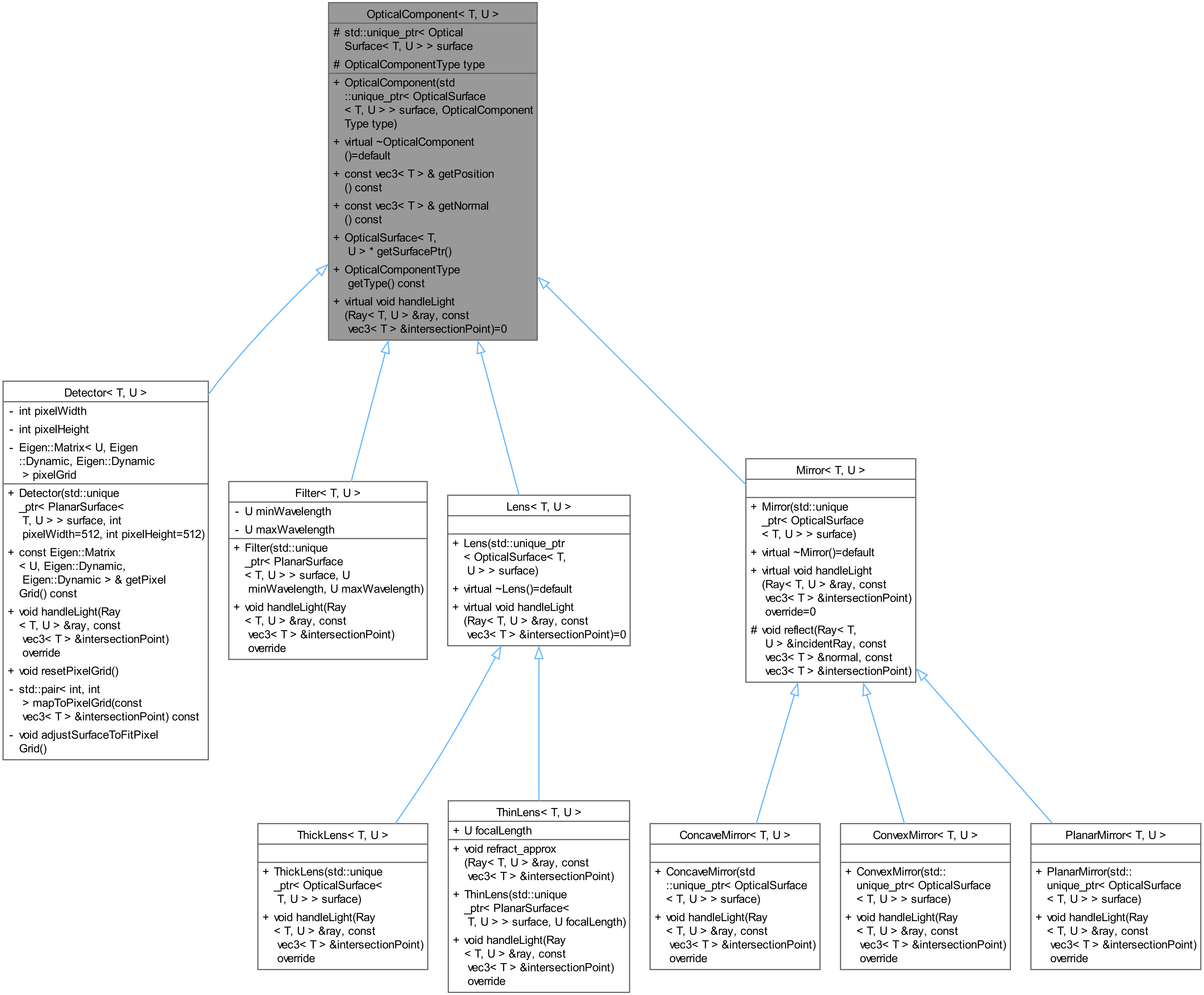
OpticalComponent< T, U >
# std::unique_ptr< OpticalSurface< T, U > > surface
# OpticalComponentType type
+ OpticalComponent(std::unique_ptr< OpticalSurface< T, U > > surface, OpticalComponentType type)
+ virtual ~OpticalComponent()=default
+ const vec3< T > & getPosition() const
+ const vec3< T > & getNormal() const
+ OpticalSurface< T, U > * getSurfacePtr()
+ OpticalComponentType getType() const
+ virtual void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint)=0
Detector< T, U >
- int pixelWidth
- int pixelHeight
- Eigen::Matrix< U, Eigen::Dynamic, Eigen::Dynamic > pixelGrid
+ Detector(std::unique_ptr< PlanarSurface< T, U > > surface, int pixelWidth=512, int pixelHeight=512)
+ const Eigen::Matrix< U, Eigen::Dynamic, Eigen::Dynamic > & getPixelGrid() const
+ void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint) override
+ void resetPixelGrid()
- std::pair< int, int > mapToPixelGrid(const vec3< T > &intersectionPoint) const
- void adjustSurfaceToFitPixelGrid()
Filter< T, U >
- U minWavelength
- U maxWavelength
+ Filter(std::unique_ptr< PlanarSurface< T, U > > surface, U minWavelength, U maxWavelength)
+ void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint) override
Lens< T, U >
+ Lens(std::unique_ptr< OpticalSurface< T, U > > surface)
+ virtual ~Lens()=default
+ virtual void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint)=0
Mirror< T, U >
+ Mirror(std::unique_ptr< OpticalSurface< T, U > > surface)
+ virtual ~Mirror()=default
+ virtual void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint) override=0
# void reflect(Ray< T, U > &incidentRay, const vec3< T > &normal, const vec3< T > &intersectionPoint)
ThickLens< T, U >
+ ThickLens(std::unique_ptr< OpticalSurface< T, U > > surface)
+ void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint) override
ThinLens< T, U >
+ U focalLength
+ void refract_approx(Ray< T, U > &ray, const vec3< T > &intersectionPoint)
+ ThinLens(std::unique_ptr< PlanarSurface< T, U > > surface, U focalLength)
+ void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint) override
ConcaveMirror< T, U >
+ ConcaveMirror(std::unique_ptr< OpticalSurface< T, U > > surface)
+ void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint) override
ConvexMirror< T, U >
+ ConvexMirror(std::unique_ptr< OpticalSurface< T, U > > surface)
+ void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint) override
PlanarMirror< T, U >
+ PlanarMirror(std::unique_ptr< OpticalSurface< T, U > > surface)
+ void handleLight(Ray< T, U > &ray, const vec3< T > &intersectionPoint) override