peaked distribution. As Gilles used a fixed size neighborhood, his algorithm selected only those salient points which were appropriate to the size of the neighborhood. As an extension, Gilles proposed the use of a global scale for the entire image, which was automatically selected by searching for peaks in the average global saliency for increasing scales.

Kadir and Brady [9] extended Gilles' algorithm by incorporating a local scale selection procedure and defining salient regions (not points) as a function of the local complexity weighted by a measure of self-similarity over scale space. Thus, the Scale-Saliency algorithm detected regions at multiple scales, locating circular salient patches on the image, where the size of the patch was determined automatically by the multiscale additions to Gilles' algorithm.

The algorithm of Kadir and Brady has a simple implementation. For each point x in the image, we calculate a histogram of the intensities in a circular region of radius (i.e. scale) s. The entropy $E(x, s)$ of each of these histograms is calculated and the local maxima are considered as candidate scales for the region. Specifically,

$$E(x, s) = -\sum_{d \in D} p(d, x, s) \log_2 p(d, x, s) \tag{1}$$

where the entropy E is defined as a function of location x and scale s and $p(d, x, s)$ is the probability of the value d occuring in the region of scale s around the pixel at location x. D is the set of all values that d can take, and assuming a histogram with a bin-width of 1, is $[0, \ldots, 255]$ for the images considered in our application. The set of scales at which the entropy peaks is defined as

$$s_p = \{s : E(x, s-1) < E(x, s) > E(x, s+1)\}. \tag{2}$$

These peaks of the entropy are weighted by the inter-scale saliency measure, which is defined as

$$W(x, s) = \frac{s^2}{2s - 1} \sum_{d \in D} |p(d, x, s) - p(d, x, s-1)| \tag{3}$$

resulting in the following definition of the Scale-Saliency at a point x

$$S(x, s_p) = E(x, s_p) \cdot W(x, s_p). \tag{4}$$

In our work, we used a scale factor of s in Equation (3) instead of $\frac{s^2}{2s-1}$ and square regions instead of circular ones.

Thus the Scale-Saliency algorithm can be summarized as follows:

Scale-Saliency Algorithm: For each pixel location x

- For each scale, s, between s_{min} and s_{max}
  - Obtain the histogram of the pixel intensities in a circular region of radius s centered at the pixel.
  - Calculate the entropy $E(x, s)$ from the histogram using Equation (1).
- Select the scale for which the entropy is peaked using Equation (2). Note that there may be no scales that satisfy this constraint.
- Weight the peaks by the appropriate inter-scale saliency measure from Equation (3).

The top n scale-salient regions, ordered based on their scale-saliency, are used for further processing. Note that since the entropy is calculated for scales ranging from s_{min} to s_{max}, the calculation of the peaks implies that they can occur only between scales $s_{min} + 1$ and $s_{max} - 1$. By their definition, scale-salient regions are invariant to planar rotation, scaling, intensity shifts, and translations. Several variants of the above algorithm are possible. We discuss a few enhancements in Section 4 motivated by the goal of reducing the computation time.

The idea of using scale-salient regions for tracking was proposed in the original paper by Kadir and Brady [9], though no results were presented. More recently, Hare and Lewis [13] have used features extracted from the scale-salient regions to perform a greedy match between the regions in one frame and those in the next. The features include the spatial location, the scale, the saliency, and the normalized intensity histogram of the region. While initial results appear promising, their simple algorithm does not perform as well as other trackers, such as those based on the Shi-Tomasi-Kanade approach [14], which incorporate additional logic to constrain the motion.

2.2. Lowe saliency regions

A recent development that has attracted much attention in the content-based image retrieval community is the work of Lowe on the Scale Invariant Feature Transform [11]. This uses a two-step process - the selection of keypoints or regions and the extraction of features for these regions which are invariant to scale, rotation, and translation. In this paper, we focus on the first of these steps, namely the selection of the regions.

Like Kadir and Brady, Lowe also uses a scale-space approach to the detection of key locations in an image. He considers locations that are maxima or minima of a difference-of-Gaussian function. In his early work [10], Lowe presents an efficient and stable method to find the maxima and minima of this function. The convolution of a 2-dimensional Gaussian function is first implemented as two passes of a 1-dimensional Gaussian function with $\sigma = \sqrt{2}$ using 7 sample points. The resulting image A is again convolved with the Gaussian function with $\sigma = \sqrt{2}$, to give an image B, which is equivalent to the original image convolved with a Gaussian of $\sigma = 2$. The difference-of-Gaussian function is obtained by subtracting image B from image A.

