Final Report 2000-08

Research

Bicycle Counter

Minnesota Guidestar

Technical Report Documentation Page

1. Report No.	2.	3. Recipient's Accession No.
MN/RC - 2000-08		

4. Title and Subtitle	5. Report Date
BICYCLE COUNTER	March 2000
	6.
7. Author(s) Scott Rogers Nikolaos P. Papanikolopoulos	8. Performing Organization Report No.
9. Performing Organization Name and Address University of Minnesota Department of Computer Science Robotics and Vision Laboratory 200 Union Street, S.E. Minneapolis, MN 55455	10. Project/Task/Work Unit No. 11. Contract (C) or Grant (G) No. c) 74708 wo) 74
12. Sponsoring Organization Name and Address Minnesota Department of Transportation 395 John Ireland Boulevard St. Paul Minnesota, 55155	13. Type of Report and Period Covered Final: Report 1997 - 1999 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract (Limit: 200 words)

This report describes a system for monitoring bicycle activity in sequences of gray scale images from a stationary camera. Applications for such a system include determining the use and congestion of bicycle paths.

The output of the system is a count of the number of bicycles detected in the image sequenc4e. The system uses a simple model of two circular objects separated by relatively known distance, with four levels of abstraction: raw images, blobs, edge images, and the bicycle model.

The system was implement4ed on a dual Pentium computer equipped with a Matrox imaging board and achieved a peak performance of eight frames per second. Experimental results based on outdoor scenes show promising results for a variety of weather conditions

17. Document Analysis/Descriptors		18. Availability Statement
Bicycle counting	Cameras	No restrictions Document available from:
Object tracking	Detection	National Technical Information Services,
		Springfield, Virginia 22161

19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	25	

BICYCLE COUNTER

Final Report

Prepared by:

Scott Rogers
Nikolaos P. Papanikolopoulos

Artificial Intelligence, Robotics and Vision Laboratory
Department of Computer Science and Engineering
University of Minnesota

March 2000

Published by:

Minnesota Department of Transportation
Office of Research & Strategic Services
Mail Stop 330
395 John Ireland Boulevard
St. Paul, MN 55155

Executive Summary

This report describes a system for monitoring bicycle activity in sequences of gray scale images acquired by a camera. The system is suitable for use in applications that aim to increase the efficiency and safety of existing traffic systems. One such application is to determine usage and congestion of multi-use trails. The output of the system is a count of the number of bicycles detected in the image sequence. The system is model-based in the sense that it uses a simple bicycle model of two circular objects separated by a relatively known distance. Based on this model, bicycles are identified in the image. The system uses four levels of abstraction: raw images, blobs (image regions), edge images, and the bicycle model. Raw images are processed on a PC computing platform and regions are extracted. Regions are areas in the image which are separate from the background. Another level consists of edge images where all the edges of a specific image are highlighted.

The proposed system has the ability not only to detect bicycles but also to detect, track, and classify vehicles, pedestrians, rollerbladers, and other traffic objects. This efficiency is achieved by a large collection of individual algorithms and models tuned to the shape characteristics of each traffic object.

The system was implemented on a dual Pentium computer equipped with a Matrox imaging board and achieved a peak performance of 8 frames per second. Experimental results based on outdoor scenes have shown promising results for a variety of weather conditions. The accuracy of the bicycle counting was close to 70% (the ground-truth was derived using manual counts). The overall objective is to have a bicycle counting system that will eventually cost around two thousand dollars.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

OVERVIEW

Modern social trends such as urbanization and migration, rapid growth of suburbs, construction of high-quality highways etc., significantly increase the complexity, cost, and congestion of transportation systems. To address the issues of cost and congestion, several transportation agencies promote the use of bicycles as an every day transportation alternative (the emphasis these days is on multi-modal travel options). Bicycles are also promoted as a good way to exercise, and funds have been spent on the construction of multi-use trails. However, there is little information and few measurements on how many bicycles use multi-use trails or how many bicycles use public streets.

There are many methods available for obtaining information regarding bicycle usage. Sensors of many types can be employed such as: loop detectors, laser triggers, camera-based detection, manual counting, etc. Cameras often provide information that is richer, more complete, and covering a larger area than other sensor-based counting methods. Cameras are also non-intrusive, and generally more mobile and less expensive to setup than other sensors. We have developed a system that can reliably count (70% accuracy) the number of bicycles using these trails (Figure 1).

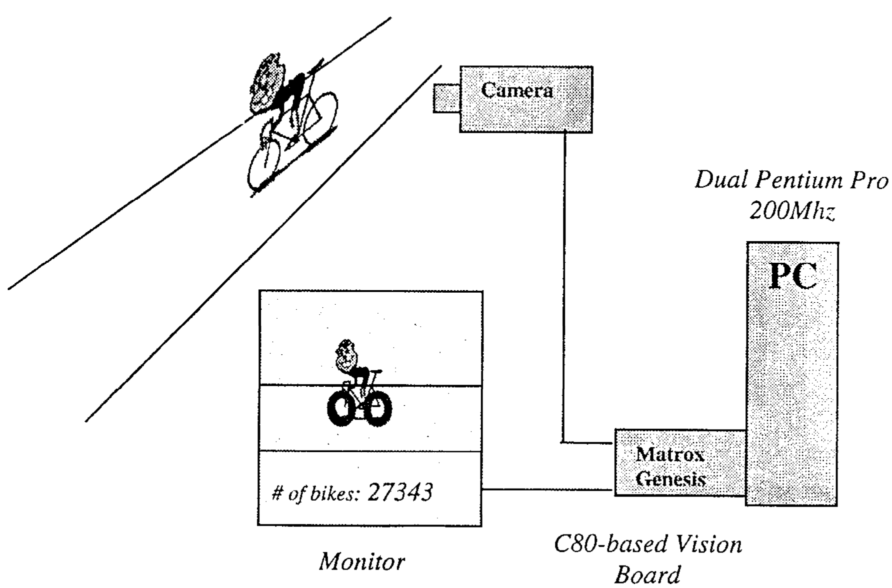


Figure 1: System design.

