HIGH SPEED AUTOMATIC DEPTH MAP GENERATION FOR 3D TELEVISION

Amisha H. Somaiya

Vivekanand Education Society's Institute of Technology, University of Mumbai, Maharashtra, India

Ramesh K. Kulkarni

Vivekanand Education Society's Institute of Technology, University of Mumbai, Maharashtra, India

Abstract

3D Television is the latest advancement in television viewing. Creating enough stereoscopic (S3D) material with stereoscopic cameras for 3DTV is time consuming & expensive. Thus the conversion of the vast collection of already existing 2D images/videos to S3D is essential. Monoscopic video content can be efficiently converted to stereo by using depth maps. The effectiveness of 2D to 3D video conversion depends on the accuracy of the generated depth maps. However, current techniques either use single monocular depth cues, which are restrained to a specific set of images/videos or combine multiple monocular cues, thus increasing the complexity and execution time of the system. The proposed algorithm alleviates these problems by using a novel depth map generation algorithm that can be used over a wide range of arbitrary videos. The high efficiency of this automatic algorithm without requiring any prior training combined with its high execution speed make it ideal for application in 3D television broadcasting industry.

Keywords: Broadcasting, Depth Map Generation, Stereo Displays, Three-Dimensional Displays, 2D-to-3D Conversion, 3D-TV, 3D Video

Introduction

Three-dimensional television (3D-TV), the latest advancement of television, increases the visual impact and the sense of presence for viewers. The supply of adequate stereoscopic 3-dimensional (S3D) content is essential to ensure that the public would be willing to spend money for 3D displays and 3D-TV services. Creating enough S3D material with stereoscopic cameras is time consuming & expensive. The conversion of the vast collection of already existing 2D images/videos to S3D is one way to alleviate this difficult problem [1].

The conversion of 2D images to stereoscopic 3D images involves horizontal shifting of pixels to create a new image, so that there are horizontal disparities between the original image and a new version of it. There are three schemes for this conversion: manual, human-assisted and automatic. The manual scheme is to shift the pixels horizontally with an artistically chosen depth value for different regions/objects in the image to generate a new image [2]. This method is very time consuming and expensive. The human-assisted scheme is to convert 2D images to stereoscopic 3D with some corrections made "manually" by an operator. Even though this scheme reduces the time consumed in comparison to the manual conversion scheme, a significant amount of human engagement is still required to complete the conversion. The automatic conversion scheme exploits depth information originated from a single image or from a stream of images to generate a new projection of the scene with a virtual camera of a slightly different (horizontally shifted) viewpoint. This scheme involves retrieving depth information from a monoscopic image or video and generating high-quality stereoscopic images at new virtual viewpoints [1].

Depth map is a greyscale picture in which a pixel's brightness specifies that pixel's distance from the viewer in the original picture. This correspondence map should be constructed for each frame of the input 2D video. The resulting stereo video is generated from the corresponding depth maps and the original 2D video by shifting each pixel of a given 2D image to the left or to the right depending on the corresponding depth value, the type of stereo view (right or left) and the generation settings.This process is simpler, more practical, more predictable and repeatable than other methods of 3D scene reconstruction [3].

Background and Related Work

Depth cues can be classified into monocular and binocular. Binocular cues provide depth information when viewing a scene with both eyes through exploitation of differences between the perceived images, while monocular cues provide depth information when viewing a scene with one eye [1].An existing 2D video when converted into individual frames for processing has only 1 view for each frame & hence binocular methods cannot be used for video. The constraints in selection of depth cues for conversion of existing videos are more restrictive than for images. The pictorial and geometric type of monocular depth cues generally used in depth map generation and their associated constraints for individual usage in 2D to 3D video conversion are as follows:

Focus/Defocus (Blur) [4]-[7]: Blur is one of the first mechanisms to be employed to recover the depth from single images.

Approach 1: Employ several images with different focus characteristics in order to extract the variation of blur for a given image feature across the available images. This approach is reliable & provides good depth estimation.

Constraints: The requirement of having several images of the same scene captured with different optical systems simultaneously is too restrictive to be of any practical application in the 2D-to-3D conversion problem.

Approach 2: Extract the blur information from a single image. This approach is relatively simple.

Constraints: The scenes captured using advanced cameras do not necessarily show background as out-of-focus regions.

Texture Gradient [7]-[9]: This method, also called shape-from-texture, aims to estimate the shape of a surface based on cues from markings on the surface or its texture. It is highly efficient for textured images & can be used to estimate distance when width, or separation of elements perpendicular to the surface slant, decreases with increasing distance and is known as perspective gradient or height, or separation of elements in the direction of surface slant, decreases with increasing distance and is known as compression gradient or density, or number of elements per unit area, increases with increasing distance and is known as density gradient.

