

# STRIPE: Supervised TeleRobotics Using Incremental Polygonal Earth Geometry

Jennifer S. Kay      Charles E. Thorpe

School of Computer Science  
Carnegie Mellon University  
Pittsburgh PA 15213

## Abstract

*We present a method of semi-autonomous teleoperation of a vehicle which allows it to accurately traverse hilly terrain while communicating with the operator across a very low bandwidth link. The operator plots the vehicle's chosen trajectory based on a single 2-D image, and the transformation of 2-D image points to 3-D world points is done in real time on the vehicle.*

*Traditional flat-earth geometry models do not work well on real world terrain. In contrast, STRIPE models the world as a collection of polygons rather than as a single plane. As the vehicle moves through the world, STRIPE continually adds new polygons to its internal world model. Each new polygon is derived by sensing the orientation of the patches of the ground beneath the vehicle's wheels as it moves. The projection onto the ground of a path, designated in a 2-D image, will become increasingly accurate as the world model is incrementally refined. While points far ahead of the vehicle will still be imprecisely projected, the incremental polygonal representation will almost always give adequate 3-D projection for the next few points used in steering the vehicle. The STRIPE method is equally applicable to on and off-road terrain, and can also be used for on-line mapping of the local terrain.*

## 1. Introduction

Semi-autonomous teleoperated driving in hilly terrain is difficult because the image of a road or desired path does not map easily from the 2-D image to the 3-D world. Typical approaches either require measuring or inferring the 3-D geometry, which is difficult and often inaccurate, or assume a planar world with no 3-D effects. The STRIPE method overcomes these problems by using an on-line approach to determine the path's geometry. Rather than requiring that we know the shape of the path before we attempt to drive along it, we are able to accurately compute the shape of the path as we go.

In STRIPE, a single image is transmitted from the vehicle to the base station. The teleoperator uses a mouse or a joystick to pick a sequence of points in the image that the vehicle should follow. This sequence of 2-D points is transmitted back to the vehicle, and the vehicle uses the points together with the incremental polygonal earth technique (described below) to project the points onto the 3-D terrain and follow the desired path. When the vehicle has moved a certain distance or

has reached the end of the path of points, it transmits another image back to the base station and the process is repeated.

The STRIPE technique is particularly useful in the case of a teleoperated vehicle which is being controlled over a low-bandwidth link. In order to make any sort of reasonable progress, we must cope with the low frequency with which images of the terrain (which typically contain a lot of data) are transmitted back to the base station. With the STRIPE system, a single image from the vehicle allows us to make considerable progress along an operator-defined path before the next image is needed. In addition, image transmission can occur in parallel with path following once the first path has been defined. The new image can be transmitted while the vehicle is only part of the way along the old path, and so the vehicle continues to make forward progress during this transmission.

## 2. Navigating in a 3-D World Using 2-D Images

There are several methods that have been used to recover 3-D geometry from 2-D images. Obviously, it is mathematically impossible to compute 3-D geometry from a single image without some additional information. Some systems use a stereo pair of images to provide sufficient information, others use a single 2-D image with additional assumed constraints about the world.