Extensive literature search showed that a fully automated, portable, and reliable bicycle counter is not commercially available. The proposed system is based on a portable time-lapse video system. This device is currently used to acquire video segments (in increments of variable size) which are later analyzed by a real-time imaging system. The output of the imaging system is the number of bicycles detected. It is desirable to have a system that differentiates bicycles, rollerbladers, and pedestrians. We believe that modern imaging has achieved the level of sophistication that allows successful completion of this task.

There are several challenges to overcome in using a vision (or camera) based counting system. First, there is no inherent way of determining one object from another, or any object from the background. Second, to compute a model that can uniquely identify a bicycle from a cluttered outdoor environment, such things as lighting, partial or complete occlusions, and shadows must be considered. Third, we need to address issues such as: computational performance of the initial image processing steps, and efficient and speedy recognition of the bicycle's characteristics in order to do the bicycle detection in real-time.

Our system extends the work of Masoud and Papanikolopoulos [1] in tracking generic objects. In particular, it detects if the object being tracked is indeed a bicycle. We are thus able to not only count the number of bicycles in each scene but also track each bicycle. In addition, the proposed system can be used to detect, track, and classify pedestrians, vehicles, rollerbladers, and other traffic objects of interest. The tracking adds flexibility to the types of applications for which this counter might be used. Thus, this report goes beyond merely counting bicycles but also tracking them.

An overview of the complete system is given in this report covering both hardware and software used, including a brief overview of how the generic object tracking works, and a detailed explanation of how it has been extended to count and track bicycles exclusively.

PREVIOUS WORK

A great deal of work has gone into bicycle detection and counting, especially in traffic situations. The vast majority of that work has dealt with sensors and triggers as opposed to image recognition. These sensory triggers offer robust accuracy for a small area. However, their use is limited.

One example of such a device is the inductive loop sensors such as provided by Traffic-2000 Ltd. These inductive loops are placed in the pavement under the places in which bicycles are expected to be detected. It takes several days of construction to place these sensors and once in place they are there for life, and once they fail it is again an expensive and time-consuming effort to replace them. In addition their effectiveness is reduced in large areas because it is possible for bikes to inadvertently go around the sensors and thus escape detection. Their advantages are that they work consistently regardless of lighting and weather conditions. Video imaging offers the advantage of being more flexible in the range of area that it can cover as well as being less permanent giving the ability to survey many different areas of concern, using the same equipment.

Our previous vision-based methods for counting bicycles had focused primarily on distinguishing bicycles from the background. By performing computationally intensive operations on the entire image, it was difficult to distinguish from one frame to the next if the bicycle detected is the same bicycle or a new bike to be counted. This new approach adds robustness by counting each bike only once and also makes it possible for detection of bicycles at more difficult angles (from the perspective of the camera) than possible before. A more comprehensive comparison is given in each section as it relates to previous work done.

MOTIVATION AND APPROACH

Our system uses a single fixed camera whose optical axis is roughly perpendicular to the bicycle path (optimal configuration). However, the proposed system can still track bicycles even if there is a deviation of +/-30° from the optimal configuration (Figure 2). We use simple rectangular patches with certain dynamic behavior for the tracking of generic objects. Overlaps and occlusions are dealt with at this generic tracking level. The cues that there are any objects to be tracked are obtained by thresholding the result of subtracting the image from the background. Once we have objects that are being tracked, these objects are tested to determine if they are bicycles or other types of objects. This testing is done by using an edge image of the area where the object is being tracked. The Hough transform is then applied to this edge image to determine the existence and location of bicycle wheels in the image, which is then used to determine if the object being tracked is indeed a bicycle.

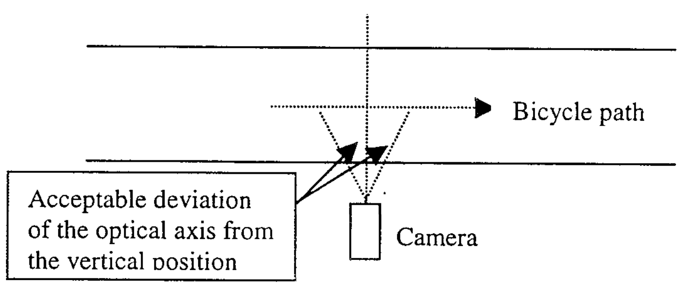


Figure 2: Top view of the experimental setup.

In our initial work in bicycle counting, we were getting an edge image from a larger image and sorting out the possible wheels and from that the number of bikes. In each of the images local to the object being tracked, we are able to localize the search for exactly two wheels (we thus exclude unicycles, which were in fact excluded in our previous work as well).

Since the motion of the object is determined before the object is determined to be a bicycle or not, we can also modify our Hough transform to look for ellipses based on the orientation of the motion. This makes it possible to more accurately detect bicycles traveling at an angle to the camera. While further work is required to implement and test this step extensively, it is important to mention that we got very promising initial results.