Constraints: This approach is normally restricted to specific types of images and cannot be applied to 2D-to-3D conversion of general video content. Also all three texture cues vary with distance according to a *power law* which depends on the surface slant of the texture and the observer's height. It has been reported that texture gradients were only useful when the surface slant was in excess of 50° from vertical and when elements of similar size, shape, and spacing repeat in the scene

Light and Shadow [10]-[11]: Light and shadow distribution refers to the information provided by shadows with respect to the position and shape of objects relative to other objects and the background. This method can be used to measure depth of various objects that are solid, have only one light source or placed lower than the ground plane (like a well).

Constraints: When utilising shadows, the visual system makes the assumptions that the light is directed from above and the objects are convex rather than concave. For attached shadows, the illumination should be uniform and the object's surface should be a uniform, diffuse reflector.

Linear Perspective [12]-[15]: Linear perspective refers to the property of parallel lines of converging at infinite distance, or equivalently, a fixed size object will produce a smaller visual angle when more distant from the eye. This characteristic is used for depth estimation by detecting parallel lines in the images and identifying the point where these lines converge (vanishing point). Then a suitable assignment of depth can be derived based on the position of the lines and the vanishing point. This is the most commonly used geometric cue.

Constraints: It alone is not sufficient for faithful depth analysis of picture.

Height [7], [16]-[17]: The height in picture denotes that objects that are closer to the bottom of the images are generally closer than objects at the top of the picture. Outdoor and landscape scenes mainly contain this pictorial depth cue. To extract this depth cue, horizontal lines usually have to be identified so that the image can be divided into stripes that go from the left border to the right border. For this purpose, a line-tracing algorithm is applied to recover the optimal dividing lines subject to some geometric constraints. A depth-refining step is further applied to improve the quality of the final depth map.

Constraints: For assigning depths to a 2D image, a pre-defined depth model, which can be adjusted according to image structure, is required. As also, it is limited to objects in contact with a level, horizontal, ground plane.

Atmospheric scattering [12],[18]-[20]: Atmospheric scattering refers to the scattering of light rays by the atmosphere producing a bluish tint and less contrast to objects that are in the far distance and a better contrast to objects that are in close range. It is a simple approach & can provide a significant enhancement to the 3D effect with respect to the perceived depth in 2D images.

Constraints: The colour rules, to divide landscape/outdoor images into six regions such as sky, farthest mountain, far mountain, near mountain, land, and other have to be learnt heuristically & it is difficult for use in studio images.

Motion Parallax [7]: Motion parallax refers to the relative motions of objects across the retina. For a moving observer, near objects move faster across the retina than far objects, and so relative motion provides an important depth cue. This is usually called the principle of the depth from motion parallax approach.

Constraints: Not all video sequences will provide motion parallax to depth. In principle, only video sequences that are captured by a freely moving camera have motion parallax that is closely related to the captured scene structure. If the camera has no motion, the captured video sequence does not have motion parallax. Even if there are some Independently Moving Objects (IMOs) in the scene, their motions will provide some cue to depth under certain circumstances, but this cue could be error-prone. Also, different camera motions will lead to different strengths of depth perception. A freely moving camera can provide more information about the depth in the scene than a camera with a zooming/translating motion along the optical axis. A camera that only rotates around the optical axis does not provide any information about the depth.

Combining Depth cues: As seen above, all monocular depth cues have limitations constraining their utility for 3D Television, thus it would be appropriate to combine cues according to a context-dependent weighted average. The majority of depth cues provide highly correlated quantitative information such as texture gradients and motion parallax. A single estimate of depth needs to be calculated from the variety of cues available. Given that the reliability of each cue is limited to certain conditions, a simple average of all the estimates would be inappropriate [7]. If the value of one variable is informative about the value of the other and the system knows their joint distribution then it would be useful to combine these signals [21]. Modified Weak Fusion (MWF) [22] and Bayesian Theory of combining depth cues [23] are powerful techniques. This may be combined with the 'Depth from X' approach to make the process more tractable for 3D Television [24]. Thus combining depth cues increases efficiency as compared to using single monocular depth cues but at the same time also increases the complexity and processing time of the system.

The Proposed Algorithm:

The proposed algorithm as explained in this section provides a novel method of providing depth information for 3D Television where the input may be arbitrary and wide range of videos without combining depth cues.

The Proposed Algorithm:

1. Convert video to frames.

   For each frame:

2. Separate background & foreground.
3. Count the number of objects in foreground.
4. For each foreground (object) component find the y-axis co-ordinate of the bottom-most pixel to know which component is the front-most & which is behind.
5. Sort the array of detected foreground components based on the y-axis co-ordinate of the bottom-most pixel in descending order. Thus, the top-most value is the front-most object & so on.

6. Count the numbers of distinct levels at which components are present by finding the unique number of y-axis co-ordinates of bottom-most pixels. This equals the number of distinct grey shades required for representing the depth map.
7. Assign white to the front-most component, intermediate grey shades to other components behind it, proceeding towards black for background.