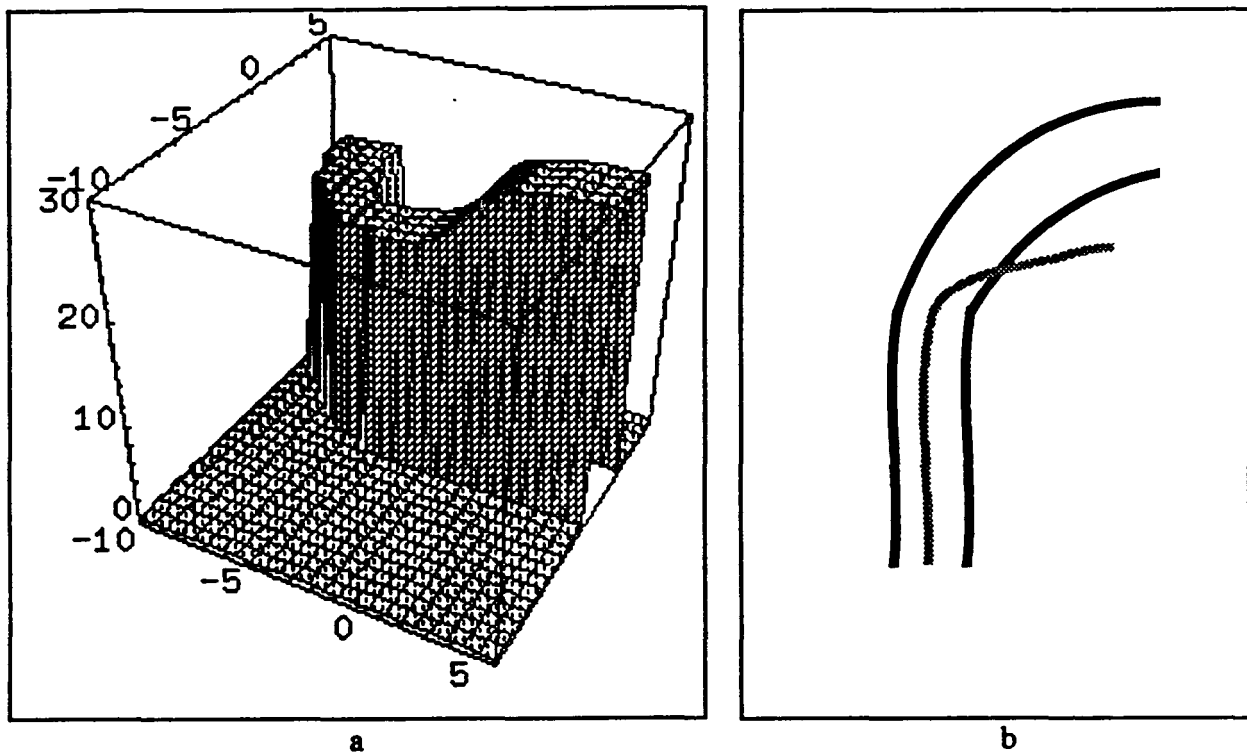
Given a stereo pair of images, it is mathematically possible to compute a 3-D description of the world, although computationally this is still a hard problem. One system similar to STRIPE [Lescoe 91] uses the human visual system to compute the 3-D geometry. The operator visually fuses the image pair, perhaps by wearing shuttered glasses, and then picks 3-D points using a 3-D mouse. Of course, if we are concerned about transmission time because we are using a limited-bandwidth link, then transmission of an image pair would take about twice as long as the transmission of a single image, and this delay may be unacceptably long.

Given only a single image, the "flat-earth" assumption is a simple way to constrain the problem sufficiently to make the transformation from 2-D to 3-D points well defined. This method assumes that all the points in the world lie on the same plane, which is known from the robot and camera geometry, and so the transformation from 2-D to 3-D is easy and well defined. Unfortunately, road following in a world which doesn't have a flat earth while using the flat-earth assumption generally gives poor results[DeMenthon 86].

Figure 1 contains an example showing a particularly hazardous road. The road travels straight ahead, down a steep hill, and then bends off to the right. Figure 1a is a view of the road from off to the side, and Figure 1b is a view from directly above the road. If the flat earth assumption was used, the length of the road before the turn would be underestimated. A vehicle using that method would take the path indicated by the gray line, with potentially disastrous results.

## 3. From Flat Earth to Polygonal Reprojection

The STRIPE technique is a new twist on the old flat earth idea that works surprisingly well. As we have shown, traditional flat-earth geometry models do not work well on real world terrain.



