AudioVision: A Stereophonic Analogue to Visual Systems

John-Paul Verkamp

December 13, 2010

Abstract

AudioVision is designed to take a visual representation of the world–in the form of one or more video feeds–and convert it into a related stereophonic audio representation. With such a representation, it should be possible for someone who has minimal or no use of their visual system to avoid obstacles using their sense of hearing rather than vision. To this end, several different vision algorithms, including single and multiple image disparity, disparity from motion and optical flow were investigated. In addition two different methods of mapping the resulting disparity map to stereophonic audio–maximal points and sonar scan–were implemented. The results are rather promising. Using Lucas-Kanade optical flow and sonar scan audio has fulfilled the aforementioned goals in simple tests.

1 Introduction

Of the five major senses, most would agree that the worst to lose would be the sense of sight. Today's world is largely a visual one, and without sight even simple everyday tasks such as walking become more difficult. To that end, a wide variety of systems for augmenting or even outright replacing the human visual system using other systems have been developed in the past. Particularly interesting systems include using an array of electrical stimulators on the tongue [19], a cane that exerts a physical force to pull it's user around obstacles [5], and theoretical work involving attaching a camera directly to the optical nerve [24].

The goal of AudioVision is to use simple, readily available components to provide a system that can provide a three dimensional audio analogue to replace some aspects of the human visual system. To date, relatively little work has been done tying the audio and visual systems together, with most tending in the opposite direction: using visual cues to place audio sources in a three dimensional environment [4] or using face detection techniques to improve the accuracy of speech recognition systems [9].

The overall goal will be to use a combination of computer vision techniques and stereophonic audio mappings to generate an audio representation of the visual field in audio. A variety of computer vision techniques such as real-time stereo vision [17], single-image stereo mapping [21, 20], stereo from motion [10, 14], and optical flow [12, 3, 6] were evaluated for use in AudioVision. Likewise, stereophonic audio techniques were investigated to generated an adequate audio representation, although none as well known as the aforementioned computer vision techniques.

2 Methods

Essentially, the entire algorithm is divided into two key components: visual and audio. Starting with 1 or more video streams, one of the following computer vision algorithms is applied creating an internal representation for the possible locations of objects. Depending on which audio algorithm will be used, this representation will be either down-sampled or converted into a one dimensional graph and played as stereophonic audio.

2.1 Visual

Four different (albeit related) computer vision algorithms were evaluated for potential use in AudioVision. Each was evaluated based on the following three criteria: does the algorithm run in real-time, how accurate is the generated depth map, and can the algorithm run on inexpensive hardware. Each of the algorithms should take a video video and produce a disparity map capable of identifying the nearest points in the image to the camera.

2.1.1 Multi-camera disparity

The first and perhaps most obvious algorithm to use for calculating disparity maps is multi-camera disparity algorithms. Using two or more cameras placed a generally known distance apart, one can use the geometry of the scene to directly infer the distance to points in the image as seen in Figure 1.

The essential idea of multi-camera disparity is that points in one image can be projected into lines in the other image (know as the epipolar line). Thus for each point in one of the image, a window around that point is matched to the corresponding point in the other image. The distance between these two points along with the distance between the two cameras gives the algorithm enough information to determine the depth to the real-world point relatively accurately [17].

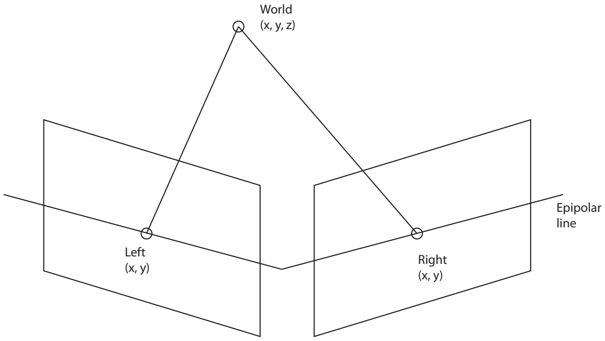
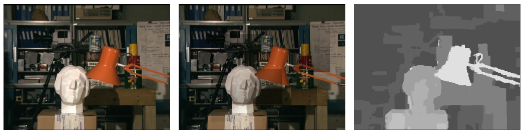


