Affine Bases for Affine Spaces

Gabriel Dogadov Marc Alexa
TU Berlin, Computer Graphics Group
cg.tu-berlin.de

Abstract

Recent work suggests quadratic distance fields (QDFs) as a representation of affine subspaces of Euclidean space (often called flats), as they enable an efficient computation of weighted means: compute the weighted mean of the QDFs and then project back onto the closest QDF representing a flat. Considering this operation as an affine combination of flats, we 1) analyze the prerequisites for flats forming a basis of the space of flats and 2) provide an algorithm for projecting flats into a such a basis. Intriguingly, the projection requires nothing more than the solution of a linear system. This allows computing with flats similar to how we commonly compute with points in Euclidean spaces, with comparable cost despite the curved nature of the manifold of flats. We demonstrate the versatility of this idea by triangulating local spaces of affine lines, to be used, e.g., in image based rendering.

1. Introduction

At the core of computer vision, as well as many other disciplines in science and engineering, is the concept of an *affine space*. Affine spaces are not only appropriate models for the idea of local coordinate systems, they also represent the basic building blocks for geometry, namely lines and (hyper)planes, not necessarily containing the origin. Moreover, affine spaces are a common representation in many processing stages of computer vision applications:

As input data: Data with invariance properties, such as faces under varying lighting conditions, facial expressions, poses, viewpoints, deformations [7, 16, 17, 19, 34]; also when some entries of the data are unknown [28].

As features or intermediate representation: Point and line features in multi-view geometry tasks such as 3D reconstruction and structure from motion (SfM), image alignment, registration, and more generally features in machine learning [14, 29, 36].

As output data: Principal Component Analysis (PCA) [27] is perhaps the most famous algorithm of such a kind, approximating point clouds with a single affine space. Subspace clustering algorithms [30] can be seen as a natural generalization of PCA. They describe point data as a mixture of affine subspaces, possibly of varying dimensions.

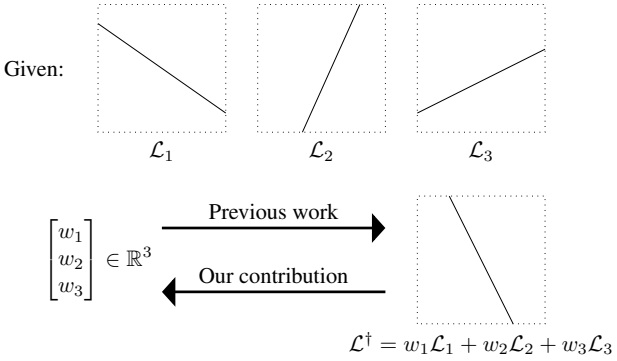


Figure 1. Previous work [4] provided a simple way to generate weighted means of flats. Here, we show what is necessary for flats to define a basis for affine spaces, and how to compute the representation of a given flat in a given basis, i.e., projecting an element.

In most cases, these affine spaces are, in fact, proper subspaces, and it has become common to refer to them based on their distinguishing geometric feature as *flats* [9].

Given their prominence in (geometric) computing, it seems desirable to be able to do computations with flats themselves. Unlike points in Euclidean spaces, however, flats of fixed dimension $k > 0$ in a space of dimension $d > k$ form a smooth but generally curved manifold (of dimension $(k+1)(d-k)$). While the manifold of linear subspaces, the *Grassmannian*, has been studied extensively [1, 32], the dedicated analysis of the space of flats has received much less attention. Flats may be represented using homogeneous coordinates in Grassmannians similar to how homogeneous coordinates in linear spaces can be used to represent projective spaces (Plücker coordinates for affine lines are the best-known example). An analysis of the space of flats without adding elements at infinity has appeared only recently [20, 21].

One may argue that flats already have a well-established representation in computer graphics: as quadratic distance fields (QDF) [10]. The central observation of Garland and Heckbert was that QDFs can be combined by simply adding them. Dogadov et al. [4] observed that the resulting sum can be projected back to the manifold of flats by simply adjusting the eigenvalues of the bilinear form. This simple procedure gives a method that allows computing weighted means of flats or fit flats of varying dimensions to other flats in the least-squares sense. Intriguingly, this method is invariant to rigid transformations and respects symmetries, unlike existing solutions based on Grassmannians.

We interpret this result as the computation of affine combinations of flats. In particular, $(k+1)(d-k)+1$ flats form a simplex in the space of flats. Since the manifold of flats contains the Grassmannian, and Grassmanianns cannot be covered by a single contractible chart, affine combinations of a fixed set of flats cannot yield all flats, yet the simplex is still an affine basis for a *local* subset of flats. To make concrete examples, in three dimensions, four affine planes span a local space of planes; or five affine lines span a local space of affine lines (we provide visual examples of how such spaces may look like in Section 5). What is missing to make flats first-class citizens for computation (at least in a local subspace) is am proper definition of local bases and then the computation of a representation in the this basis: Given the $(k+1)(d-k)+1$ basis elements of a local subspace of flats, when do they properly span a subspace of all flats of this dimension and how do we find the affine combination for a given flat to be represented in that basis (see Fig. 1)? We discuss the property of bases in Sec. 3 and derive method for finding the coefficients in Sec. 4. Interestingly, despite the seemingly non-linear nature of the problem, it requires nothing more than the solution of a small linear system, similar to the linear case. We show several illustrative examples of what this could be used for in Section 5, and conclude with some remarks on avenues in real applications.

2. Background: Quadratic Distance Fields

In general, a quadratic distance field (QDF) of a (connected) set $\Omega \in \mathbb{R}^d$ is a scalar field that maps every point in space onto its squared orthogonal distance to Ω:

$$d^2_\Omega : \mathbb{R}^d \to \mathbb{R}, \quad \mathbf{x} \mapsto \min_{\mathbf{y}\in\Omega} d^2(\mathbf{x},\mathbf{y}). \tag{1}$$

A quadratic distance field (QDF) is essentially the square function of the signed distance field (SDF), a popular tool to represent geometry in computer graphics and vision [23]. While the latter may be more useful for most applications due to several desired properties (Lipschitz-continuity and notion of inside and outside to name two), the former comes in handy when working with flats.

2.1. Flat Representations

For geometric computations, a k-dimensional flat $\mathcal{F}$ is often represented in either parametric form or normal form. The parametric representation consists of a matrix $\mathbf{A} \in \mathbb{R}^{d\times k}$ whose columns form a basis of the corresponding linear subspace and a vector $\mathbf{b} \in \mathbb{R}^d$ representing its shift from the origin. In normal form, the matrix $\mathbf{N} \in \mathbb{R}^{d\times d-k}$ contains the normal vectors and a signed distance vector $\mathbf{c} \in \mathbb{R}^{d-k}$ encodes the shift from the origin. One typically imposes orthogonality constraints on the respective matrices and vectors.