**Figure 1. Why the flat earth assumption doesn't work for a non-planar world. a: A hazardous road. b: Aerial view of the same road with the route that the flat earth assumption would take in gray.**

STRIPE, however, models the world as a collection of polygons rather than as a single plane. As the vehicle moves through the world, STRIPE continually adds new polygons to its internal world model. Each new polygon is derived by sensing the orientation of the patches of the ground beneath the vehicle's wheels as it moves. The projection onto the ground of a path, designated in a 2-D image, will become increasingly accurate as the world model is incrementally refined. While points far ahead of the vehicle will still be imprecisely projected, the incremental polygonal representation will almost always give adequate 3-D projection for the next few points used in steering the vehicle.

As we have demonstrated, it is unreasonable to assume that the entire road lies on the same plane. However it is usually safe to assume that the next little bit of road immediately in front of the vehicle lies on approximately the same ground plane as the vehicle itself.

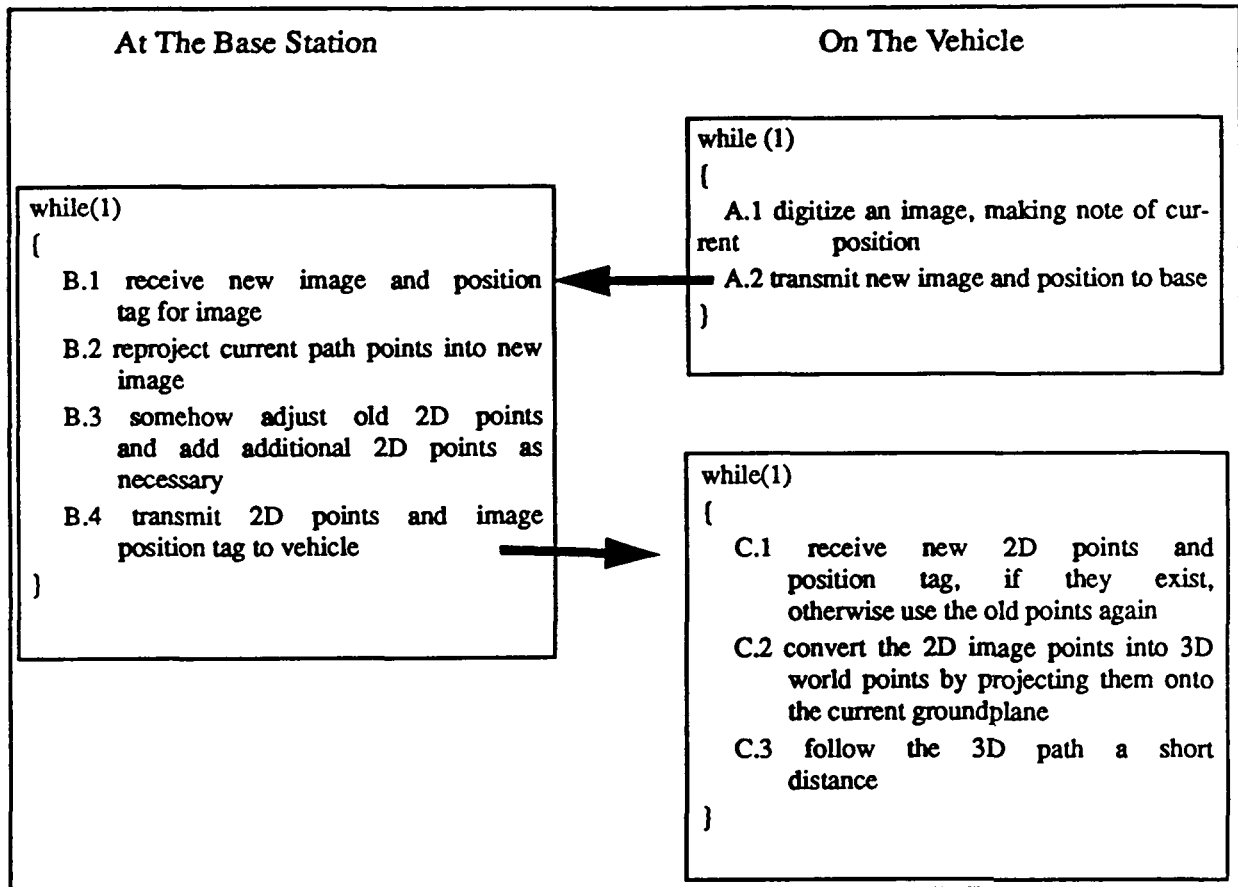
This leads to an obvious algorithm:

```
while (not there yet)
{
    1. digitize an image of the current scene
    2. somehow plot the 2D path you wish to follow in that image
    3. convert the 2D path points in the image into 3D world points using the flat earth assumption
       to project that image onto the vehicle's current groundplane
    4. from the 3D points, plan a trajectory
    5. follow that trajectory for a short distance
}
```

