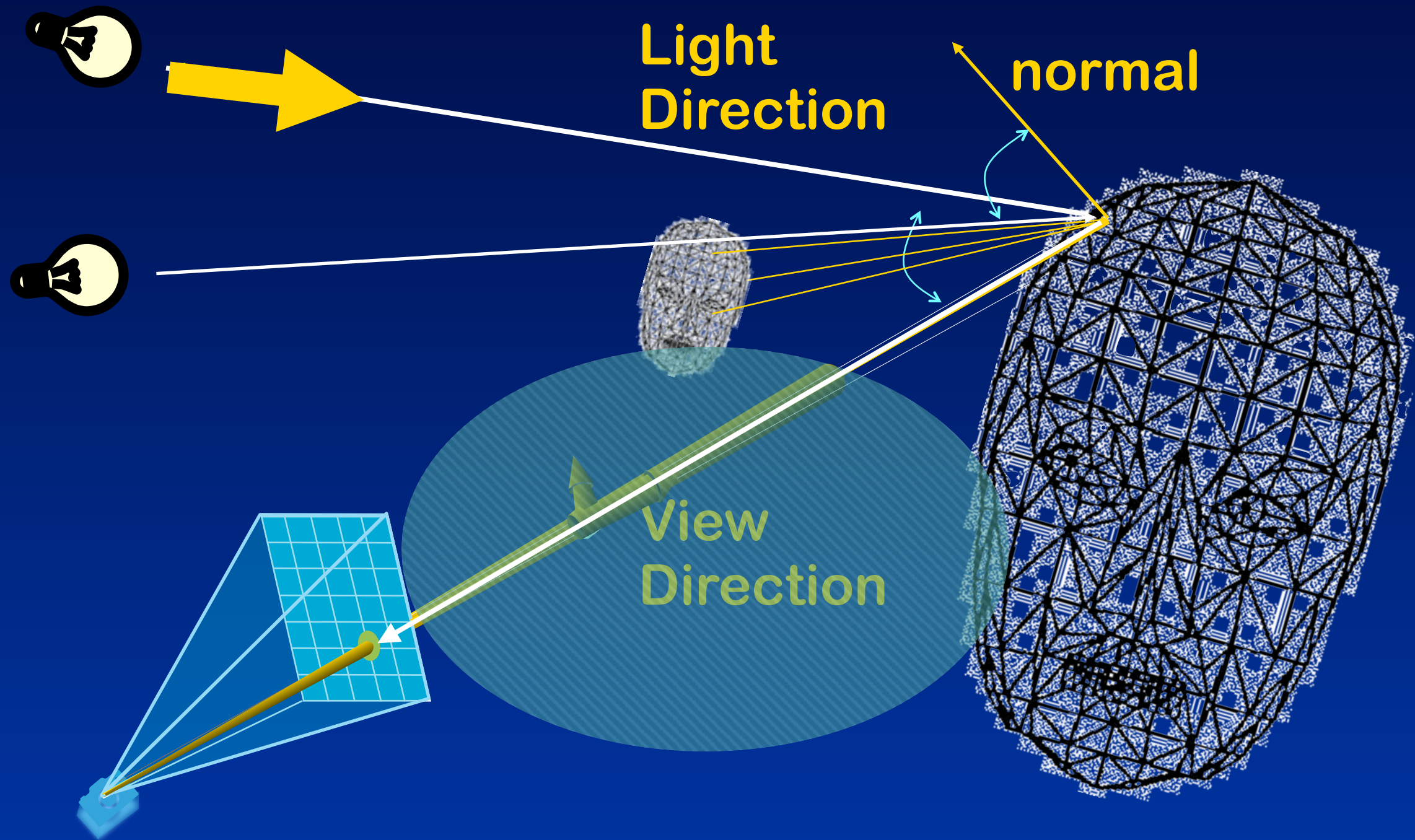


SUBODH KUMAR

HIGH PERFORMANCE COMPUTATION & GRAPHICS

RAY TRACING