The idea of scale-space is next introduced by considering a pyramid of such difference-of-Gaussian images. The image B is resampled using bilinear interpolation with a pixel spacing of 1.5 in each direction and the process is repeated on this new image. The 1.5 spacing implies that each new sample will be a linear combination of 4 adjacent pixels. Next, maxima and minima of this scale space function are determined. A pixel is retained if it is a maxima or minima when compared with the 8 neighbors at its level and the nine neighbors at each of the levels above and below it.

In his later work [11], Lowe discusses enhancements to the original method that improve the way in which the sampling is done in the image and the scale domains. In addition, instead of just locating the keypoints at the location and scale of the center sample point, further improvements are proposed to more accurately locate the keypoints and remove points that are poorly localized along an edge or have low contrast.

The software incorporating these enhancements, as well as the the second step of extracting features from the keypoint regions, is available in the form of an executable from Lowe's web site [15]. The executable takes

as input an image in the form of a pgm file and returns a file with the list of all keypoints found in the image. Associated with each keypoint is its location (a floating point value as the location is determined to sub-pixel accuracy), the scale, the orientation, and the list of features extracted for the region. For our work, we assume that the salient region is centered at the integer value of the location of the keypoint, ignoring sub-pixel accuracy and regions whose scale is less than 1. The scale of the region is the same as the integer value of the scale of the keypoint i.e. we consider a square region centered at the keypoint with the side of the square equal to twice the scale. We could also have considered a circular region for rotation invariance.

3. TRACKING METHODOLOGY

The output of the Scale-Saliency algorithm and Lowe's algorithm is a set of regions for each frame in the video sequence. These regions are referred to as observations. The goal of the tracking stage is to use these observations to generate the tracks that correspond to the moving objects. Some of the issues that make this a challenging problem include missed observations due to poor video quality, noisy observations due to changes in illumination, occlusions, objects that are moving in close proximity to each other, etc.

The problem of assigning observations to tracks is known as the motion correspondence problem. In this work, we use a graph matching approach to perform this assignment. We form a weighted bipartite graph in which the first set of nodes corresponds to the observations in the current frame and the second set of nodes corresponds to the current tracks. The edges between these two sets of nodes are assigned weights that indicate how well a given observation matches a given track. An optimal assignment is then made using the Hungarian Algorithm [16] which has complexity $O(n^3)$ where n corresponds to the number of nodes.

The similarity between observations and tracks is based on a number of features, namely, the centroid of the observation, the size of the observation, and the average intensity of the observation. These are easily extracted given the masks corresponding to the salient regions and the original video frames. These features are represented in vector form and a weighted Euclidean distance is used to compute the similarity between the values for an observation and the values for a current track. Tracks are assigned the feature values of the last observation to which they were matched; i.e., the feature values are not predicted.

We include a few enhancements to improve the quality of the assignments resulting from the bipartite graph matching. We include a gating function that restricts how far an object can move from one frame to another. Functionally, this removes many of the graph edges and greatly speeds up the Hungarian Algorithm. We also allow tracks to split and merge due to shadows, multiple observations, etc.

4. EXPERIMENTAL RESULTS

In this section, we compare the performance of the two salient regions when used in tracking using motion correspondence. We conduct our study using the "Bright" sequence from the publicly-available urban traffic video from the website maintained by KOGS/IAKS Universitaet Karlsruhe*. This sequence is 1500 frames long and shows a traffic intersection in bright daylight. In our work, we use frames 400-499, which includes some cars moving through the scene without being occluded, some cars going behind occlusions such as trees or lamp-posts, and some cars coming to a stop at a traffic light. Three sample frames (frames 400, 450, and 499) are shown in Figure 1. Each frame is 740×560 pixels. Note that there is dark band on the left side of the sequence - this causes problems with the Scale-Saliency approach and was removed by cropping each frame prior to the application of the algorithm.

4.1. Post-processing of the Salient Regions

The algorithms for both the Scale-Saliency regions and the Lowe's keypoint regions return far more salient regions than can be used in motion correspondence. For example, Fig. 2 displays all the regions for frame 450 for Lowe's algorithm and the top 100 regions for the Scale Saliency algorithm. We used the implementation in Algorithm 1, using $s_{min} = 14$, $s_{max} = 26$, and 256 bins in the histogram. This scale range was selected to approximate the sizes of the vehicles which are the objects of interest in the scene.

*The URL is http://i21www.ira.uka.de/image_sequences. All sequences are copyrighted by H.-H. Nagel of KOGS/IAKS Universitaet Karlsruhe.

(a) Frame 400 (b) Frame 450 (c) Frame 499

Figure 1. *Sample frames from the Bright test sequence.*

We observe that there are several very small Lowe's keypoint regions, many of which, such as the regions around the markings on the road, are not relevant to the task of tracking the moving vehicles. There are also some very large keypoint regions, which are much larger than the size of a vehicle. Hence, we first process the Lowe keypoint regions to keep only those with scale greater than or equal to 3 and less than or equal to 15.

