



HAL
open science

Two Cryptomorphic Formalizations of Projective Incidence Geometry

David Braun, Nicolas Magaud, Pascal Schreck

► **To cite this version:**

David Braun, Nicolas Magaud, Pascal Schreck. Two Cryptomorphic Formalizations of Projective Incidence Geometry. *Annals of Mathematics and Artificial Intelligence*, In press. hal-01887000

HAL Id: hal-01887000

<https://hal.science/hal-01887000v1>

Submitted on 3 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Two Cryptomorphic Formalizations of Projective Incidence Geometry

David Braun · Nicolas Magaud · Pascal Schreck

the date of receipt and acceptance should be inserted later

Abstract Incidence geometry is a well-established theory which captures the very basic properties of all geometries in terms of points belonging to lines, planes, etc. Moreover, projective incidence geometry leads to a simple framework where many properties can be studied. In this article, we consider two very different but complementary mathematical approaches formalizing this theory within the Coq proof assistant. The first one consists of the usual and synthetic geometric axiom system often encountered in the literature. The second one is more original and relies on combinatorial aspects through the notion of rank which is based on the matroid structure of incidence geometry. This paper mainly contributes to the field by proving the equivalence between these two approaches in both 2D and 3D. This result allows us to study the further automation of many proofs of projective geometry theorems. We give an overview of techniques that will be heavily used in the equivalence proof and are generic enough to be reused later in yet-to-be-written proofs. Finally, we discuss the possibilities of future automation that can be envisaged using the rank notion.

Keywords. automation, Coq, formalization, incidence, matroid, projective geometry, ranks

1 Introduction

One of the most elementary geometries that can be studied is incidence geometry which basically analyzes the incidence relation between two types of objects: points and lines. This geometry is ubiquitous, and collinearity issues are found in various contexts such as graph theory, experiments planification, subway maps or even children games like Dobble¹. Moreover, incidence geometry can be defined by a

David Braun · Nicolas Magaud · Pascal Schreck
ICube, UMR 7357 CNRS, University of Strasbourg,
david.braun@unistra.fr
magaud@unistra.fr
schreck@unistra.fr

¹ Dobble is a board game in which players must find drawings in common between two cards. Such a deck of cards can be designed by taking the lines in $P^2(K)$ where K is a finite field.

fairly simple set of axioms both in the plane and in a three dimensional setting [7, 10]. Let us note that it is possible to formalize such a geometry in several ways.

Our main long-term motivation is the mechanization of demonstrations in order to ease the task of mathematicians but also to study more effortlessly the degenerate cases. Formalizing and proving geometric theorems is not an easy task. It requires being very rigorous to take into account all degeneracy conditions that can occur in this context. These conditions often lead to long and tedious technical proofs. That is why, many papers have emphasized the importance of investigating the problem of degenerate cases in formal geometry [11, 25, 32]. In an elementary geometry such as incidence geometry, it is conceivable to efficiently automate some tedious part of the geometric reasoning.

To specialize primitive incidence geometry, we work in the specific context of projective geometry in which the notion of parallel does not exist. In other words, two coplanar lines always meet. This simple framework remains powerful enough to study the formalization and proof of complex problems such as those suggested in [26]. In addition, there is no loss of generality since it is possible to switch from projective geometry to affine geometry by adding a chosen line at infinity.

In this paper, we present two equivalent formalizations of projective incidence geometry. The first one follows the classical way: it is based on a geometric characterization and is widely described in the literature [7, 10]. The second one is more original, it relies on the combinatorial notion of rank provided by the matroid structure of incidence geometry. These two different approaches are complementary when dealing with geometric problems. Indeed, the classical geometric definition makes it possible to describe, through an intuitive terminology, a large majority of theorems using powerful notions such as lines, planes, collinearity and coplanarity relations. These concepts allow us to build complex geometric configurations but are relatively difficult to handle when it comes to solving a problem. It is in this sense that the second approach is complementary to the first one. Using the rank notion, it is possible to express in a homogeneous way all kind of incidences of this geometry (point-point, point-line, point-plane, line-line, plane-plane) as well as relations of collinearity, coplanarity, etc. Thanks to the unique representation of all incidence relations as ranks, we can thus consider a more systematic automation.

But, to do this, it is necessary to rigorously prove the equivalence between these two formalizations in order to be able to use them jointly. The main original contribution of this paper is the equivalence proof between these two approaches through the Coq proof assistant [4, 9]. This demonstration is not trivial and requires some sophisticated techniques to establish the equivalence between the two axiom systems in both 2D and 3D. We give some of the proof engineering methods used to simplify demonstrations steps and to highlight the automation.

Some ideas behind this work have their origins in previous studies about combinatorial aspects of proofs in incidence geometry as described in [22, 28]. More specifically, a proof of the well-known Desargues' theorem using only the rank approach provides a better understanding of the advantages and disadvantages of ranks. All these results are based on a partial formalization and an intuition of what might be the equivalence proof between these two systems. To our knowledge, mathematicians have not studied extensively the details of such equivalence.

Related Work Geometry is a good candidate to be studied through proof assistants like Coq. Many formalizations of plane geometry like Hilbert and Tarski axiomatization have already been performed [12, 15, 25, 32]. In the narrower context of projective geometry, some work in constructive projective geometry was initiated by Von Plato [18, 37]. Our formalization differs by considering a decidable incidence relation. In the field of automatic methods in projective geometry, we have to mention the algebraic formalization of Grassmann-Cayley algebra [13, 21, 34]. Finally, to our knowledge there is no work treating incidence geometry in a projective framework using the equivalence between a geometric characterization and a combinatorial approach.

Outline of the paper This paper is an extended version of the contribution presented but not published at Automated Deduction in Geometry 2016 [6]. The paper is organized as follows. In Section 2, we present classical axioms for projective incidence geometry and we give a preview of our Coq formalization. In Section 3, we expose our axiomatization based on the concept of rank from matroid theory. Section 4 investigates in detail the equivalence proof between these two formalizations. Finally, automation possibilities and future work are discussed in Section 5.

Notation We use the naming convention $AXYN$ for our axioms. The letters correspond to A for axiom, X for the axiom number, Y may take two values (P = projective, R = rank) and N designs the dimension.

2 Axiom Systems of Projective Geometry

In a general framework, geometry such as Euclidean geometry is a complex subject mixing objects like points, lines and planes with concepts of distance, angle, continuity, incidence etc. Among all these notions, only two kinds of objects, points and lines, and the incidence relation between them are kept to form the elementary incidence geometry.

2.1 Incidence Geometry

Incidence geometry is the study of a triple (Ω, Δ, Φ) known as incidence structure. Ω refers to a set of points, Δ is a disjoint set composed of lines and Φ is the binary relation that unites them. Elements which are mutually incident constitute a subset $\Phi \subseteq \Omega \times \Delta$. Intuitively, a point and a line are within this subset if and only if the point is on the line. In the following, the incidence relation Φ will be noted in an infix way by \in . It is possible to informally describe this geometry with three rules [3].