   Shade assignment for foreground components is done as follows:

   Let n: number of distinct levels at which component present

   j: component number

   If j = 1, (nearest component)

   Assign the shade white = 255 (maximum intensity of any greyscale image)

   Else for other components:

   If current component is at same level as previous component,

   Intermediate Grey Shade = ((n-j-1)/n)*255

   Else,

   Intermediate Grey Shade = ((n-j)/n)*255.
8. The resulting image is the generated depth map.

Any existing (including High Definition) video when converted to frames consists of images that have main objects in front of the background. For the front-most object, the rest of the image becomes the background, then for the object behind the front-most, the remaining part of the image becomes the background and so on till we are only left with an approximately single intensity background (for example: Sky in outdoor images or a wall in indoor images).

In addition, it has been observed that the front-most objects which are closest to the viewer will start at the maximum y-co-ordinate as compared to others. For example, consider the following two images:

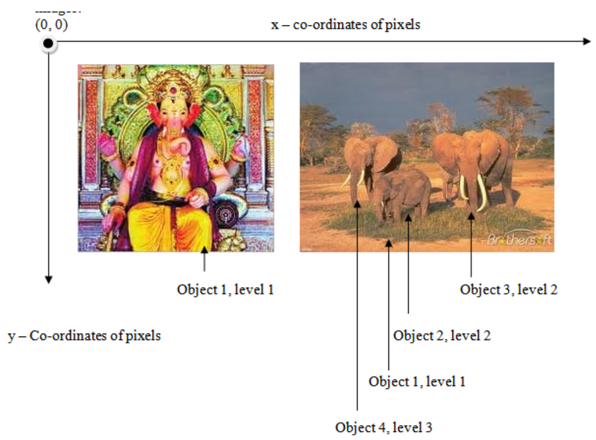


Fig1.Some Identified Objects and Levels in Two Examples of Video Images

As it can be seen from the above images, the objects closest to the viewer have pixels with maximum y-co-ordinates which decrease as the objects move farther away.

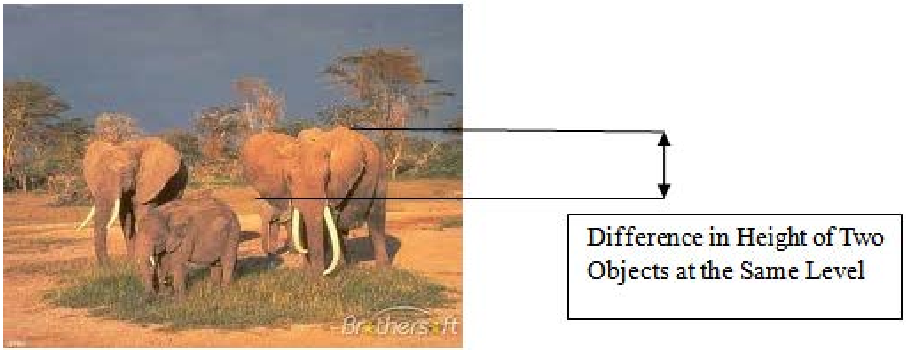


Fig2.Two Objects at the Same Level but with Different Heights

As seen in Figure 2, the two identified objects are at the same level(y-co-ordinate level) i.e. at the same distance from the viewer & hence at the same depth but they have different heights. Hence the bottom-most pixels of each object are considered & not the top-most since though the levels are different both the objects have the same y-co-ordinates for the bottom-most pixels.

Thus, by sorting the array that contains all objects in the descending order results in objects with maximum value of y-co-ordinates i.e. the objects closest to the viewer at the top.

A depth map represents objects in decreasing intensity in greyscale with the objects at the same level requiring the same grey shade to be assigned to them. Hence, after sorting, the numbers of distinct levels are found by comparing the y-co-ordinates of the pixels.

After this, shade assignment i.e. white to the front-most component, intermediate shades to other components behind it, proceeding towards black for background is done to generate the final depth map.

Experimental Results:

Programming Tool: Matlab R2010b

Video input: “Wildlife in HD”, sample video in Windows 7 library

Type : Windows media audio/video file

Size : 25.0 MB

Length : 00:00:30

Step 1 output: 900 frames

Frames 124 to 128 are as shown below:

Fig3.Frame Nos. 124 to 128 of the 900 Frames Generated as Step 1 Output

The maintained High Definition quality in each frame implies the conversion has been efficient. The following are screenshots of step-wise output for frame no. 127 as input.