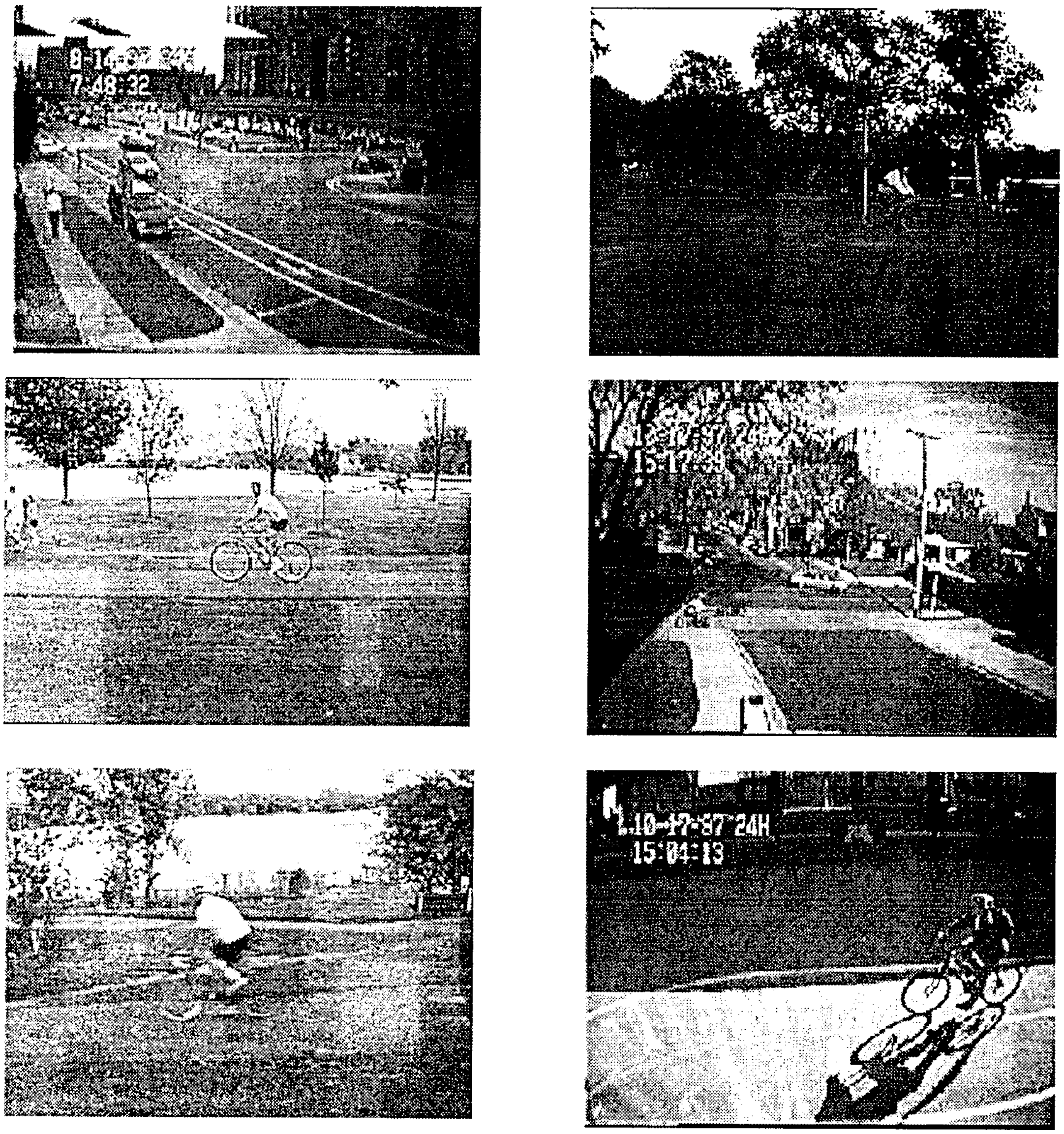


Figure 3: Input images from different scenarios.

CHAPTER 2

GENERIC OBJECT TRACKING

GENERIC BLOB ANALYSIS

We present an overview of the novel approach presented by Masoud and Papanikolopoulos [1] to track blobs (regions) in the image regardless of what they represent. The objective is that by tracking the blobs, we can track objects such as bicycles. In this case, bicycles will consist of several blobs. The tracking scheme attempts to describe changes in the difference image in terms of motion of blobs and by allowing blobs to merge, split, appear, and vanish. A diagram of our approach is shown in Figure 4.

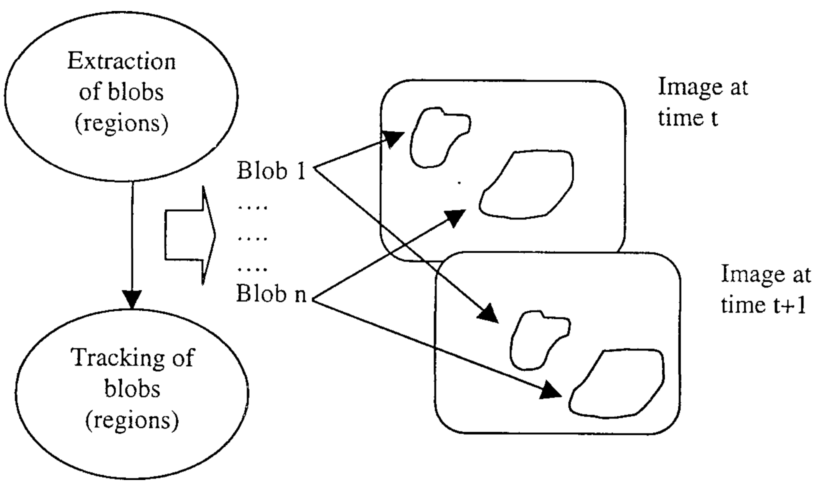


Figure 4: Overview of the tracking system.

Blobs can be extracted from the image using simple blob extraction techniques. The extracted blobs correspond to different traffic objects in the image. This section in the report describes how we move from the extraction of blobs to the counting and tracking of bicycles. By extracting and following the blobs, we can locally search for shape characteristics that correspond to bicycles (circles/wheels) in each image frame.

Whenever, a characteristic that corresponds to a bicycle is detected, we try to look at additional image parameters which indicate the existence of a bicycle.

BLOB EXTRACTION

Once a feature image is computed, connected segments are extracted efficiently using border following. Border following has the advantage of being more efficient then other methods such as raster scanning. While raster scan algorithm has to traverse every pixel in the image, with border following, the interior of blobs does not need to be considered. Thus, the larger the total area of all the blobs in the image, the faster the segmentation process becomes.

BLOB TRACKING

When a new set of blobs is computed for a given image frame, an association with the previous frame's set of blobs is sought. Ideally, this association can be an unrestricted relation. With each new image frame, blobs can split, merge, appear or disappear. The relation among blobs can be represented by an undirected bipartite graph, with one frame on one side and the next frame on the other. We will refer to this graph as a *blob graph*. Since there is a one-to-one correspondence between the blobs in first frame and the vertices representing them in the blob graph, we will use the terms blob and vertex interchangeably. The process of blob tracking is then equivalent to computing the *blob graph*. To find the optimum correspondence between blobs in one frame and those in the next, we need to define a cost function so different graphs can be compared. A graph with no edges, is one extreme solution in which all blobs in the first frame disappear and all blobs in the second frame appear. This solution has no association among blobs and therefore has a high cost. In order to proceed with our formulation of the cost function we penalizes graphs in which blobs change significantly in size. A perfect match would be one in which blob sizes remain constant (e.g., the size of a blob that splits equals to the sum of the sizes of blobs it split into). Using this cost function, we can proceed to compute the optimum graph.