We accomplish Step 2 by having a human pick points as described above. It could, however, be completely automatic, first segmenting out the road in the image and then picking a path down the center of that road.

This algorithm was shown to work well in simulation, but isn't really very practical. Step 2 is very slow, especially using a human operator, and so the algorithm wouldn't really work in real time.

STRIPE uses a similar, but much faster, three part algorithm:



With this approach, the points are chosen in the B module, while the path following is done in the C module. Road following can continue at high speed so long as there are some points remaining in the C module that are ahead of the vehicle. As soon as new points are chosen, they are transmitted from B to C, and C can immediately begin to follow these new points instead.

## 4. Coordinate Frame Transformation Details

STRIPE needs to keep track of certain coordinate frame transformations in order to do the incremental reprojection of a path created at an old position onto a new ground plane. In particular, in order to project an image taken in a previous location to the groundplane of the vehicle's current location, we need to know the transformation between the old camera's coordinate system and the

current vehicle coordinate system. In this discussion we use the following notation: The transformation between  $\alpha$  and  $\beta$  is denoted by  $T_{\alpha}^{\beta}$ . The equation  $q = (T_{\alpha}^{\beta})p$  transforms the point  $p$  (currently in  $\alpha$ 's coordinate frame) to a point  $q$  in  $\beta$ 's coordinate frame.

The transformation between any vehicle coordinate frame and its corresponding camera coordinate frame is constant, since the camera is fixed on the vehicle, and we will refer to it as  $T_v^c$ .

Assume that we know the transformation between some world coordinate frame,  $w$ , and the vehicle position,  $a$ , where the image was taken ( $T_w^a$ ), and the transformation between the same world coordinate frame and the current vehicle location,  $b$  ( $T_w^b$ ).

So first we use  $(T_v^c)^{-1}$  to put the point in  $a$ 's coordinates. Next we use  $(T_w^a)^{-1}$  to put the point in world coordinates, and finally  $T_w^b$  transforms the point out of world coordinates and into  $b$ 's coordinate frame. Summarizing, if  $p$  is a point in the camera's coordinate frame, and  $q$  is the point in  $b$ 's coordinates, we have:

$$q = (T_w^b) (T_w^a)^{-1} (T_v^c)^{-1} p \quad (1)$$

Note that relative position information, such as data from an INS, is sufficient for the STRIPE system. No global positioning information is necessary, the actual location of the origin of the world coordinate frame is irrelevant.

## 5. Examples

Figure 2 through Figure 5 show a series of snapshots of the STRIPE simulator as it follows a path marked out in 2-D on a copy of the image by a human operator. In each diagram, the large window on the left is a simulation of a path as seen from a camera mounted on the roof of the vehicle. The small window in the upper left hand corner of the quartet is the image of the road as seen by a camera flying 30 meters above the van pointing straight down along the  $z$  axis in our world coordinate system. The "roll" of the camera is independent of the orientation of the vehicle, and is fixed in the world coordinate system throughout the sequence of snapshots. The conical blob on the road represents the current location of the vehicle. The lower left small square is the road as mapped by the STRIPE system. Similarly, the upper right window depicts an image of the road as seen by a camera 30 meters to the side of the vehicle (i.e. along the  $x$  axis in our world coordinate system). Again the roll of the camera is fixed with respect to our world coordinate system, and the lower right window is the corresponding vehicle generated map.