The QDF representation follows from the normal form. Given a point $\mathbf{x}$ in space, its distance to $\mathcal{F}$ is given by $\|\mathbf{N}^\mathsf{T}\mathbf{x} - \mathbf{c}\|$, and, thus, the quadratic distance field is given by

$$\begin{aligned} q(\mathbf{x}) = \|\mathbf{N}^\mathsf{T}\mathbf{x} - \mathbf{c}\|^2 &= \mathbf{x}^\mathsf{T}\mathbf{N}\mathbf{N}^\mathsf{T}\mathbf{x} - 2\mathbf{x}^\mathsf{T}\mathbf{N}\mathbf{c} + \mathbf{c}^\mathsf{T}\mathbf{c} \\ &= \mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{x} - 2\mathbf{r}^\mathsf{T}\mathbf{x} + s. \end{aligned} \tag{2}$$

The points on the flat satisfy $q(\mathbf{x}) = 0$ as well as $\mathbf{Q}\mathbf{x} = \mathbf{r}$. This property holds up to scaling, thus, any nonzero multiple of $(\mathbf{Q}, \mathbf{r})$ can be treated as the same flat. The QDF and thus the flat itself can be parameterized as a pair $(\mathbf{Q}, \mathbf{r})$. The following relations hold:

$$\mathbf{Q} = \mathbf{N}\mathbf{N}^\mathsf{T} = \mathbf{I} - \mathbf{A}\mathbf{A}^\mathsf{T} \quad \text{and} \quad \mathbf{r} = \mathbf{N}\mathbf{c} = \mathbf{b}, \tag{3}$$

while the scalar $s \in \mathbb{R}$ can be omitted since it can be derived from $s = \mathbf{r}^\mathsf{T}\mathbf{r}$ and is not required for most computations, as argued in [4]. Note that not all choices of $(\mathbf{Q}, \mathbf{r}) \in \mathbb{R}^{d\times d} \times \mathbb{R}^d$ represent a QDF of a k-flat. They have to satisfy the following properties:

(C1) The matrix $\mathbf{Q}$ is symmetric, rank $d-k$, and idempotent. Its null space (0-eigenspace) is $\mathrm{span}(\mathbf{A})$. Its image (1-eigenspace) is $\mathrm{span}(\mathbf{N})$.

(C2) The vector $\mathbf{r}$ is a fixed point of $\mathbf{Q}$, i.e., $\mathbf{Q}\mathbf{r} = \mathbf{r}$.

We refer to [4] for more details.

Note: Unlike the previously discussed representations, the QDF is uniform for flats of all dimensions given the same ambient space $\mathbb{R}^d$. Although this enables computations with flats of varying dimensions, in the following, we will deal only with flats of the same fixed dimension k.

2.2. Affine Combinations of QDFs

As QDFs are scalar fields, they can be linearly combined. A linear combination $q^*(\mathbf{x}) = \sum_{i=1}^m w_i q_i(\mathbf{x})$ can be attained by a weighted sum of the respective representations:

$$\mathbf{Q}^* = \sum_{i=1}^m w_i \mathbf{Q}_i \quad \text{and} \quad \mathbf{r}^* = \sum_{i=1}^m w_i \mathbf{r}_i. \tag{4}$$

Due to the scaling invariance of the representation, the coefficients can be constrained to sum up to one: $\sum_{i=1}^m w_i = 1$.

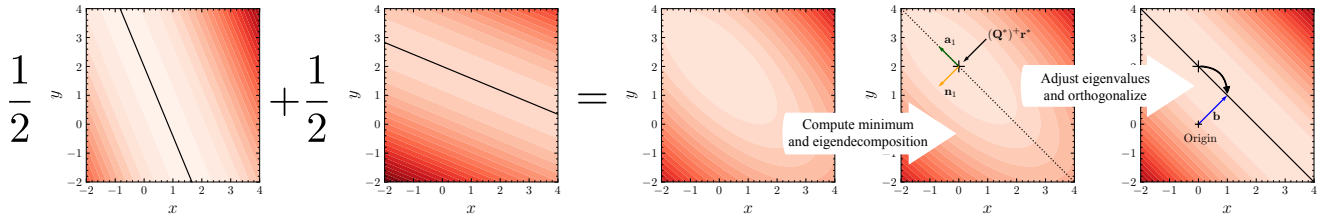


Figure 2. Stepwise visualization of the mean computation of two line QDFs followed by the projection onto the space of valid $(\mathbf{Q}, \mathbf{r})$-representations.

Such a weighted sum is called an *affine combination*. To the best of our knowledge, the idea of summing QDFs dates back to Garland and Heckbert [10] in the context of triangle mesh simplification under the name *Quadric Error Metrics (QEMs)*. In their proposed algorithm, each triangle of the mesh is abstracted by its plane and represented as a QDF, which is parameterized by a $(d+1) \times (d+1)$ matrix $\mathbf{Q}_H$ in homogeneous coordinates with

$$\mathbf{Q}_H = \begin{bmatrix} \mathbf{Q} & -\mathbf{r} \\ -\mathbf{r}^\mathsf{T} & s \end{bmatrix}. \tag{5}$$

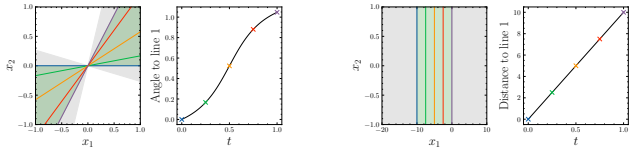
Similar to Eq. (4), the sum of QDFs corresponds to the sum of the respective homogeneous matrices. Although the proposed algorithm computes unweighted sums, they discuss weighting the QDF with the surface area of the triangle it belongs to. In the following years, QDFs and their weighted sums have been used in similar ways for (attribute-aware) surface simplification and compression [11, 12, 22], mesh segmentation [33], shape reconstruction [35] and recently line triangulation from planes [4].

2.3. Projection

An affine combination of QDFs of flats is generally not a QDF of a flat. This, for instance, can be seen in the spectrum of the respective matrix $\mathbf{Q}^*$: the eigenvalues are not necessarily exactly zero or one [8, 15, 31]. Dogadov et al. [4] suggest to map a sum of QDFs to a the QDF of a flat, which is a projection $\mathcal{P}$ that maps $q^* \mapsto \tilde{q}$ and the respective representation $(\mathbf{Q}^*, \mathbf{r}^*) \mapsto (\tilde{\mathbf{Q}}, \tilde{\mathbf{r}})$.

