A Pyramid Approach to Stereo Coorespondence

Piper L. Canfield

Anupam Joshi

Report Number:
95-007
A PYRAMID APPROACH TO
STEREO CORRESPONDENCE

Piper L. Canfield
Anupam Joshi

CSD-TR-95-007
February 1995
A Pyramid Approach to Stereo Correspondence

Piper L. Canfield
Amoco Corporation
Chicago, IL 60601 USA
and
Anupam Joshi
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907 - 1398 USA

Abstract

The problem of establishing correspondence between stereo images is a fundamental problem in many areas of computational vision. In this paper, we present a new approach to this problem of establishing correspondence and using it to obtain depth-based segmentation. The proposed algorithm uses the pyramid data structure and can be implemented in parallel. This representation was chosen because of its robust nature. Results from simulations are presented at the end of this paper.

1 Introduction

Stereo vision uses the disparity between two images of the same object, taken by imaging devices placed some distance from each other, to recover depth information about the object. It is much the same principle that is used by animals having binocular vision. In order to recover disparity, and hence depth, we need to know which points in the two images represent the same point in the scene. Establishing correspondence between points across image frames is thus a prerequisite for computing depth.

There is literature aplenty dealing with the subject of obtaining stereo correspondence. Much of it however rests on the cardinal assumption that all points are present in both the frames. Since the problem of correspondence can also be recast as an optimization problem, solutions using Neural Networks have also been proposed. Chellappa and Zhou[7] have proposed a three-dimensional Neural Network that minimizes an energy function containing terms for disparity values across regions and smoothness. It uses a sequence of frames. Nasrabadi and Choo[8] have used a Hopfield net to address this problem recently. In previous works[1,5], we also introduce a technique that uses Elastic nets to obtain stereo matching.

In [1] Joshi and Lee introduced an efficient stereo matching technique that uses elastic networks. This technique is robust even when there are a few missing points. Using the ideas presented in [1] and a pyramid approach inspired by recent psychophysical studies[2], a new algorithm has been formulated. This algorithm is relatively insensitive to a small amount of noise and to missing points. These features make this algorithm quite robust.

2 Pyramid Algorithm

The inspiration for this algorithm came from research into the human visual system. It is known that there are cells in area V2 of the primate cortex that code for depth[9]. These fall in the motion pathway that leads to Area 7 of the parietal cortex where motion correspondence is established. The receptive fields of these cells cover corresponding areas on the retinal field. Also, the sizes of the receptive fields increase the further one goes into the visual system. Pizlo et al. [2] posit that an exponential pyramid scheme models this feature of the visual system. The visual system thus has hardwired into it the broad knowledge of corresponding regions on the left and right retina. This idea has been used to obtain the displacements at a region level, which is then averaged to obtain the overall displacement vector [1]. Such a scheme allows for points that may exist in one frame, but not
the other. This can arise due to a number of reasons, including occlusion, noise, and the object moving out of the
camera's field of view. An image of \(n \times m\) pixels can be divided into \(m^2\) patches of \(n/m \times n/m\) pixels each.
Analogous to the brain, we assume that correspondence between patches is already known. We can apply the
method outlined above to each pair of corresponding patches, and obtain \(T_i, i = 1, \ldots, m^2\). We expect that each
\([T_i] \) will be close to the global \([T]\) which relates the two frames. For disparities that are small compared
to the size of the patch, most of the points from a given patch will be found in its corresponding patch in the next
frame. Some points will of course cross frame boundaries. This will result in some "missing" points in each frame.
A few missing points could also be introduced due to imaging errors or errors in feature detection. Such errors are
easily handled by our algorithm, since in the formulation of the energy function, we do not assume anything about
the number of points in the two frames. However, the gradient descent adjusts \([T]\) so that each point in the first
frame finds a match in the second, so we must consider the frame with the fewer as the first. This is WLOG, since
the translation vector in the two cases is simply related by -1. Note however that as the number of missing points
increase, the estimate to \([T]\) will get worse.

The idea of patches helps by spreading out the missing points amongst different patches. However, this idea is of
singular elegance when we consider the other major source of missing points, namely occlusion. Due to occlusion,
it is likely that some patch in one of the frames will have most of its points missing. In such cases, we can simply
ignore this patch-pair. A predefined limit can be fixed to determine when a patch pair has "too many" missing
points. Moreover, the computation of \([T_i] \) on each patch is independent of the other. Thus they can be
computed in parallel. The simplest way to obtain the global \([T]\) is to perhaps average the \([T_i] \). However, we could use more sophisticated techniques, such as an average that ignores the highest and lowest
estimates, or which ignores estimates that are far away from the mean or the median. Correspondence is trivially
obtained using this \([T]\) by the method outlined at the end of section-

2.1 Pyramid Structure

The pyramid has a base that has \(n\) regions per side. For simplicity, we assume \(n\) to be a power of two. The
base of the pyramid thus has \(n^2\) regions. Each of these regions is assumed to cover its corresponding part of the
image. The objective of the scheme is to obtain the translation vector representing the displacement in each of
these regions. To obtain these initial translation vectors, the points in the corresponding regions of each of the two
images are run through the algorithm described in [1]. As mentioned earlier, this uses an elastic net[6] based
scheme to obtain approximations to the displacement without knowing the correspondence a priori.

Each region in the pyramid contains a list of translation vectors and which points are associated with that
translation vector. At the base of the pyramid, this list contains only one translation vector and points from that
specific base region. That is, the points from the base region and the computed translation vector for the region are
stored in the T-list of the base region.