- There is always a line passing through two points.
- On any line, there are at least two points.
- There exist three points that are not collinear.

This characterization is already sufficient to prove many theorems [7]. In addition, the enrichment of incidence geometry by new concepts preserves many

fundamental results. Projective geometry relies on the basics of incidence geometry. It is a general setting in the hierarchy of geometries which assumes that two lines in a plane always meet [10]. Planes can be defined within our theory but as opposed to Hilbert's formalization they are not primitive objects. With the concepts of incidence geometry, we can describe properly projective incidence geometry in 2-dimensional space and in 3-dimensional space.

2.2 Axiom System for Projective Plane Geometry

One of Coxeter's axiom systems [10] for projective plane geometry consists of five axioms presented in Table 1 and illustrated in Table 2.

The first two axioms (A1P2, A2P2) deal with construction of points and lines. We do not need to specify that the involved points (resp. lines) should be different in axiom *Line-Existence* (resp. *Point-Existence*). Indeed if these points (resp. lines) are equal, such line (resp. point) still exists. In fact, there exists an infinity of lines (resp. points) passing through the two equal points (resp. lines). This theoretical choice is significant and defines the degenerate cases that one wishes to maintain. Indeed, it allows us to consider a geometry where it is always possible to show that there exists a line passing through a point without extending our axiom system. In the same way, there always exists an intersection point between two merged lines. These considerations make it feasible to provide witnesses of existence in some degenerate configuration proofs. Nevertheless in most books, these degeneracy conditions are excluded to be consistent with the introduction of finite geometries where the number of distinct points is specified. The next axiom (A3P2) concerns uniqueness of the two defined objects. Then, axiom (A4P2) states that each line contains at least three points. This axiom differs from the second informal rule cited above by preventing some small and exceptional systems from being called projective planes. Finally, axiom (A5P2) expresses that there always exist two distinct lines, which means that the dimension is at least 2. Together with axiom *Point-Existence* it shows that the dimension is at most 2. Therefore the dimension is exactly 2.

- (A1P2) **Line-Existence** : $\forall A B : \text{Point}, \exists l : \text{Line}, A \in l \wedge B \in l$
- (A2P2) **Point-Existence** : $\forall l m : \text{Line}, \exists A : \text{Point}, A \in l \wedge A \in m$
- (A3P2) **Uniqueness** : $\forall A B : \text{Point}, \forall l m : \text{Line}, A \in l \wedge B \in l \wedge A \in m \wedge B \in m \Rightarrow A = B \vee l = m$
- (A4P2) **Three-Points** : $\forall l : \text{Line}, \exists A B C : \text{Point}, A \neq B \wedge B \neq C \wedge A \neq C \wedge A \in l \wedge B \in l \wedge C \in l$
- (A5P2) **Lower-Dimension** : $\exists l m : \text{Line}, l \neq m$

Table 1 Standard axiom system for projective plane geometry

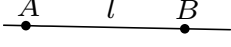
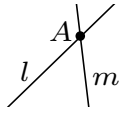
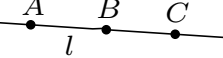
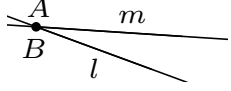
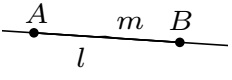

A1P2	A2P2	A4P2
		
A3P2		A5P2
		

Table 2 Illustrations of the standard axiom system for plane geometry

The formalization of this axiom system is straightforward; textbooks [7, 10] often use some variants of this system (see Table 3). There is an equivalence between axioms (A4P2, A5P2) and (A4P2') which states that there exists at least four distinct points, no three of them being collinear. This equivalence proof is available in [22]. We choose the first axiom system to ease decomposition and mechanization of proofs. Indeed, the axiom (A4P2') is often more complicated to manipulate within the proofs than the decomposition presented in Table 1.

(A1P2) Line-Existence : $\forall A B : \text{Point}, \exists l : \text{Line}, A \in l \wedge B \in l$

(A2P2) Point-Existence : $\forall l m : \text{Line}, \exists A : \text{Point}, A \in l \wedge A \in m$

(A3P2) Uniqueness : $\forall A B : \text{Point}, \forall l m : \text{Line}, A \in l \wedge B \in l \wedge A \in m \wedge B \in m \Rightarrow A = B \vee l = m$

(A4P2') Four-Points : $\exists A B C D : \text{Point},$
 $A \neq B \wedge A \neq C \wedge A \neq D \wedge B \neq C \wedge B \neq D \wedge C \neq D \wedge$
 $(\forall l : \text{Line},$
 $(A \in l \wedge B \in l \Rightarrow C \notin l \wedge D \notin l) \wedge$
 $(A \in l \wedge C \in l \Rightarrow B \notin l \wedge D \notin l) \wedge$
 $(A \in l \wedge D \in l \Rightarrow B \notin l \wedge C \notin l) \wedge$
 $(B \in l \wedge C \in l \Rightarrow A \notin l \wedge D \notin l) \wedge$
 $(B \in l \wedge D \in l \Rightarrow A \notin l \wedge C \notin l) \wedge$
 $(C \in l \wedge D \in l \Rightarrow A \notin l \wedge B \notin l))$

Table 3 Another axiom system for projective plane geometry

2.3 Axiom System for Projective Space Geometry

In the same way, we declare an axiom system for projective space geometry in Table 4 and Table 5 by extending the previous one. The system still contains five axioms with three of them remaining unchanged (A1P3, A3P3, A4P3). *Pasch's* axiom replaces (A2P2) assuming that two coplanar lines always meet. Furthermore, we modify the axiom *Lower-Dimension* to capture projective geometry in an at least 3-dimensional space (noted $\geq 3D$). For this, we assume that there exist two lines which do not meet. This time, *Pasch's* axiom does not allow us to limit the upper dimension, that is why we consider a geometry $\geq 3D$.

However, it is possible to limit this spatial geometry by adding the optional axiom (A6P3) to constrain the dimension to be exactly 3. This axiom specifies that there is always a line intersecting three other non-coplanar lines.