1. The matrix $\tilde{\mathbf{Q}}$ results from taking the smallest k eigenvalues of $\mathbf{Q}^*$ and setting them to zero, while setting the remaining eigenvalues to one. The corresponding eigenvectors to the smallest/largest eigenvalues form the spanning/normal vectors of the resulting flat, respectively.
2. The minimizer $\mathbf{x}^*$ (saddle point if we allow negative coefficients) of q^* is computed as the solution to $\mathbf{Q}^*\mathbf{x} = \mathbf{r}^*$. Afterward, it has to be projected into the 1-eigenspace of $\tilde{\mathbf{Q}}$, resulting in $\tilde{\mathbf{r}} = \tilde{\mathbf{Q}}(\mathbf{Q}^*)^+\mathbf{r}^*$, where $(\cdot)^+$ denotes the pseudo-inverse.

The first step is essentially the computation of a chordal flag average [5, 25], in which the subspaces are represented through their orthogonal complement. Fig. 2 visualizes the described computations at the example of two lines in 2-space.



(a) Intersecting lines (b) Parallel lines

Figure 3. Linear interpolation between two lines ($\mathcal{L}_1$ in blue and $\mathcal{L}_2$ in purple) in QDF representation. Left images: The lightgreen region indicates the convex hull of the two lines, while the gray region indicates their affine hull. Right images: Angle and distance of intermediate lines to $\mathcal{L}_1$.

3. (Affine) Bases of Flats

By combining the ideas of linearly combining QDFs and projecting them onto a proper representation, the respective coefficients $w_1, \ldots, w_m$ can be interpreted as the coordinates of the resulting flat with respect to the given "basis" flats. Fig. 3 shows, at least for simple cases, the predictable interpolation behavior at the example of two intersecting (left) and parallel (right) lines in $\mathbb{R}^2$. While the distance between the two lines is interpolated linearly in the parameter, the angle is interpolated in a sinusoidal shape (which is expected, since the procedure is optimizing for chordal distances of the orientation [4, 5, 21, 24]). However, the interpolation behavior of more complex cases (e.g., two or more skew lines in $\mathbb{R}^3$) may not be as clear (see Fig. 6 and Sec. 5). Similarly to the Euclidean case, we can now define the notion of the affine hull of a set of flats/QDFs $q_1, \ldots, q_m$ as the set of their projected affine combinations:

$$\mathcal{A}(q_1, \ldots, q_m) = \left\{ \mathcal{P}\left(\sum_{i=1}^{m} w_i q_i \right) \;\middle|\; \sum_{i=1}^{m} w_i = 1 \right\}. \tag{6}$$

Similarly, we denote the convex hull as $\mathcal{C}(\cdot)$, where the weights have to be non-negative, i.e., $w_i \geq 0$. As an example, Fig. 3 shows the convex and affine hulls of two lines.

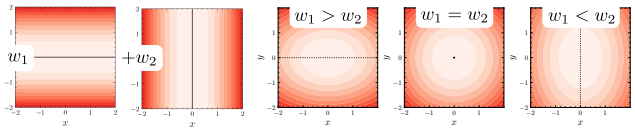


Figure 4. Basis of orthogonal lines and their possible combinations.

3.1. One Dimension

In the following, we discuss some properties of the sets of flats spanned by two basis elements. We start by noticing that two orthogonal flats are not spanning a one-dimensional space (see Fig. 4 and [4]). We claim that the space spanned by two non-orthogonal flats is injective, i.e., different weights imply different flats. This is non-trivial, because the projection appears to be not necessarily unique if the linear combination of the QDFs has coinciding kth and $k+1$st smallest eigenvalues. Consequently, our first goal is proving that these eigenvalues do not coincide.

The statement about eigenvalues refers to the directions of the flats, so it makes sense only for flats that intersect. We can always translate this intersection to the origin, so it suffices to consider the matrices $\mathbf{Q}_0, \mathbf{Q}_1$, resp. their affine span

$$\mathbf{Q}(\mu) = (1-\mu)\mathbf{Q}_0 + \mu\mathbf{Q}_1. \tag{7}$$

We make the following interesting observation:

Lemma 1. *Given two $\mathbf{Q}_0, \mathbf{Q}_1$ that are part of a QDF representation of two flats that are not orthogonal, the matrices $\mathbf{Q}(\mu)$ do not have the eigenvalue μ (except for $\mu \in \{0, 1\}$).*

The proof follows from the fact $\mathbf{Q}_0, \mathbf{Q}_1$ are projection matrices whose null space and column space are orthogonal, and is detailed in the supplementary material.

By symmetry, $\mathbf{Q}(\mu)$ also has no eigenvector to the eigenvalue $(1-\mu)$ (except for $\mu \in \{0, 1\}$). These 'forbidden' eigenvalues lead to the following result:

Lemma 2. *The eigenvalues λ_k and λ_{k+1} of $\mathbf{Q}(\mu)$ are different (assuming non-orthogonal flats and ascending eigenvalues, as above).*

The proof follows from simple continuity arguments about the eigenvalues (the eigenvalues for $\mu \in [0, 1]$ are illustrated in Fig. 5) and is also detailed in the supplemental material.

With that we have proved that the degenerate case discussed by Dogadov et al. [4], namely that the smallest k eigenvalues might not be unique (so that the resulting flat is not unique) cannot happen, if the two flats are not orthogonal. Moreover, this implies that all flats generated by affine combinations are different:

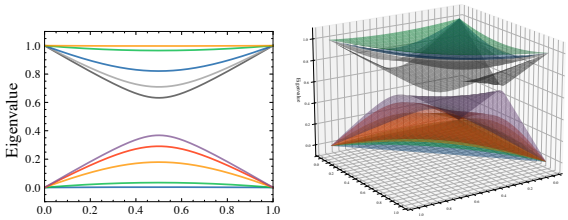


Figure 5. Spectrum of $(1-t)\mathbf{Q}_1 + t\mathbf{Q}_2$ with $t \in [0, 1]$ (left) and $(1-s-t)\mathbf{Q}_1 + s\mathbf{Q}_2 + t\mathbf{Q}_3$ with $s, t \in [0, 1],\ s+t \leq 1$ (right), the matrices $\mathbf{Q}_1, \mathbf{Q}_2\mathbf{Q}_3 \in \mathbb{R}^{10\times 10}$ having rank 5.

Theorem 1. *The flats represented by $\mathbf{Q}(\mu)$ and $\mathbf{Q}(\mu')$ for $\mu \neq \mu'$ are different if $\mathbf{Q}_0$ and $\mathbf{Q}_1$ represent different flats and are not orthogonal.*

This follows from an extension of the fact that if two matrices share an eigenvector, all their linear combinations have this eigenvector, resp. the contrapositive statement. The details are in the supplemental material.