Our algorithm to compute the optimum graph works as follows: A graph is constructed such that the addition of any edge makes it violate the locality constraint. There can be only one such graph. Note that the graph may violate the parent structure constraints at this moment. The next step in our algorithm systematically eliminates just enough edges from it to make it satisfy the parent structure constraint. The resulting graph is valid and also dense. The process is repeated so that all possible dense graphs are generated. The optimum graph is the one with the minimum cost. By systematically eliminating edges, we are effectively enumerating valid graphs. The computational complexity of this step is highly dependent on the graph being considered. If the graph already satisfies the parent structure constraint, it is linear. On the other hand, if we have a fully connected graph, the complexity is exponential in the number of vertices (bounded by 2^{mn} where m is the number of blobs in the first frame and n is the number of blobs in the second frame). Fortunately, because of the locality constraint and the high frame rate, the majority of graphs considered already satisfy the parent structure constrained. Occasionally, a small cluster of the graph may not satisfy the parent structure constraint and the algorithm will need to enumerate a few graphs. In practice, the algorithm never took more than a few milliseconds to execute even in the most cluttered scenes. Other techniques to find the optimum (or near optimum) graph (e.g., stochastic relaxation using simulated annealing) can also be used. The main concern, however, would be their efficiency, which may not be appropriate for this real-time application due to their iterative nature.

At the end of this stage, we use a simple method to calculate the velocity of each blob, based on the velocities of the blobs at the previous stage and the computed blob graph. The blob velocity will be used to initialize generic models. If a blob is the outcome of a splitting operation, it will be assigned the same velocity as the parent blob. If it is the outcome of a merging operation, it will be assigned the velocity of the biggest child blob. If it is a new blob, it will be assigned zero velocity. Finally, if there is only one blob, related to it, the velocity is computed as the displacement of the center of weight of the blob from one frame to the next and compared with its previous velocity.

CHAPTER 3
BICYCLE DETECTION

EDGE DETECTION

The raw images captured by the camera (Figure 5) pass through the Sobel edge detection algorithm (Figure 6). In previous work that we have done, the whole image was passed through the edge detection. However, since we have specific locations of all of the objects of interest (objects not part of the background) we localize the edge detection to a specific area right around each object of interest (Figures 8 and 9).

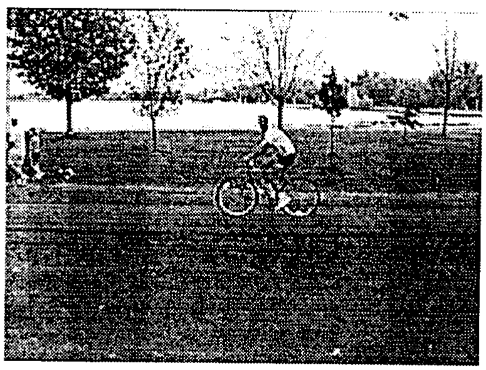


Figure 5: Raw image.

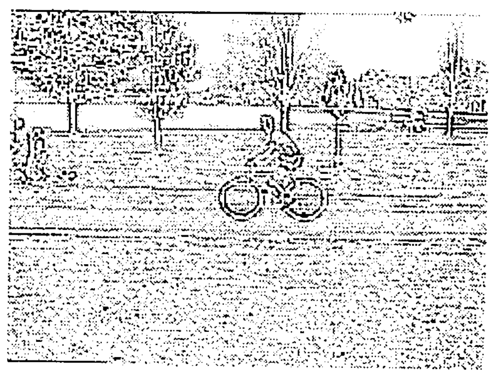


Figure 6: Edge image.

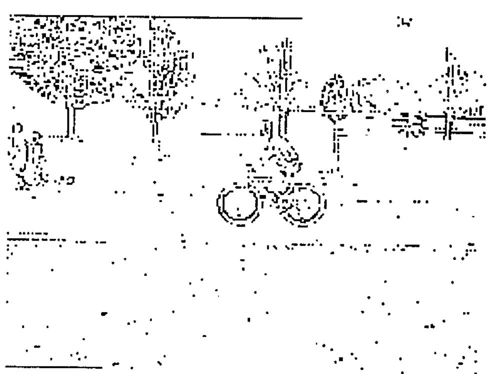


Figure 7: Thresholded edge image.

It is important that we not only determine the position of the edges but also the direction -- this information will simplify and speed up the Hough transform used in the next stage for detecting circles. The edge detection is then broken down into horizontal and vertical edge detection where the value of the directional gradient at each pixel of the raw image replaces the image value.

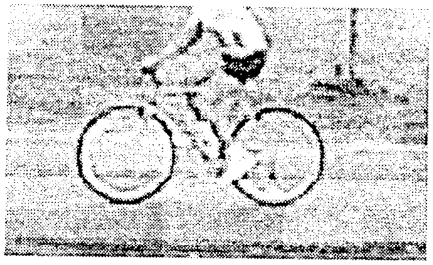


Figure 8: Raw subimage.

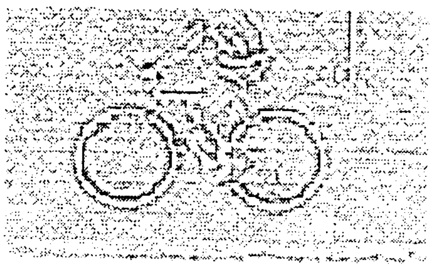


Figure 9: Edge subimage.

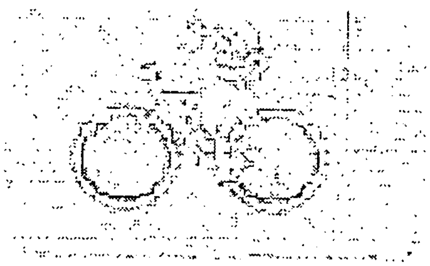


Figure 10: Thresholded edge subimage.

