SEGMENTS
(Chapter 7 in *Computer Graphics*)

- Segment Concepts
- Segment Files
- Segment Attributes
- Multiple Workstations

Segment Concepts

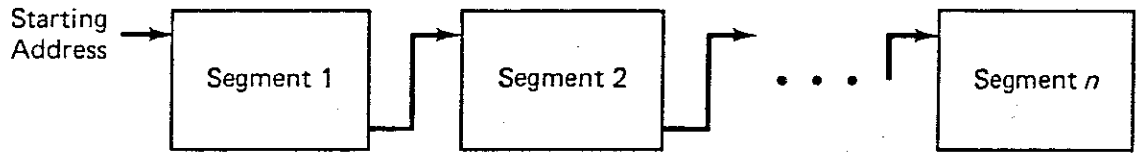
- efficient to define and modify a picture as a set of subpictures
- a segment is a set of output primitives that are joined for modification purposes
- segment commands
  - create_segment (id)
  - close_segment
  - delete_segment (id)
  - rename_segment (id_old, id_new)
- example:
  - delete_segment (6);
  - create_segment (6);
    polyline (n, x, y);
    text (xt, yt, "graphics");
  - close_segment;
  - rename_segment (6, 9);

Segment Concepts, continued

- sometimes provided
  - copy_segment (id) (into an open segment)
- generally not provided
  - reopen_segment (id)
  - append_to_segment (id)

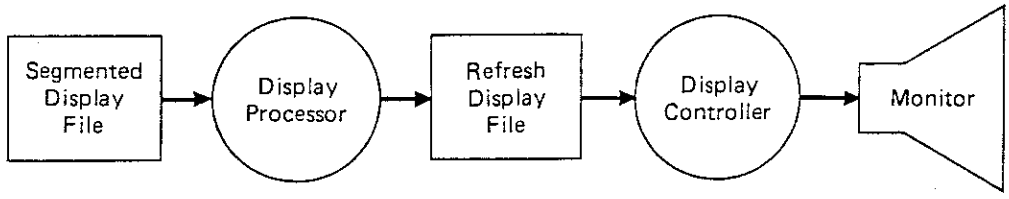
Segment Files

- a segment file is any list of segments maintained by a graphics system
- segment files often are stored as linked structures

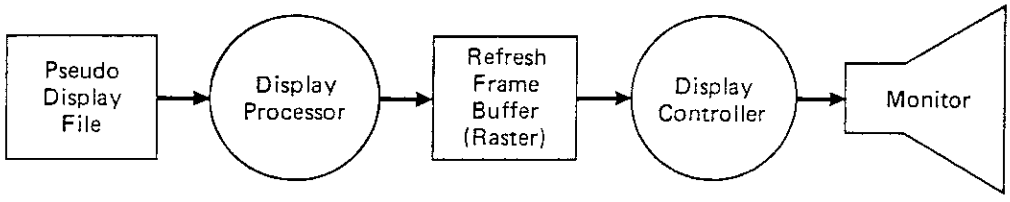


Segment Files, continued

- several forms
  - a segmented display file is a display file program for a simple vector system



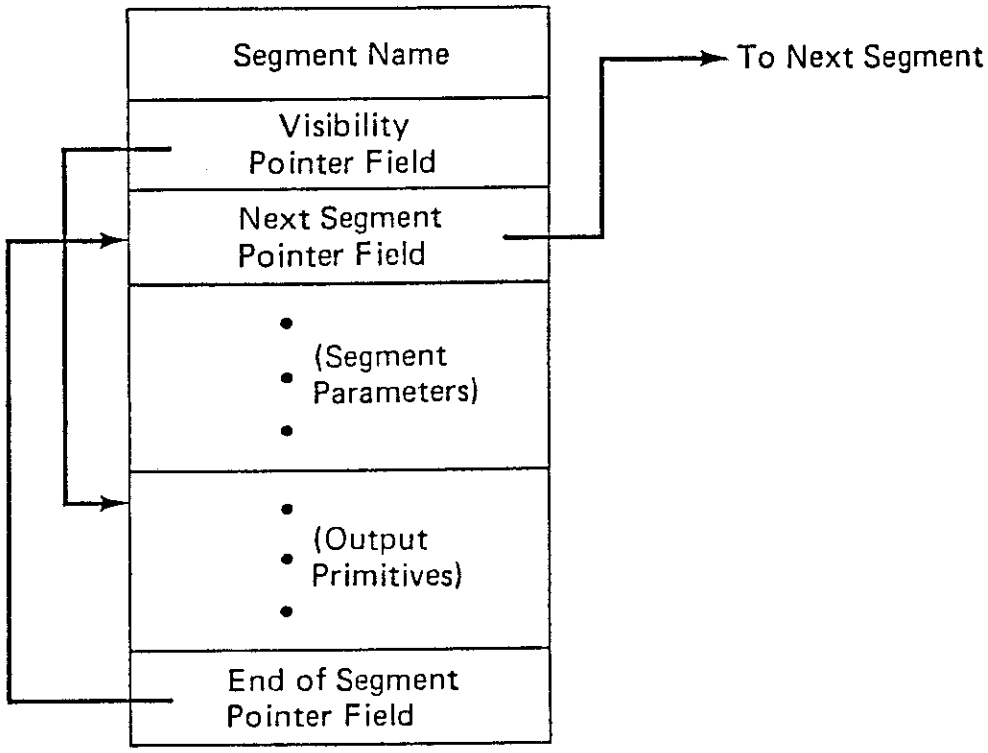
- a pseudo display file contains segment definitions from which appropriate bits in the frame buffer are set