(A1P3) Line-Existence :	$\forall A B : \text{Point}, \exists l : \text{Line}, A \in l \wedge B \in l$
(A2P3) Pasch :	$\forall A B C D : \text{Point}, \forall l_{AB} l_{CD} l_{AC} l_{BD} : \text{Line},$ $A \neq B \wedge A \neq C \wedge A \neq D \wedge B \neq C \wedge B \neq D \wedge C \neq D \wedge$ $A \in l_{AB} \wedge B \in l_{AB} \wedge C \in l_{CD} \wedge D \in l_{CD} \wedge$ $A \in l_{AC} \wedge C \in l_{AC} \wedge B \in l_{BD} \wedge D \in l_{BD} \wedge$ $(\exists I : \text{Point}, I \in l_{AB} \wedge I \in l_{CD}) \Rightarrow$ $(\exists J : \text{Point}, J \in l_{AD} \wedge J \in l_{BC})$
(A3P3) Uniqueness :	$\forall A B : \text{Point}, \forall l m : \text{Line},$ $A \in l \wedge B \in l \wedge A \in m \wedge B \in m \Rightarrow A = B \vee l = m$
(A4P3) Three-Points :	$\forall l : \text{Line}, \exists A B C : \text{Point},$ $A \neq B \wedge B \neq C \wedge A \neq C \wedge A \in l \wedge B \in l \wedge C \in l$
(A5P3) Lower-Dimension :	$\exists l m : \text{Line}, \forall p : \text{Point}, p \notin l \vee p \notin m$
(A6P3) Upper-Dimension :	$\forall l_1 l_2 l_3 : \text{Line}, l_1 \neq l_2 \wedge l_1 \neq l_3 \wedge l_2 \neq l_3 \Rightarrow$ $\exists l_4 : \text{Line}, \exists P_1 P_2 P_3 : \text{Point}, P_1 \in l_1 \wedge P_1 \in l_4 \wedge$ $P_2 \in l_2 \wedge P_2 \in l_4 \wedge P_3 \in l_3 \wedge P_3 \in l_4$

Table 4 Standard axiom system for projective space geometry

A2P3	A5P3	A6P3

Table 5 Illustrations of the standard axiom system for projective space geometry

2.4 Formalization in Coq

Implementing these two systems of axioms is rather immediate in Coq; we present below our formalization of the projective space. To enhance modularity, we take advantage of the *type classes* and functors of Coq.

To analyze dependencies between these axiom systems and to observe equivalences between different variants of our axiomatizations, we split our axiom sets in different classes. These classes are constructed incrementally by adding one by one each axiom. Therefore, it is possible to examine in great details the equivalence between each class of axioms within the two formalizations. Moreover, this decomposition allows us to bring out the minimal axiom system required to demonstrate a set of properties. Finally, type classes in Coq make it possible to verify that we have well instantiated what we were trying to prove.

First, we build type classes with common axioms between the dimensions namely *Line-Existence*, *Uniqueness* and *Three-Points* (see Table 7). Then, we construct three classes, the first one captures the whole projective plane (see Table 8), the second one (see Table 9) describes the projective space $\geq 3D$ and the latter (see Table 10) captures exactly the projective space. For the sake of clarity, we present here a simplified version of our implementation. We define point, line, incidence relation and its decidability in the signature `ProjectiveStructure` (see Table 6).

```
(* Types *) Class ProjectiveStructure := {
Point : Set;
Line : Set;
Incid : Point -> Line -> Prop;

(* decidability for incidence relation *)
incid_dec : forall (A : Point)(l : Line), {Incid A l} + {~Incid A l}
}.

```

Table 6 Type class for projective structure

```
(* ND *) Class ProjectiveStructureLEU '(PS : ProjectiveStructure) := {

(* A1P2-P3 Line-Existence *)
a1_exist: forall (A B : Point) , exists l : Line, Incid A l /\ Incid B l;

(* A3P2-P3 Uniqueness *)
uniqueness: forall (A B : Point)(l1 l2 : Line), Incid A l1 -> Incid B l1 ->
  Incid A l2 -> Incid B l2 -> A [==] B \/ l1 = l2;

(* A4P2-P3 Three-Points *)
a3_1: forall l : Line, exists A : Point, exists B : Point, exists C : Point,
  (~ A [==] B /\ ~ A [==] C /\ ~ B [==] C /\
  Incid A l /\ Incid B l /\ Incid C l)
}.

```

Table 7 Type class for projective structure independent of dimension

```
(* 2D *) Class ProjectivePlane '(PSLEU : ProjectiveStructureLEU) := {
(* A2P2 Point-Existence *)
a2_exist : forall (l1 l2 : Line), exists A : Point, Incid A l1 /\ Incid A l2;

(* A5P2 Lower-Dimension *)
a3_2 : exists l1 : Line, exists l2 : Line, l1 <> l2;
}.
```

Table 8 Type class for projective plane

```
(* >= 3D *) Class ProjectiveSpaceOrHigher '(PSLEU : ProjectiveStructureLEU) := {

(* A2P3 Pasch *)
a2: forall A B C D : Point, forall lAB lCD lAC lBD : Line,
  ~ A [==] B /\ ~ A [==] C /\ ~ A [==] D /\
  ~ B [==] C /\ ~ B [==] D /\ ~ C [==] D ->
  Incid A lAB /\ Incid B lAB ->
  Incid C lCD /\ Incid D lCD ->
  Incid A lAC /\ Incid C lAC ->
  Incid B lBD /\ Incid D lBD ->
  (exists I : Point, (Incid I lAB /\ Incid I lCD)) ->
  exists J : Point, (Incid J lAC /\ Incid J lBD);

(* A5P3 Lower-Dimension *)
a3_2: exists l1 : Line, exists l2 : Line, forall p : Point,
  ~(Incid p l1 /\ Incid p l2)
}.
```

Table 9 Type class for projective space at least in dimension 3

```
(* 3D *) Class ProjectiveSpace '(PSH : ProjectiveSpaceOrHigher) := {

Intersect_In (l1 l2 : Line) (P : Point) := Incid P l1 /\ (Incid P l2);

(* A6P3 Upper-Dimension *)
a3_3 : forall l1 l2 l3 : Line, ~ l1 = l2 /\ ~ l1 = l3 /\ ~ l2 = l3 ->
  exists l4 : Line, exists J1 : Point, exists J2 : Point, exists J3 : Point,
  (Intersect_In l1 l4 J1) /\ (Intersect_In l2 l4 J2) /\ (Intersect_In l3 l4 J3)
}.
```

Table 10 Type class for projective space exactly in dimension 3

The main difference between our formalization and the axiom system shown above comes from equality relations and decidability issues. The equality, denoted `[==]`, is an equivalence relation and a parameter of our theory formally defined in `ProjectiveStructure` type class. This equality allows us to make transparent the way in which objects are built. In other words, this equality makes it possible to replace one object with another of the same type if they are constructively equiv-

alent but not equal. The manipulation of this particular equivalence relation will lead to some technical subtleties which we detail in the following parts. We choose to use this parametric equality only for points. Indeed lines play a less important role, they can be represented as a set of points in the other axiomatization. In all cases, we can consider a specific equality for each type of object without conflict since we do not compare the objects between them. In theory, these two objects do not need to be treated differently, we plan to exploit same equality for lines in a future update also.

As the underlying logic of the Coq system is by default intuitionist, we have to declare explicitly which predicates are decidable. That is, we must indicate to the system which concepts are based on the excluded middle property from classical logic. Assuming only the decidability of the predicate `Incid` in our theory, we can prove the decidability of equality for both points and lines [22]. This design choice in no way complicates the proofs and we will not discuss this further.