Notice how the road weaves about and banks at the same time as the ground plane is changing. The vehicle remains well centered on the road, and does a good job of mapping the terrain already covered. Even though at the beginning, in Figure 2, STRIPE's estimated position of the last rung in Figure 3 may not be very accurate, as we approach that spot the estimate gets better and we accurately follow the path as can be seen in the STRIPE's mapping of the road.

Figure 4 is an example of an image that's not terribly useful. We can only pick out points for a very short path because we have a very limited view of the road. While a human driver can adjust his gaze in order to gain a better view of the road, STRIPE currently has to put up with the occa-

sional image like that of Figure 4. Because of the bandwidth constraint, the vehicle has to somehow know that it should pan before digitizing an image. We intend to investigate this problem in the future.

## 6. Conclusion

We have shown how the STRIPE method models the world as a collection of polygons rather than as a single plane. With this model, it is possible to drive along winding paths that bank and go over hills, on and off road. The STRIPE model is particularly useful for semi-autonomous situations where a vehicle is tele-operated across a very low bandwidth link, however it is also applicable in autonomous road following situations where the edges of the road can be accurately segmented out from an image.

STRIPE is currently being tested on the Carnegie Mellon Navlab vehicle [Thorpe 91]. Arguably the hardest problem for the current version of STRIPE is the fact that the camera is mounted in a fixed position relative to the vehicle. Computationally, adding transformations to cope with a new camera position is quite trivial, however the problem of deciding how to move the camera is less so.

## 7. Acknowledgments

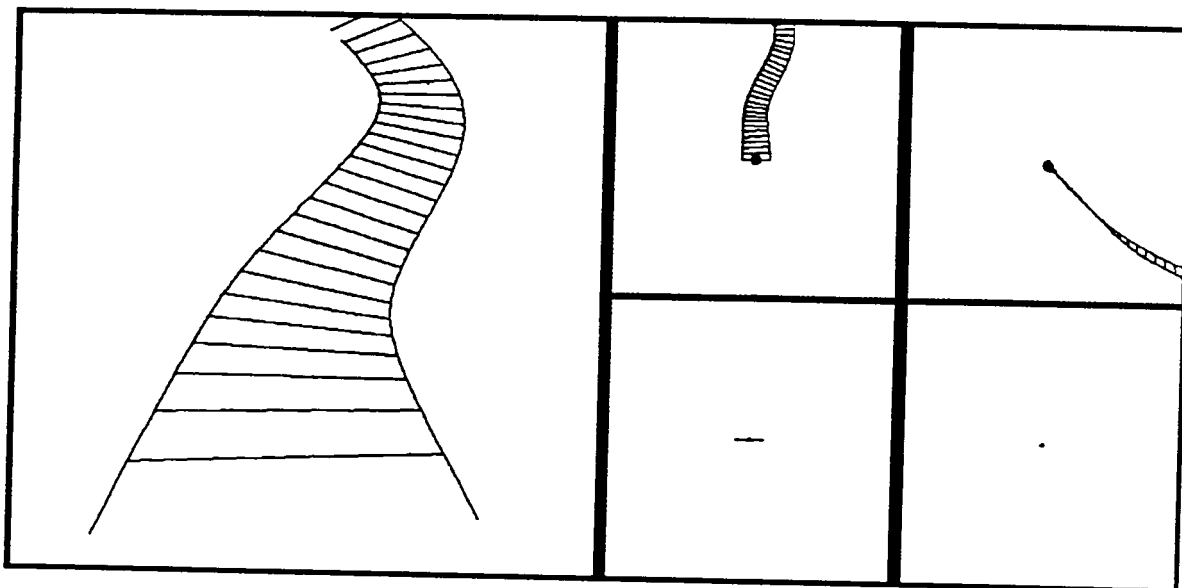
This research was partly sponsored by DARPA, under contracts "Perception for Outdoor Navigation" (contract number DACA76-89-C-0014, monitored by the US Army Topographic Engineering Center) and "Unmanned Ground Vehicle System" (contract number DAAE07-90-C-R059, monitored by TACOM). Thanks to Redmond English for the 3-D flat earth graphic.