Figure 1: Multi-camera disparity

Furthermore, several possible improvements are possible over naively comparing a window with all possible windows in the other image. For example, using the idea that the disparity map will generally be smooth with the exception of occlusion and folds, each line can be modeled as a Markov Chain (or the entire image as a Markov Random Field [22]) and solved using belief propagation [22], graph cuts [23], or similar algorithms. Each of these concepts improves the accuracy of the algorithm without increasing or sometimes even decreasing the run-time.

For an example of the disparity maps generated by the aforementioned belief propagation based multi-camera disparity algorithm, see Figure 2.



(a) Left (b) Right (c) Result

Figure 2: Multi-camera disparity example[2, 22]

Overall, multi-camera disparity succeeded on delivering adequate disparity maps given proper input and did manage to run in real-time although the frame-rate was not spectacular. Using Markov Random Fields with belief propagation, it was possible to get 2-3 frames per second on a moderately powerful computer. The biggest problem with using this algorithm is that it is rather sensitive to the quality of hardware and the calibration of the system. The best speeds are obtained when the image planes of the two cameras are parallel (so that no projective transformation is necessary to make it so); however, with cheap, off the shelf web-cams, such calibration is difficult or impossible.

2.1.2 Single-camera disparity

The second potential computer vision algorithm is based on the Make3D project originally from Stanford University [21, 20]. The essential idea is, rather than needing a pair of images to infer depth, to be able to treat the entire image as a Markov Random Field and to learn a series of parameters that can determine an approximate depth at each point directly as shown in Figure 3.

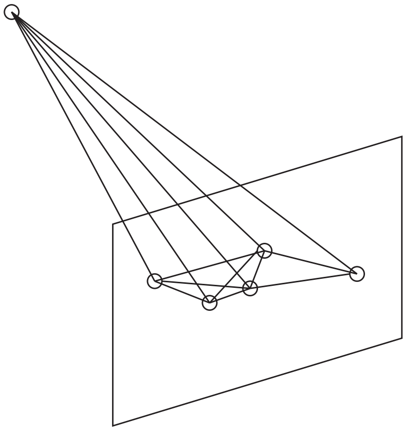
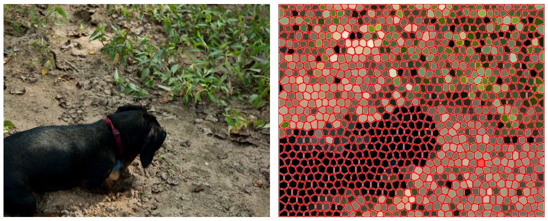


Figure 3: Single camera disparity

The first step of the algorithm is to over-segment the image into a collection of super-pixels as shown in Figure 4. This has the advantage of drastically reducing the size of the result graph while mitigating most of the problems with inaccurate segmentation by purposely over segmenting the image.



(a) Before (b) After

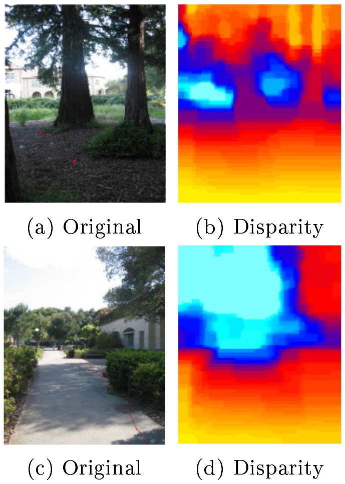
Figure 4: Over segmentation[21, 20]

After that, five assumptions were made about the super-pixels [21, 20]:

- Features (such as color and intensity of the super-pixels) are related to depth
- Neighboring super-pixels are likely to be connected, with the exception of occlusion
- Long lines on the edges of super-pixels are likely lines in the image
- If neighboring pixels have the same features, they likely belong to a plane
- If there is a line between similar pixels, it likely belongs to a fold / corner in the image

Using these rules as the basis for a Markov Random Field generated from the super-pixels and several hundred test images with ground truth based on a laser range finder, it was possible to generate relatively accurate disparity maps and three-dimensional models on a variety of test images as shown in Figure 5.