3 A Rank-based Axiom System

Section 2 described a standard axiomatization for projective geometry as a reference. We propose here an alternative axiom system based on the notion of rank.

3.1 Matroid Properties

The concept of rank comes from matroid theory [33]. Matroids were introduced by Whitney in 1935 to abstractly capture the essence of dependence. Whitney's definition embraces a surprising diversity of combinatorial structures. Matroid allow us to capture and generalize the main set properties of linear dependence in vector space. When combined with a finite set of points, it catches incidence (collinearity, coplanarity, ...) between these points.

However matroids apply to a much larger class of objects. Other natural examples are obtained from graph theory, fields and greedy algorithms. There are several cryptomorphic ways to define a matroid. In our context, we use the definition based on rank notion. Using ranks allows us to deal only with points which makes proofs automation easier because we do not handle directly lines or planes. It is thus possible to combine all the concepts encountered in incidence geometry only with the rank function. In this way, most proof schemes are reduced to the essentials.

Formally, an integer function rk on a finite set E is the rank function of a matroid if and only if conditions of Table 11 are satisfied.

$$\text{(A1R2-R3) non-negative and subcardinal : } \forall X \subseteq E, 0 \leq rk(X) \leq |X|$$

$$\text{(A2R2-R3) non-decreasing : } \forall X \subseteq Y, rk(X) \leq rk(Y)$$

$$\text{(A3R2-R3) submodular : } \forall X, Y \subseteq E, rk(X \cup Y) + rk(X \cap Y) \leq rk(X) + rk(Y)$$

Table 11 Rank function properties

3.2 Rank to Describe Projective Incidence Geometry

In the framework of projective geometry, we define a rank function on finite sets of points which checks the three conditions as in Table 11. We specify the notions of closure and flat which are alternative axiomatizations of the matroids.

Let M be a matroid on a finite set E with the rank function rk as above. The closure cl of a subset A of E is the set: $cl(A) = \{x \in E \mid rk(A) = rk(A \cup \{x\})\}$. A set whose closure equals itself is said to be closed or a flat.

A set is a flat if it is maximal for its rank, meaning that the addition of any other element to the set would increase the rank. In other words, the rank of a flat E is the cardinal of a smallest set generating E (see Table 12 for some examples).

$rk\{A,B\} = 1$	$A = B$
$rk\{A,B\} = 2$	$A \neq B$
$rk\{A,B,C\} = 2$	A,B,C are collinear with at least two of them distinct
$rk\{A,B,C\} \leq 2$	A,B,C are collinear
$rk\{A,B,C\} = 3$	A,B,C are not collinear
$rk\{A,B,C,D\} = 3$	A,B,C,D are coplanar, not all collinear
$rk\{A,B,C,D\} = 4$	A,B,C,D are not coplanar

Table 12 Some rank statements and their geometric interpretations

Using this definition, it can be shown that every projective space has a matroid structure, but the converse is not true. To capture projective geometry, we need to introduce some additional axioms to that of the matroid's one.

3.3 2D Rank-based Axiom System

Table 13 presents a rank-based axiom system to describe the projective plane. The first two axioms establish the non-degeneracy of the rank function. The others are more or less direct translations of the axioms of plane projective geometry.

(A4R2) Rk-Singleton : $\forall P : \text{Point}, rk\{P\} = 1$

(A5R2) Rk-Couple : $\forall P Q: \text{Point}, P \neq Q \Rightarrow rk\{P, Q\} = 2$

(A6R2) Rk-Inter : $\forall A B C D : \text{Point}, \exists J : \text{Point}, rk\{A, B, J\} = rk\{C, D, J\} = 2$

(A7R2) Rk-Three-Points : $\forall A B : \text{Point}, \exists C, rk\{A, B, C\} = rk\{B, C\} = rk\{A, C\} = 2$

(A8R2) Rk-Lower-Dimension : $\exists A B C : \text{Point}, rk\{A, B, C\} \geq 3$

Table 13 Rank axiom system for projective plane geometry

3.4 3D Rank-based Axiom System

In the same way, we define a rank-based axiom system to describe projective space in Table 14. Again, we modify only the axioms of *Pasch* and *Lower-Dimension* to capture at least a three dimensional setting ($\geq 3D$). To limit this axiomatization to 3D, we add the axiom (A9R3).

- (A4R3) **Rk-Singleton** : $\forall P : \text{Point}, \text{rk}\{P\} = 1$
- (A5R3) **Rk-Couple** : $\forall P Q : \text{Point}, P \neq Q \Rightarrow \text{rk}\{P, Q\} = 2$
- (A6R3) **Rk-Pasch** : $\forall A B C D : \text{Point}, \text{rk}\{A, B, C, D\} \leq 3 \Rightarrow \exists J : \text{Point},$
 $\text{rk}\{A, B, J\} = \text{rk}\{C, D, J\} = 2$
- (A7R3) **Rk-Three-Points** : $\forall A B : \text{Point}, \exists C, \text{rk}\{A, B, C\} = \text{rk}\{B, C\} = \text{rk}\{A, C\} = 2$
- (A8R3) **Rk-Lower-Dimension** : $\exists A B C D : \text{Point}, \text{rk}\{A, B, C, D\} \geq 4$
-
- (A9R3) **Rk-Upper-Dimension** : $\forall A B C D : \text{Point}, \text{rk}\{A, B, C, D\} \leq 4$

Table 14 Rank axiom system for projective space geometry

The implementation in Coq follows exactly the same process; we use *type classes* to increase modularity of the code depending on the dimension.

Now that we have specified the axiomatization, we will focus on proving the equivalence between these two axiom systems. In previous work [23, 24], only the implication from the ranks to $\geq 3D$ projective geometry has been studied. It is sufficient to allow the formalization of Desargues theorem with the ranks. To allow bilateral translation and to increase the possibilities of automation within proof steps, we carry out this complete equivalence proof.

4 Equivalence Proof

We prove the following statement:

Theorem 1 *The synthetic axiom system on projective incidence geometry and rank-based axiom system are equivalent respectively in 2D, $\geq 3D$ and 3D.*

To achieve this proof, we split the demonstration of the equivalence depending on the dimension and the direction. To represent sets in Coq, we use the *Containers* library [20]. It allows us to abstractly reason about sets or to use implementations such as lists or AVLs. For performance reasons in execution time and given that we are working on sets of small sizes, we prove one of the two implications by using an implementation based on the lists rather than the abstract sets. Since it is necessary to perform more set operations to prove implication from projective incidence geometry to the ranks, we prefer to manipulate the implementation on the lists. In addition, we evaluate and compare the abstract approach and one of its implementations for the automation of reasoning.

4.1 From Ranks to Projective Geometry

We begin with the implication from ranks to projective geometry in 2D, ≥ 3 D and 3D. The proofs remain the same no matter the dimension.

