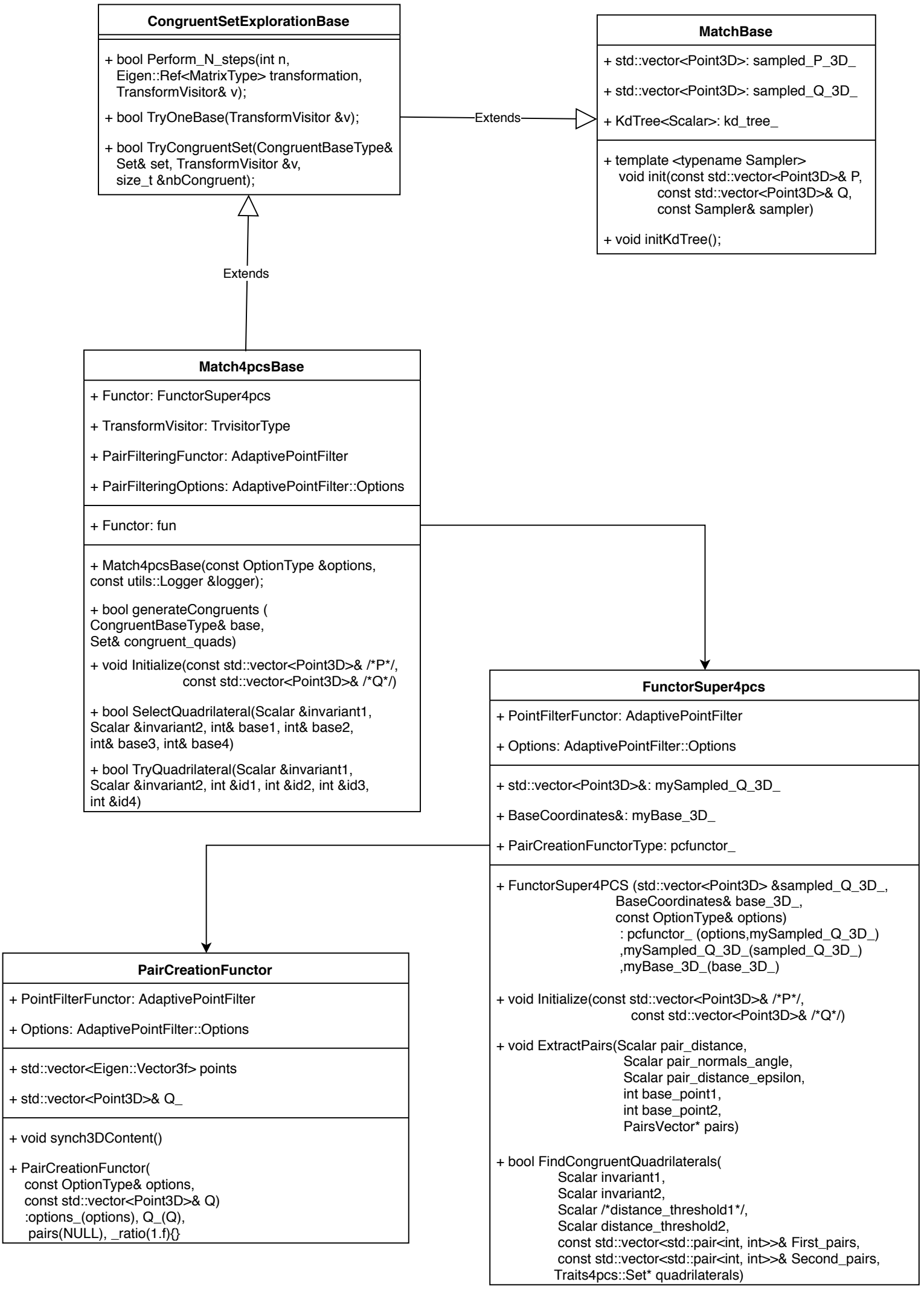
CongruentSetExplorationBase
+ bool Perform_N_steps(int n, Eigen::Ref<MatrixType> transformation, TransformVisitor& v);
+ bool TryOneBase(TransformVisitor &v);
+ bool TryCongruentSet(CongruentBaseType& Set& set, TransformVisitor &v, size_t &nbCongruent);
Extends
MatchBase
+ std::vector<Point3D>: sampled_P_3D_
+ std::vector<Point3D>: sampled_Q_3D_
+ KdTree<Scalar>: kd_tree_
+ template <typename Sampler> void init(const std::vector<Point3D>& P, const std::vector<Point3D>& Q, const Sampler& sampler)
+ void initKdTree();
Extends
Match4pcsBase
+ Functor: FunctorSuper4pcs
+ TransformVisitor: TrvisitorType
+ PairFilteringFunctor: AdaptivePointFilter
+ PairFilteringOptions: AdaptivePointFilter::Options
+ Functor: fun
+ Match4pcsBase(const OptionType &options, const utils::Logger &logger);
+ bool generateCongruents (CongruentBaseType& base, Set& congruent_quads)
+ void Initialize(const std::vector<Point3D>& /*P*/, const std::vector<Point3D>& /*Q*/)
+ bool SelectQuadrilateral(Scalar &invariant1, Scalar &invariant2, int& base1, int& base2, int& base3, int& base4)
+ bool TryQuadrilateral(Scalar &invariant1, Scalar &invariant2, int &id1, int &id2, int &id3, int &id4)
FunctorSuper4pcs
+ PointFilterFunctor: AdaptivePointFilter
+ Options: AdaptivePointFilter::Options
+ std::vector<Point3D>&: mySampled_Q_3D_
+ BaseCoordinates&: myBase_3D_
+ PairCreationFunctorType: pcfunctor_
+ FunctorSuper4PCS (std::vector<Point3D> &sampled_Q_3D_, BaseCoordinates& base_3D_, const OptionType& options) : pcfunctor_ (options,mySampled_Q_3D_) ,mySampled_Q_3D_(sampled_Q_3D_) ,myBase_3D_(base_3D_)
+ void Initialize(const std::vector<Point3D>& /*P*/, const std::vector<Point3D>& /*Q*/)
+ void ExtractPairs(Scalar pair_distance, Scalar pair_normals_angle, Scalar pair_distance_epsilon, int base_point1, int base_point2, PairsVector* pairs)
+ bool FindCongruentQuadrilaterals(Scalar invariant1, Scalar invariant2, Scalar /*distance_threshold1*/, Scalar distance_threshold2, const std::vector<std::pair<int, int>>& First_pairs, const std::vector<std::pair<int, int>>& Second_pairs, Traits4pcs::Set* quadrilaterals)
PairCreationFunctor
+ PointFilterFunctor: AdaptivePointFilter
+ Options: AdaptivePointFilter::Options
+ std::vector<Eigen::Vector3f> points
+ std::vector<Point3D>& Q_
+ void synch3DContent()
+ PairCreationFunctor(const OptionType& options, const std::vector<Point3D>& Q) :options_(options), Q_(Q), pairs(NULL), _ratio(1.f){}