As can be seen in Figure 5, the results were similar to those obtained using multiple cameras. Although the disparity



(a) Original (b) Disparity

(c) Original (d) Disparity

Figure 5: Sample images from Make-3D[21, 1, 20]

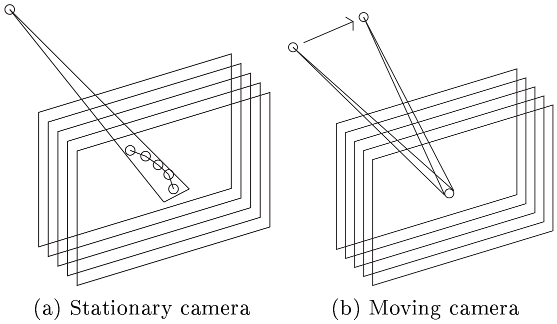
map was not as detailed in some cases, there is more than enough detail for the purposes of AudioVision. So far as run-time is concerned, the original training that is necessary for the Markov Random Field to produce acceptable answers takes quite a while to run; however, it only needs to be run once. Once the algorithm has been trained, it is possible for it to run in real-time. In addition, because of the over-segmentation in the first step many of the problems with noise and expensive hardware setups that other setups have had were removed. The only problem with the project is a lack of portability. Written as a combination of C++ and MATLAB, it is unfortunately necessary to have a MATLAB license to run the algorithm. While this restriction can be overcome, it is not ideal.

2.1.3 Disparity from motion

The third possibility is to use a single camera but to take advantage of the sequential frames of the video feed. The strength of the idea comes from the idea that tiny eye movements are a strong factor in stereo vision in humans and other animals [11], especially in the ability of a human to determine depth even with only one functioning eye [16].

There are two possible cases for how disparity from motion can work. In the first case, as shown in Figure 6a, the camera remains stationary and observes an object in motion. Using other methods (such as object recognition), the general size and/or velocity of the object is estimated. From this information it is possible to recover the depth of the object [11]. The second option relies observing a stationary object and moving or panning the source of the video tiny amounts as shown in Figure 6b[10, 14]. In this case, the same geometry used in the multi-camera model can be used with a single camera.

Disparity from motion has yet to be evaluated; however, it is likely that it will be similar in performance to the following optical flow algorithms due to their similar reliance on matching image windows and single video feeds.



(a) Stationary camera (b) Moving camera

Figure 6: Disparity from motion

2.1.4 Optical flow

The fourth and final potential computer vision algorithm is optical flow. Based on the idea that, with a small enough difference between frames, the motion of individual pixels within an image is directly related to the motion of the objects they represent [12, 3]. An example of optical flow field for a rotating object is shown in Figure 7. Two possible optical flow algorithms were tested: Horn-Schunk [12] and Lucas-Kanade [3]. See Figure 8 below for another example.

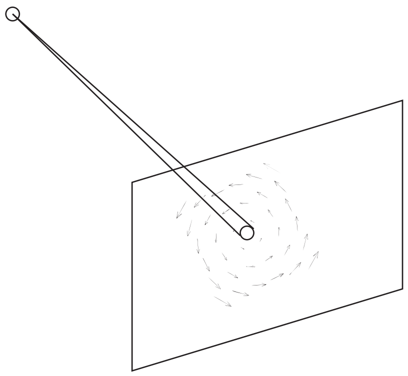


Figure 7: Optical flow

In Lucas and Kanade's method, the assumption is made that flow is essentially constant in a local neighborhood (similar to the window matching used in multi-camera stereo). Using this idea, a window around each pixel is compared to all of the nearby windows, looking for an optical match [3]. Because it doesn't rely on feature detection, this method is both less sensitive to noisy images and able to distinguish between textured regions by preferring the nearest such regions. Conversely, because it is a local algorithm, regions with minimal or no texture are not matched.

