Actuality
Systems

Public Information Disclosure

Actuality Ref:	ASI-0066
Title	INTERLACED MULTIPLANAR VOLUMETRIC DISPLAYS
Publication Date	13 November 2005
Inventors	Won-Suk Chun (Cambridge, MA)
Point of Contact	Gregg Favalora, CTO favalora [at] actuality-systems.com +1 (781) 541-6016 x102
Notes	

INTERLACED MULTIPLANAR VOLUMETRIC DISPLAYS

[0001] Using interlacing to increase the performance and quality of volumetric displays.

BACKGROUND

[0002] There are many types of 3D displays, including stereoscopic displays, multiplanar volumetric displays (for example U.S. 6,554,430, "Volumetric three-dimensional display system") holographic video systems (for example U.S. 5,172,251, "Three dimensional display system"), and multi-view 3D displays. Applications of 3D displays include the depiction of medical images, such as a transparent CT image of a patient's anatomy which includes vasculature and tumors; geophysical data for the petroleum industry, such as seismic data overlaid with drill paths; or 3D luggage scan data, for example the CT scan of luggage in which each 3D pixel ("voxel") is color coded as a function of effective atomic number.

DESCRIPTION OF THE FIGURES

[0003] Fig. 1 is a view of an image displayed on a 3D display.

[0004] Fig. 2 is a chart illustrating the relationship between slice location and slice number.

[0005] Fig. 3 is a chart illustrating the relationship between slice location and slice number.

[0006] Fig. 4 is a chart illustrating the relationship between slice location and slice number.

[0007] Fig. 5 is a chart illustrating the relationship between slice location and slice number.

[0008] Fig. 6 is a diagram of color slice display.

[0009] Fig. 7 is an RGB LCD display.

[0010] Fig. 8 is a top view of a 3D display.

[0011] Fig. 9 is a top view of a 3D display showing color interlacing.

DESCRIPTION OF THE INVENTION

[0012] 3D volumes can be decomposed into a sequence of 2D slices. Physical realizations of 3D displays (so-called multiplanar volumetric displays) frequently multiplex the sequence of 2D slices in various positions over time. For example, the Actuality Systems, Inc. Perspecta display (see ‘430) uses a high-speed projector synchronized with a spinning screen. For example, a volumetric 3D image composed might be composed of 200 radially-disposed slices, each with a resolution of 768 x 768 pixels. A volume is fully projected when the sequence of 200 slices is optically projected on to the spinning screen as it rotates 180 degrees. If each projected 2D slice is approximately centered on the spinning screen, each 2D slice will subtend slightly less than 1 degree of a cylindrical coordinate system.

[0013] Normally, the volume is optically refreshed at 30 Hz. This implies that the 2D refresh rate of the embedded 2D projector is (30 volumes / second) x (200 2D slices / volume) = 6000 2D slices / second.

[0014] In this particular system, there are several bottleneck regimes. Similar multiplanar displays suffer from similar performance-limiting factors.

[0015] Problem: Limited Radial Resolution

[0016] The embedded 2D projector’s frame bandwidth can limit the radial resolution of the 3D display. The number of voxels in a volumetric display can be several orders of magnitude greater than that of a typical 2D display. Since display technologies are developed towards 2D requirements, it is a recurring challenge to achieve a data bandwidth surplus to 2D spatial light modulators (SLM) so that the excess can be used to multiplex the slices into a 3D display. Because SLM data communication bandwidth can be a limiting factor, volumetric display performance often scales with SLM bandwidth. Factors that improve the use of SLM bandwidth translates directly to improved volumetric display performance.

[0017] Problem: Computational bottleneck to embedded projector (SLM)

[0018] Data communication to the SLM can be limited by the rate at which the volumetric data can be computationally generated. Oftentimes, the format of the volumetric image does not match the native format of the volumetric display. This is typically true when rendering images describes as curves and surfaces (as is generated by the OpenGL API), or when the display employs an unusual coordinate system. In this case, some kind of raster conversion or image resampling must take place. Since graphics processing technologies (so-called graphics processing units or GPUs perform 3-D computation but ultimately project their results into 2D) are developed towards 2D requirements, it is a recurring challenge to achieve a computational bandwidth surplus to 2D graphics processors so that the excess can be used to perform computation on the multiplexed slices.

[0019] There are also qualitative performance issues.

[0020] Problem: Spoked appearance when looking down on display

[0021] There is an unwanted sampling artifact in the radial dimension which appear as “spokes” in the display space when looking down from above. Certain SLMs, particularly the high-speed digital mirror devices (DMDs), do not continuously transition from one image to another. Instead, there is a limited duty cycle during which the SLM can produce lit voxels. This is manifest as gaps between image slices. If the slices can be arranged such that they partially overlap, then these gaps can be eliminated.