Fig4.Frame No. 127: Input Frame for Steps 2 – 8

```
info =

           Filename: 'C:\Users\raj\Documents\MATLAB\frames\depth127.jpeg
        FileModDate: '30-Aug-2012 16:05:59'
           FileSize: 129393
             Format: 'jpg'
      FormatVersion: ''
              Width: 1280
             Height: 720
           BitDepth: 24
          ColorType: 'truecolor'
    FormatSignature: ''
    NumberOfSamples: 3
       CodingMethod: 'Huffman'
      CodingProcess: 'Sequential'
            Comment: {}
```

Fig5.Input Frame No. 127 Information

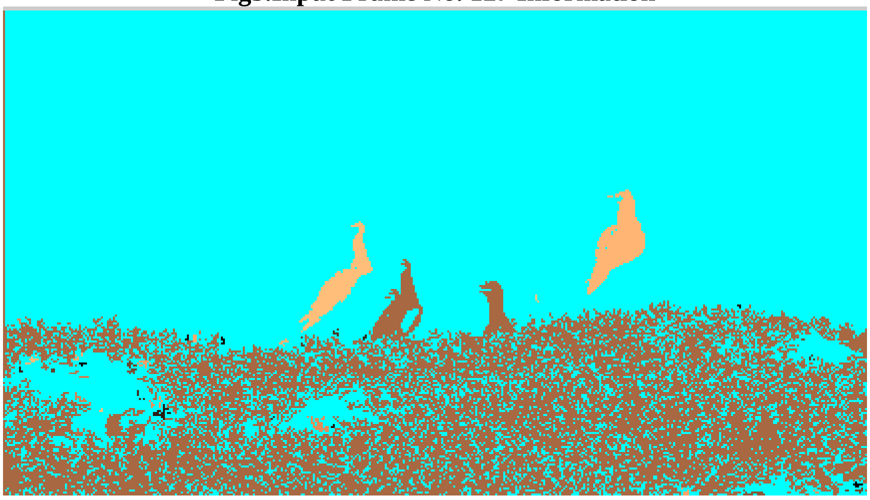


Fig 6.Step 2 Output: Background and Foreground Separated.Light Blue Region Indicates Separated Background and Shades of Brown Indicate the Various Objects in Foreground

```
Command Window

num_objects =

   601


STATS =

601x1 struct array with fields:
    PixelIdxList
    PixelList
    PixelValues
```

Fig7. (a)Step 3 Output: Total Number of Objects in Foreground (b) Statistical Structure Generated for Further Processing based on Total Number of Objects

The statistical structure STATS generated as above gives information about properties of each object such as the list of all pixels in each object, the total number of pixels in each object and so on. These properties are used to find out the bottom-most pixel which is required. The complete list of all pixels present in the 20^{th} object as an example is shown below.