In the Horn-Schunk method, optical flow is treated as a global optimization problem on the smoothness of the image. It starts with the assumption that regions in the image are moving together and seeks to minimize the number of breaks between regions [12]. A large number of features are detected and tracked through multiple frames, using this assumption. The primary advantage of the Horn-Schunk method is that, due to it's global nature, information missing from non-textured regions is filled in from the edges. On the other hand, because of the global constraints, it is relatively more susceptible to noise than the Lucas-Kanade method [3].

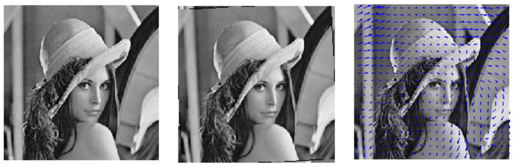

(a) Frame 1 (b) Frame 2 (c) Result

Figure 8: Optical flow example[18, 3]

In order to estimate depth from optical flow, a similar method to that used in disparity from motion is used. Small movements in the camera result in optical flow around still objects; the closer the object, the greater the magnitude of the motion vector. Theoretically, two cameras working as a stereo pair would have the same effect (and in fact use a very similar if not the same basic algorithm) as the multi-camera disparity model mentioned earlier.

Both algorithms run quickly enough to satisfy the real-time requirement, running faster than any of the other methods tested so far. However, due to the necessity of using inexpensive components, the Horn-Schunk method had problems with the noisy video stream. The Lucas-Kanade method performed much better under noisy conditions. Because of the speed at which the algorithm can run and the accurate results, optical flow using the Lucas-Kanade method was chosen for AudioVision.

2.2 Audio

Once a disparity map has been generated, the second half of AudioVision is to generate a three-dimensional audio representation. Using the difference in time between each ear when a sound is received, the brain is able to place sounds in a three-dimensional space [8, 15]. Using this idea, it is possible to play a sound into a pair of headphones and have the brain automatically place it into three-dimensional space.

One problem with this representation is that this audio illusion is just that: an illusion. There are several possible situations that the brain cannot deal with correctly. For example, several points directly in front of the user but at different vertical positions will not be distinguished. In addition, if two sounds of equal volume are played at equal angles from directly forward (see Figure 9), the will cancel each other out and appear to be directly in front of you.

So far, there are two possible methods for placing the sounds in three dimensional space: maximal placement and a scanning field similar to a sonar ping.

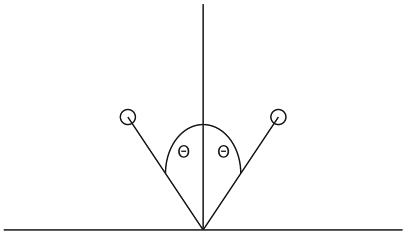


Figure 9: Equal angels

2.2.1 Maximal points

In the maximal placement algorithm, a set number of maximal points (the points registered as closest to the user) are detected in the disparity map, suppressing nearby points so that the maximal points are not clustered. See Figure 10b for an example of the maximal points.

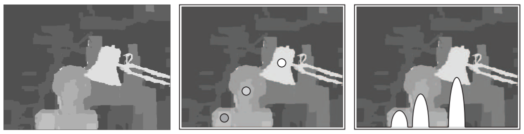
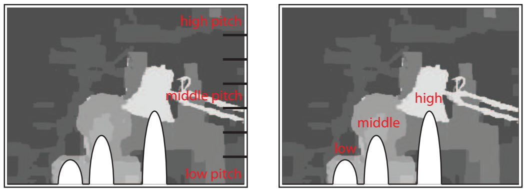

(a) Disparity (b) Points (c) Volume

Figure 10: Maximal placement

Using these points, a set of tones are played at each of the x-coordinates. The strongest match has the highest volume with descending volume for the rest of the points. This has the advantage of relaying the most important information to the user; however, it doesn't account for either vertical displacement or equal angles.