memory management for segment storage

- blocks must be assigned as segments are created
- blocks must be returned to the storage pool as segments are deleted
- options
  - fixed-size blocks
    - easy to manage
    - lead to fragmentation
  - variable-sized blocks
    - avoid fragmentation
    - more memory management

segment format



- make changes only at the end of the refresh cycle

Segment Attributes

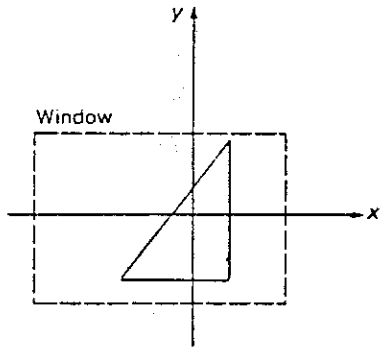
- visibility
  - set_visibility (id,v)
    v = visible (posted) or invisible (unposted)
- priority
  - set_segment_priority (id,p)
  - used for raster-scan systems
- highlight
  - set_highlight (id,h)
    h = highlighted or normal
- size, position and orientation
  - set_segment_transformation (id,matrix)

Segment States

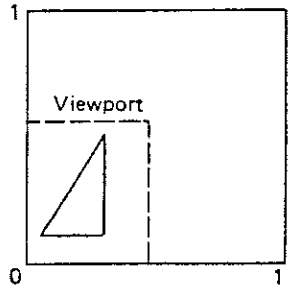
- a segment can be
  - painted
  - unpainted

avoiding repeated window-to-viewport mappings

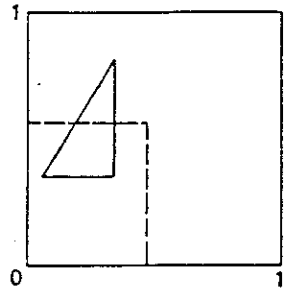
- convert from world coordinates to normalized device coordinates
- transform in normalized device coordinates
- clip against viewport boundaries
- store in a refresh file or refresh frame buffer



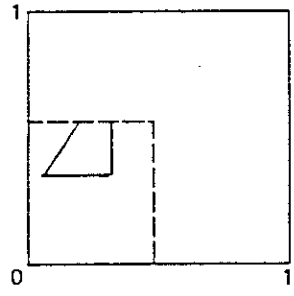
(a)
World Coordinates
Description



(b)
Segment Mapped to
Normalized Coordinates



(c)
Transformation Matrix
Applied in Normalized
Coordinates



(d)
Segment Clipped in
Normalized Coordinates

Multiple Workstations

- each output device is identified with a unique workstation number
- controlling the display of segments is accomplished by activating and deactivating workstations
- example

```
activate_workstation (5);
create_segment (12);
        .
        .
        .
close_segment;
activate_workstation (2);
create_segment (13);
        .
        .
        .
deactivate_workstation (5);
        .
        .
        .
```

Multiple Workstations, continued

- additional commands
  - clear_workstation (ws)
  - delete_segment_from_workstation (ws, id)
  - redraw_segments_on_workstation (ws)
    - when an overlapping segment is erased, this command restores the overlapped segments on raster-scan systems
  - copy_segment_to_workstation (ws, id)

long-term storage

- a metafile is a file used for long-term storage of graphical information
  - used by several graphics packages

SEGMENTS

- Segment Concepts
- Segment Files
- Segment Attributes
- Multiple Workstations