[0022] Problem: Jittery image due to mechanical and software-bsaed miscalibration

[0023] Some 3D displays with imperfect "calibration" produce images that appear to jitter back and forth. In some 3D displays because the image slices are rotating about a shared axis, they form a double covering of the volume. This means that pairs of slices 180 degrees apart actually completely overlap. Because of this, the images projected on these slices must be carefully calibrated so that the images are aligned to each other. In practice, a perfect alignment is unlikely. Misalignment between these pairs of images results in apparent oscillations within the volume ("jitter"). Furthermore, this exact double coverage also creates a waste in spatial resolution that exacerbates the SLM bandwidth performance constraints.

[0024] It is possible to achieve a partial alignment and maintain the double coverage, eliminating both gaps and jitter in the volume while improving the bandwidth efficiency of the volumetric display.

[0025] This disclosure describes several methods to:

[0026] Interlace slices in volumetric displays to:

- In a compute-limited volumetric display scenario:
  - Provide progressive updates of a 3D image
  - Increase temporal resolution
- In a transfer-limited volumetric display scenario:
  - Increase spatial resolution
- Improve voxel connectivity between 2D slices
- Decrease apparent motion-jitter of 3D images

Interleave color channels in volumetric displays to:

Increase spatial resolution of time-multiplexed volumetric displays

Decorrelate luma and chroma perceptual motion cues

Combining these methods

[0027] See Figure 1, which illustrates a 3-D image (10) created by a 3-D display (15) represented as a sequence of 2-D slices (20). The 3-D display generates the 3-D image by rapidly displaying the individual 2-D slices in different positions rapidly enough so that they all appear to be simultaneously displayed to a human observer through persistence effects. This is the method of operation of the Perspecta display as taught in U.S. Pat. No. 6,554,430.

[0028] See Figure 2, which illustrates the relationship between slice number and slice location. The first graph (30) illustrates that in the case of Perspecta (‘430), slice number *q* corresponds to the angle of rotation *theta* about an axis of rotation. However, because *theta* represents the angle of rotation for a diameter rather than a radius, any angle *rho* represents the same physical location as *rho* + pi. The second graph (40) illustrates the relationship between slice number and slice location accounting for this detail.

[0029] One solution is the use of slice-based interlacing, as discussed here.

[0030] There are several approaches to slice-based interlacing.

[0031] See Figure 3, which illustrates the relationship between time and slice number. Because this relationship is periodic in time, only one period is shown. Typical 3D displays will iterate through each 2D slice in a fixed, cyclic order. The first graph (50) illustrates the relationship between displayed time *t* and the 2D slice *q* in the obvious, linear scan order. The second graph (60) illustrates the relationship between displayed time *t* and the 2D slice *q* with 2-way interlacing, and the third graph (70) with 3-way interlacing.

[0032] See Figure 4, which composes the functions described in figures 2 and 3. The first graph (80) illustrates the relationship between displayed time *t* and 2D slice location in linear order, the second graph (90) with 2-way interlacing, and the third graph (100) with 3-way interlacing.

[0033] The first important feature of Figure 4 to observe is that most of the range of the 3D display is covered in a fraction of the time in the interlaced cases compared to the linear scan order in an amount equal to the reciprocal of the degree of interlacing.

[0034] Additional benefits of the disclosure

[0035] One can exploit this result in several ways. In a compute-limited 3-D display scenario, if each slice can be computed independently but slowly, then incrementally displaying slices in an interlaced order is superior to the linear scan order because the user can see more different parts of the 3-D image. The user is free to interact with the 3-D image before it has completed drawing, so getting a broad sampling of the complete scene as early as possible increases the rate off user interaction.

[0036] If a large fraction (e.g. half) of the slices can be rendered during a display period, then this technique can be used to provide animation at the full display rate by updating an interlaced portion of the scene at a time. There is a limit to using this animation technique; certain scenes can create motion artifacts, just as there are motion artifacts due to interlacing in NTSC. These can be reduced or eliminated by image processing that band-limits the image in space and/or time.

[0037] The second feature of Figure 4 to observe is that in the case when the degree of interlacing is an even divisor of the number of slices as in graph (90), the actual physical locations are not effectively interleaved. In some displays, this is further exacerbated by the double coverage of slices to physical locations.