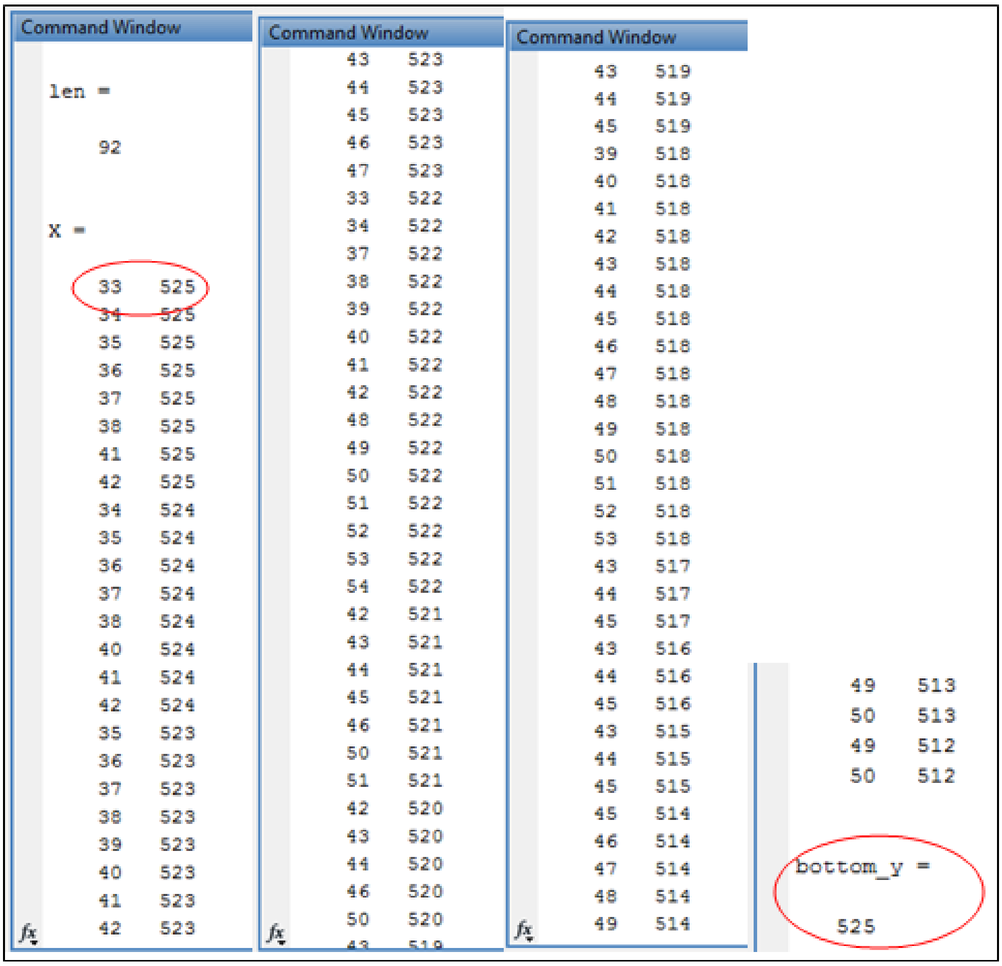
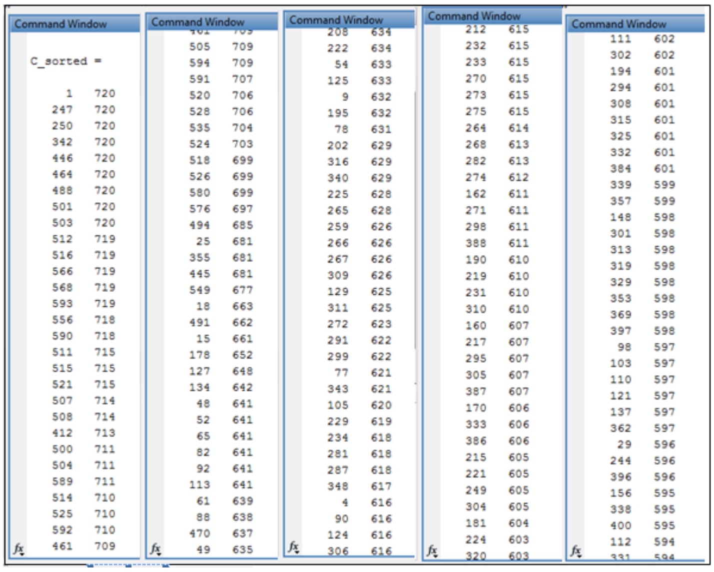
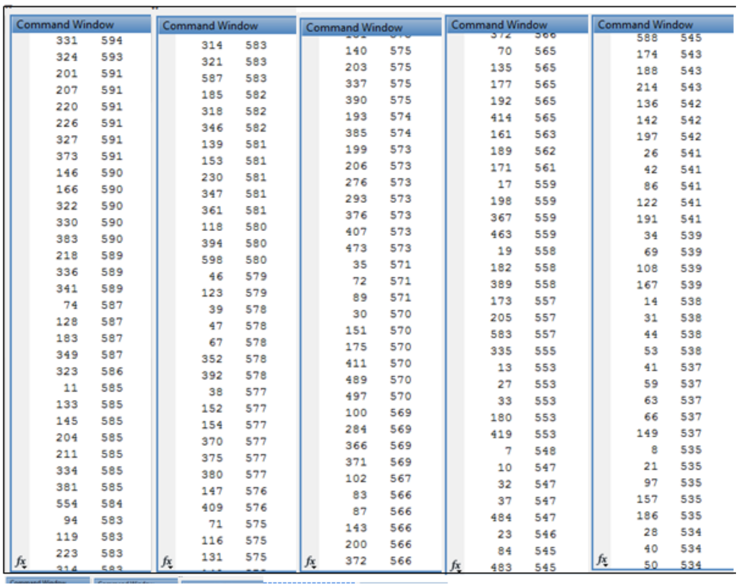


Fig8. Step 4 Output Example: Count of the Number of Pixels in the 20th Object, Sorted Array of All Pixels of the 20th Object, Y-Co-ordinate of the Bottom-most Pixel of the 45th Object

Similar arrays are obtained for all 601 foreground components to identify their respective bottom-most pixels' y-co-ordinates. This array is again sorted as per the y-axis co-ordinates to get the step 5 output as follows:





Command Window

331	594
324	593
201	591
207	591
220	591
226	591
327	591
373	591
146	590
166	590
322	590
330	590
383	590
218	589
336	589
341	589
74	587
128	587
183	587
349	587
323	586
11	585
133	585
145	585
204	585
211	585
334	585
381	585
554	584
94	583
119	583
223	583
314	583

Command Window

314	583
321	583
587	583
185	582
318	582
346	582
139	581
153	581
230	581
347	581
361	581
118	580
394	580
598	580
46	579
123	579
39	578
47	578
67	578
352	578
392	578
38	577
152	577
154	577
370	577
375	577
380	577
147	576
409	576
71	575
116	575
131	575

Command Window

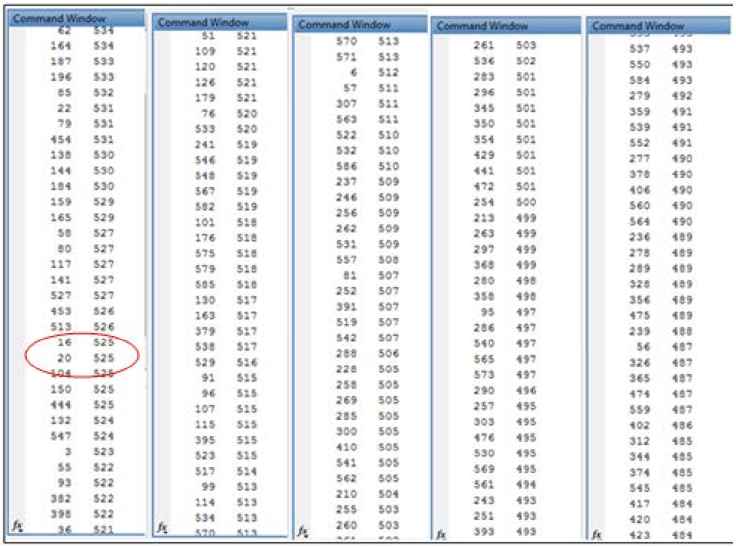
140	575
203	575
337	575
390	575
193	574
385	574
199	573
206	573
276	573
293	573
376	573
407	573
473	573
35	571
72	571
89	571
30	570
151	570
175	570
411	570
489	570
497	570
100	569
284	569
366	569
371	569
102	567
83	566
87	566
143	566
200	566
372	566