For the Scale-Saliency case, if we consider only the 100 most salient regions, the algorithm does select many of the vehicles, but several of the regions have a large overlap, and some vehicles are missed. Hence, we consider the 200 most salient regions and exploit the overlap to reduce the computational cost (Section 4.2).

After this minor post-processing, we observe that there are still too many salient regions being identified by both Lowe's algorithm and the Kadir and Brady algorithm. Some of these regions are stationary and additional post-processing is used to remove them. First, focusing on the salient regions, we consider the difference in the pixel value relative to the pixel value in the previous frame. If this is greater than or equal to 3, we keep the pixel for further processing. These initial masks for frame 450 are shown in Fig. 3, panels (a) and (c). Next, we use simple morphological operations to remove isolated small regions of size 2-3 pixels and to combine several clumps of small regions. This results in the masks in panels (b) and (d). These masks are then used for extracting the features necessary in tracking using motion correspondence.

4.2. Improving the computational efficiency of Scale-Saliency regions

In our experiments with the two types of salient regions, we observed that while the calculation of the Lowe keypoints was very fast (taking about a second for each frame), the calculation of the Scale-Saliency regions was far more expensive. For example, the processing of a single frame using a histogram of 256 bins, $s_{min} = 14$ and $s_{max} = 26$ took 865 seconds. Since all regions in the frame must be calculated before the most salient ones are selected, this computational time is independent of the number of regions actually used in the application.

We considered several ways to reduce this computational time for the Scale-Saliency algorithm without adversely affecting the results, and even improving them in some cases. These include:

- **Using a spatial stride:** We noticed that even when we considered the 200 most salient regions, the algorithm tended to select regions which had a large overlap with existing regions, ignoring regions in other areas of the image. This is because some areas of the image are strongly salient, and if we consider regions whose centers are only a pixel apart, they will cluster in the strongly salient areas. One solution to this problem is not to consider all pixels for processing. Instead of calculating the scale-saliency for every pixel in the image, we can calculate it for every n-th pixel in the x and y directions. Figure 4 shows the top 100 salient points and the masks generated using the top 200 salient points when we consider a stride of 3, 6, and 10 in the x and y directions. The time for calculating the scale-salient regions goes down from a high of 865s (n=1), to 161s (n=3), 65s (n=6), and 35s (n=10). Note that the accuracy actually improves as some objects such as the pedestrian on the lower left corner are selected when n=10, but not at lower values. From here on, we will consider a spatial stride of 10 pixels in calculating the scale-salient regions.

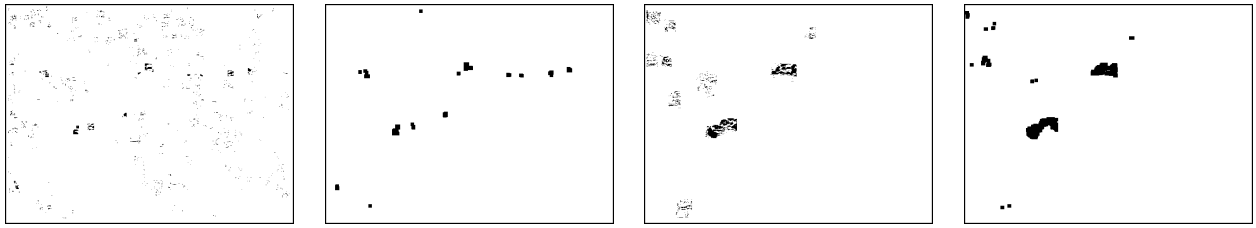
(a) Lowe's keypoints

(b) Top 100 Scale-Saliency regions

Figure 2. *Salient regions (highlighted in white) for frame 450.*

- **Using fewer bins in the histogram:** Another approach to reducing the computation time is to consider fewer than 256 bins in the histogram. Figure 5 shows the results using 20, 100, and 200 bins. A spatial stride of 10 pixels is used in both the x and y directions. The total time for the calculation of the scale-salient regions is 28s (20 bins), 32s (100 bins), 34s (200 bins), and 35s (256 bins). In our experience we found that reducing the bins often gave poorer quality results (e.g. not identifying the motorcycle in the center of the image). We therefore decided that the reduction in time was not worth the loss of key salient regions and therefore used 256 bins in all our experiments.