## 8. References

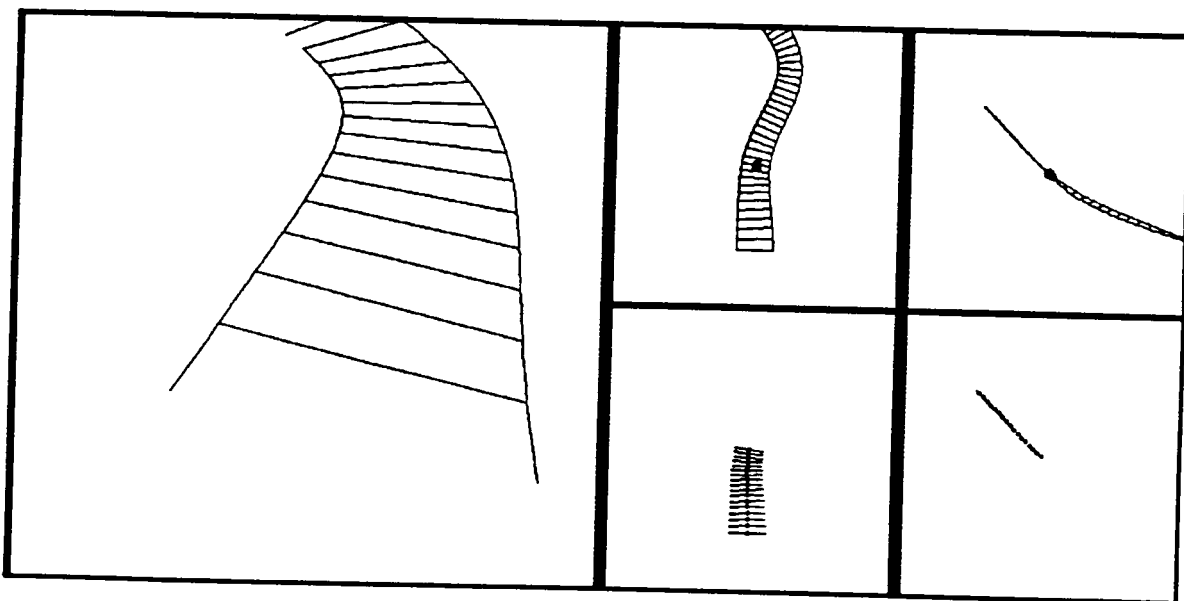
[DeMenthon 86] Daniel Dementhon. *Inverse Perspective of a Road from a Single Image*. Technical Report CAR-TR-210, Computer Vision Laboratory, Center for Automation Research, University of Maryland, July 1986.

[Lescoe 91] Paul Lescoe, David Lavery, Roger Bedard, *Navigation of Military and Space Unmanned Ground Vehicles in Unstructured Terrains*, Third Conference on Military Robotic Applications, September 1991.