Command Window

372	566
70	565
135	565
177	565
192	565
414	565
161	563
189	562
171	561
17	559
198	559
367	559
463	559
19	558
182	558
389	558
173	557
205	557
583	557
335	555
13	553
27	553
33	553
180	553
419	553
7	548
10	547
32	547
37	547
484	547
23	546
84	545
483	545

Command Window

588	545
174	543
188	543
214	543
136	542
142	542
197	542
26	541
42	541
86	541
122	541
191	541
34	539
69	539
108	539
167	539
14	538
31	538
44	538
53	538
41	537
59	537
63	537
66	537
149	537
8	535
21	535
97	535
157	535
186	535
28	534
40	534
50	534



Command Window

62	534
164	534
187	533
196	533
85	532
22	531
79	531
454	531
138	530
144	530
184	530
159	529
165	529
58	527
80	527
117	527
141	527
527	527
453	526
513	526
16	525
20	525
104	525
150	525
444	525
132	524
547	524
3	523
55	522
93	522
382	522
398	522
36	521

Command Window

51	521
109	521
120	521
126	521
179	521
76	520
533	520
241	519
546	519
548	519
567	519
582	519
101	518
176	518
575	518
579	518
585	518
130	517
163	517
379	517
538	517
529	516
91	515
96	515
107	515
115	515
395	515
523	515
517	514
99	513
114	513
534	513
570	513

Command Window

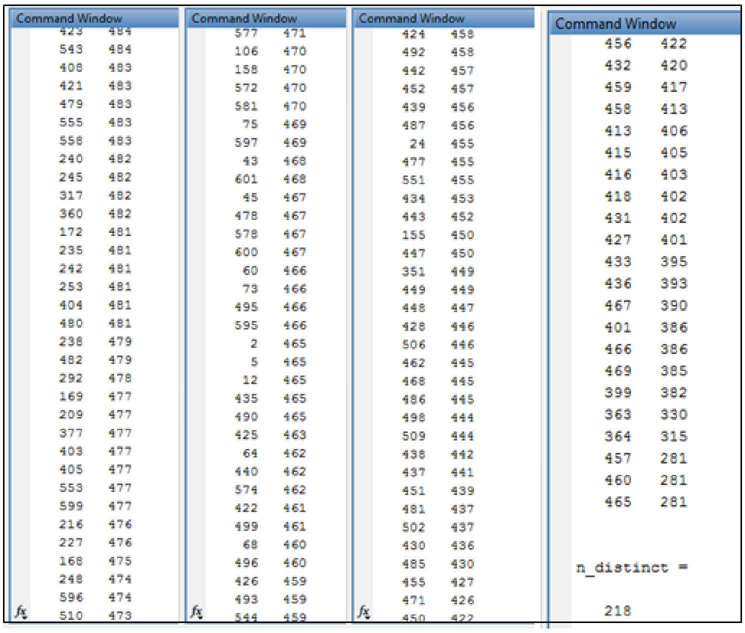
570	513
571	513
6	512
57	511
307	511
563	511
522	510
532	510
586	510
237	509
246	509
256	509
262	509
531	509
557	508
81	507
252	507
391	507
519	507
542	507
288	506
228	505
258	505
269	505
285	505
300	505
410	505
541	505
562	505
210	504
255	503
260	503

Command Window

261	503
536	502
283	501
296	501
345	501
350	501
354	501
429	501
441	501
472	501
254	500
213	499
263	499
297	499
368	499
280	498
358	498
95	497
286	497
540	497
565	497
573	497
290	496
257	495
303	495
476	495
530	495
569	495
561	494
243	493
251	493
393	493

Command Window

537	493
550	493
584	493
279	492
359	491
539	491
552	491
277	490
378	490
406	490
560	490
564	490
236	489
278	489
289	489
328	489
356	489
475	489
239	488
56	487
326	487
365	487
474	487
559	487
402	486
312	485
344	485
374	485
545	485
417	484
420	484
423	484



Command Window

423	484
543	484
408	483
421	483
479	483
555	483
558	483
240	482
245	482
317	482
360	482
172	481
235	481
242	481
253	481
404	481
480	481
238	479
482	479
292	478
169	477
209	477
377	477
403	477
405	477
553	477
599	477
216	476
227	476
168	475
248	474
596	474
510	473