Thresholding (Figures 7 and 10) is necessary to determine which values represent strict edges and which represent gradual changes in the image, since only the strict edges need to be considered as bicycle wheels. It is possible that the wheels of the bicycle are very similar in color to the background behind it, thus making it difficult to extract the edges of the wheel. However, we have found that the majority of the bicycle wheels differ distinctly from the rims in color, making it much easier to avoid such difficulties.

Lighting conditions can also play a role in how difficult it is to separate edges from a raw image. There is no perfect solution to this problem but several researchers have attempted to overcome some of its aspects. For example, some research groups have used adaptive background update to capture new global changes in the image sequence such as lighting conditions [2]. Noise reduction is usually taken care of by using a low-pass filter on the entire image before the Sobel is applied.

CIRCLE EXTRACTION

In our previous work, our first, fundamental goal was to identify unobstructed bicycles in each frame [3]. The assumptions are: 1) the camera position should be close to be perpendicular (or more accurately the algorithm can work with angle deviations of ±30° from the optical axis of the camera) to the bicycle path, 2) the traffic should be light to moderate, and 3) the occlusions should be minimal. The algorithm now first calculates positions and velocities and then determines the type of the object. Our goal here is merely to determine what the object is. And since the velocity of the object is computed, the orientation of the bicycle can be determined from this velocity. Thus, we can relax the assumptions about the orientation of the camera.

While there are many ways to visually identify a bicycle, for our first experiment the identification of their wheels is chosen. This approach has the advantage of being computationally simpler than a stricter modeling technique. For example, consider the fact that bicycles are made in a wide variety of shapes, sizes, and colors. In contrast, bicycle tires and wheels are made in a limited range of standard sizes, of colors, and most importantly they have a single, circular shape. Therefore, rather than develop a complex model-based system, we attempt to identify circles whose size corresponds to the average bicycle wheel and tire.

A version of the Hough transform [4] is used in order to detect circles whose sizes correspond to a specific range in diameter (Figure 11). Processing begins with the set of points that form edges determined in the previous step. The magnitude and direction of each point is derived from the edge detection. For each point, a line segment equivalent to the length of the radius of a bicycle tire is projected perpendicularly to the edge's direction. The point at which this segment ends receives an increment by one vote. The process continues for the rest of the edge points and the output matrix accumulates all votes. These votes represent the likelihood of a point in the image as being the center of a wheel. A proposed center point may not be singular due to mechanical error or the camera not being exactly perpendicular to the wheels, but it does exist as a region. Therefore, process continues with blob detection whereby each blob of

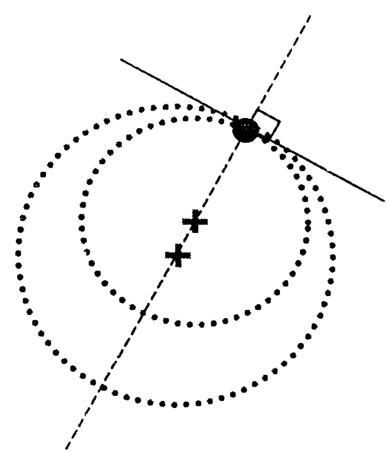
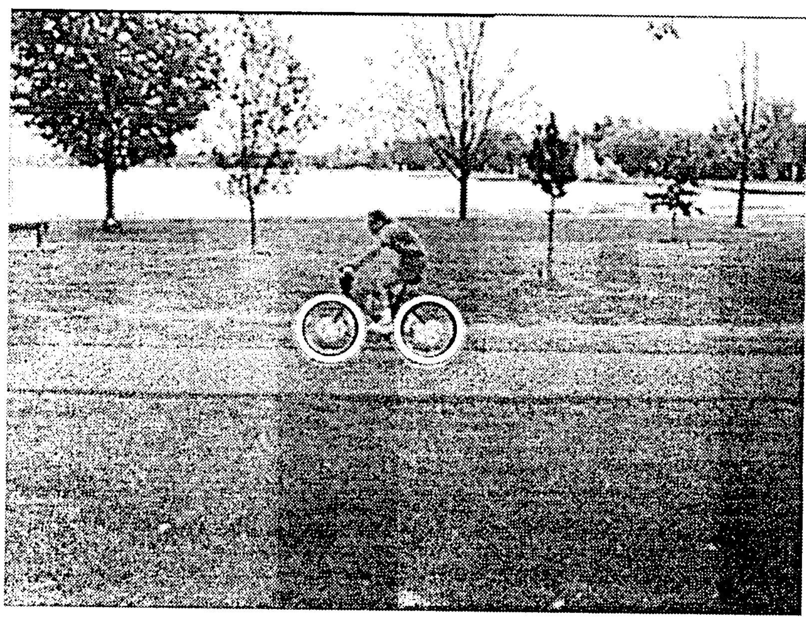


Figure 11: Example of voting for Hough transform.

significant size or weight (as determined by the number of votes in that region) represents the center-point of a circle.

Previously, the Hough transform was used in the following way: each point in the edge image may vote for several points each corresponding to different radius within a range of likely radii for bicycle tires. To then better determine the major contributing radius for each wheel, the center point of the blob obtained for each possible circle is used. Then, by employing the edge image, the algorithm decides which radius from that center point will give a circle that hits the most points in the edge image. From this, a center point and a radius for each circle in the image are obtained. See Figure 13 for an example image and the resulting blob and radius pairs. Currently, since the approximate size, orientation, and position are known, the proposed algorithm computes if the object is indeed a bicycle merely by determining if there are two circles (or ovals) in positions relative to the object that would suggest that it is indeed a bicycle.

Figure 12: Example of Hough blobs and pairing compared with positional blob finding.



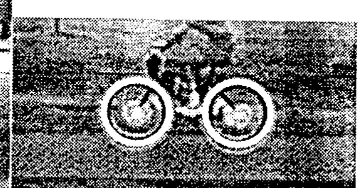


Figure 14: Example of focused search area and Hough blobs.

Figure 13: Example image with Hough blobs and radius pairs.