Concluding, two flats that are not identical and not orthogonal span a one-dimensional space of flats. If the flats are not parallel, the elements share the common subspace of the two flats and span a set of direction within an angle of π, covering (at least) the smaller principal angle between the two flats. Note that while there is a one-to-one correspondence between affine coefficients and flats, the one-dimensional affine space only covers a part of the whole space of flats.

3.2. Multidimensional Spaces

For a non-degenerate base of an affine m-space, we would ask that any of the $m+1$ basis elements cannot be represented as an affine combination of the remaining m basis elements. We naturally ask the same property for any basis of a space of flats, yet observe that this applies to *flats* – the $\mathbf{Q}$ part of the representation may well represent the same linear subspace, i.e. the affine combination of m flats may be parallel to the last basis element.

In addition, we have to deal with the requirement that basis elements are not orthogonal. Similar to affine independence, we ask that no affine combination of m basis elements generates a flat that is orthogonal to the remaining basis element.

We note that even for more than one dimension, the restriction of the flats to directional intervals of π remains. It may be tempting to use three affine lines forming a regular triangle as bases. But notice that the midpoint between any two of them is orthogonal to the third one. The resulting space is fragmented, containing, for example, only some of the lines parallel to the bases intersecting the triangle (see Fig. 6, left). A basis satisfying our criteria results from starting with any two affine lines and then adding a third that is in the span of the directions of the first two, yet displaced

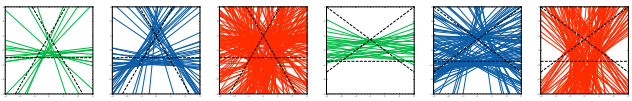


Figure 6. Two different bases of affine lines (black, dashed) in 2-space forming a triangle. Other lines are colored green if they lie inside the convex hull of the three lines, blue if they lie inside the affine but not convex hull and red if neither is the case.

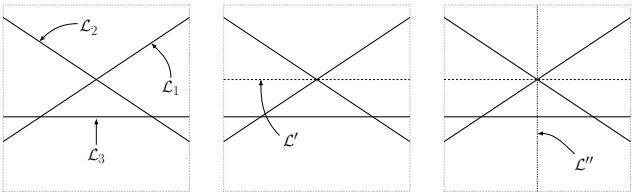


Figure 7. A basis of three lines $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ (left), of which two elements $\mathcal{L}_1, \mathcal{L}_2$ can be used to produce a line $\mathcal{L}'$ (middle, dashed) that is parallel to the remaining line $\mathcal{L}_3$. Affinely combining the unprojected representation of $\mathcal{L}'$ with $\mathcal{L}_3$ can produce an orthogonal line $\mathcal{L}''$ (right, dotted).

so that there is no common intersection. This results in an obtuse triangle and all lines in the directional span intersecting the triangle are affine and, in fact, convex combination of the three lines (see Fig. 6, right).

However, this construction also reveals a fundamental problem for bases with more than two elements: we can construct parallel flats, whose $\mathbf{Q}$ representation (prior to projection) is generally different. The affine span of such representations has elements with different eigenvalues to the same eigenspaces, so it can also have coinciding eigenvalues (as shown in Fig. 7). So, while we have found it useful to allow working with general affine space for practical applications, in order to prove anything further restrictions are necessary, and we find it natural to consider only convex combinations.

Under this assumption, we expect that Theorem 1 can be extended from the one-dimensional case to arbitrary dimensions. Specifically, let $\mathbf{Q}^*$ be a convex combination of $\mathbf{Q}$-matrices of m non-orthogonal flats. It then remains to show that the kth eigenvalue of $(1-\mu)\mathbf{Q} + \mu\mathbf{Q}^*$ is not repeated for $0 < \mu < 1$, provided that the kth eigenvalue of $\mathbf{Q}^*$ itself is simple and that $\mathbf{Q}$ represents another non-orthogonal k-flat not contained in the convex hull of the remaining m flats. Under this non-degeneracy assumption, the result would follow inductively, using arguments about continuity of the eigenvalues and corresponding eigenspaces analogous to those in the one-dimensional case. The eigenvalues for a two-dimensional basis are visualized in Fig. 5 (right).

4. Computing a Representation

Although the (affine) Grassmannian is by no means a linear space, we established its usefulness to work with affine combinations using the $(\mathbf{Q}, \mathbf{r})$-representation, with the help of which a flat can be approximately represented as the weighted sum of (other) flats by interpreting the weights as their coordinates. In this context, the following natural question arises:

> Given an affine basis $q_1, \dots, q_m$ of flats in QDF representation as $(\mathbf{Q}_1, \mathbf{r}_1), \dots, (\mathbf{Q}_m, \mathbf{r}_m)$ and a target QDF $q^\dagger$ as $(\mathbf{Q}^\dagger, \mathbf{r}^\dagger)$, is it possible to express $q^\dagger$ as a convex (or affine) combination of $q_1, \dots, q_m$, and, if yes, what are the respective coordinates $w_1, \dots, w_m$?

We derive a method to solve this problem that only requires a system of linear equations. To simplify notation, we define $\mathbf{Q}^* = \sum_{i=1}^m w_i \mathbf{Q}_i$ and $\mathbf{r}^* = \sum_{i=1}^m w_i \mathbf{r}_i$, of which the coefficients $w_1, \dots, w_m$ are unknown.

4.1. Linear subspaces

We first describe our method for linear subspaces, ignoring the shift from the origin ($\mathbf{r}_i = \mathbf{0}$), which we will afterward extend to general flats. As the dimension of $\mathrm{Gr}(k, d)$ is $k(d-k)$, we assume the number of basis elements is no more than $m = k(d-k) + 1$.

Due to the non-linear structure of the projection matrix manifold, it is unlikely that $\mathbf{Q}^*$ equals $\mathbf{Q}^\dagger$ for any coefficients. Instead, we aim to find a set of coefficients $w_1, \dots, w_m$ so that $\mathbf{Q}^*$ has a specific eigenbasis. Since the matrix $\mathbf{Q}^\dagger$ is known and a QDF representation of a flat, we know its eigenvectors and eigenvalues: spanning vectors $\mathbf{A}^\dagger = [\mathbf{a}_1^\dagger, \dots, \mathbf{a}_k^\dagger]$ to the eigenvalue zero, and $\mathbf{N}^\dagger = [\mathbf{n}_1^\dagger, \dots, \mathbf{n}_{d-k}^\dagger]$ to the eigenvalue one, of which we take any orthonormal set. In contrast, the eigenvalues and -vectors of $\mathbf{Q}^*$ are yet unknown, but since $\mathbf{Q}^*$ is symmetric, its eigenspaces are pairwise orthogonal. This also holds for non-intersecting (Minkowski) sums of eigenspaces.