Command Window

577	471
106	470
158	470
572	470
581	470
75	469
597	469
43	468
601	468
45	467
478	467
578	467
600	467
60	466
73	466
495	466
595	466
2	465
5	465
12	465
435	465
490	465
425	463
64	462
440	462
574	462
422	461
499	461
68	460
496	460
426	459
493	459
544	459

Command Window

424	458
492	458
442	457
452	457
439	456
487	456
24	455
477	455
551	455
434	453
443	452
155	450
447	450
351	449
449	449
448	447
428	446
506	446
462	445
468	445
486	445
498	444
509	444
438	442
437	441
451	439
481	437
502	437
430	436
485	430
455	427
471	426
450	422

Command Window

456	422
432	420
459	417
458	413
413	406
415	405
416	403
418	402
431	402
427	401
433	395
436	393
467	390
401	386
466	386
469	385
399	382
363	330
364	315
457	281
460	281
465	281

n_distinct =

218

Fig10. Step 5-6 Output: Sorted List of All Objects in the Descending Order of Y-Co-ordinate of the Bottom-most Pixel and the Number of Distinct Levels. Note the highlighted component no. 20 and its pixel as found in the previous step

This list is used to find position of each object with respect to the viewer. The objects with same bottom-most pixels values indicate they lie at the same level i.e. the same distance from the viewer. Thus the same grey shade is allotted to them. In order to find the number of shades required for this image, the number of distinct levels is found as shown in the above figure. After shade assignment as explained in the algorithm, the final depth map generated is as follows:

Fig11. Step 7-8: Generated Depth Map for Frame No. 127

For more testing, a synthetic image was generated with objects of different shapes, sizes, colours and positions. The above procedure was repeated for this image. The image with its corresponding depth map is as follows:

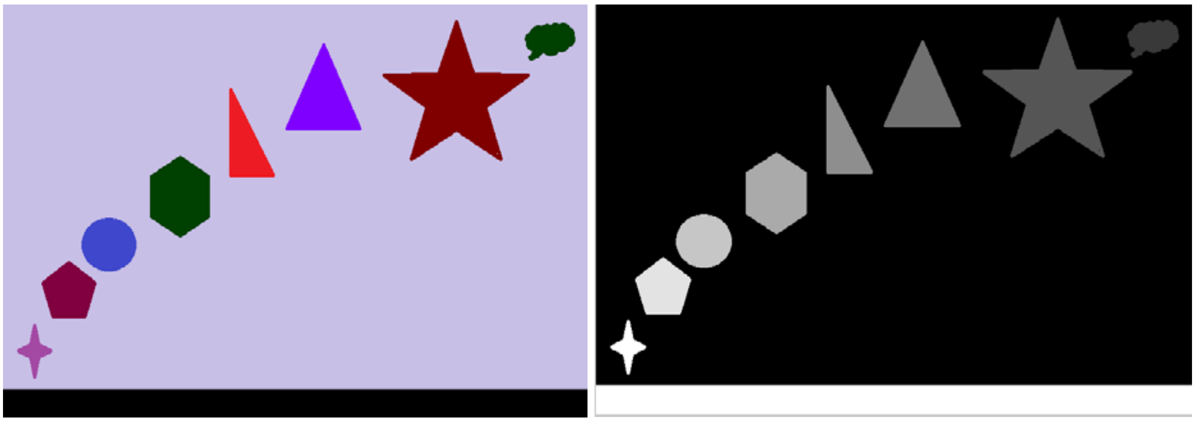


Fig12.Experimental Image No.2: Created Synthetic Image and Its Generated Depth Map

The experimental results for a variety of video frames are as follows:

Fig13.Experimental Video Frames and Generated Depth Maps[1]

The above results on various different types of inputs highlights the efficiency of the algorithm to work on arbitrary videos without requiring any assumptions. Observe the slight tilt in the first video frame (figure 13). The system has taken into consideration this tilt by assigning intermediate grey shades to objects along the tilt thus efficiently works at the slightest difference in pixel levels. The objects in the background do not require any processing for conversion to stereo hence even when identified form redundant information. This algorithm excludes background objects from shade assignment, thus increasing execution speed further.

Conclusion

Creating enough S3D material with stereoscopic cameras for 3D television is time consuming & expensive. Thus the conversion of the vast collection of already existing 2D images/videos to S3D is essential for 3D television. However extracting 3D information from arbitrary 2D video is intractable at present since methods either make strong assumptions on the 2D video (e.g., a static scene) or use human interactions to train a huge database of prior knowledge. The proposed algorithm alleviates this problem by using a novel technique that can be used over a wide range of videos without the constraints faced when using a single monocular depth cue. At the same time, it avoids the usage of multiple depth cues, thus

[1]Observe the slight tilt in the first video frame (figure 13). The system has taken into consideration this tilt by assigning intermediate grey shades to objects along the tilt thus efficiently works at the slightest difference in pixel levels. Also this algorithm excludes background objects from shade assignment, thus increasing execution speed further, reducing the time consumed without affecting the efficiency of the depth maps produced. The experimental results of the proposed algorithm tested on a High Definition video as also a variety of video frames verify the effectiveness of the technique. The accuracy of converting video to frames, background & foreground separation as also identify foreground objects play a crucial role in the depth map generated. This algorithm excludes background objects from shade assignment, which form redundant information, thus increasing execution speed further.