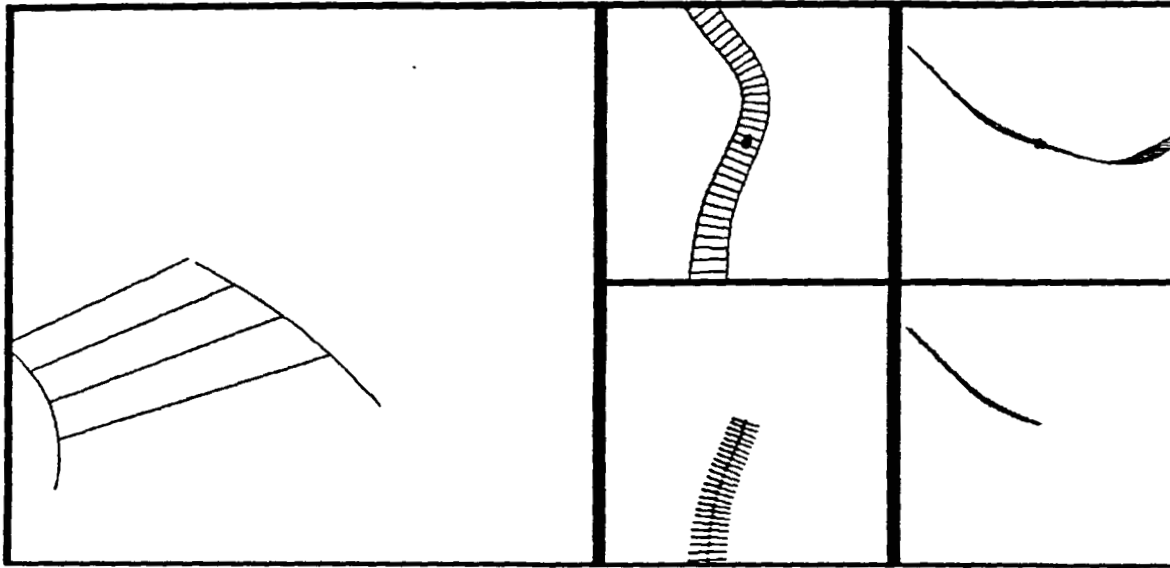
[Thorpe 91] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer. *Toward Autonomous Driving: The CMU Navlab*. IEEE Expert, V 6 # 4 August 1991.



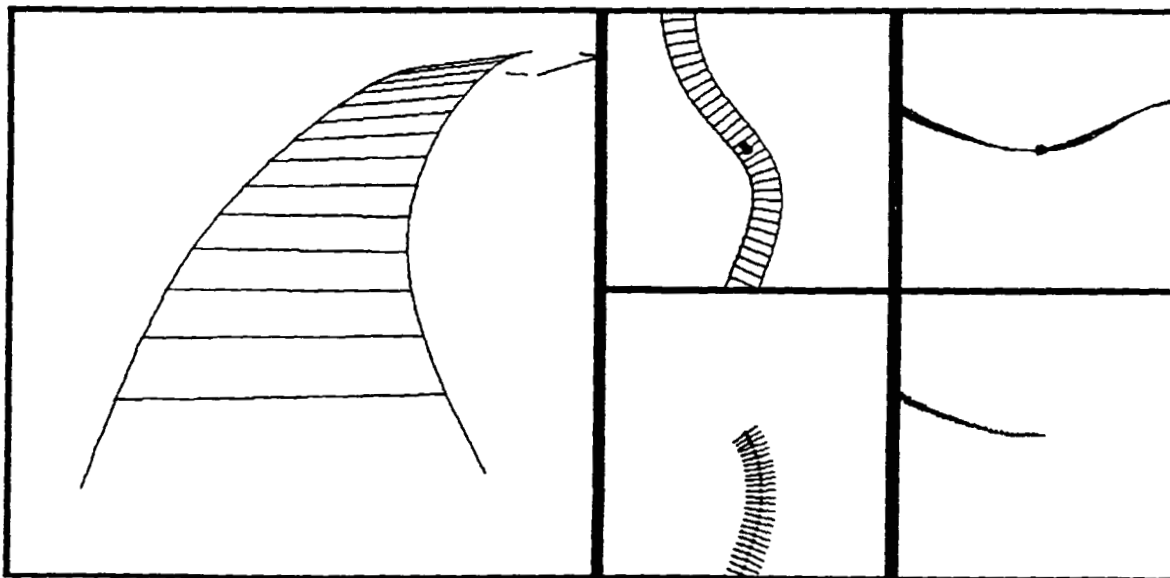
**Figure 2. At the Start of the Road.**



**Figure 3. About 8 Meters Along.**



**Figure 4. Not Much Road Visible.**



**Figure 5. Further Along the Road.**