Preliminaries

To begin, we need to characterize the concept of line and incidence that does not exist in the axiom system based on ranks (see Table 15). We build, using an inductive definition, a line from two distinct points. A point P is incident to a line, if the rank of the triple formed by the two points constituting the line and the point P remains equal to 2. Finally, two lines l and m are equal, if the rank of the quadruple formed by the two points constituting each line remains equal to 2.

```
Definition Point := Point.
```

```
Inductive LineInd : Type :=
|Cline : forall (A B : Point)(H : ~ A[=]B), LineInd.
```

```
Definition Line := LineInd.
```

```
Definition Incid (P : point)(l : Line) := rk(triple (fstP l)(sndP l) P) = 2.
```

```
Definition line_eq (l m : Line) :=
rk(quadruple (fstP l)(sndP l)(fstP m)(sndP m)) = 2.
```

Table 15 Characterization of the projective incidence geometry from rank axiom system

From there we can express the axioms of projective geometry and prove them using rank axioms.

Submodularity

We detail some proof techniques used to demonstrate the five axioms in projective geometry in both 2D and ≥ 3 D. Firstly, we usually prove equalities on ranks ($rk(a) = rk(b)$) in two steps : first that $rk(a) \leq rk(b)$, then $rk(a) \geq rk(b)$.

Second, we work systematically with the axiom of submodularity (A3R2-R3). Determining the intersection of two finite sets of points is an issue, we need to distinguish cases about the equalities of points. The resulting proofs become more complex with these distinctions. Therefore, we do not consider the theoretical intersection but a lower approximation of this intersection (denoted by \sqcap). This intersection will eliminate most case distinctions, thus facilitating intersection between sets.

Definition 1 (Theoretical Intersection).

Let L_1 and L_2 be two sets of points. By definition $L_1 \cap L_2$ is the exact intersection of the two sets of points considered. Any points that are syntactically identical or that are equal by construction appear only once in the result set.

For example, let $L_1 = \{A, C\}$ and $L_2 = \{B, C\}$ be two sets of points. We add the hypothesis that $A = B$, then the theoretical intersection of these two sets is $L_1 \cap L_2 = \{A, C\}$.

Definition 2 (Literal Intersection).

Let L_1 and L_2 be two sets of points. By definition $L_1 \sqcap L_2$ is the intersection of the two sets of points syntactically considered. Any points that are syntactically identical appear only once in the result set. In practice, $L_1 \sqcap L_2 \subseteq L_1 \cap L_2$. Using axiom A2R2-3, we deduce that: $rk(L_1 \sqcap L_2) \leq rk(L_1 \cap L_2)$.

For example, let $L_1 = \{A, C\}$ and $L_2 = \{B, C\}$ be two sets of points. We maintain the hypothesis that $A = B$ even if it is not useful in this case, then the literal intersection of these two sets is $L_1 \sqcap L_2 = \{C\}$.

From this literal intersection, we can derive a more appropriate version of the axiom A3R2-R3 ignoring case distinctions:

Lemma 1 (*A3R2-R3-lit*),

$$\forall X Y, rk(X \cup Y) + rk(X \sqcap Y) \leq rk(X) + rk(Y).$$

However, Coq does not make it possible to easily define the notion of literal intersection in a computational way. We prefer to define a derived version to capture the meaning of this lemma:

Lemma 2 (*A3R2-R3-alt*)²,

$$\forall X Y I, I \subseteq (X \sqcap Y) \subseteq (X \cap Y) \Rightarrow rk(X \cup Y) + rk(I) \leq rk(X) + rk(Y).$$

This functional version of the submodularity will be extensively used in each proof. Unlike the intersection, it is not necessary to define the theoretical union and the literal union since they always capture the same set of points regardless of the equalities between the points.

Third, the knowledge about the dimension of a set of points can be incomplete leading to the representation of its value by an integer interval in absence of additional information or evidence of a contradiction. In other words, the rank of a set of points is not always exact, the set is not constrained enough in the statement to represent a specific dimension and the system needs more information or a contradiction to move forward. In most cases, the only possible method to handle this is to do a case-by-case reasoning for each of the dimensions. The demonstration of the axiom *Upper-Dimension* is an example which illustrates this proof pattern. The three lines which are cut by a fourth one have only the constraint of being different two by two. To perform this demonstration, it is mandatory to make case distinctions on whether this couple of lines is coplanar or not.

Proof of the uniqueness property

² Lemma `matroid3_useful` is available in the file `matroid_to_matroid.p`.

To illustrate the proof mechanism of the last subsection, we detail the demonstration of the Uniqueness property within the ranks framework.

Lemma 3 (*A3P2-P3*) *Uniqueness*³, $\forall A B : \text{Point}, \forall l m : \text{Line},$
 $A \in l \wedge B \in l \wedge A \in m \wedge B \in m \Rightarrow A = B \vee l = m$

Proof We begin this proof by performing a case distinction on the equality between A and B.

If $A = B$, then the conclusion holds trivially.

If $A \neq B$, by unfolding definitions of **Line** and **Incid**, it follows that :

$$\left. \begin{array}{l} \text{We have } A \langle \rangle B \\ \text{Let } P \in l, Q \in l \text{ and } P \langle \rangle Q \\ \text{Let } R \in m, S \in m \text{ and } R \langle \rangle S \\ \text{Incid } A \ l \Rightarrow \text{rk}\{P, Q, A\} = 2 \\ \text{Incid } B \ l \Rightarrow \text{rk}\{P, Q, B\} = 2 \\ \text{Incid } A \ m \Rightarrow \text{rk}\{R, S, A\} = 2 \\ \text{Incid } B \ m \Rightarrow \text{rk}\{R, S, B\} = 2 \end{array} \right\} \text{Assumptions}$$

$$l = m \Rightarrow \text{rk}\{P \ Q \ R \ S\} = 2 \quad \} \text{Goal}$$

To continue, we determine the rank of two new sets using A3R2-R3-alt: $\text{rk}\{P, Q, A, B\}$ and $\text{rk}\{R, S, A, B\}$.

$$\begin{aligned} & \text{rk}(\{P, Q, A\} \cup \{P, Q, B\}) + \text{rk}(\{P, Q, A\} \cap \{P, Q, B\}) \\ & \leq \text{rk}\{P, Q, A\} + \text{rk}\{P, Q, B\} \\ & \Rightarrow \text{rk}\{P, Q, A, B\} + \text{rk}\{P, Q\} \leq \text{rk}\{P, Q, A\} + \text{rk}\{P, Q, B\} \\ & \Rightarrow \text{rk}\{P, Q, A, B\} \leq 2 \end{aligned}$$

Analogously, we compute that $\text{rk}\{R, S, A, B\} \leq 2$. Then we establish that $\text{rk}\{P, Q, R, S, A, B\} \leq 2$.

$$\begin{aligned} & \text{rk}(\{P, Q, A, B\} \cup \{R, S, A, B\}) + \text{rk}(\{P, Q, A, B\} \cap \{R, S, A, B\}) \\ & \leq \text{rk}\{P, Q, A, B\} + \text{rk}\{R, S, A, B\} \\ & \Rightarrow \text{rk}\{P, Q, R, S, A, B\} + \text{rk}\{A, B\} \leq \text{rk}\{P, Q, A, B\} + \text{rk}\{R, S, A, B\} \\ & \Rightarrow \text{rk}\{P, Q, R, S, A, B\} \leq 2 \end{aligned}$$

Using A2R2-R3, we prove that:

$$\{P, Q, R, S\} \subset \{P, Q, R, S, A, B\} \Rightarrow \text{rk}\{P, Q, R, S\} \leq \text{rk}\{P, Q, R, S, A, B\}$$

So $\text{rk}\{P, Q, R, S\} \leq 2$ and as the set $\{P, Q, R, S\}$ contains at least two distinct points, we conclude that $\text{rk}\{P, Q, R, S\} = 2$.