Each computing node in layer \(i\) (parent) has a receptive field that covers the receptive fields (regions) of
four nodes underneath it in layer \(i-1\) (children). Nodes in the base of the pyramid have no children, as they have
no regions below them. Each parent node uses the translation vector of its children to obtain either a translation
vector representing the whole region, or decides that the regions underneath it contain objects at different depths.
In the latter case, it identifies the vectors in each of the regions and propagates these upwards. The method used to
combine these translation vectors will be discussed in the next section.
2.2 Traversing the Pyramid

Once the base of the pyramid has been filled with initial translation vectors, the algorithm begins to process one level above the base. The translation vectors from a parent's four children must be combined in some meaningful way. This algorithm combines the lists in the following manner:

- A new, empty T-list is created for the parent node. For each of its four children,
  - If the translation vector of the child is close to a translation vector already in the T-list of the parent then
    - append the point list of the child's T-set (translation vector and associated points) to the point list of the parent's T-set
  - Otherwise
    - create a new T-set and appended it to the parent's current T-list (list of T-sets)

* A translation vector is close to another if it lies within some predetermined epsilon neighborhood the other.

When the top of the pyramid is reached, there will be one list that contains one or more translation vectors and their corresponding points. This list can then be traversed to see what those values are and what regions in the image are associated with it.

2.3 Enhancement to the original algorithm

The original algorithm used the first non-close translation vector from a child as the translation vector used for comparison of other children's translation vectors. There was a bit of difficulty with this concept because it did not represent the translation vector that was obtained after adding each child's points (and slightly different translation vectors) to the original. To rectify this problem, when a translation vector is determined to be close enough to a translation vector already in the parent's T-list, the translation vector that is stored changes (where it would have remained the same in the original algorithm). The new translation vector is the arithmetic average of the parent's current translation vector and the child's translation vector. This new scheme more accurately represents the translation vector of the points in the point list of the T-set in which this newly computed translation vector is to be stored.

3 Results of simulations

To test the proposed algorithm, several simulations were performed on synthesized data. For each run, two files were generated. These files contain simulated data points from each of two hypothetical cameras. The first file contains pairs of random real numbers within some known range. These pairs correspond to the x and y coordinates of the simulated points. The second image is created by shifting the first image by some translation vector. If, after a point has been shifted by the translation vector, it is no longer in the acceptable range, the point is discarded. The algorithm that was used also allows for noise to be added to the second image. This noise simulates what would be seen in real images be it from data capture irregularities or other noise sources.

The following table summarizes our results. It shows the translation vectors representing disparity in the original images, and the recovered values for the same. This is shown when there are upto three different depths for the objects in the image. The noise range shows the range of values in which values were chosen according to a uniform distribution and added to the data points. If a region had fewer than 10 points in the first or second image, a translation vector was not computed for that region. This is because with fewer than 10 points to analyze, the generated translation vector will not be accurate enough.
<table>
<thead>
<tr>
<th>Translation Vector(s)</th>
<th>Noise Range</th>
<th>Computed Translation Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3, 0)</td>
<td>none</td>
<td>(-2.936206, 0.030140)</td>
</tr>
<tr>
<td>(3, 0)</td>
<td>-0.7745 to 0.7745</td>
<td>(-2.083346, 0.716432)</td>
</tr>
<tr>
<td>(3, 0)</td>
<td>-0.2449 to 0.2449</td>
<td>(-2.723125, 0.110390)</td>
</tr>
<tr>
<td>(3, 0)</td>
<td>none</td>
<td>(-2.905216, -0.007830)</td>
</tr>
<tr>
<td>(1, 0)</td>
<td>-0.796115 to 0.19172</td>
<td>(-2.723125, 0.110390)</td>
</tr>
<tr>
<td>(3, 0)</td>
<td>none</td>
<td>(-2.846041, 0.085013)</td>
</tr>
<tr>
<td>(1, 0)</td>
<td>none</td>
<td>(-0.882990, -0.009923)</td>
</tr>
<tr>
<td>(2, 0)</td>
<td>-0.244949 to 0.244949</td>
<td>(-2.250100, 0.512935)</td>
</tr>
<tr>
<td>(5, 0)</td>
<td>-0.244949 to 0.244949</td>
<td>(-4.433579, 0.257409)</td>
</tr>
<tr>
<td>(4, 0)</td>
<td>-0.244949 to 0.244949</td>
<td>(-3.304881, 0.341543)</td>
</tr>
<tr>
<td>(1, 0)</td>
<td>none</td>
<td>(-1.032595, 0.126086)</td>
</tr>
<tr>
<td>(5, 0)</td>
<td>none</td>
<td>(-3.419695, 0.149729)</td>
</tr>
<tr>
<td>(3, 0)</td>
<td>none</td>
<td>(-0.796115, 0.19172)</td>
</tr>
<tr>
<td>(5, 0)</td>
<td>none</td>
<td>(-4.789385, -0.112133)</td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper, we have introduced an efficient stereo matching algorithm. This algorithm is relatively insensitive to noise and to missing points. It uses the idea of exponential pyramids and multiresolution to expand on an earlier algorithm proposed by the authors in [1] and allows it to handle the case of images containing objects at multiple depths. One possible enhancement to the current algorithm involves using regions that overlap as does the human visual system. The use of overlapping regions may increase the accuracy obtained by this algorithm for large translation vectors.

References