- **Using a scale stride:** Similar to the idea of using a spatial stride, we can also use a scale stride to calculate the saliency across scales at a given pixel. Figure 6 shows the results using a scale stride of 2, 3, and 4 pixels. A spatial stride of 10 pixels is used in both the x and y directions and 256 bins are used in the histogram. The total time for the calculation of the scale-salient regions is 19s (stride 2), 14s (stride 3), and 11s (stride 4), in comparison with 35s (stride 1). We consider the effect of this use of a stride in



(a) Lowe's keypoints (b) Lowe's keypoints (c) Scale-Saliency (d) Scale-Saliency

Figure 3. *Post-processing of the salient regions for frame 450. (a) and (c) Masks after removal of pixels in the salient regions that do not differ from the pixel in the previous frame by more than 2. (b) and (d) Masks after further processing using morphological operations to remove small regions and fill holes.*

the scale on the tracking in Section 4.3.

- **Using a narrow range of scales:** To reduce computational costs but get accurate results, we need to select the range of scales carefully. If s_{min} is too small, the algorithm selects only parts of a vehicle instead of the whole vehicle. On the other hand, if s_{max} is too large, the computational costs can be high. For our experiments, we selected $s_{min} = 14$ and $s_{max} = 26$.
- **Using the variance instead of the entropy:** We also tried using the variance instead of the entropy. However, the results were not accurate enough as several of the cars were not selected as scale-salient.

4.3. Results of tracking

Figure 7 shows the tracking results at frame 470. We focus on five of the moving objects: two pedestrians in the top left that walk from the middle of the street to the sidewalk; two cars just below that come to a stop at the traffic light; a pedestrian in the lower left; a dark car that moves from the center to the left edge of the frame, disappearing behind a tree; and a white van that follows a curved path in the upper portion of the frame.

Results are shown for three salient region detection schemes: Lowe's keypoints; Scale-Saliency with a scale stride of 1; and Scale-Saliency with a scale stride of 4. We also consider two tracking methodologies, one where only the observation centroid is used in solving the motion correspondence problem, and another where the observation centroid *and the average intensity as computed using the original video frames* are used. Figure 7, panels (a) and (b) show the results for Lowe's keypoints with and without incorporating intensity, respectively. Figure 7, panels (c) and (d) show the results for Scale-Saliency with scale stride 1 with and without incorporating intensity, respectively. And, Figure 7, panels (e) and (f) show the results for Scale-Saliency with scale stride 4 with and without incorporating intensity, respectively.

These results demonstrate that salient regions obtained using the Lowe keypoints algorithm and the Scale-Saliency algorithm can be used to successfully track moving vehicles, and potentially even smaller objects such as pedestrians, in moderate resolution video. We make the following specific observations based on these results:

- Lowe's keypoints result in jagged tracks. Spurious keypoints displace the observation centroids and prevent the tracks from being smooth.
- Lowe's keypoints often result in multiple tracks per object. This is not necessarily a problem but would need to be addressed in a post-processing step if, for example, knowing the exact number of vehicle was part of the analysis.
- Lowe's keypoints have difficulty tracking through occlusions e.g. when the white van goes behind a pole.
- Lowe's keypoints fail to detect the pedestrians in the top left.
- Scale-Saliency results in smoother tracks, especially when the scale stride is 1.
- Increasing the scale stride from 1 to 4 for Scale-Saliency degrades performance. Tracks are more likely to be segmented, possibly due to missed observations.

- Using intensity to help solve the motion correspondence problem improves the tracking results for all three detection schemes. The tracks tend to be less segmented (compare the tracks for the car in the middle of the frame in figures 7, panels (c) and (d)).

5. CONCLUSIONS

Our early work in the use of Lowe's keypoints algorithm and the Kadir and Brady Scale-Saliency algorithm shows that they can be used in tracking moving objects in video using motion correspondence. The Scale-Saliency approach, though somewhat more expensive, is a little better as it results in smoother, more complete tracks. We tried several options to reduce the computational time, such as increasing the spatial stride. This led to improved results, as well as a substantial reduction in compute time (from 865s for stride 1 to 35s for stride 10). Increasing the scale stride from 1 to 4, which results in a further reduction of computation time (to 11s), however yielded poorer results.

We plan to extend this work in several ways. These include the use of circular regions to support rotational invariance, improved post-processing of the regions to localize the vehicles better, and improved logic to handle the tracking of vehicles, especially as they move behind occlusions. We also plan to pursure additional ways of reducing the time for the Scale-Saliency regions, such as using a scale stride of 2, which reduces the time somewhat, but may give comparable results to a scale stride of 1.

ACKNOWLEDGMENTS

We would like to acknowledge discussions with Samson Cheung during the early stages of this work. Parts of this work were done by George Marlon Roberts while a student intern at Lawrence Livermore National Laboratory.