The main idea of our method is to demand that $\mathbf{Q}^*$ has a sum of eigenspaces of dimension k that coincides with $\mathrm{img}(\mathbf{A}^\dagger)$ and the sum of the remaining eigenspaces (of dimension $d-k$) coincides with $\mathrm{img}(\mathbf{N}^\dagger)$. In equations, this means $\mathbf{Q}^*\mathbf{N}^\dagger\mathbf{v} = \mathbf{N}^\dagger\mathbf{v}'$ and $\mathbf{Q}^*\mathbf{A}^\dagger\mathbf{w} = \mathbf{A}^\dagger\mathbf{w}'$ for all $\mathbf{v} \in \mathbb{R}^{d-k}$ and $\mathbf{w} \in \mathbb{R}^k$. Taking the latter equation and multiplying $\mathbf{N}^\dagger\mathbf{v}$ on both sides from the left yields (recall $(\mathbf{N}^\dagger)^\mathsf{T}\mathbf{A}^\dagger = \mathbf{0}$)

$$\mathbf{v}^\mathsf{T}(\mathbf{N}^\dagger)^\mathsf{T}\mathbf{Q}^*\mathbf{A}^\dagger\mathbf{w} \stackrel{!}{=} 0 = \mathbf{v}^\mathsf{T}(\mathbf{N}^\dagger)^\mathsf{T}\mathbf{A}^\dagger\mathbf{w}' \qquad (8)$$

for all $\mathbf{v} \in \mathbb{R}^{d-k}$ and all $\mathbf{w} \in \mathbb{R}^k$. The bilinear map is always zero if and only if $(\mathbf{N}^\dagger)^\mathsf{T}\mathbf{Q}^*\mathbf{A}^\dagger = \mathbf{0}$.

This means that each pair of spanning and normal vector $(\mathbf{a}_i^\dagger, \mathbf{n}_j^\dagger)$ of the given $\mathbf{Q}^\dagger$ has to be orthogonal w.r.t. the

$$\begin{bmatrix} (\mathbf{n}_1^\dagger)^\mathsf{T}\mathbf{Q}_1\mathbf{a}_1^\dagger & (\mathbf{n}_1^\dagger)^\mathsf{T}\mathbf{Q}_2\mathbf{a}_1^\dagger & \dots & (\mathbf{n}_1^\dagger)^\mathsf{T}\mathbf{Q}_m\mathbf{a}_1^\dagger \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{n}_{d-k}^\dagger)^\mathsf{T}\mathbf{Q}_1\mathbf{a}_1^\dagger & (\mathbf{n}_{d-k}^\dagger)^\mathsf{T}\mathbf{Q}_2\mathbf{a}_1^\dagger & \dots & (\mathbf{n}_{d-k}^\dagger)^\mathsf{T}\mathbf{Q}_m\mathbf{a}_1^\dagger \\ (\mathbf{n}_1^\dagger)^\mathsf{T}\mathbf{Q}_1\mathbf{a}_2^\dagger & (\mathbf{n}_1^\dagger)^\mathsf{T}\mathbf{Q}_2\mathbf{a}_2^\dagger & \dots & (\mathbf{n}_1^\dagger)^\mathsf{T}\mathbf{Q}_m\mathbf{a}_2^\dagger \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{n}_{d-k}^\dagger)^\mathsf{T}\mathbf{Q}_1\mathbf{a}_k^\dagger & (\mathbf{n}_{d-k}^\dagger)^\mathsf{T}\mathbf{Q}_2\mathbf{a}_k^\dagger & \dots & (\mathbf{n}_{d-k}^\dagger)^\mathsf{T}\mathbf{Q}_m\mathbf{a}_k^\dagger \end{bmatrix}$$

(a) Linear equations for the ortientation.

$$\begin{bmatrix} (\mathbf{n}_1^\dagger)^\mathsf{T}(\mathbf{Q}_1\mathbf{r}^\dagger - \mathbf{r}_1) & \dots & (\mathbf{n}_1^\dagger)^\mathsf{T}(\mathbf{Q}_m\mathbf{r}^\dagger - \mathbf{r}_m) \\ \vdots & \ddots & \vdots \\ (\mathbf{n}_{d-k}^\dagger)^\mathsf{T}(\mathbf{Q}_1\mathbf{r}^\dagger - \mathbf{r}_1) & \dots & (\mathbf{n}_{d-k}^\dagger)^\mathsf{T}(\mathbf{Q}_m\mathbf{r}^\dagger - \mathbf{r}_m) \end{bmatrix}$$

(b) Additional linear equations for the displacement.

Figure 8. The linear system to acquire the coefficients $w_1, \dots w_m$ of a flat w.r.t. a basis of other flats. The right-hand side is zero for all equations.

bilinear form induced by $\mathbf{Q}^*$, mathematically

$$(\mathbf{n}_j^\dagger)^\mathsf{T}\mathbf{Q}^*\mathbf{a}_i^\dagger = \sum_{\ell=1}^{m} w_\ell (\mathbf{n}_j^\dagger)^\mathsf{T}\mathbf{Q}_\ell\mathbf{a}_i^\dagger = 0 \quad \substack{i\in\{1,\dots,k\},\\ j\in\{1,\dots,d-k\}}. \tag{9}$$

Recall that $(\cdot)^\dagger$ does not mean adjoint in this context, but represents the given target QDF. With the partition of unity property for the weights, this forms a system of $k(d-k)+1$ linear equations, whose solution yields a set of barycentric coordinates.

4.2. General flats

We now extend the described approach to flats that do not necessarily intersect the origin. As $\dim \mathrm{Graff}(k,d) = (k+1)(d-k)$, we have another $d-k$ degrees of freedom that we use to formulate linear equations. We observe that, during the projection, the extremum of q^*, i.e., the solution to $\mathbf{Q}^*\mathbf{x} = \mathbf{r}^*$, lies on the resulting flat, but is not necessarily equal to the resulting $\mathbf{r}$ as it has yet to be orthogonalized. We ask that said extremum is an element of $\mathcal{F}^\dagger$, which can be formulated as linear equations:

$$\mathbf{r}^* = \mathbf{Q}^*(\mathbf{A}^\dagger\mathbf{y} + \mathbf{b}^\dagger) \quad \text{for some } \mathbf{y} \in \mathbb{R}^k. \tag{10}$$

With the expansion of the parenthesis and the fact that $\mathbf{b}^\dagger = \mathbf{r}^\dagger$, we get the homogeneous linear system

$$\mathbf{Q}^*\mathbf{A}^\dagger\mathbf{y} + \mathbf{Q}^*\mathbf{r}^\dagger - \mathbf{r}^* = \mathbf{0}. \tag{11}$$

