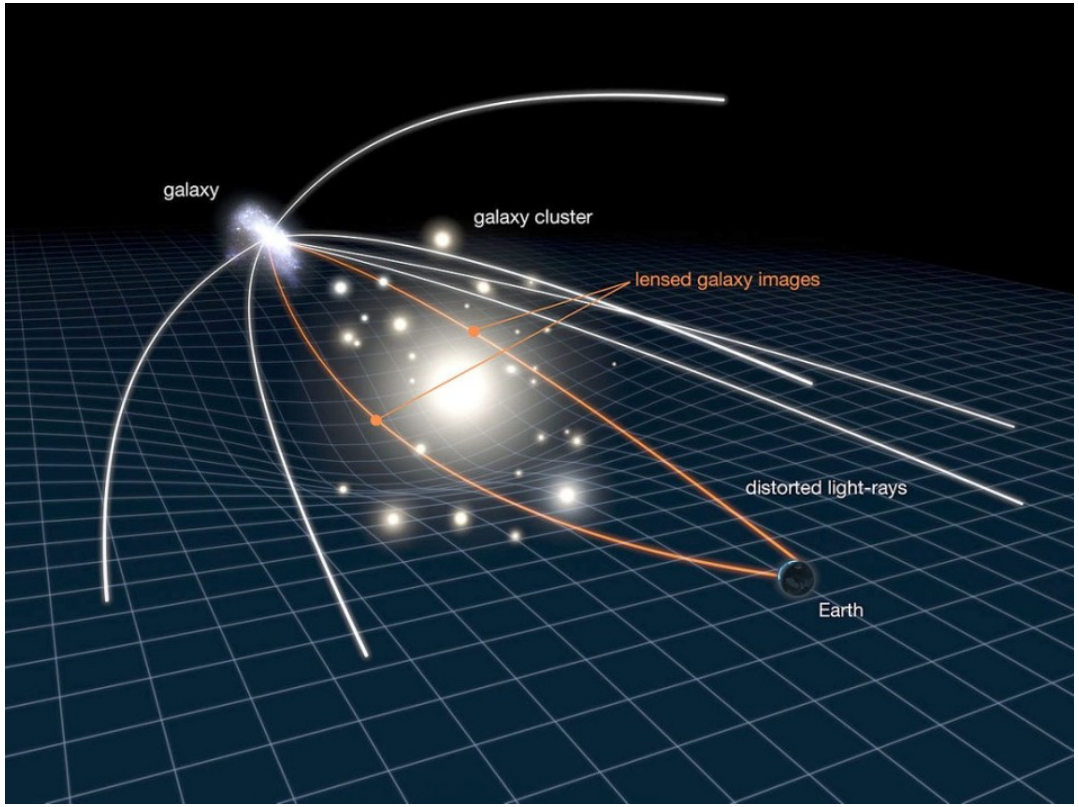
블랙홀 실험실

2018 국가슈퍼컴퓨팅 청소년캠프 실습 과제

2018.07.17. @ 울산과학기술원

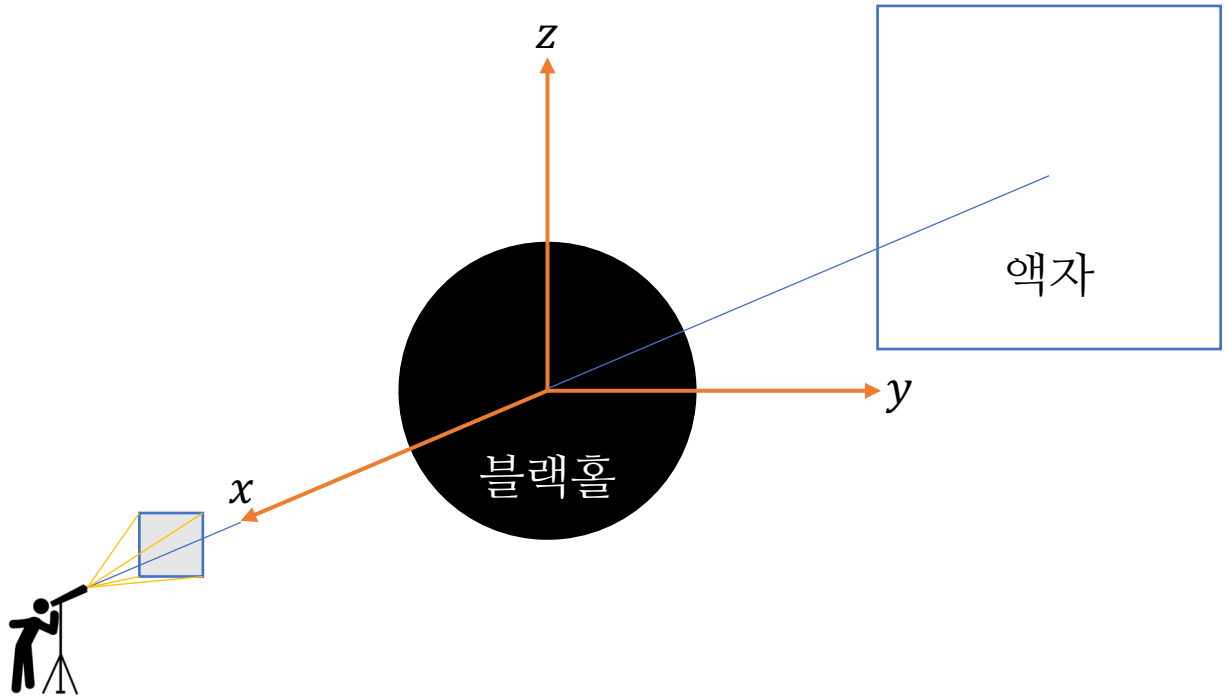
블랙홀

- 한번 들어가면 절대 빠져 나올 수 없는 시공간 영역