UCRL-CONF-208738 This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

REFERENCES

1. S.-C. Cheung and C. Kamath, "Robust background subtraction with foreground validation for urban traffic video," *EURASIP Journal on Applied Signal Processing* . To appear.
2. C. Schmid and R. Mohr, "Local gray-value invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(5), pp. 530–534, 1997.
3. S. Smith, "Reviews of optic flow, motion segmentation, edge finding, and corner finding," Technical report TR97SMS1, Oxford University, Oxford, 1997. `http://www.fmrib.ox.ac.uk/~steve`.
4. S. S. Mohith, "Real-time interactive object tracking," Master's thesis, University of Manchester Institute of Science and Technology, Manchester, U.K., 1998. `http://smohith.tripod.com/proj/riot/thesis/riot.html`.
5. S. T. Barnard and C. M. Brown, "Disparity analysis of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2**(4), pp. 333–340, 1980.
6. L. Shapiro, H. Wang, and J. Brady, "A matching and tracking strategy for independently-moving, non-rigid objects," in *Proceedings, Second British Machine Vision Conference*, pp. 306–315, 1992.
7. S. Smith, "ASSET-2: Real-time motion segmentation and object tracking," Technical report TR95SMS2, Defence Research Agency, Surry, UK, 1995. `http://www.fmrib.ox.ac.uk/~steve`.
8. D. Charnley and R. J. Blissett, "Surface reconstruction from outdoor image sequences," *Image and Vision Computing* **7**(1), pp. 10–16, 1989.
9. T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision* **45**(2), pp. 83–105, 2001.
10. D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings, International Conference on Computer Vision*, pp. 1150–1157, September 1999.
11. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* **60**(2), pp. 91–110, 2004.
12. S. Gilles, *Robust matching and description of images*. PhD thesis, Oxford University, Oxford, U.K., 1998.
13. J. S. Hare and P. H. Lewis, "Scale saliency: Applications in visual matching, tracking and view-based object recognition," in *Proceedings, Distributed Multimedia Systems 2003 / Visual Information Systems 2003*, pp. 436–440, 2003.
14. J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, (Seattle), June 1994.
15. D. G. Lowe, "Demo software: SIFT keypoint detector," 2004. `http://www.cs.ubc.ca/~lowe/keypoints/`.
16. D. Jungnickel, *Graphs, Networks and Algorithms*, Springer-Verlag, 1999.