However, at this point, we added another d equations and k variables to the system. Recall that $(\mathbf{N}^\dagger)^\mathsf{T}\mathbf{Q}^*\mathbf{A}^\dagger = \mathbf{0}$. By

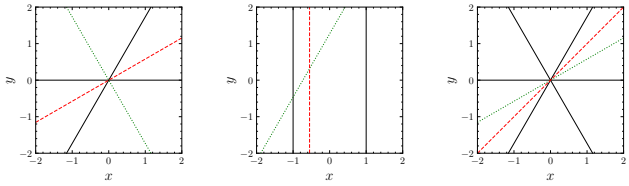


Figure 9. Some cases in which our method computes no or wrong coefficients: the basis (black, solid lines), the target $q^\dagger$ (green, dotted lines) and the result of recombination after computing the coefficients with our method (red, dashed lines).

multiplying with the normal vectors $\mathbf{N}^\dagger$ from the left side, we omit the need of the auxiliary variables and reduce the amount of additional equations to $(d-k)$:

$$(\mathbf{N}^\dagger)^\mathsf{T}\mathbf{Q}^*\mathbf{r}^\dagger - (\mathbf{N}^\dagger)^\mathsf{T}\mathbf{r}^* = \mathbf{0} \Leftrightarrow \sum_{i=1}^{m} w_i (\mathbf{N}^\dagger)^\mathsf{T}(\mathbf{Q}_i\mathbf{r}^\dagger - \mathbf{r}_i) = \mathbf{0}. \tag{12}$$

The solution to the linear system in Fig. 8 under the partition of unity constraint yields the coefficients $w_1, \dots, w_m$.

4.3. Properties and Experiments

Our method consists of solving a linear system with $(k+1)(d-k)+1$ equations/rows, which becomes square if the number m of basis elements matches this count. The first $k(d-k)$ equations are constructed so that they restrict the eigenspaces of the resulting combination – they do not impose any constraints on the eigenvalues. Ideally, the sum of eigenspaces to the smallest k eigenvalues of $\mathbf{Q}^*$ corresponds to $\mathrm{img}(\mathbf{A}^\dagger)$, but this is not the case for certain inputs that are discussed in the following.

Properties Interestingly, our method returns coefficients, even if the target $q^\dagger$ is not an element of the convex (or even affine) hull of $q_1, \dots, q_m$. An example for this phenomenon is shown in the left of Fig. 9: the target line is projected onto its orthogonal line. Using a basis of two parallel lines, one can only represent lines also parallel to them (which is the affine hull of the two in this case). Using a non-parallel target, the linear system has no solutions. Using a least-squares solver projects it onto a parallel line (middle). However, it is unclear to us which line exactly it is projected to. Furthermore, one does not have to use $m = (k+1)(d-k)+1$ (or $m = k(d-k)+1$ in the case of linear subspaces) basis elements. If one uses less than m basis elements but $q^\dagger$ is inside the convex hull, we still get the correct coefficients. However, using more basis elements than necessary can lead to wrong coefficients, even if the target is inside the convex hull (right). This results from the fact that the affine sum of two possible coordinates is not necessarily a valid coordinate, as we are projecting onto a curved manifold.

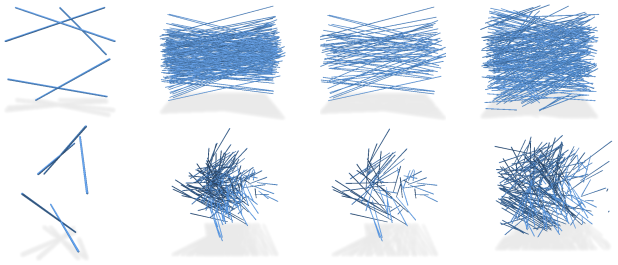


Figure 10. A simplex in the 4-dimensional space of affine lines centered around lines in the x-direction (left), shown from the z and x direction (top and bottom). The images in the middle show random points in the simplex (middle left) and the the subset spanning the convex hull (middle right), computed from the Delaunay triangulation in the space of affine lines. The right images show random affine combination, including points outside the simplex.

Experiments We first generate a d-simplex of affinely independent points $\mathbf{v}_1, \dots, \mathbf{v}_{d+1} \in [-1, 1]^d$. Then, we generate $m = (k+1)(d-k)+1$ pairwise distinct index sets $V_i \subseteq \{1, \dots, d+1\}$ of size $d+1$. As the basis, we use the QDFs q_i of the flats $\mathcal{F}_i$ that contain the points whose indices are in the corresponding V_i. We generate a number of coefficients $\alpha'_1, \dots, \alpha'_m \in [0, 1]$ and divide by their sum, so that the resulting coefficients $\alpha_1, \dots, \alpha_m$ sum up to one. Afterward, we compute the convex combination $q^\dagger(\mathbf{x}) = \mathcal{P}\left(\sum_{i=1}^m \alpha_i q_i\right)$, and give the basis $q_1, \dots, q_m$ as well as the combination $q^\dagger$ to our method as input, receiving coefficients $w_1, \dots, w_m$ as output. In every single case, the resulting coefficients $w_1, \dots, w_m$ of our method coincided with the randomly generated coefficients $\alpha_1, \dots, \alpha_m$.

5. Application: Triangulations

We believe computing with flats has many interesting applications. One particular case we imagine is that one wants to represent a function $f : \mathrm{Graff}(k, d) \to \mathcal{M}$ based on samples $((\mathbf{Q}_i, \mathbf{r}_i), f_i)$. For this, we will assume that the samples are in the affine hull of a simplex. Then, we can use classic techniques from scattered data interpolation [6], in particular, a triangulation.

Recall that our method provides affine coordinates $\mathbf{a}_i \in \mathbb{R}^m$ for each sample location $(\mathbf{Q}_i, \mathbf{r}_i)$. These affine coordinates lie on a specific hyperplane ($\sum_{i=1}^m w_i = 1$) of dimension $m-1$ in $\mathbb{R}^m$, and multiplying them with the vertex positions of a regular simplex in $\mathbb{R}^{m-1}$ yields Euclidean coordinates. These Euclidean coordinates can be treated as points in space and used to compute a Delaunay triangulation, using an existing implementation (we use CGAL [3]).

In the triangulation, we attach the function values f_i to the corresponding vertices. This defines a piecewise linear function, approximating the underlying sampled function. Any parameter value in the form of a flat $(\mathbf{Q}, \mathbf{r})$ can be turned into an affine and then Euclidean coordinate in the same way as explained above. Delaunay triangulations provide fast look-ups of the simplex that contains the point using *walking* [2]. In the simplex of the triangulation, the function values can be interpolated based on the barycentric coordinates of the query point $(\mathbf{Q}, \mathbf{r})$, providing the interpolated function value.