HAUSDORFF METRIC

The Hausdorff metric [5] has been employed for model matching. The Hausdorff metric is a method of comparing edges of one image with the edges of another to determine the difference between the two. If the difference is small, then there is a high likelihood of the objects being the same. This can be applied to the frame of the bike, and though there are still some occlusions to the frame of the bike (i.e. the leg of the rider, carriers on the back of the bike, etc.), the overall shape of the bike is not dramatically changed. In Figure 17, the Hausdorff metric is applied to the bike frame (Figure 15) and the edge image (Figure 16).

Figure 15: Bicycle model used in Hausdorff.

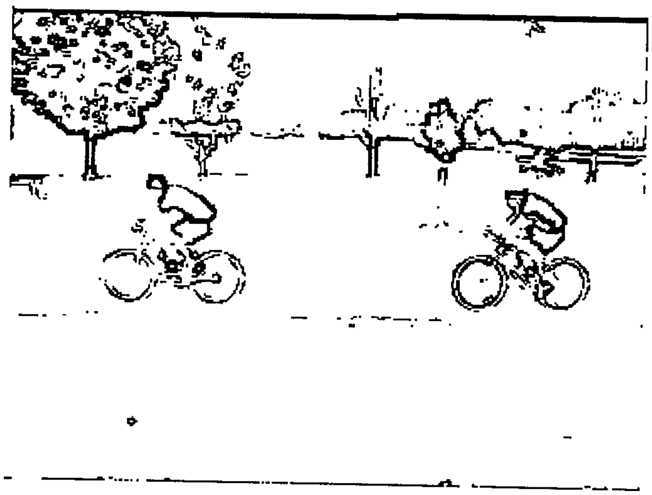


Figure 16: Sobel edge image searched for model matches.

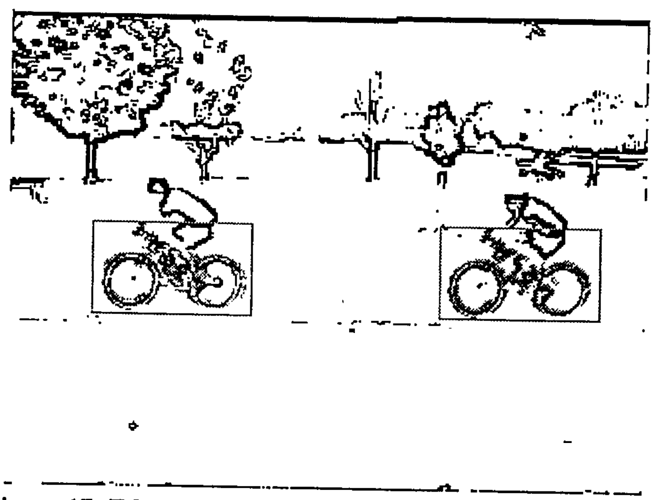


Figure 17: Edge image with highlighted Hausdorff matches.

In implementing the Hausdorff however, it was found that it took several minutes to look for a match to a model in just one frame. Another difficulty was to find a good model that would match with different bicycles consistently (this problem still has not yet been solved.) When a generic model created from basic shapes was used, the edges of the bicycle in the image did not correspond accurately to provide good detection. When a sub-sample image of bicycles from one frame was employed, we got very consistent detection, but other bikes might or might not match within the given tolerance. When i the tolerance was increased (making the detection less stringent), dubious results were obtained.

It is clear from this experience that even by limiting the area in which the Hausdorff metric is used, this method is still too computationally heavy to make the Hausdorff metric suitable for the bicycle detection task.

BICYCLE RECOGNITION

In order to use the generic object tracking, the camera tilt, the camera height, and an approximate size of the objects to track need to be provided. The tilt and height are with respect to the earth. In addition, the size is a 2-dimensional size as well as a coverage criterion (how much of the size is covered by the blob/s). And from the generic tracking we obtain the location of all the objects being tracked and the velocity (direction and magnitude) of the object. From the direction of velocity and the camera tilt we can directly determine the orientation of the bike with respect to the camera. From the position and orientation, we can determine approximately where the wheels should be if it were a bicycle. Thus, we conduct the edge detection on the original image at the location of the potential bicycle, we then perform the Hough transform, and look for peek votes at the desired locations indicating the presence of circular objects there. This test is performed several times and if successful at least once it is determined that the object is indeed a bicycle.

CHAPTER 4

HARDWARE FOR VISION-BASED DETECTION

EQUIPMENT SETUP

The initial hardware system used for experimentation is a JVC color video camera, a Panasonic VHS video cassette recorder, and a dual Pentium Pro 200 MHz personal computer with a Matrox Genesis imaging board which uses the Texas Instruments TMS320C80 DSP processor. The video camera was used simply to capture the image sequences; these video sequences were then fed into the computer via the video cassette recorder. We also used the Path video system to monitor multi-use trails for long periods. This is a time-lapse video unit equipped with a marine battery and a camera in a