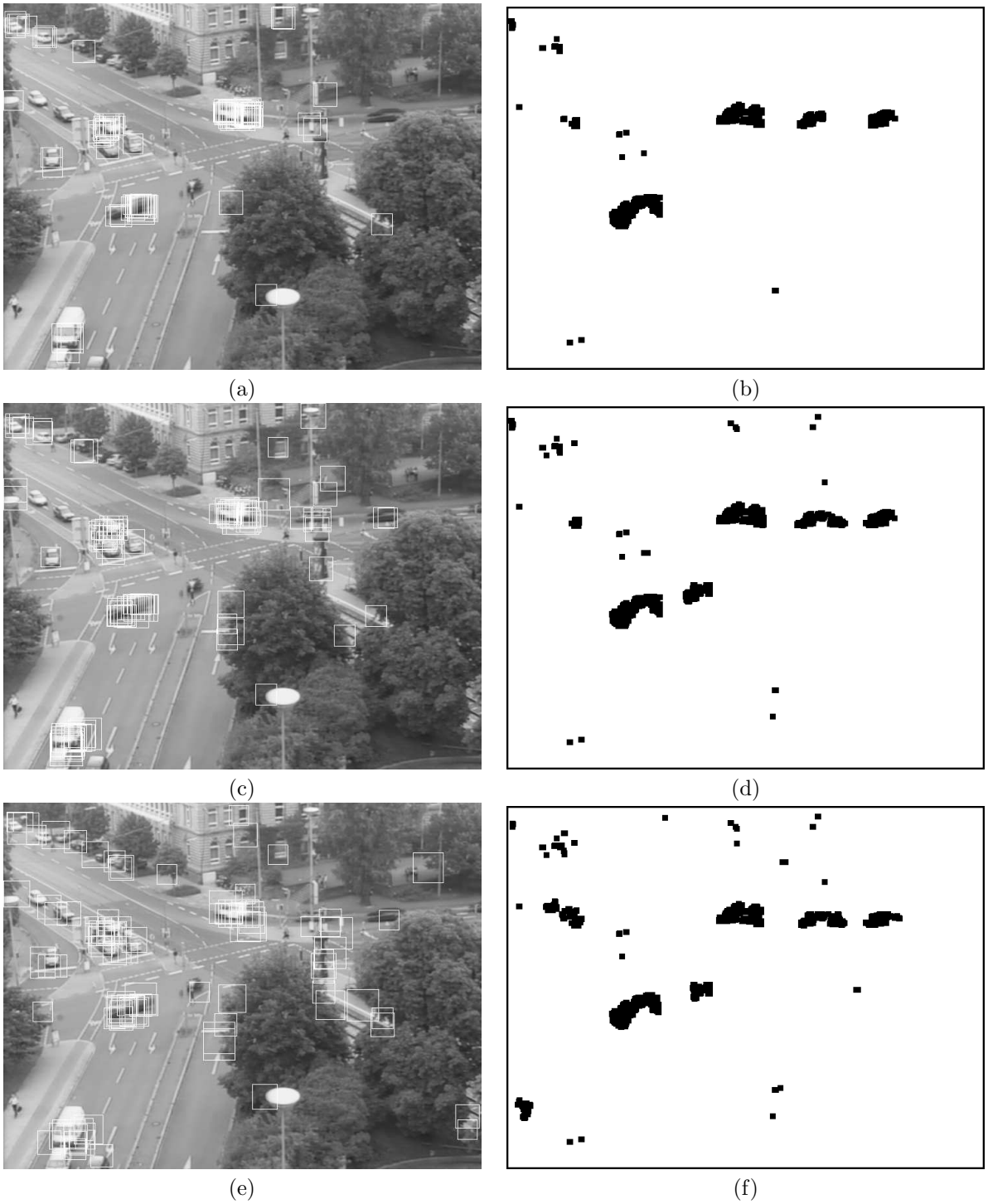


Figure 4. *Calculating the salient points by considering every n-th pixel in the image. The left column shows the top 100 scale-salient regions and the right column shows the corresponding masks calculated using the top 200 scale-salient regions. (a) and (b) Results for a stride of n=3. (c) and (d) Results for a stride of n=6. (e) and (f) Results for a stride of n=10.*