[0038] Several issues can be solved simultaneously by eliminating the double-coverage of slice numbers to slice locations. See Figure 5. The diagram illustrates the arrangement of 2D slices in another 3-D display. If only an odd number of slices per revolution are used, then each location is visited once instead of twice. Effectively, this doubles the number of spatially addressable locations in the display because slices do not overlap in a single rotation. The graph illustrating the relationship between displayed time *t* and spatial location *theta* (110) illustrates that, furthermore, the slices are effectively already interlaced in the linear scan order.

[0039] This has a number of repercussions on the properties of the 3-D display. Since slices no longer exactly overlap, there is no need to align the images on them, simplifying calibration and reducing or eliminating jitter caused by slice misalignment. Because the number of physically addressable locations effectively doubles, the SLM bandwidth requirements are effectively halved for a given resolution. The spare bandwidth (slices/second) can be used to increase the spatial resolution (slices/volume) and/or the refresh rate (volumes/second) of the 3D display yielding a higher quality image.

[0040] The partial overlap also eliminates gaps in the volumetric image due to the limited SLM duty-cycle as shown in Figure 8.

[0041] Interlacing can also occur between color channels as shown in Figure 9.

[0042] Another opportunity for interlacing is between color channels. In some displays, each color channel is handled in parallel by separate DMDs. See Figure 6. If the timing of each of the slices is offset relative to each other, then 1 three-color slice (120) can generate 3 single-color slices (130, 140, 150). If the offset between two of these single-color slices is less than the offset between two single-color slices of the same color, then each of the single-color slices occupies a unique physical position, effectively increasing the resolution of the display.

[0043] A similar technique is used in 2D displays. See Figure 7. An LCD flatscreen monitor (160) has separate red, green and blue subpixels (180, 190, 200) for each pixel (170), typically arranged as regular columns. It is possible to increase the apparent resolution of an LCD by treating each colored subpixel as a distinct location. For example, this is used in ClearType and other font anti-aliasing schemes. Because each subpixel can only produce color of a single channel, the image must be band-limited to the original, full pixel resolution.

[0044] However, this band-limiting step doesn't entirely remove the benefit of the higher subpixel resolution. First of all, because the image is effectively super-sampled, the band-limited image is higher quality than an image naively created at the original resolution. Second of all, the phase resolution is actually higher, because a full color pixel can be displaced at the subpixel level. Third of all, band-limiting can be performed perceptually; each of the perceptual color channels do not need to be filtered to the same degree. These factors, in combination, increase the perceptual resolution of an LCD display.

[0045] Sub-slice interlacing can also be used in multiplanar volumetric displays.

[0046] An analogous technique can be used to perform sub-slice interlacing on a multi-planar 3-D display to increase perceptual resolution. Just as ClearType and other font anti-aliasing schemes band-limit supersampled columns of colored subpixels, you can band-limit supersampled slices of colored subslices.

[0047] The different techniques for interlacing are essentially independent and can be combined.