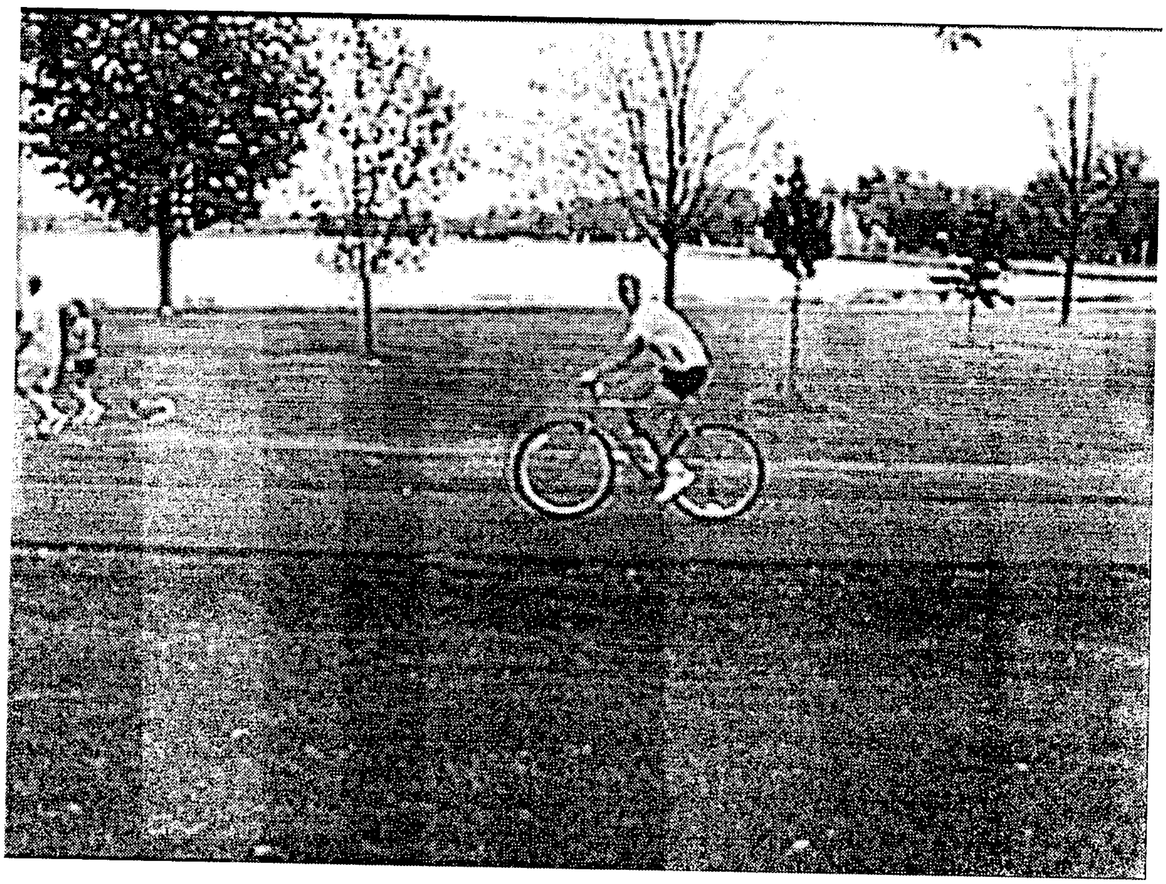


Figure 18: Example image from Lake Calhoun showing an instance captured when the optical axis of the camera is perpendicular to the trajectory of the bicycles.

weather-hardened box. A characteristic image from this system is shown in Figure 18.

CHAPTER 5
RESULTS AND CONCLUSIONS

EXPERIMENTAL RESULTS

The generic object tracking combined with the wheel detection proved successful. Actual video sequences were used in order to evaluate the proposed techniques. Figure 6 shows an image of the bicycle path at Lake Calhoun in Minneapolis where the optical axis of the camera is perpendicular to the trajectory of the bicycles. The results of processing demonstrate the successful tracking and detection of bicycles in real-time (Figure 19). The system has an average performance of 8 frames per second, with as many as two bicycles per image (the most tested for, with as many as 4 other objects being tracked at the same time). We saw successful counting of bicycles even when the back tire was partly occluded (also shown in Figure 19). We also tested the detection of bicycles whose path was not perpendicular to the optical axis of the camera. The results were similar to the first experiment as long as the angle of deviation is less than ±30°.

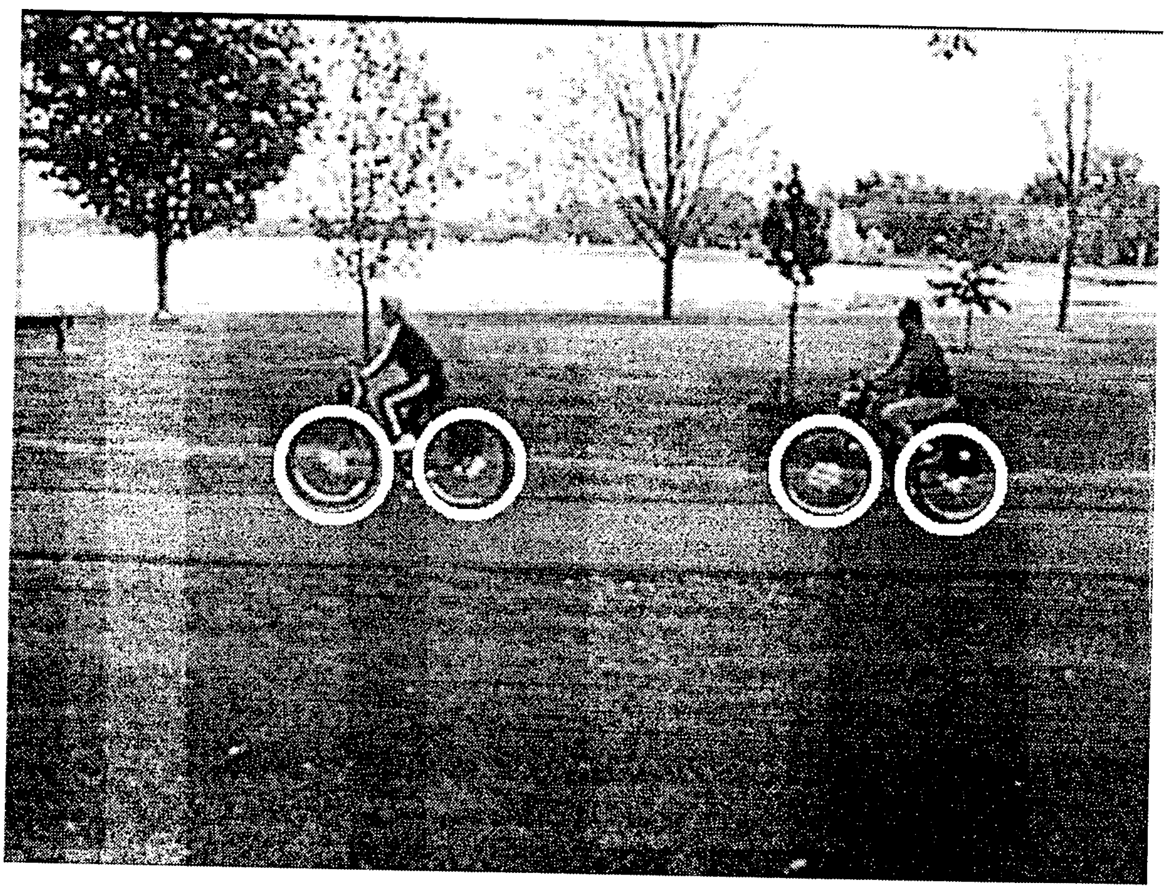


Figure 19: Example of bicycles detected.