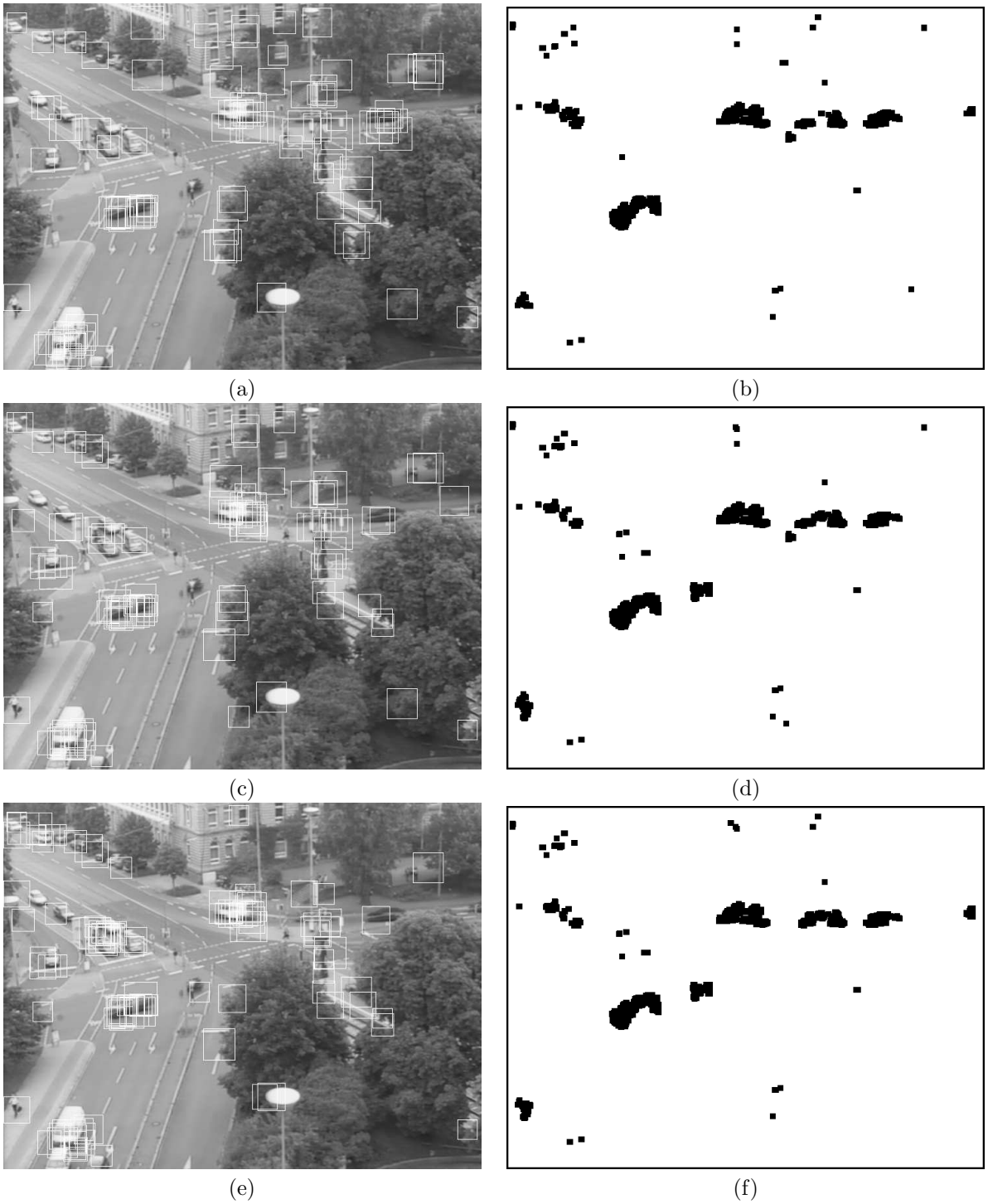


Figure 5. *Calculating the salient points by considering fewer bins in the histogram. The left column shows the top 100 scale-salient regions and the right column shows the corresponding masks calculated using the top 200 scale-salient regions. (a) and (b) Results for 20 bins. (c) and (d) Results for 100 bins. (e) and (f) Results for 200 bins.*

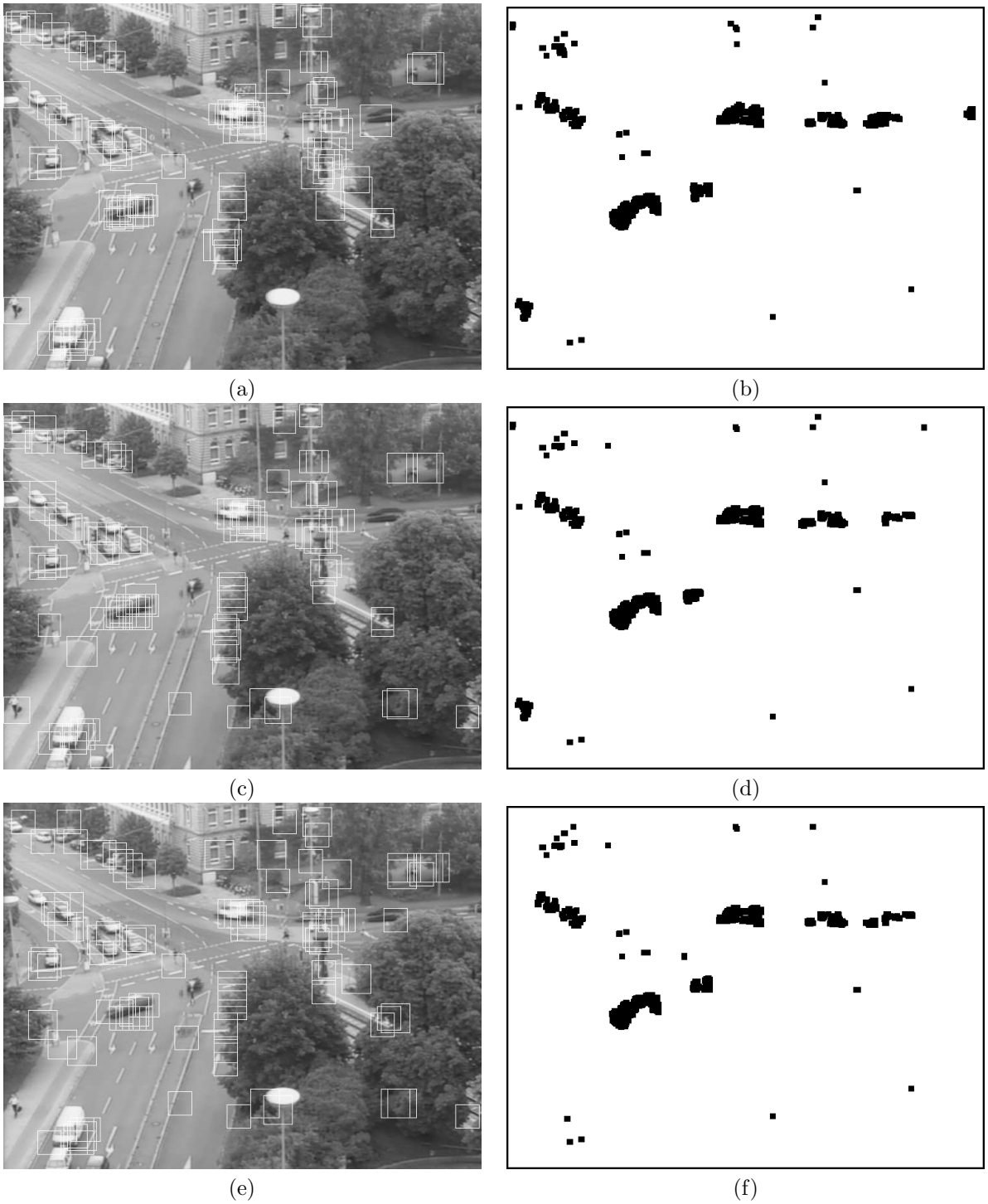


Figure 6. *Calculating the salient points by considering a stride for the scale range. The left column shows the top 100 scale-salient regions and the right column shows the corresponding masks calculated using the top 200 scale-salient regions. (a) and (b) Results for a stride of n=2. (c) and (d) Results for a stride of n=3. (e) and (f) Results for a stride of n=4.*