In the following, we make this process concrete for the case of affine lines in 3D, which are not only relevant in practice, but also cover all non-trivial aspects of the approach, as both the orientation component as well as the offset component are more than one-dimensional. In fact, this space is the product of the real projective plane (the space of orientations in 3D) and the 2D Euclidean plane, so the intrinsic dimension is 4. The scattered data interpolation task we consider is inspired by the surge of representing the visual information of the 3D world based on images [26]. For simplicity, we associate each affine line to a single color value, in the spirit of classic view interpolation based on the assumption of low depth complexity [13, 18].

5.1. Base simplex

The choice of a suitable base simplex is crucial, as this determines the relative distances of angles and distances in Euclidean space. We have found in experiments that the wrong choice of simplex may lead to interpolating essentially only in the space of orientation, or only in the space of offsets.

We have constructed a quasi-regular simplex in the space of affine lines as follows: we start by considering a fixed line, typically one of the canonical axes (i.e., zero offset). Then, we compute the 4D tangent plane in the ambient $(\mathbf{Q}, \mathbf{r})$-space for this line and then take a regular 4-simplex centered at the origin of this tangent space.

To represent the tangent space, let the direction of the line be $\mathbf{a}$ and two vectors $\mathbf{n}_1, \mathbf{n}_2$ spanning the orthogonal complement. Then, we have

$$(\mathbf{Q}, \mathbf{r}) = (\mathbf{n}_1\mathbf{n}_1^\mathsf{T} + \mathbf{n}_2\mathbf{n}_2^\mathsf{T}, \alpha_1\mathbf{n}_1 + \alpha_2\mathbf{n}_2), \tag{13}$$

which we flatten to a vector in $\mathbb{R}^{12}$, with α_1, α_2 representing the offset. We now take the four partial derivatives w.r.t. varying $\mathbf{a}$ along $\mathbf{n}_1, \mathbf{n}_2$ and varying α_1, α_2. This gives the gradient (each line flattened again)

$$\nabla(\mathbf{Q}, \mathbf{r}) = \begin{bmatrix} -\mathbf{n}_1\mathbf{a}^\mathsf{T} - \mathbf{a}\mathbf{n}_1^\mathsf{T} + \mathbf{a}\mathbf{a}^\mathsf{T}, \alpha_1\mathbf{a} \\ -\mathbf{n}_2\mathbf{a}^\mathsf{T} - \mathbf{a}\mathbf{n}_2^\mathsf{T} + \mathbf{a}\mathbf{a}^\mathsf{T}, \alpha_2\mathbf{a} \\ \mathbf{0}, \mathbf{n}_1 \\ \mathbf{0}, \mathbf{n}_2 \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{12\times 4}. \tag{14}$$

A regular 4-simplex centered at the origin is

$$\mathbf{S}_4 = \begin{bmatrix} 1 & 1 & -1 & -1 & 0 \\ 1 & -1 & 1 & -1 & 0 \\ 1 & -1 & -1 & 1 & 0 \\ \frac{-1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{4}{\sqrt{5}} \end{bmatrix} \in \mathbb{R}^{4\times 5}. \tag{15}$$

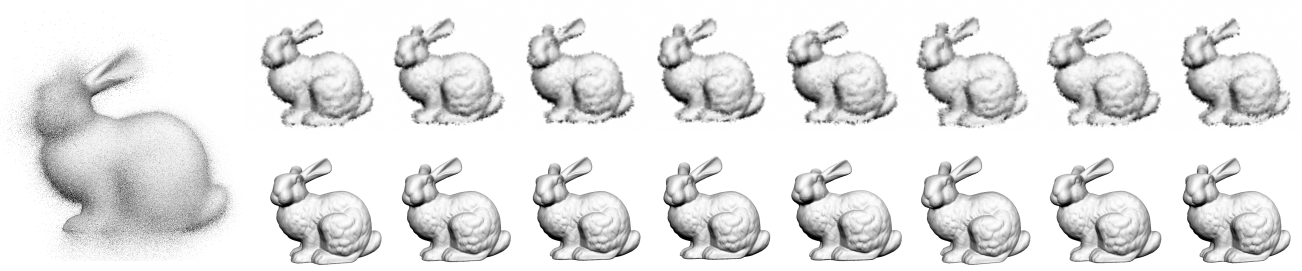


Figure 11. The BUNNY is sampled on a fixed 1000^2 raster screen in the xy-plane with eye positions chosen uniformly random in $[-1,1]^2 \times [-4.5,-1.5]$. The resulting gray values (displayed on the fixed grid on the left) are associated to a the vertex of a Delaunay triangulation in the space of affine lines. Specific images can be reconstructed by choosing a fixed eye point and image plane and looking up the necessary rays in the Delaunay triangulation by barycentric interpolation. The top row shows reconstructions for varying focal lengths (first four images) and rotating the image screen and eye by $\pm 10^\circ$ along the major axes (last four images). The bottom row shows ground truth. Reconstructions are at the resolution 250^2 and use four-times oversampling.

Multiplying the gradient with the coordinates of the simplex yields five directions in the tangent space of $(\mathbf{Q},\mathbf{r})$. Unflattening them and adding a small multiple of each to the line's $(\mathbf{Q},\mathbf{r})$ gives the five vertices of a simplex in the $(\mathbf{Q},\mathbf{r})$-representation. As we constructed it on the tangent plane, it is regular in the space of affine lines only for infinitesimal scale. We have verified that the distances among the vertices are close to identical for small scale factors also in floating point representation. In practice, it is advisable to use the smallest possible scale so that the affine hull still spans the orientations of the sample set. For now, we have no better method than bisection search for the optimal scale parameter. This seems tolerable, as it is a one-time process for each application. One such basis is shown in Fig. 10 (left).

5.2. Triangulation of view rays

We create a virtual scene with an object centered at the origin, the camera plane containing the z-axis as the up-direction, assuming the eye is along the (negative) y-direction. Then, we create an eye ray for each pixel of the camera plane by drawing the eye position uniformly random from $[-1,1]^2 \times [-4.5,-1.5]$. For each eye ray, we compute an intersection with the object and use diffuse shading to compute a gray value. The result could be considered a local but random light-field sampling; it is shown for 1000^2 pixels on the camera plane in Fig. 11 (left).

For the simplex covering the lines, we use the method above for a line along the y-direction, the base simplex being multiplied with 0.01. This includes all but very few of the line samples (the number of outliers in a sample depends on the random choice of eye points). Based on the simplex, we compute Euclidean coordinates by first projecting into the affine hull of the simplex (as in Sec. 4.2) and then turning the affine coordinate into a linear one, as explained above. The resulting coordinates are then Delaunay-triangulated with CGAL's dD-triangulation package [3]. To each vertex in the triangulation, we attach the corresponding gray value.