One solution to account to solve either vertical displacement or equal angles (but not both) is to use different pitches to represent either the vertical dimension as in Figure 11a or the different points as in Figure 11b; however, this cannot solve both problems at the same time.



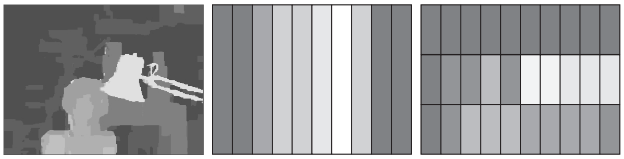
(a) Vertical (b) Equal angles

Figure 11: Varying pitch in maximal placement

The different pitches do help with either problem, but the layering of sounds has proven to be jarring and too loud to listen to for long stretches at a time.

2.2.2 Sonar scan

The second style audio representation uses a scanning pulse similar to sonar to overcome the problem with multiple sounds at equal angles. Essentially, the entire image is compressed into a series of vertical bars (taking the average and then normalizing, for example) as shown in Figure 12b. After that a single tone is played for each bar from left to right, with the volume of the bar corresponding to its intensity. Because only one sound is played at a time, the problem with equal angles is suppressed. In addition, the scanning audio actually enhances the audio illusion making placing the sounds easier.



(a) Disparity (b) Vertical bars (c) Blocks

Figure 12: Sonar scan

One problem that still exists with this solution is again vertical displacement; however, a similar solution to above exists. Simply break down the image into blocks instead of bars (as in Figure 12c) and play a chord for each bar while scanning. Because the problem with equal angles has already been solved by the sonar scan, the inability to solve both issues that the maximal points algorithm had is no longer a problem.

One variable to tune when using a sonar scan is the speed of the scan. If the scan is too slow, than objects may appear between scans. In a system designed to work under the same circumstances as visual systems, this is a problem. Conversely, if the scan is too fast the sounds will get jumbled and it becomes impossible to distinguish between them. Both problems are easily solved by allowing the user to control this speed, either speeding up or slowing down playback as necessary.

Sonar scan using vertical blocks is the currently implemented method as it solves both the vertical displacement and equal angle problems.

3 Results

One of the primary goals was to design an algorithm capable of taking a live video feed and generating an equivalent stereophonic audio representation in real-time to the extent that navigation in a simple environment is possible.

To that end, a number of different computer vision algorithms including multi-image disparity mapping [17], single-image stereo disparity [21, 20], disparity from motion [10, 14], and optical flow [12, 3, 6] were evaluated. As shown in Table 1, the three goals of the visual algorithms were to be accurate, fast, and inexpensive.

Optical flow using the Lucas-Kanade method was selected as the best method, although it is possible that single-image disparity maps using Make-3D [20, 21] or using the Horn-Schunk optical flow method [12] are possible in the future.

Algorithm	Accuracy	Speed	Cost
Multi-image	×	×	[1]
Single-image	×	×	×[2]
Motion[3]			
Optical flow (HS)	[4]	×	×
Optical flow (LK)	×	×	×

Table 1: Possible visual algorithms

[1] Requires calibrated stereo cameras
[2] Requires MATLAB license
[3] Not currently implemented
[4] Sensitive to noise

In addition to the computer vision algorithms, two different algorithms for producing stereophonic audio were tested–maximal points and sonar scan–each with multiple variations.

In this case, two problems were tested against: the ability to account for vertical resolution and to correct for two points at equal angles. In addition, the speed of audio feedback is important. The results are shown in Table 2 with sonar scan using blocks being the currently optimal solution.