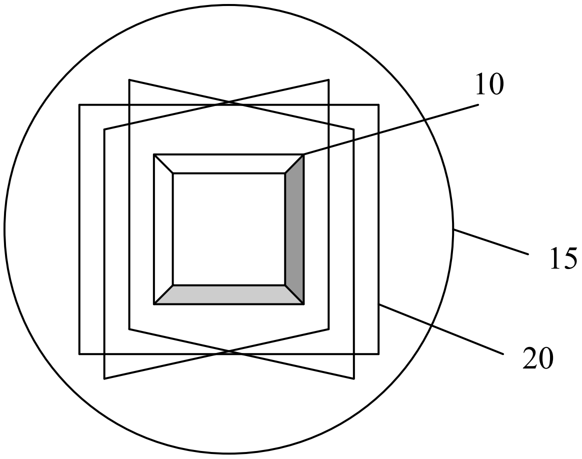
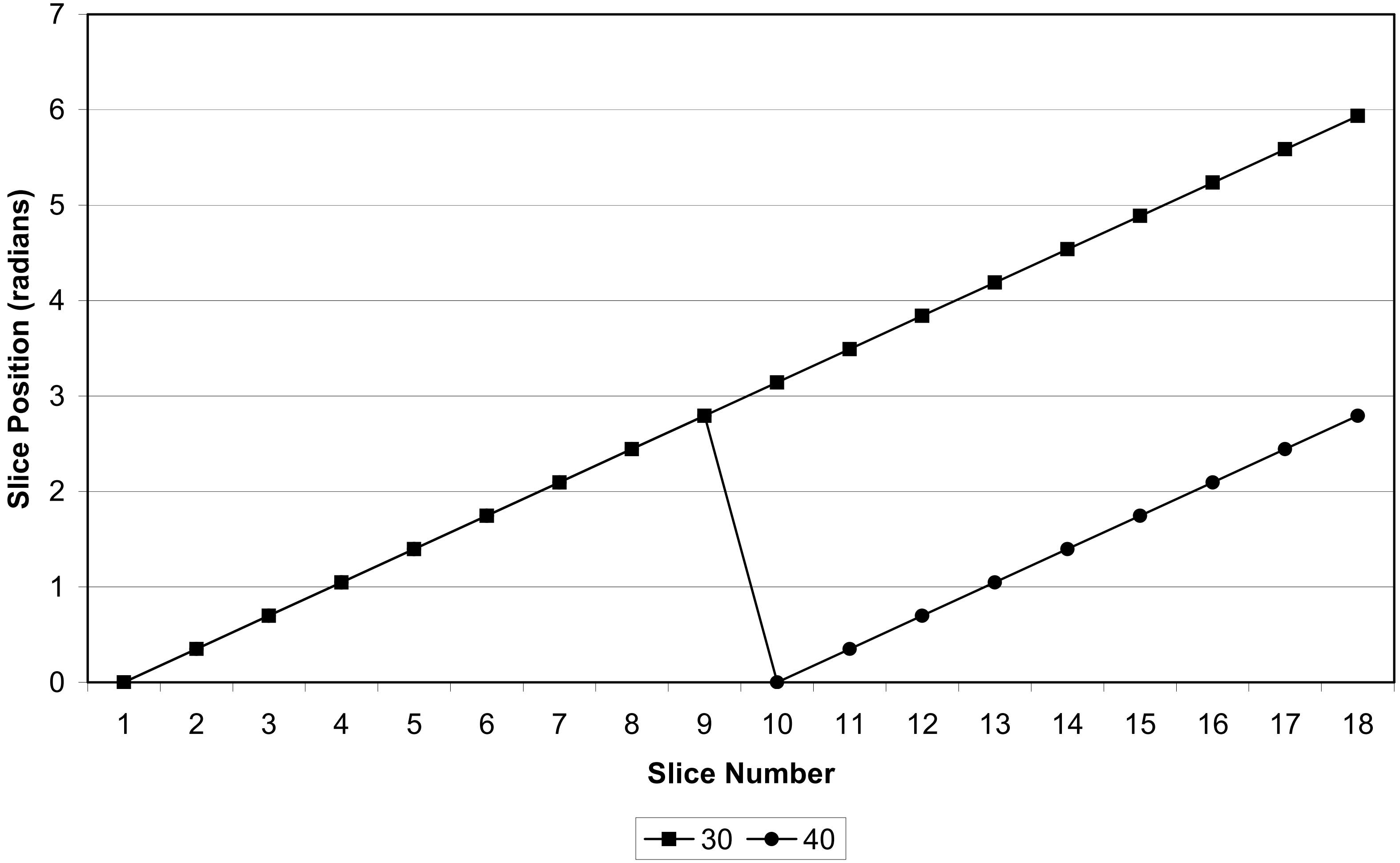


FIGURE 1

Figure 2


Slice Position (radians)
0
1
2
3
4
5
6
7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Slice Number
30
40