The triangulation provides a piecewise linear approximation of the map from affine lines to color values. We can now generate images using this approximation by fixing an eye point and collecting the color values for each line through the eye and a pixel by barycentric interpolation in the triangulation. Several such reconstructions for varying focal lengths and angles towards the original image plane are shown in Fig. 11 and the supplemental material.

6. Discussion

We have found a simple and efficient way to enable computation in the non-linear space of flats and demonstrated, albeit at simplistic examples, that it can be used in practice. There are still many open questions, both theoretical as well as practical.

It would be interesting to compute Delaunay triangulations based on the intrinsic metric of the space, yet we currently lack the equivalent of an in-sphere test, even assuming that we are restricted to an affine hull (for the compact space of orientations, this is generally difficult). It is already unclear whether the distance between two flats in $(\mathbf{Q},\mathbf{r})$-space can be computed in a closed form.

Another interesting problem is that of finding useful bases (simplices) for given problems, e.g., containing all data points if they are given. Furthermore, is there a basis that fulfills additional desired properties (e.g., it can be used as a reduced basis like in PCA)?

We have also not yet used the full potential of the representation for combining flats of varying dimension. Understanding how the convex and affine hulls of a set of flats look like has already proven to be challenging, however, for flats of varying dimensions it is wide open.

Acknowledgements

The authors thank Jonas Engler (TU Berlin) for maintaining and extending the code base, as well as for conducting additional experiments with parameter optimization leading to the visualizations shown in the appendix. This work was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement No. 101055448, ERC Advanced Grand EMERGE).

References

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80:199–220, 2004. 1

[2] Olivier Devillers, Sylvain Pion, and Monique Teillaud. Walking in a triangulation. Technical Report RR-4120, INRIA, 2001. 7

[3] Olivier Devillers, Samuel Hornus, and Clément Jamin. dD triangulations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 6.0.1 edition, 2024. 7, 8

[4] Gabriel Dogadov, Ugo Finnendahl, and Marc Alexa. Fitting flats to flats. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 3, 4

[5] Bruce Draper, Michael Kirby, Justin Marks, Tim Marrinan, and Chris Peterson. A flag representation for finite collections of subspaces of mixed dimensions. *Linear Algebra and its Applications*, 451:15–32, 2014. 3

[6] Richard Franke and Gregory M. Nielson. Scattered data interpolation and applications: A tutorial and survey. In *Geometric Modeling*, pages 131–160, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg. 7

[7] B.J. Frey, A. Colmenarez, and T.S. Huang. Mixtures of local linear subspaces for face recognition. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, pages 32–37, 1998. 1

[8] William Fulton. Eigenvalues of sums of hermitian matrices. *Séminaire Bourbaki*, 40(255-269):5, 1998. 3

[9] Jean Gallier. Basics of affine geometry. *Geometric Methods and Applications: For Computer Science and Engineering*, pages 7–63, 2011. 1

[10] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997. 2, 3

[11] Michael Garland and Paul S Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *Proceedings Visualization'98 (Cat. No. 98CB36276)*, pages 263–269. IEEE, 1998. 3

[12] Michael Garland and Yuan Zhou. Quadric-based simplification in any dimension. *ACM Transactions on Graphics (TOG)*, 24(2):209–239, 2005. 3

[13] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, page 43–54, New York, NY, USA, 1996. Association for Computing Machinery. 7

[14] Jun He, Dejiao Zhang, Laura Balzano, and Tao Tao. Iterative online subspace learning for robust image alignment. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8. IEEE, 2013. 1

[15] Alfred Horn. Eigenvalues of sums of hermitian matrices. 1962. 3

[16] Hyung-Soo Lee and Daijin Kim. Expression-invariant face recognition by facial expression transformations. *Pattern Recognition Letters*, 29(13):1797–1805, 2008. 1

[17] Kuang-Chih Lee, J. Ho, and D.J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005. 1

[18] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, page 31–42, New York, NY, USA, 1996. Association for Computing Machinery. 7

[19] Stan Z Li, Anil K Jain, Gregory Shakhnarovich, and Baback Moghaddam. Face recognition in subspaces. *Handbook of face recognition*, pages 141–168, 2005. 1

[20] Lek-Heng Lim, Ken Sze-Wai Wong, and Ke Ye. Numerical algorithms on the affine grassmannian. *SIAM Journal on Matrix Analysis and Applications*, 40(2):371–393, 2019. 1

[21] Lek-Heng Lim, Ken Sze-Wai Wong, and Ke Ye. The grassmannian of affine subspaces. *Foundations of Computational Mathematics*, 21(2):537–574, 2021. 1, 3

[22] Hsueh-Ti Derek Liu, Xiaoting Zhang, and Cem Yuksel. Simplifying triangle meshes in the wild. *arXiv preprint arXiv:2409.15458*, 2024. 3

[23] Ravi Malladi, James A Sethian, and Baba C Vemuri. Shape modeling with front propagation: A level set approach. *IEEE transactions on pattern analysis and machine intelligence*, 17(2):158–175, 1995. 2

[24] Nathan Mankovich and Tolga Birdal. Chordal averaging on flag manifolds and its applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3881–3890, 2023. 3

[25] Nathan Mankovich, Emily J King, Chris Peterson, and Michael Kirby. The flag median and flagirls. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10339–10347, 2022. 3

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, 2021. 7

[27] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901. 1

[28] Tomer Perets. *Clustering of lines*. Open University of Israel Ra'anana, Israel, 2011. 1

[29] Yohann Salaün, Renaud Marlet, and Pascal Monasse. Line-based robust sfm with little image overlap. In *2017 International Conference on 3D Vision (3DV)*, pages 195–204. IEEE, 2017. 1

[30] René Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011. 1

[31] Hermann Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912. 3

[32] Yung-Chow Wong. Differential geometry of grassmann manifolds. *Proceedings of the National Academy of Sciences*, 57(3):589–594, 1967. 1

[33] Dong-Ming Yan, Wenping Wang, Yang Liu, and Zhouwang Yang. Variational mesh segmentation via quadric surface fitting. *Computer-Aided Design*, 44(11):1072–1082, 2012. 3

[34] Ming-Hsuan Yang, N. Abuja, and D. Kriegman. Face detection using mixtures of linear subspaces. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 70–76, 2000. 1

[35] Tong Zhao, Laurent Busé, David Cohen-Steiner, Tamy Boubekeur, Jean-Marc Thiery, and Pierre Alliez. Variational shape reconstruction via quadric error metrics. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 3

[36] Yingying Zhu, Dong Huang, Fernando De La Torre, and Simon Lucey. Complex non-rigid motion 3d reconstruction by union of subspaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1542–1549, 2014. 1