- 블랙홀 주변을 지나는 빛은 구부러진 시공간을 따라가며 경로가 휘게 됨.



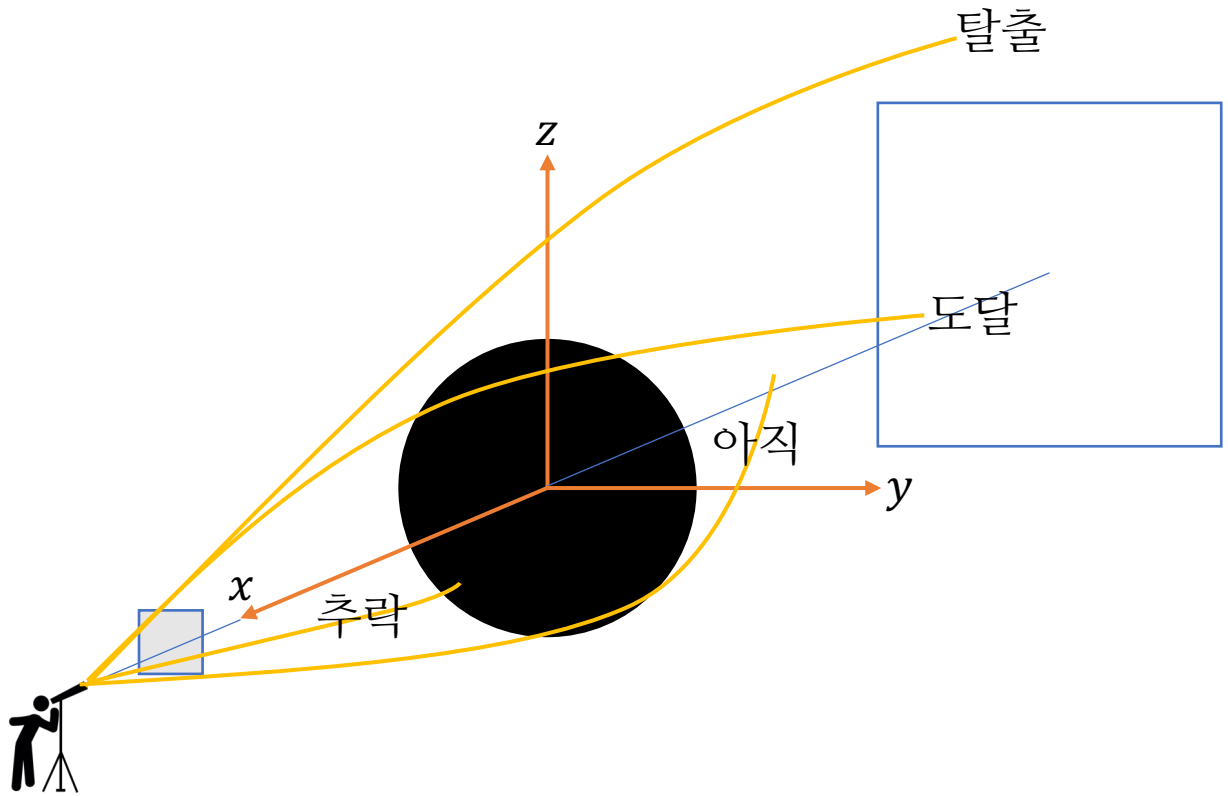
실험실 구조

- 그림이 그려진 액자와 이를 바라보고 있는 관찰자 사이에 블랙홀을 둔다.
- 블랙홀로 인해 관찰자가 보게될 왜곡된 이미지를 시뮬레이션 한다.