³ Lemma **uniqueness** is available in the file `rk_equiv_to_pp`.

Coq implementation

At the implementation level, to make the deduction on abstract sets, we use the tactic `fsetdecide`⁴ provided by the library about *Containers*. We also use setoids to make substitutions when we have to indicate to the system that the sets are identical. Finally to deal with inequalities, we use the `omega`⁵ tactic. The other lemmas of this implication are dealt with similar techniques.

4.2 From Projective Geometry to Ranks

The opposite direction is more difficult to handle since we need to specify the concept of rank from projective geometry. Let us recall that this rank function regroups by itself all the objects that can be found in projective geometry but also notions of incidence, collinearity and coplanarity.

Preliminaries

Defining the rank function requires to write some intermediate definitions representing the different returned values of the rank function. To characterize the rank (see Table 16), we are inspired by the concept of flat from the matroid theory. A set of points is either empty or it represents a point, a line, a plane or a space.

```

Definition rk1 s := match s with
| nil => 0
| x :: nil => 1
| s => if contains_four_non_coplanar_points s then 4 else
      if contains_three_non_collinear_points s then 3 else
      if contains_two_distinct_points s then 2 else 1 end.

```

Table 16 Characterization of the rank function from projective incidence geometry⁶

The three predicates `contains_four`, `contains_three` and `contains_two` (see Table 17) put bounds on the dimension of the considered set. The system first checks the coplanarity: either there is a quadruple of non-coplanar points and the set represents a space, or it continues analyzing collinearity. These predicates form what is called an intermediate layer. They help the transition between the two axiom systems and are often accompanied by dozens of lemmas. It is desirable to be careful during the development of such definitions to take into account all the degenerate cases (mainly coincident points).

⁴ `fsetdecide` : decision procedure for a class of propositions involving finite sets.

⁵ `Omega`: a solver of quantifier-free problems in Presburger Arithmetic.

⁶ Definition `rk1` is available in the file `psoh_equiv_rk_lemmas`.

```

Fixpoint contains_four_non_coplanar_points l := match l with
| nil => false
| a::r => if coplanar_with_all a (all_triples r)
          then contains_four_non_coplanar_points r else true end.

```

Table 17 Recursive definition of predicate `contains_four_non_coplanar_points`. The predicate `coplanar_with_all` tests if the first point is coplanar with all other points⁶

Proof techniques

Similarly, we present the main technical aspects that one encounters when performing the demonstrations of the nine axioms of ranks characterization from the set of geometric axioms. We exploit mostly the induction mechanism on natural numbers to do a case analysis on the dimension of a set of points. Handling the three conditionals in the characterization of rank definition and the management of degenerate cases yield many goals which significantly increase the size of the proofs. The resulting proofs are often tedious, although it is possible to automate many steps.

For this, we design tactics using the Ltac tactic language provided by Coq. To do that, we identify specific patterns in the context and the goal statement that we simplify by a sequence of tactics and lemmas. The development of tactics remains a relatively tricky and challenging problem. The aim is to treat all possible configurations. This requires a good scheduling of simplifications and attempts to resolve the problem. If one simplifies too fast or too deeply, the system will not be able to prove the result, otherwise the search tree can quickly become large and considerably slow down the use of such tactics.

Tactics like the one depicted in Table 18 allow us to unfold as much as possible the goal while introducing as many assumptions as possible. If the goal has become trivial with a simple contradiction or equality/inequality, the work is done. Otherwise one has to make explicit a contradiction in the assumptions. This logic work is the core of the proof and must generally be performed by hand. To raise contradictions in the context, it makes sense to write intermediate lemmas.

```

Ltac my_rank :=
repeat match goal with
| [H : _ |- _] => [progress intros|progress intro]
| [H : _ |- _] => solve[intuition]
| [H : _ |- _] => progress contradiction
| [H : _ = _ |- _] => solve[inversion H]
| [H : ?X[==]?X -> False |- _] => apply False_ind;apply H;reflexivity
| [H : _ |- (if if ?X then _ else _ then _ else _) = _ \/_] => case_eq X
...
end.

```

Table 18 Example of simplification tactics⁸. For a more detailed understanding of the main Coq commands see [4]

⁶ Definition `contains_four_non_coplanar_points` is available in the file `psoh_equiv_c4p`. Definition `coplanar_with_all` is available in the file `psoh_equiv_copwa`.

Proof of the non-decreasing property

In order to exemplify the proof methodology from the geometric axiom system to the matroid theory, we detail the proof of the non-decreasing property.

Lemma 4 (*A2R2-R3*),

$$\forall X \subseteq Y, rk(X) \leq rk(Y).$$

In our context, the rank function can take five possible values from 0 to 4, thus resulting in twenty five different cases when we make a structural induction on X and Y which are lists. First case: $0 \leq 0$, second case: $0 \leq 1$, etc. Thanks to tactics, we eliminate fifteen trivial goals immediately wherein X is included in Y . To prove the remaining ones, we build a contradiction and we simplify the goal using the tactic `my_inA` (see Table 19) which automatically handles the calculation of inclusion. For example, we make clear that a plane cannot be included in a line. It is therefore impossible to reach the case $3 \leq 2$.