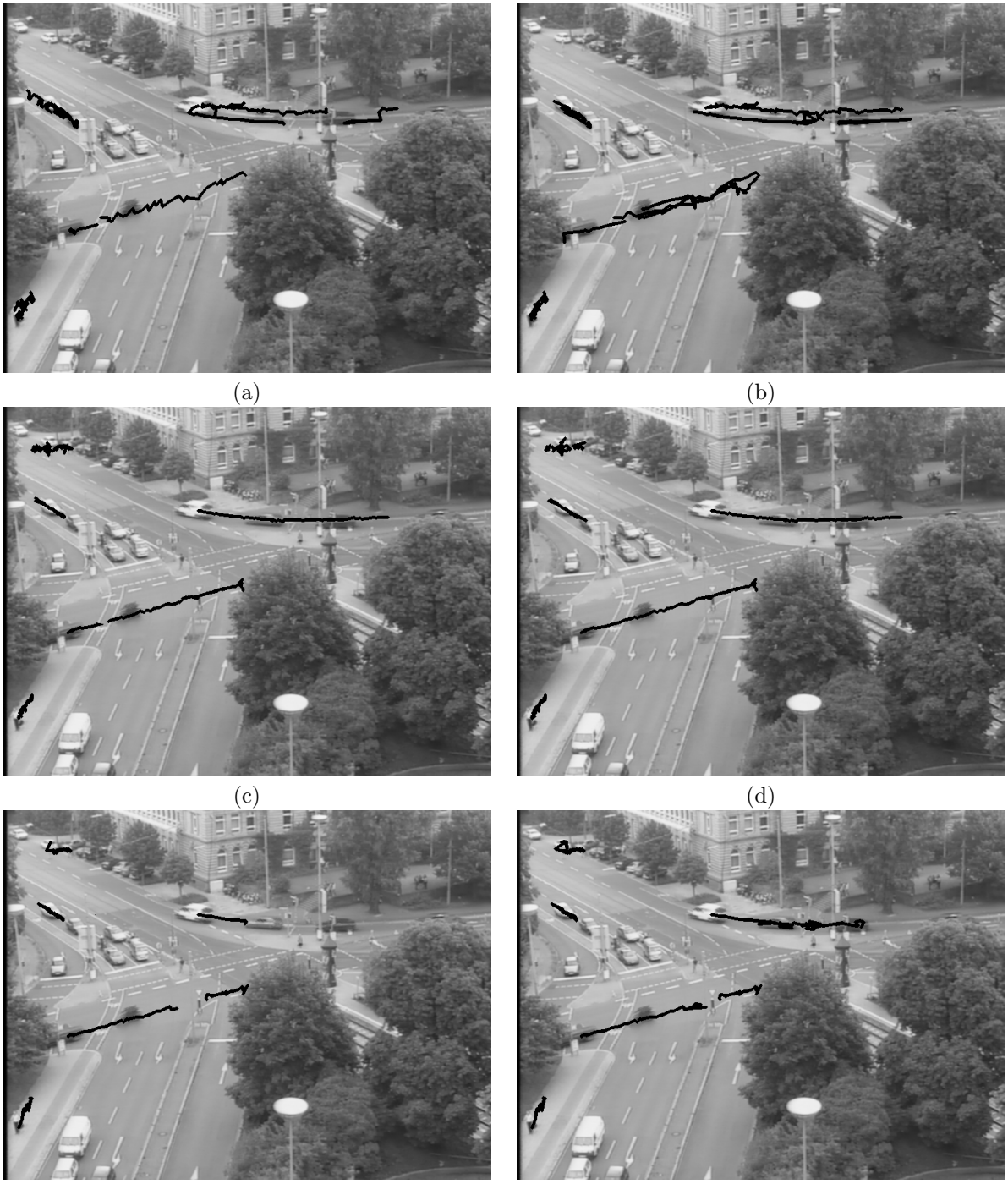


Figure 7. *Result of tracking at frame 470, showing tracks for the following five objects: two pedestrians at the top left; two cars at the top left; a pedestrian in the lower left; a dark car that moves from the center to the left; and a white van that follows a curved path in the upper portion of the frame. (a) and (b) Lowe regions with and without intensity features. (c) and (d) Scale-Saliency, scale stride 1, with and without intensity features. (e) and (f) Scale-Saliency, scale stride 4, with and without intensity features.*