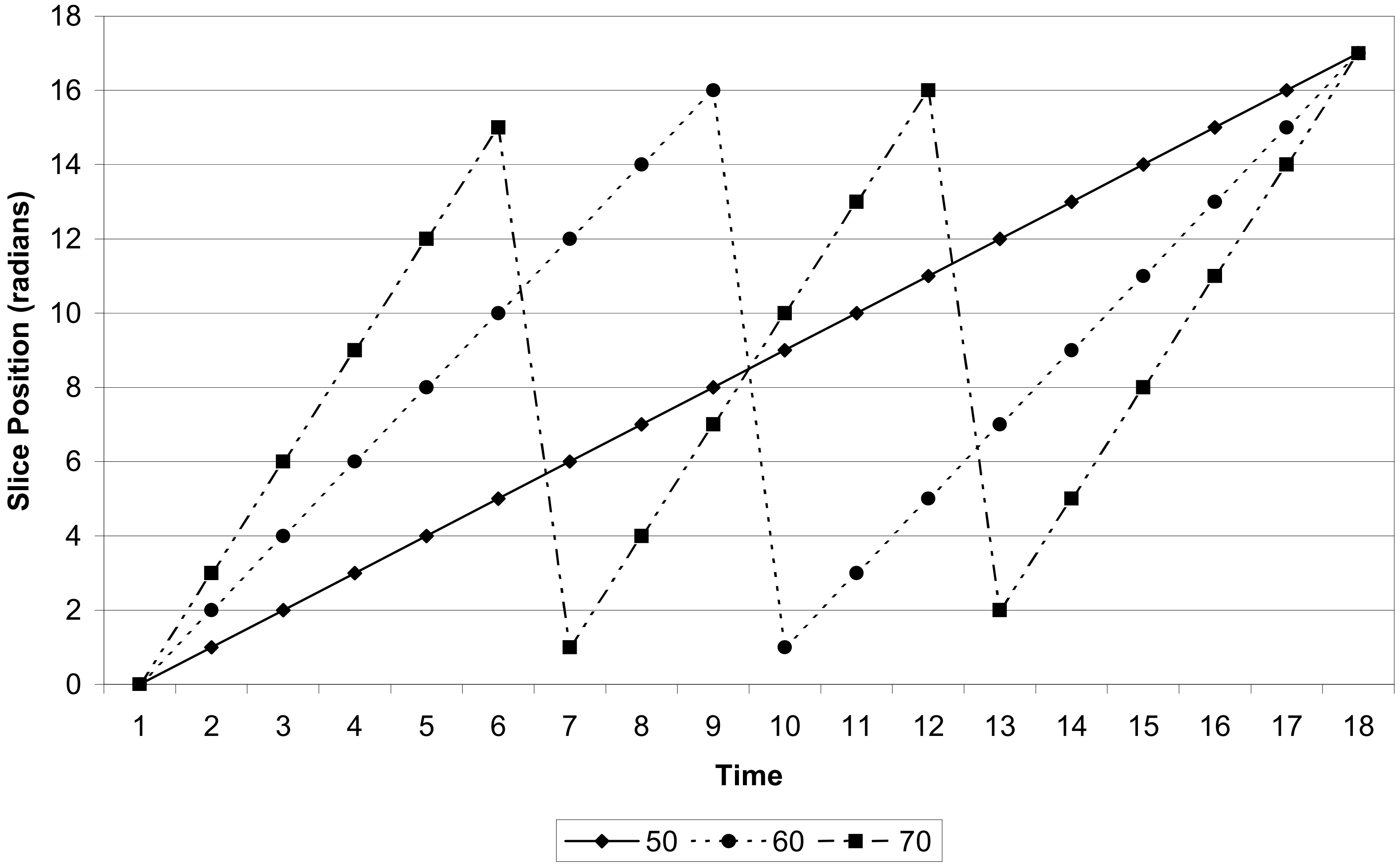
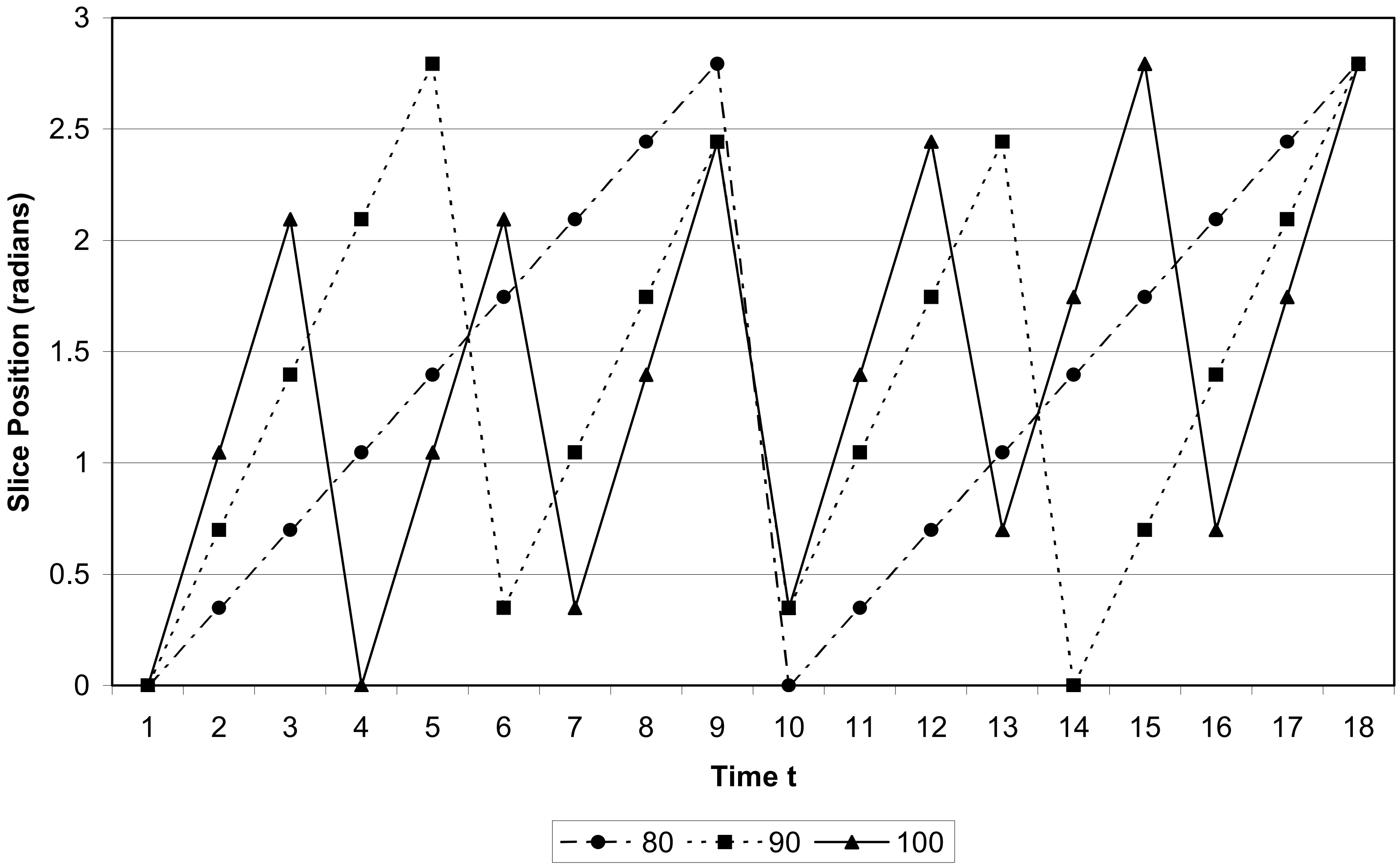