The notation `InA` denotes the inclusion relation over type A in the case of lists with respect to a setoid equality. Setoids in Coq make it possible to declare a new equivalence relation that can be used in certain tactics and especially during rewrites. When it can be proved that two terms are equivalent but not necessarily equal, it is possible to replace one of the terms by the other. In our case, we use a particular equality `[]=` different from the usual equality of Coq. We must therefore specify the predicates for which rewriting is allowed using this equality. The `InA` predicate only accepts rewrites if the equality provided is an equivalence relation. This tactic proposes an analogous resolution to `fsetdecide` in the case where sets are represented by lists while being much more efficient in time. Indeed, all the abstract layers of the *Containers* library are eliminated, thus facilitating the verification of the proof term.

```

Ltac my_inA :=
  intuition;unfold equivlistA in *; unfold inclA in *;
  repeat match goal with
  |[H : _ |- _] => progress intros
  |[H : _ |- _] => progress intro
  |[H : _ |- _] => progress intuition
  |[H : _ |- _] => progress my_subst
  |[H : _ |- Equivalence eq] => apply eq_equiv
  |[H : InA eq _ _ |- InA eq _ ( app _ _ )] => apply InA_app_iff
  |[H : InA eq _ ( app _ _ ) |- _] => apply InA_app_iff in H
  |[H : _ |- _] => split;intuition
  |[H : InA eq _ ( ?P :: ?Q ) |- _] => inversion H; clear H
  |[H : InA eq _ nil |- _] => inversion H
  |[H : _ = nil |- _] => inversion H
  |[H : InA eq ?P ( ?P :: ?Q ) -> _ |- _] => let T:=fresh in
    assert(T : InA eq P ( P :: Q)) by (intuition)
  |[H : InA eq ?P ( ?Q :: ?P :: ?Z ) -> _ |- _] => let T:=fresh in
    assert(T : InA eq P ( Q :: P :: Z)) by (apply InA_cons_tl;intuition)
  end.

```

Table 19 Tactic of automatic computation of inclusion⁹

⁸ Ltac `my_rank` is available in the file `psoh.equiv_rk.lemmas`.

Coq implementation

The submodularity lemma (A3R2-R3) is the hardest property to prove with regards to the management of the intersection and the union. It contains many non-trivial cases that we decompose into several lemmas as in Table 20. We note `list_inter` the literal intersection between two lists. The first example indicates that if two lines l and m are merged, it is possible to represent the problem by a single line containing all the points of the two lines. The second one specifies that if a line m is contained in a plane p then all the points of the line m can be added to the plane p to form a single object. All these usual lemmas of projective incidence geometry describing results of the different intersections of possible objects must be defined in order to be able to characterize all the cases of the submodularity lemma.

```
(* matroid3_rk2_rk2_interrk2_to_unionrk2 *)
Lemma matroid3_aux17_bis_aux :
forall l m,
rk1 l = 2 ->
rk1 m = 2 ->
rk1 (inter l m) = 2 ->
rk1 (union l m) = 2.
```

```
(* matroid3_rk3_rk2_interrk2_to_unionrk3 *)
Lemma matroid3_aux17_bis_bis_aux :
forall l m,
rk1 p = 3 ->
rk1 m = 2 ->
rk1 (list_inter p m) = 2 ->
rk1 (p ++ m) = 3.
```

Table 20 Examples of intermediate lemmas¹⁰

Finally, when we define a new predicate with respect to a setoid equality that does not belong to the standard library, we must write a morphism. This morphism is the proof that makes it possible to rewrite with our equivalence relation `[==]`. Thus for each definition involving the point type, it is necessary to establish a morphism. This ensures that our equality is compatible and consistent with the definition in which it is used. In this way, it is possible to substitute a set X similar to a set Y directly in the rank definition. The morphism, as expressed in Coq, is given in Table 21.

⁹ Ltac `my_inA` is available in the file `p_equiv_list_additions`.

¹⁰ Lemma `matroid3_aux17_bis_aux` and `matroid3_aux17_bis_bis_aux` are available in the file `psoh_equiv_rk_lemmas`.

```
Instance rank_morph : Proper (@equivlistA Point eq => (@Logic.eq nat)) rkl.
Proof.
...
Qed.
```

Table 21 Example of morphism¹¹

Once the `Proper` predicate is unfolded, the following lemma, where the predicate `equivlist` indicates that the two lists are equal, must be proved:

Lemma 5 *rank_morph*,

$\forall x y : \text{list Point}, \text{equivlist } x y \rightarrow \text{rkl } x = \text{rkl } y$

The Coq proofs of the other lemmas of projective geometry follow the same principles.

4.3 Statistics

Finally, we give some data about our library, and we also detail the impact of automation that we have put in place throughout the process. Overall, our development consists of more than 35 000 lines and 1 000 lemmas. As shown in the Table 22, the proof of the equivalence breaks down to around 15 000 lines. By using tactics and splitting the demonstration into lemmas, the size of the formalization can be reduced by a factor of 3. With the sole aim of establishing equivalence and maximizing automation, it is still possible to improve this result. This development aims at being a library as exhaustive as possible for projective incidence geometry.

	From rank to PG		From PG to rank	
	2D	3D	2D	3D
lines of Coq specs	250	400	650	1 200
lines of Coq proofs	300	1 500	2 600	12 500

Table 22 Organization of equivalence proof in Coq

5 Conclusion and Future Work

This paper explains how projective geometry can be formalized in Coq using two different axiom systems in both 2D, $\geq 3D$ and 3D. We first described the original axiomatization based on the notion of rank from matroid theory. Then, we rigorously proved the equivalence between these two formalizations. We highlighted the tools and techniques used to achieve it by presenting a typical proof. Finally we give some thoughts on proof automation in the presence of ranks.

¹¹ Morphism `rank_morph` is available in the file `psoh_equiv_rk_lemmas`.

We discuss here the automation possibilities that can be envisaged by combining the two approaches. For this, we provide an overview of a work in progress and tracks that we wish to explore.

Automation in incidence geometry quickly reaches its limits when the user has to manipulate several objects in dimensions greater than 2. Developing lemmas and tactics becomes a tedious task since it is necessary to duplicate the treatments for all combinations of processed objects. At each transition to a higher dimension, it is mandatory to add a new object, new predicates, and update the set of simplification rules. From this point of view, the approach of geometry with ranks seems much more promising.

As we mentioned in the first implication of equivalence, submodularity plays a decisive role in the proof of lemmas involving the concept of rank. This is the key concept to master and to manipulate in order to automate proofs. By itself, it allows us to control dimensions and intersections of flats. If we want to automate proofs involving ranks, we need to formalize a procedure that applies the submodularity axiom to be able to discover as much information as possible about the dimensions of different sets of points. In practice, this process is already in place. However, there is still a major problem: How can we guide the automation to compute the relevant sets of points? This problem already exists in the case of projective incidence geometry when it comes to creating new objects. At present, the rank-based approach facilitates proof simplifications while maintaining control over the direction of calculated sets. Nevertheless, we plan to use combinatorial solutions to compute the ranks of numerous subsets before attempting to solve the theorem. The underlying problem in this case is to avoid the combinatorial explosion. Indeed if we consider a geometric problem with n points, computing the set of flats of this theorem amounts to calculating partitions of the set containing all points, ie 2^n flats.

In the work currently in progress, we study these problems of combinatorial explosion in the specific case of finite geometries. Preliminary results indicate that it is possible to obtain far more satisfactory results by using the ranks in the most complicated cases.

In the future, we plan to carry on with our investigations in two main directions. On the one hand, we expect to write a reliable algorithm for performing a bilateral translation between the two approaches. In this way we can alternate between geometry for visualization and combinatorics for automation and computation.