Initial experiments (without tracking) had failed with such instances as the Figure 20 demonstrates: where edges form a circular pattern of appropriate size and expected location in frame, yet they do not correspond to a bicycle wheel. This was remedied by first applying independent object tracking. However, tracking fails to detect bicycles that overlap significantly. In addition, it has been found that if the camera is too close to where the bicycle passes, the bicycle might escape detection by either: a) passing by too quickly to be tracked, or b) the bicycle might just be starting to be tracked as it exits the view of the camera, thus escaping the detection phase. The average accuracy of the counting is around 70% with zero double- or triple-counting (as this was the case in our initial attempts in bicycle counting).

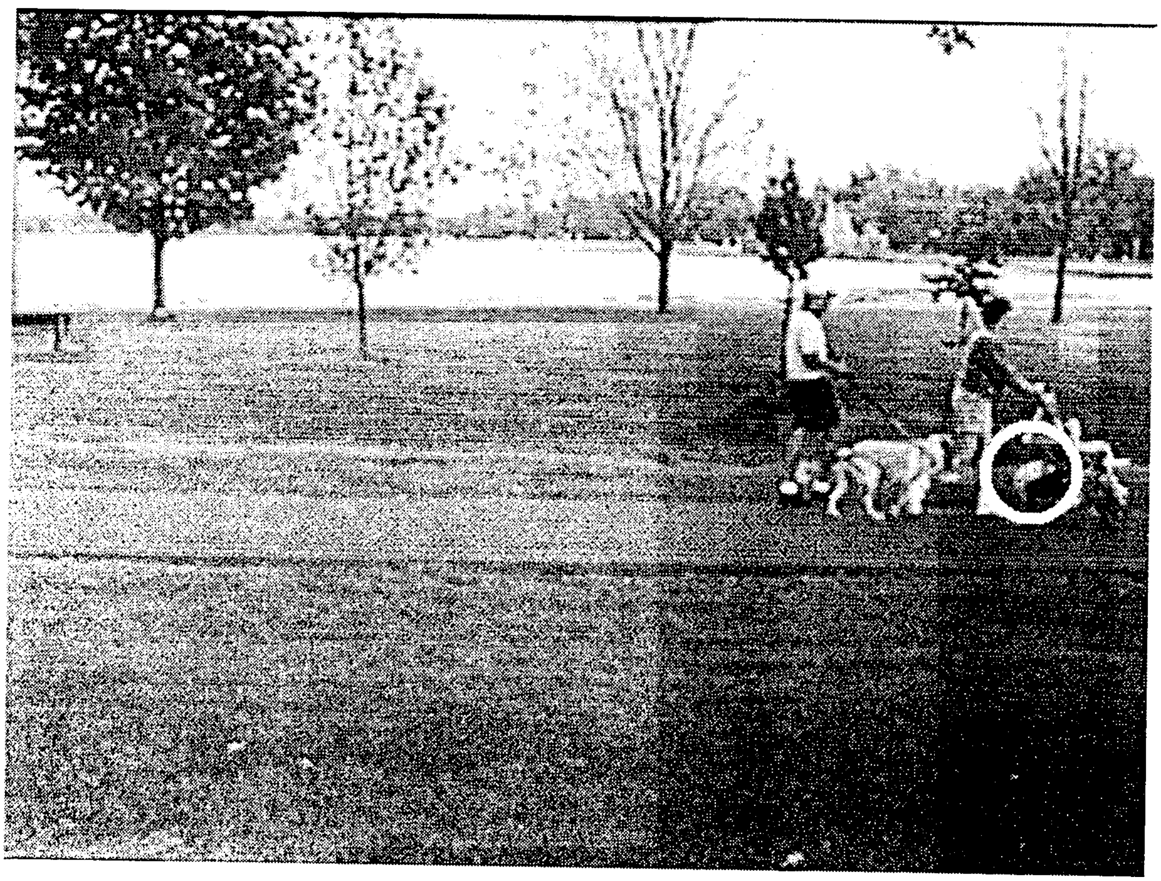


Figure 20: Example of a false detection.

CONCLUSIONS AND FUTURE WORK

The proposed system is able to successfully detect, track and count multiple bicycles in an image. However, there are still a small number of false detections and a small number of bicycles that pass through undetected. Much of this might be reduced by moving the camera back from where the bicycles. For reasons such as safety, security, occlusions, and practicality, the camera might be mounted considerably higher than the road surface. Some work is also needed to detect bicycles whose path is not perpendicular to the optical axis of the camera. Due to the fact that the Hough algorithm requires specific circle diameters, our final system must scale this size relative to the mounted camera distance. During initialization of the system, the user may define a range of wheel diameters in order to increase the processing speed. However, it should be mentioned that this step is suggested but not necessary.

Robustness is recognized as an important characteristic of the final system. Particular attention will be paid to occlusions such as those caused by uneven lighting due to variable weather conditions, and by other objects such as automobiles and trees. While occlusions are not a concern with the generic object tracking, it is possible for the detection to be taking place while the bicycle is either partially or completely blocked from view of the camera. This will require some further research work and experimental testing.

Finally, Mn/DOT decided to focus on phone-surveys in order to gather data about multi-use trails. Thus, the research on automatic bicycle detection did not continue.

REFERENCES

1. O. Masoud and N.P. Papanikolopoulos, Pedestrian Control at Intersections (St. Paul, MN: Mn/DOT MN/RC-1998/17, 1998).
2. G.X. Ritter and J. Wilson, Computer Vision Algorithms in Image Algebra (Boca Raton, FL: CRC Press, 1996).
3. S. Rogers and N.P. Papanikolopoulos, "A Robust Video-Based Bicycle Counting System", Proceedings of the ITS America Ninth Annual Meeting (Washington, DC: April 19-22, 1999).
4. E.R. Davies, Machine Vision: Theory, Algorithms, Practicalities (San Diego: Academic Press, 1990).
5. W. Rucklidge, Efficient Visual Recognition Using the Hausdorff Distance (Berlin: Springer-Verlag, 1996).