David W. Murray and Bernard F. Buxton, Experiments in the Machine Interpretation of Visual Motion (MIT Press, Cambridge, MA, 1990) 236 pages

Chapter 1: *Image, Scene and Motion*. Exegesis. Scene and image motion. Paradigms for computing visual motion. Visual versus projected motion. Remarks. *Chapter* 2: *Computing Image Motion*. Edgels as weak tokens. An edgel matching algorithm. The algorithm in detail. Computer experiments. *Chapter* 3: *Structure from Motion of Points*. The aperture problem. 3D constraints on the aperture problem. Point structure from known motion. Pointwise depths. Straight edges with known motion. Remarks. *Chapter* 4: *The Structure and Motion of Edges*. Motion segmentation using edges. The structure from motion algorithm. Computer experiments. How many edges?. *Chapter* 5: *From Edges to Surfaces*. Imposing polyhedral constraints. The polyhedral motion algorithm. Computer experiments. Remarks. *Chapter* 6: *Structure and Motion of Planes*. Planar scenes. Recovering planar structure and motion. Planar facets with known motion. 3D reconstructions: computer experiments. Failures of the planar facet algorithms. Reconstructing full visual motion. *Chapter* 7: *Visual Motion Segmentation*. Global segmentation. Local segmentation. Remarks. *Chapter* 8: *Matching to Edge Models*. Model and data specification. Matching in overview. The constraints in detail. Sign management within search. Location stage and testing global validity. Computer experiments. Remarks. Appendix. *Chapter* 9: *Matching to Planar Surfaces*. The matching constraints. Location stage. An experimental example. *Chapter* 10: *Commentary*. Sensing, perception and action. Representation. Computing motion and depth. Object recognition and location. What next?. Perception begets action. Dynamic vision. Reactive vision. Vision and control. Recognition. Shape and motion.

Piergiorgio Odifreddi, ed., Logic and Computer Science (Academic Press, London, 1990) 430 pages

Introduction (P.G. Odifreddi). Recursion theory as a theory of computers. Lambda calculus as a theory of functions. Typed theories. Deductions as computations. The Curry–Howard isomorphism. Automazing mathematics. *Two Extensions of Curry's Type Inference System* (F. Cardone and M. Coppo). The basic system (A kernel functional language. Models. Typed λ-calculus and type inference. Curry's type inference system. Principal type schemes). Recursive types (Recursive and infinite types. Semantics). Intersection types (Type assignment systems with intersection types. Basic syntactic properties. Semantics and filter models. Principal type schemes and type checking). *Nuprl as a General Logic* (R.L. Constable and D.J. Howe). Introduction (Overview of the syntactic method. Implementations. Faithfulness). The classical predicate calculus in Nuprl (Background. The Nuprl library. Semantics). Some advantages. *Metamathematical Investigations of a Calculus of Constructions* (T. Coquand). Informal motivation. The system (terms and typing rules) (Formal presentation. Type inference rules. Some syntactic remarks. Relationship with Automath languages). Semantics (Proof-irrelevance semantics. Realizability semantics. Models in domain theory). Some properties (Conservativity over F_ω. Connection with higher-order logic. Categorical semantics. Representation of data types. Inconsistent extensions). Some possible extensions (Addition of universes. Strong sums. Inductive types). Conclusions. *On Girard's "Candidats de Reductibilité"* (J.H. Gallier). Introduction. Syntax of the second-order polymorphic lambda calculus. Substitution and α-equivalence. Type assignments and type-checking. Reduction and conversion. An untyped version of the candidates of reducibility. A typed version of the candidates of reducibility. Families of sets of saturated sets. Families of sets of Girard sets. Girard's fundamental theorem. A comparison of proofs. Syntax of the higher-order polymorphic lambda calculus F_ω. Substitution and α-equivalence. Contexts, kind-checking and type-checking. Reduction and conversion. The method of candidates. Appendix 1: Product types in F_ω. Appendix 2: Formal details. *Logical Aspects of Computation: Contributions and Distractions* (G. Kreisel). Logical generality (Logical classifications: prefix classes and logical closure. Dismantling universal logical schemes: picking useful modules. Higher set theory: infinitistic

methods and computation). Computation and proofs (Standard generalities: to be remembered, but not dwelled upon. Markets for unwinding. Would-be computational versions of three heroic enterprises). Looking back: from foundations to technology (Scientific theory of familiar phenomena. Possibilities-in-principle. Two gems: going back to Cantor and Dedekind. What of a world without heroic idea(l)s?). *Contracting Proofs to Programs* (D. Leivant). A contraction homomorphism from second order deductions to λ-programs. Convergence proofs as polymorphic λ-programs. Programs over data systems. Representation of recursive functionals of order 2. Contraction to programs for generative axiomatizations. Inductive axiomatizations. Controlled abstraction. Appendix 1: Herbrand–Gödel computability. Appendix 2: Extensions of κ in M2L. Appendix 3: Natural deductions for totality of numeric functions. Appendix 4: Type containment. *Abstractions in Logic Programs* (D. Miller). Introduction. A first-order logic. First-order Horn clauses. Providing scope to program clauses. Providing scope to individual constants. Providing scopes to function and predicate symbols. A higher-order logic. Two higher-order logic programming languages. Higher-order programming. Conclusions. *Isabelle: The Next* 700 *Theorem Provers* (L.C. Paulson). Introduction. The LCF approach to theorem proving. Proof construction by unification. Early experiments. Extending the rule calculus. A formalization of first-order logic. Soundness and completeness. The Edinburgh logical framework. Automated proof in Isabelle. Some Isabelle object logics. Conclusions. *A Guide to Polymorphic Types* (A. Scedrov). Second-order polymorphic lambda calculus. Church–Rosser confluence property. Strong normalization. An overview of topics in semantics (Domain-theoretic and category-theoretic models. PER models and realizability. Some current research topics). Calculus of constructions. An interpretation of the calculus of constructions.