On the other hand, we shall study how to efficiently implement the search for new flats to automate proofs about ranks at different levels while managing combinatorial complexity. First, we will finish our case study on the possibilities of full automation in the case of finite geometries. Secondly, we are interested in a partial automation of many steps in the Desargues' theorem presented in [23, 24] using mainly the submodularity. To extend this analysis, we will examine two other consequent theorems in projective incidence geometry: Dandelin-Galluci theorem [1, 2] and the harmonic conjugate. The aim will be to make proof as readable as possible by removing technical details, thus being as close as possible to a mathematical proof.

Availability The last version of full Coq development is available from github repository in the branch *amai2018* at:

<https://github.com/ProjectiveGeometry/ProjectiveGeometry>

References

1. Baker, Henry Frederick: Principles of geometry, vol. 1. Cambridge University Press (1925)
2. Bamberg, John and Penttila, Tim: Completing Segre's proof of Wedderburn's little theorem. *Bulletin of the London Mathematical Society* 47(3), 483–492 (2015)
3. Batten, Lynn Margaret: Combinatorics of finite geometries. Cambridge University Press (1997)
4. Bertot, Yves and Castéran, Pierre: Interactive Theorem Proving and Program Development, Coq'Art: The Calculus of Inductive Constructions. Texts in Theoretical Computer Science, Springer Science & Business Media (2004)
5. Boutry, Pierre and Narboux, Julien and Schreck, Pascal and Braun, Gabriel: Using small scale automation to improve both accessibility and readability of formal proofs in geometry. *Automated Deduction in Geometry 2014* pp. 1–19 (2014)
6. Braun, D., Magaud, N., Schreck, P.: An Equivalence Proof Between Rank Theory and Incidence Projective Geometry. In: *Proceedings of Automated Deduction in Geometry 2016*. pp. 62–77 (2016), <https://hal.inria.fr/hal-01334334>
7. Buekenhout, Francis: Handbook of Incidence Geometry: buildings and foundations. Elsevier (1995)
8. Castéran, Pierre and Sozeau, Matthieu: A Gentle Introduction to Type Classes and Relations in Coq (2016)
9. Coq development team: The Coq proof Assistant Reference Manual. <https://coq.inria.fr/distrib/current/files/Reference-Manual.pdf> (2016)
10. Coxeter, Harold Scott Macdonald: Projective Geometry. Springer Science & Business Media (2003)
11. Dehlinger, Christophe and Dufourd, Jean-François and Schreck, Pascal: Higher-Order Intuitionistic Formalization and Proofs in Hilbert's Elementary Geometry. *Automated Deduction in Geometry* pp. 306–324 (2000)
12. Duprat, Jean: Une axiomatique de la géométrie plane en Coq. *Journées Francophones des Langages Applicatifs* pp. 123–136 (2008)
13. Fuchs, Laurent and Thery, Laurent: A formalization of grassmann-cayley algebra in Coq and its application to theorem proving in projective geometry. *Automated Deduction in Geometry 6877*, 51–67 (2010)
14. Génevaux, Jean-David and Narboux, Julien and Schreck, Pascal: Formalization of Wu's simple method in Coq. *Certified Programs and Proofs 7086*, 71–86 (2011)
15. Guillot, Frédérique: Formalisation en Coq et visualisation d'un cours de géométrie pour le lycée. *Journées Francophones des Langages Applicatifs 7*, 15 (2004)
16. Janičić, Predrag and Narboux, Julien and Quaresma, Pedro: The Area Method : a Recapitulation. *Journal of Automated Reasoning* 48(4), 489–532 (2012)
17. Jermann, Christophe and Trombettoni, Gilles and Neveu, Bertrand and Mathis, Pascal: Decomposition of Geometric Constraint Systems: a survey. *International Journal of Computational Geometry & Application* 16(05n06), 379–414 (2006)
18. Kahn, Gilles: Constructive geometry according to Jan von Plato (1995)
19. Kusak, Eugeniusz: Desargues theorem in projective 3-space. *J. of Formalized Mathematics* 2 (1990)
20. Lescuyer, Stéphane: First-Class Containers in Coq. *Stud. Inform. Univ.* 9(1), 87–127 (2011)
21. Li, Hongbo and Wu, Yihong: Automated short proof generation for projective geometric theorems with Cayley and bracket algebras: I. Incidence geometry. *Journal of Symbolic Computation* 36(5), 717–762 (2003)
22. Magaud, Nicolas and Narboux, Julien and Schreck, Pascal: Formalizing Projective Plane Geometry in Coq. *Automated Deduction in Geometry 6301*, 141–162 (2008)
23. Magaud, Nicolas and Narboux, Julien and Schreck, Pascal: Formalizing Desargues theorem in Coq using ranks. In *Proceedings of the 2009 ACM symposium on Applied Computing* pp. 1110–1115 (2009)

24. Magaud, Nicolas and Narboux, Julien and Schreck, Pascal: A Case Study in Formalizing Projective Geometry in Coq: Desargues Theorem. *Computational Geometry : Theory and Applications* 45(8), 406–424 (2012)
25. Meikle, Laura and Fleuriot, Jacques: Formalizing Hilbert’s Grundlagen in Isabelle/Isar. *Theorem proving in higher logics* 2758, 319–334 (2003)
26. Michelucci, Dominique and Fofou, Sebti and Lamarque, Loïc and Schreck, Pascal: Geometric constraints solving:some tracks. *ACM* pp. 185–196 (2006)
27. Michelucci, Dominique and Schreck, Pascal: Detecting induced incidences in the projective plane. *isiCAD Workshop*. Citeseer (2004)
28. Michelucci, Dominique and Schreck, Pascal: Incidence constraints : a combinatorial approach. *International J. of Computational Geometry & Application* 16(05n06), 443–460 (2006)
29. Moulton, Forest Ray: A Simple Non-Desarguesian Plane Geometry. *Transactions of the American Mathematical Society* 3(2), 192–195 (1902)
30. Narboux, Julien: A Decision Procedure for Geometry in Coq. *Theorem Proving in Higher Order Logics* 3223, 225–240 (2004)
31. Narboux, Julien: Formalisation et automatisation du raisonnement géométrique en Coq. Ph.D. thesis, Université Paris Sud-Paris XI (2006)
32. Narboux, Julien: Mechanical Theorem Proving in Tarski’s Geometry. *Automated Deduction in Geometry* 4869, 139–156 (2006)
33. Oxley, James: *Matroid Theory*, vol. 3. Oxford University Press, USA (2006)
34. Richter-Gebert, Jürgen: Mechanical theorem proving in projective geometry. *Annals of Mathematics and Artificial Intelligence* 13, 139–172 (1995)
35. Sozeau, Matthieu and Oury, Nicolas: First-class type classes. *Theorem Proving in Higher Order Logics* 5170, 278–293 (2008)
36. Tarski, Alfred: *What is Elementary Geometry ?* (1983)
37. Von Plato, Jan: The axioms of constructive geometry. *Annals of pure and applied logic* 76(2), 169–200 (1995)