Figure 3

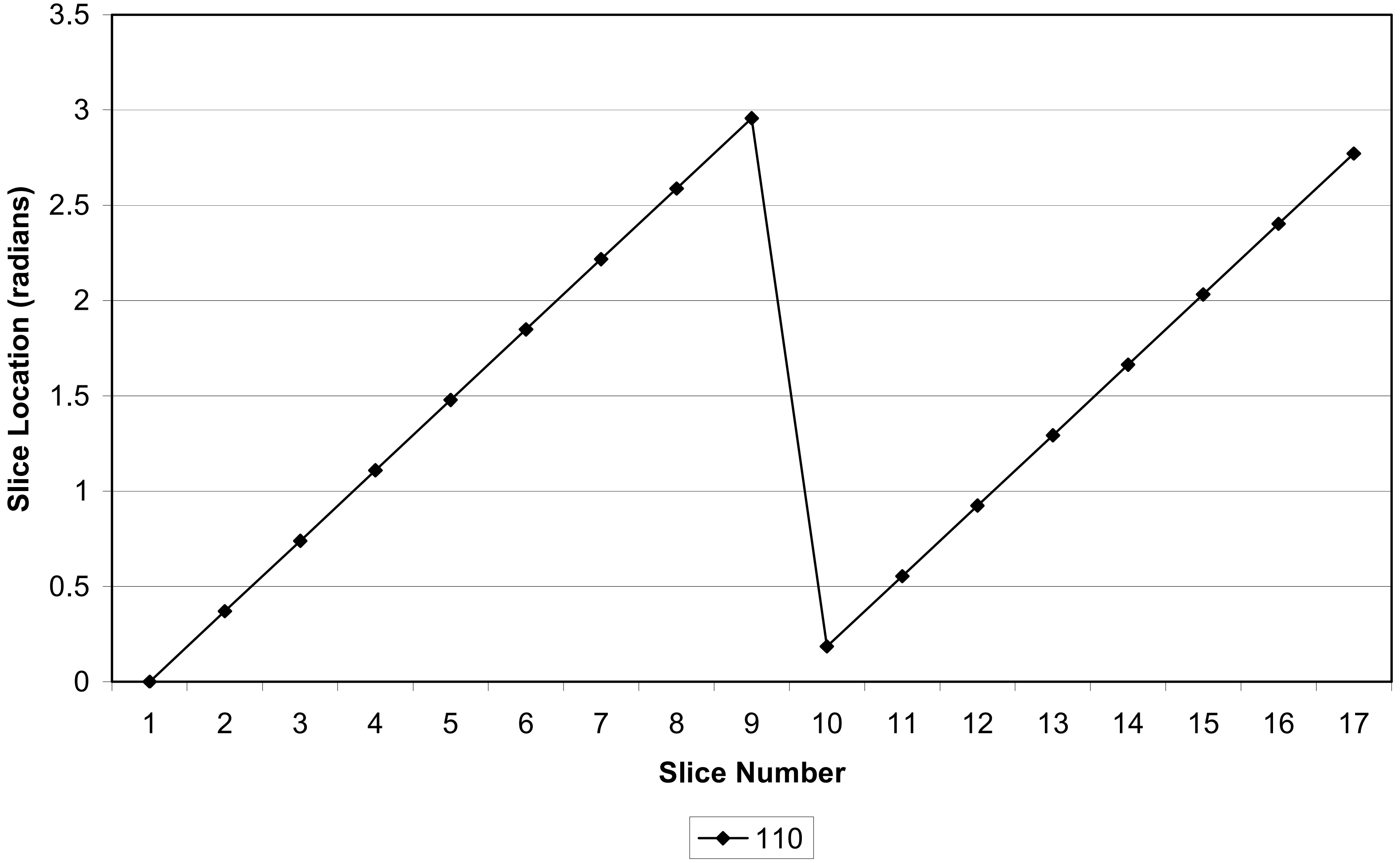
Slice Position (radians)
0
2
4
6
8
10
12
14
16
18
Time
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
50
60
70