Furthermore, it is intended to generate depth map for inter-frame information to increase the execution speed and ytilize the generated depth maps along with original video frames to create stereoscopic videos.

References:

Liang Zhang, Senior Member, IEEE, Carlos Vázquez, Member, IEEE, and Sebastian Knorr, "3D-TV Content Creation: Automatic 2D-to-3D Video Conversion", in IEEE Transactions on Broadcasting, Vol. 57, June (2011)

G. Guo, N. Zhang, L. Huo, and W. Gao, "2D to 3D conversion based on edge defocus and segmentation," in IEEE Int. Conf. Acoust., Speech Signal Process., Mar. 31–April, 4 (2008), pp. 2181–2184

YUVSOFT 3D video, 2d to 3d conversion process. Available: http://www.yuvsoft.com/stereo-3d-technologies/2d-to-s3d-conversion-process/(2012)

J. Ens and P. Lawrence, "An investigation of methods of determining depth from focus," IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, no.2, pp. 523–531, (1993)

G. Guo, N. Zhang, L. Huo, and W. Gao, "2D to 3D conversion based on edge defocus and segmentation," in IEEE Int. Conf. Acoust., Speech Signal Process., Mar. 31–April, 4 (2008), pp. 2181–2184

Favaro, "Shape from focus/defocus," Washington University, Department of Electrical Engineering Electronic Signals and Systems Research Lab, Jun. 25, (2002)

George Mather, Depth Perception in "Foundations of sensations and perception", Psychology Press, 2nd edition, UK., 8th December, (2008) pp. 296 – 318

D. A. Forsyth, "Shape from texture without boundaries," in ECCV02, (2002), pp. 225–239.

A. M. Loh and R. Hartley, "Shape from non-homogeneous, non-stationary, anisotropic, perspective texture," in Proc. British Mach. Vis. Conf., (2005)

R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 8, pp. 690–706, (1999)

F. R. R. Mallory, Types of Monocular Depth Cues Available: http://www.ehow.com/about_5042449_types-monocular-depth-cues.html (2011)

S. Battiato, S. Curti, M. La Cascia, M. Tortora, and E. Scordato, "Depth map generation by image classification," in Proc. SPIE, Apr. (2004), vol. 5302, pp. 95–104

X. Huang, L. Wang, J. Huang, D. Li, and M. Zhang, "A depth extraction method based on motion and geometry for 2D to 3D conversion," in 3rd Int. Symp. Intell. Inf. Technol. Appl., (2009), pp. 294–298

John H. Krantz, Ph.D.,June (2012), Linear Perspective, Sensation & perception tutorials Available : http://psych.hanover.edu/krantz/art/linear.html

Ton Derksen," Discovery of linear perspective & its limitations" in Philosophica 63 (1999, 1) pp. 19-50 Available: http://logica.ugent.be/philosophica/fulltexts/63-3.pdf

Y.-J. Jung, A. Baik, and D. Park, “A novel 2D-to-3D conversion technique based on relative height-depth-cue,” in SPIE Conf. Stereoscopic Displays Appl. XX, San José, CA, (2009), vol. 7237, 72371U

K. Yamada, K. Suehiro, and H. Nakamura, “Pseudo 3D image generation with simple depth models,” in Int. Conf. Consum. Electron., Dig. Techn. Papers, Jan. 8–12, (2005), pp. 277–278

F. Cozman and E. Krotkov, “Depth from scattering,” in IEEE Conf.Comput. Vis. Pattern Recog. (CVPR), (1997), pp. 801–806.

W. J. Tam, C. Vázquez, and F. Speranza, “3D-TV: A novel method for generating surrogate depth maps using colour information,” in SPIE Conf. Stereoscopic Displays Appl. XX Conf., San José, CA, (2009), vol. 7237.

C. Vázquez and W. J. Tam, “CRC-CSDM: 2D to 3D conversion using colour-based surrogate depth maps,” in Int. Conf. 3D Syst. Appl. (3DSA), Tokyo, Japan, May (2010).

Frank Jäkel and Marc O. Ernst, “Learning to Combine Arbitrary Signals from Vision and Touch “

Michael S Landy, Laurence T Maloney, Elizabeth B Johnston, Mark Young, “Measurement and modelling of Depth Cue Combination: in defense of weak fusion”, psych.nyu.edu

David C. Knill, “Motor Perception”, Journal of Vision

Paul R. Schrater and Daniel Kersten, “The Role of Task Specification in Optimal Cue Integration”