HIGH PERFORMANCE GRAPHICS

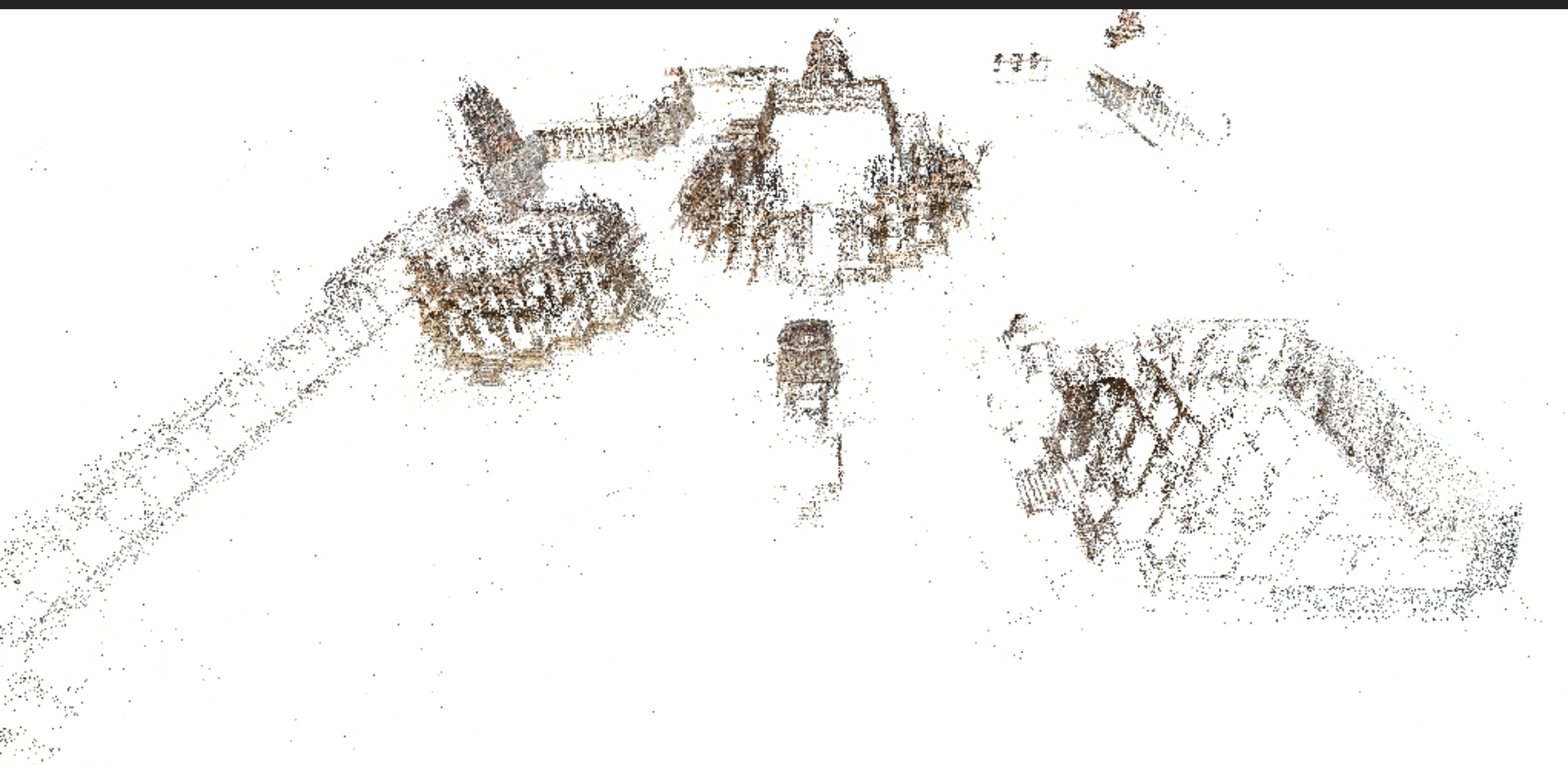
VIRTUAL REPLICAS

Historic Vittala Temple (Vijayanagara)