Figure 4



Slice Position (radians)
3
2.5
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Time t
80
90
100

Figure 5


Slice Location (radians)
3.5
3
2.5
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
Slice Number
110

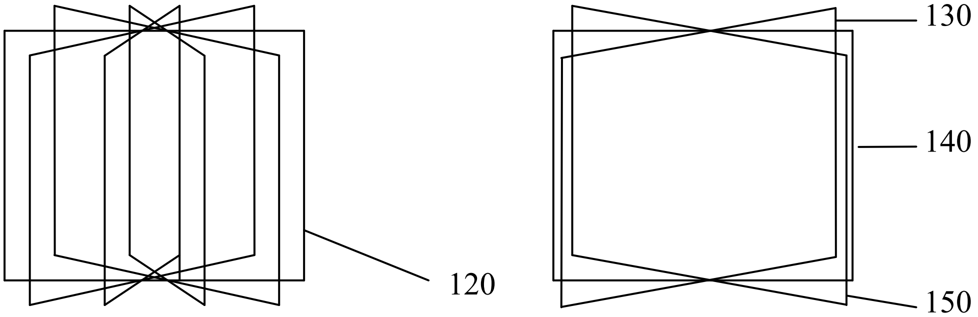


FIGURE 6

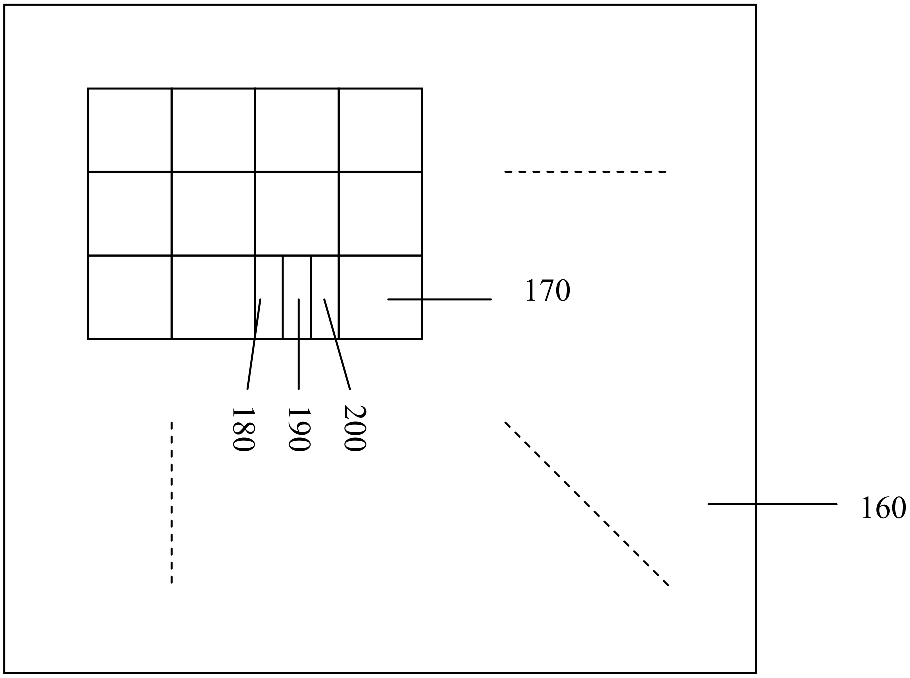


FIGURE 7

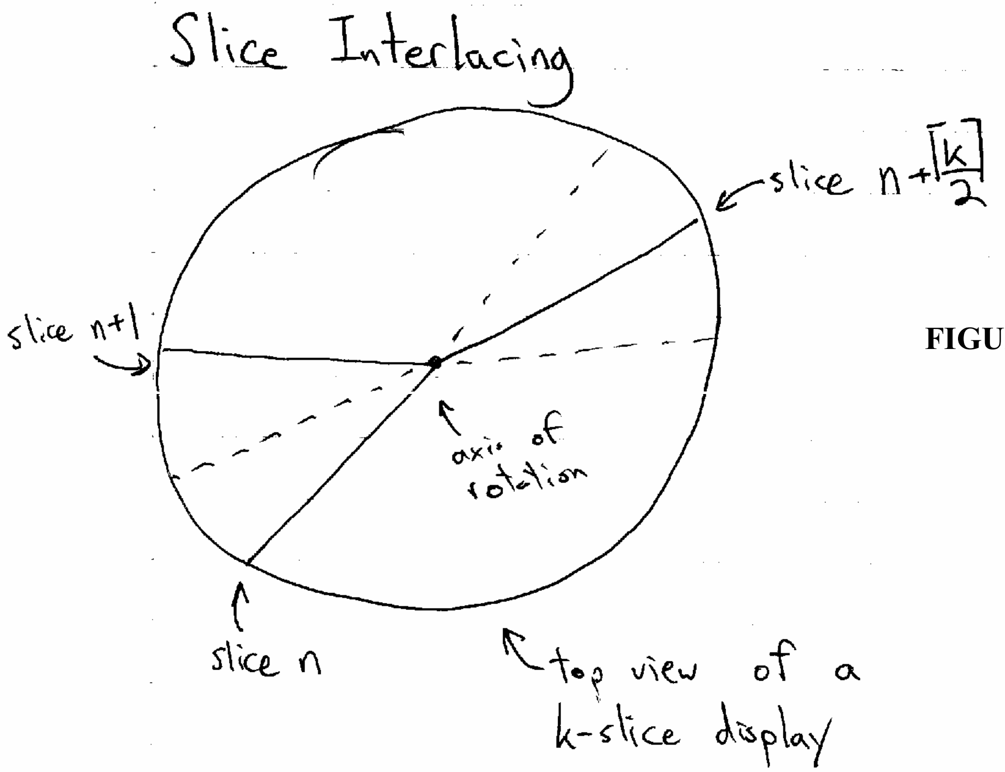


FIGURE 8

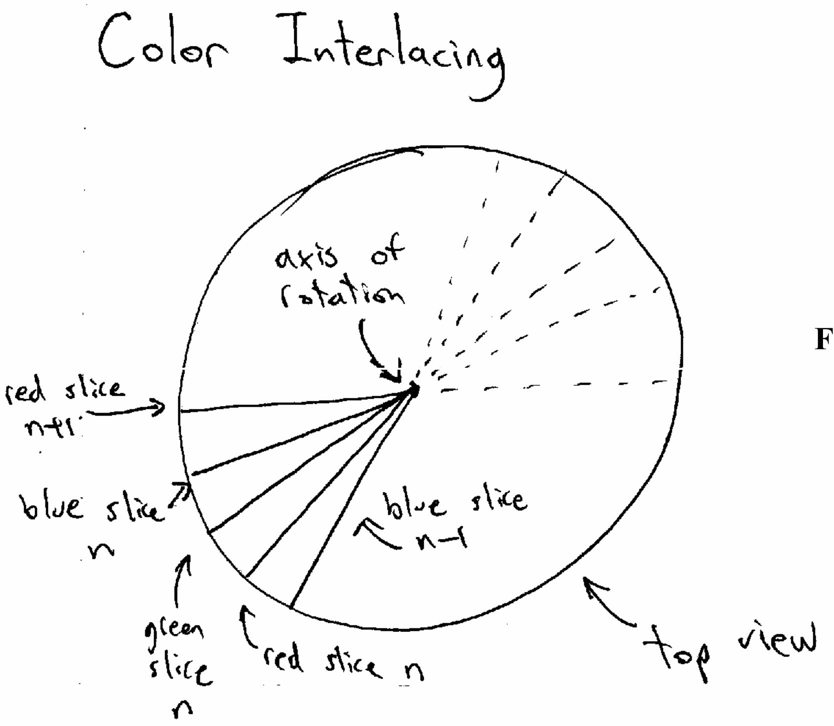


FIGURE 9