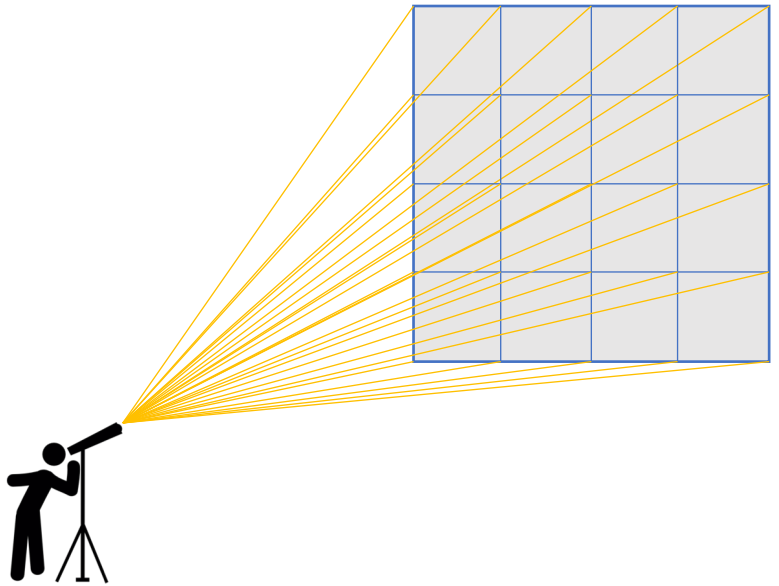
시뮬레이션 방법

- 관찰자 앞에 놓인 화면으로 입사되는 빛 다발의 과거 궤적을 추적하여 어느 지점에서 온 빛인지를 판단한다.



이미지를 얻는 방법

- 화면의 해상도가 가로 세로 픽셀수로 주어진다.
- 모든 픽셀마다 해당 지점으로 입사되는 빛의 과거 궤적을 추적한다.
- 액자에서 나온 빛으로 판별되면 해당 위치로 부터 색 정보를 얻어와 픽셀에 표시한다.



실습 1 : 컴파일 및 실행

- 압축 해제 : `$ tar -xf assignment.tar`
- 디렉토리 이동 : `$ cd assignment/code`
- 컴파일 : `$ make`
- 실행 : `$ ./bhlab`

실습 2 : 이미지 만들기

- 디렉토리 이동 : `$ cd ../tool/`
- 컴파일 : `$ make`
- 데이터 검증 : `$ ./tool ../code/data`
- 이미지 제작 : `$ ./tool ../code/data input1.png output1.png`
- 이미지 제작 : `$ ./tool ../code/data input2.png output2.png`