Algorithm	Vertical	Angles	Speed
Maximal			×
Maximal, vertical	×		×
Maximal, pitch		×	×
Scan, bars		×	×[1]
Scan, blocks	×	×	×[1]

Table 2: Possible audio algorithms

[1] Adjustable

4 Future Work

There are several components of AudioVision that would benefit from future work, including various vision algorithms, audio algorithms, and human testing.

From the vision algorithms, the first improvement would be to investigate other optical flow algorithms such as the Buxton-Buxton method [7] which uses moving edge features or the Black-Jepson method [13] using correlation. Judging from the success of the implemented optical flow methods, either or both of these may prove successful.

The audio algorithms could also use some more tuning. There are a number of parameters used in either algorithm–the number of points in maximal points, the number of columns or rows in sonar scan–that are set essentially arbitrarily at this point for which optimal values should be determined. In addition, there are many more possibilities for more soothing audio playback options in place of the current tones, such as playing back music or instrumental patterns.

Finally, once more has been done with both the visual and audio algorithms, the ultimate end goal would be to test the algorithm with volunteers unfamiliar with the implementation in a controlled environment. With feedback from such volunteers, the various parameters could be tuned even more or placed under direct control of the user.

References

[1] Learning Depth from Single Monocular Images.

[2] Stereo Matching on Semi-Synthetic Scenes.

[3] S. Baker and T. Kanade. Super-resolution optical flow. 1999.

[4] S. Basu, M. Casey, W. Gardner, A. Azarbayejani, and A. Pentland. Vision-steered audio for interactive environments. *Proceedings of IMAGECOM*, 96, 1996.

[5] J. Borenstein and I. Ulrich. The guidecane-a computerized travel aid for the active guidance of blind pedestrians. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 2, pages 1283–1288. IEEE, 2002.

[6] A. Bruhn, J. Weickert, and C. Schn
"orr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[7] BF Buxton and H. Buxton. Computation of optic flow from the motion of edge features in image sequences. *Image and Vision Computing*, 2(2):59–75, 1984.

[8] J.M. Chowning. The simulation of moving sound sources. *Computer Music Journal*, 1(3):48–52, 1977.

[9] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *Multimedia, IEEE Transactions on*, 2(3):141–151, 2002.

[10] C. Eveland, K. Konolige, and RC Bolles. Background modeling for segmentation of video-rate stereo sequences. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 266–271. IEEE, 2002.

[11] W.C. Gogel and J.D. Tietz. Absolute motion parallax and the specific distance tendency. *Perception and Psychophysics*, 13(2):284–292, 1973.

[12] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[13] A. Jepson and M.J. Black. Mixture models for optical flow computation. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 760–761. IEEE, 2002.

[14] T. Kanade, H. Kano, S. Kimura, A. Yoshida, and K. Oda. Development of a video-rate stereo machine. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 95–100. IEEE, 2002.

[15] P.P. Lennox, T. Myatt, and J.M. Vaughan. From surround to true 3-D. In *Proceedings of the AES 16th International Conference on Spatial Sound Reproduction*, pages 10–12, 1999.

[16] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 204(1156):301–328, 1979.

[17] D. Murray and J.J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171, 2000.

[18] Gabriel Peyre. Optical Flow Computation, December 2010.

[19] E. Sampaio, S. Maris, and P. Bach-y Rita. Brain plasticity: visual acuity of blind persons via the tongue. *Brain research*, 908(2):204–207, 2001.

[20] A. Saxena, S.H. Chung, and A.Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision*, 76(1):53–69, 2008.

[21] A. Saxena, M. Sun, and A.Y. Ng. Make3d: learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, pages 824–840, 2008.

[22] J. Sun, N.N. Zheng, and H.Y. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 787–800, 2003.

[23] M.F. Tappen and W.T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. 2003.

[24] E. Zrenner. Will retinal implants restore vision? *Science*, 295(5557):1022–1025, 2002.