Simplypush.com





HIGH PERFORMANCE GRAPHICS

VIRTUAL SURGERY & TRAINING

SUPERCOMPUTING@IIT DELHI

- ▶ 10,000 CPU cores, 350 GPUs
- ▶ Nodes with up to 512GB memory
 - ▶ 30TB cumulative main memory
- ▶ 1.5 Petabytes of parallel file system
- ▶ 56Gbps non-blocking network

Parallel algorithms & data structures
Scaling with processors

Parallel IO, Disk structure

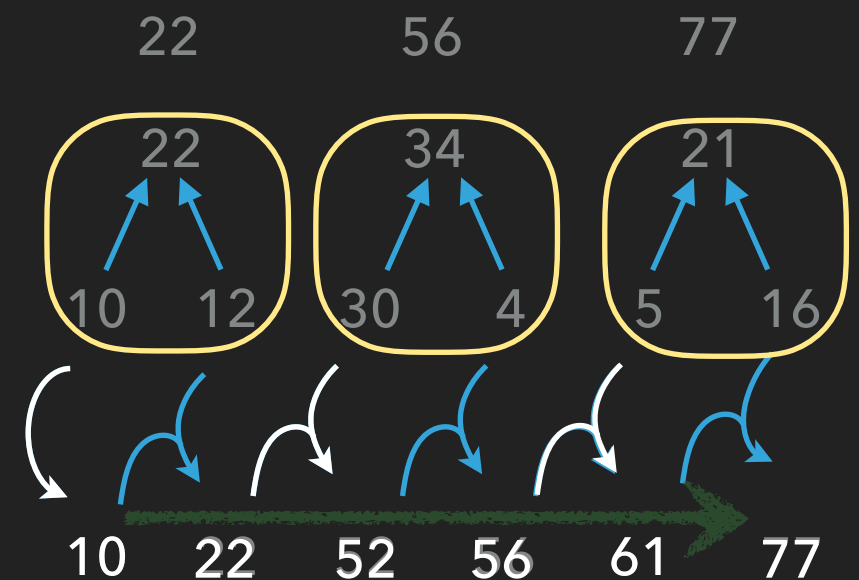
Languages, Runtime and Middleware
Job partitioning and scheduling
Load balancing, Latency hiding

PREFIX SUM

```
def fpsum(lst):
    psum = [lst[0]]
    for i in xrange(1, len(lst)):
        psum.append(psum[i-1] + lst[i])
    return psum
```

```
inline int psum(int *a, int *p, int n)
{
    int *e;
    for(e=a+n, *p=*a++; a < e; a++, p++)
        p[1] = p[0] + a[0];
}
```

$$\begin{aligned}
 T(n) &= 1 + T(n/2) + 1 \sim T(n/2) + K \\
 &= T(n/4) + K + K \\
 &= T(n/8) + K + K + K \\
 &= T(n/2^i) + iK \\
 &= K \log n
 \end{aligned}$$




```
int ppsum(int *a, int *p, int n)
{
    if(n < MIN_PAR_COUNT)
        return psum(a, p, n);

    int n1 = n/2;
    int s1;

    #pragma omp task shared(s1)
    {ppsum(a, p, n1);
    s1 = p[n1-1];}

    #pragma omp task
    ppsum(a+n1, p+n1, n-n1);

    #pragma omp taskwait

    #pragma omp task
    for(int *pp = p+n1, *ep = p + n1 + n1/2; pp < ep; *pp++ += s1);

    #pragma omp task
    for(int *pp = p+n1+n1/2, *ep = p + n; pp < ep; *pp++ += s1);

    # pragma omp taskwait
}
```