실습 3 : 해상도 변경 및 블랙홀 회전 추가

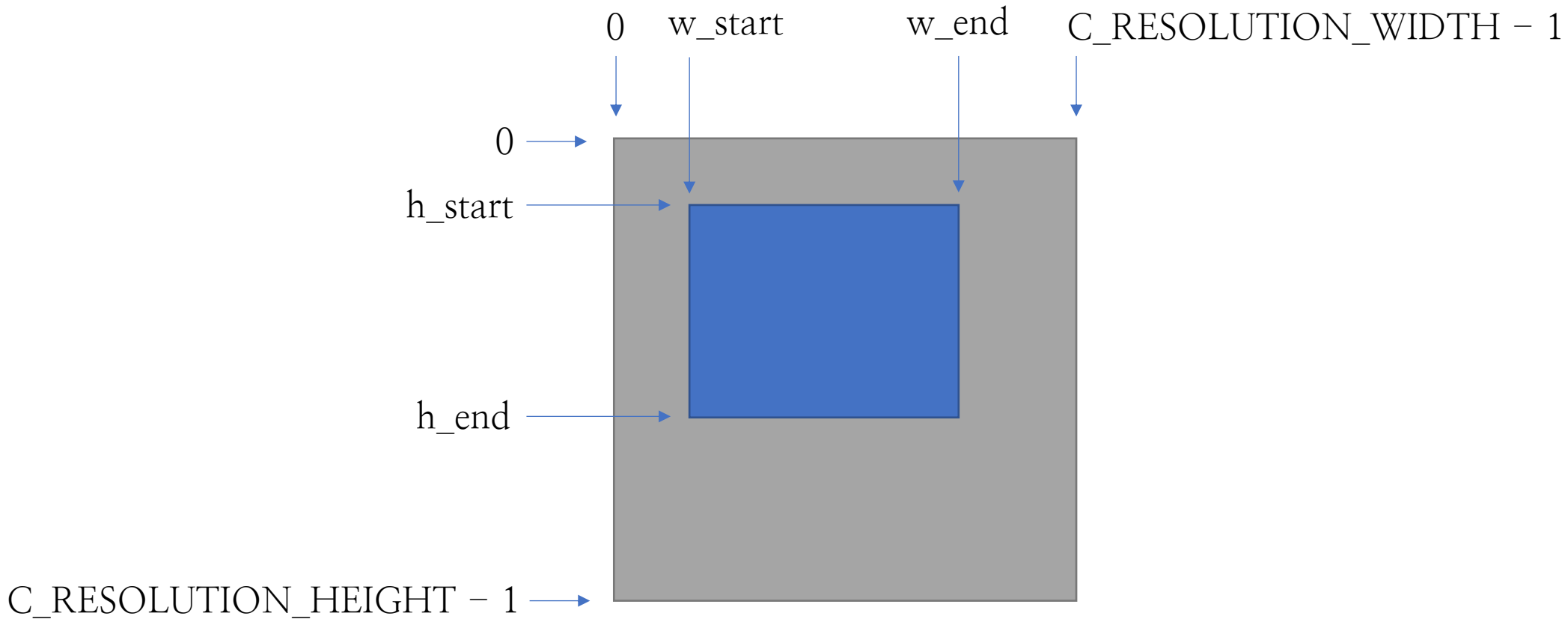
- 디렉토리 이동 : `$ cd ../code/`
- Makefile 수정 : `$ vi Makefile`
  - KERR = -DKERR_SPACETIME : 회전 여부
  - WIDTH = -DC_RESOLUTION_WIDTH=400 : 가로 해상도
  - HEIGHT = -DC_RESOLUTION_HEIGHT=400 : 세로 해상도
- 다시 컴파일 : `$ make clean; make`
- 기존 data 보관 : `$ cp data ../data_serial`
- 실행 : `$ ./bhlab`
- 기존 data와 비교 : `$ ../tool/tool data ../data_serial`

소스코드 소개 : main.c의 run 함수

- `void run(int w_start, int w_end, int h_start, int h_end, int *status, double *y, double *z)`
  - (w_start, w_end, h_start, h_end)의 픽셀 범위에 대해 빛의 과거 경로를 추적하여 추적 결과(도달, 탈출, 추락, 아직)를 status에 저장하고, 액자에서 나온 빛의 경우 출발 지점의 좌표를 y와 z에 저장한다.
  - w_start와 h_start는 각각 가로, 세로 픽셀의 시작 인덱스를 나타내며 인덱스는 0부터 시작한다.
  - w_end와 h_end는 각각 가로, 세로 픽셀의 끝 인덱스를 나타낸다.
  - status, y, z는 각각 독립된 배열 공간을 가리키고 있어야 한다.
  - 배열의 크기는 모두 총 픽셀수(C_RESOLUTION_TOTAL_PIXELS)여야 한다.

소스코드 소개 : main.c의 run 함수

- (w_start, w_end, h_start, h_end)의 픽셀 범위



소스코드 소개 : main.c의 export 함수

- `void export(int *status, double *y, double *z)`
  - status, y, z에 저장된 정보를 파일(C_OUTPUT_FILENAME)로 출력한다.
  - status, y, z는 각각 배열 공간을 가르키고 있어야 한다.
  - 배열의 크기는 모두 총 픽셀수(C_RESOLUTION_TOTAL_PIXELS)여야 한다.

과제2: MPI 병렬화 (점대점 통신)

- 주어진 serial 코드를 MPI 병렬화 하시오.
- 수정 부분 : main.c의 main 함수만 수정
- MPI_Recv와 MPI_Send를 사용
- 실행 : `$ mpirun -np 16 -host host01,host02, host03,host04 ./bhlab`

과제3: MPI 병렬화 (집합 통신)

- 주어진 serial 코드를 MPI 병렬화 하시오.
- 수정 부분 : main.c의 main 함수만 수정
- MPI_Gather를 사용
- 세로 픽셀의 갯수는 MPI 프로세스 숫자로 나누어 떨어진다.
- 실행 : `$ mpirun -np 16 -host host01,host02,host03,host04 ./bhlab`